Diversity Transfer Network for Few-Shot Learning

Mengting Chen,[1*†] **Yuxin Fang,**[1*] **Xinggang Wang,**[1‡] **Heng Luo,**[2] **Yifeng Geng,**[2]
Xinyu Zhang, Chang Huang,[2] **Wenyu Liu,**[1‡] **Bo Wang**[3,4]

[1]School of EIC, Huazhong University of Science and Technology [2]Horizon Robotics, Inc. [3]Vector Institute [4]PMCC, UHN
{mengtingchen, yuxin_fang, xgwang, liuwy}@hust.edu.cn {heng.luo, yifeng.geng, chang.huang}@horizon.ai
xyzhang.me@foxmail.com bowang@vectorinstitute.ai

Abstract

Few-shot learning is a challenging task that aims at training a classifier for unseen classes with only a few training examples. The main difficulty of few-shot learning lies in the lack of intra-class diversity within insufficient training samples. To alleviate this problem, we propose a novel generative framework, Diversity Transfer Network (DTN), that learns to transfer latent diversities from known categories and composite them with support features to generate diverse samples for novel categories in feature space. The learning problem of the sample generation (i.e., diversity transfer) is solved via minimizing an effective meta-classification loss in a single-stage network, instead of the generative loss in previous works. Besides, an organized auxiliary task co-training over known categories is proposed to stabilize the meta-training process of DTN. We perform extensive experiments and ablation studies on three datasets, i.e., *mini*ImageNet, CIFAR100 and CUB. The results show that DTN, with single-stage training and faster convergence speed, obtains the state-of-the-art results among the feature generation based few-shot learning methods. Code and supplementary material are available at: `https://github.com/Yuxin-CV/DTN`.

Introduction

Deep neural networks (DNNs) have shown tremendous success in solving many challenging real-world problems when a large amount of training data is available (Krizhevsky, Sutskever, and Hinton 2012; Simonyan and Zisserman 2014; He et al. 2015). Common practice suggests that models with more parameters have the greater capacity to fit data and more training data usually provide better generalization ability. However, DNNs struggle to generalize given only a few training data while humans excel at learning new concepts from just a few examples (Bloom 2000). Few-shot learning has therefore been proposed to close the performance gap between machine learner and human learner. In the canonical setting of few-shot learning, there are a training set $\mathcal{D}_{train}$ (seen, known) and a testing set $\mathcal{D}_{test}$ (unseen, novel) with *disjoint* categories. Models are trained on the training set while

tested in an N-way K-shot scheme (Vinyals et al. 2016) where the models need to classify the queries into one of the N categories correctly when only K samples of each novel category are given. This unique setting of few-shot learning poses an unprecedented challenge in fully utilizing the prior information in the training set $\mathcal{D}_{train}$, which corresponds to the known information or historical information of the human learner. Common approaches to address this challenge either learn a good metric for novel tasks (Snell, Swersky, and Zemel 2017; Vinyals et al. 2016; Sung et al. 2018) or train a meta-learner for fast adaptation (Finn, Abbeel, and Levine 2017; Munkhdalai and Yu 2017; Ravi and Larochelle 2017).

Recently, the generation based approach is becoming an effective solution for few-shot learning (Hariharan and Girshick 2017; Schwartz et al. 2018; Wang et al. 2018; Zhang et al. 2018), since it directly alleviates the problem of lacking training samples. We propose a Diversity Transfer Network (DTN) for sample generation. In DTN, the offset between a random sample pair from the known category is composited with a support sample in the novel category in the latent feature space. Then, the generated features, as well as the support features, are averaged as the proxy of the novel category. At last, query samples are evaluated by the proxy. Only if the generated samples follow the distribution of the real samples to be diverse, can the meta-classifier (i.e., the proxy) be robust enough to classify queries correctly.

In addition to the new sample generation scheme, we utilize an effective meta-training curriculum called OAT (Organized Auxiliary task co-Training), inspired by the auxiliary task co-training in TADAM (Oreshkin, Rodríguez López, and Lacoste 2018) and curriculum learning (Bengio et al. 2009). OAT organizes auxiliary tasks and meta-tasks reasonably and effectively reduces training complexity. Experiments show that by applying OAT, our DTN converges much faster compared with the naïve meta-training strategy (i.e., meta-training from scratch), the multi-stage training strategy used in Δ-encoder (Schwartz et al. 2018) and the auxiliary task co-training strategy used in TADAM.

The main components of DTN are integrated into a single network and can be optimized in an end-to-end fashion. Thus, DTN is very simple to implement and easy to train. Our experimental results show that this simple method outperforms many previous works on a variety of datasets.

Related Work

Metric Learning Based Approaches

Metric learning is the most common and straightforward solution for few-shot learning. An embedding function can be learned by a myriad of instances of known categories. Then some simple metrics, such as Euclidean distance (Snell, Swersky, and Zemel 2017) and cosine distance (Vinyals et al. 2016; Qiao et al. 2018; Qi, Brown, and Lowe 2018), are used to build nearest neighbor classifiers for instances in unseen categories. Furthermore, to model the contextual information among support images and query images, bidirectional LSTM and attention mechanism are adopted in Matching Network (Vinyals et al. 2016). Besides measuring the distances of a query to its support images, there is a new solution that compares the query to the center of the support images of each class in feature space, such as Snell, Swersky, and Zemel; Qiao et al.; Qi, Brown, and Lowe; Gidaris and Komodakis. The center is usually termed as a proxy of the class. Specifically, squared Euclidean distance is used in Prototypical Network (Snell, Swersky, and Zemel 2017), and cosine distance is used in the other works. Snell, Swersky, and Zemel; Qi, Brown, and Lowe directly calculate proxies by averaging the embedding features, while Qiao et al.; Gidaris and Komodakis take a small network to predict proxies. Based on Prototypical Network, TADAM (Oreshkin, Rodríguez López, and Lacoste 2018) further proposes a dynamic task conditioned feature extractor by predicting the layer-level element-wise scale and shift vectors for each convolutional layer. Different from simple metrics, Relation Network (Sung et al. 2018) takes the neural network as a non-linear metric and directly predicts the similarities between the query and support images. TPN (Liu et al. 2019) performs transductive learning on the similarity graph contains both query and support images to obtain high-order similarities.

Meta-Learning Based Approaches

Meta-learning approaches have been widely used in few-shot learning scenarios by optimization learning for fast adaptation, aiming to learn a meta-learner that can solve the novel task quickly. Meta Network (Munkhdalai and Yu 2017) and adaResNet (Munkhdalai et al. 2018) are memory-based methods. Example and task level information in Meta Network are preserved in fast and slow weights, respectively. AdaResNet performs rapid adaptation by mimicking conditionally shifted neurons which modify activation values with task-specific shifts retrieved from a memory module. An LSTM-based update rule of the parameters of a classifier is proposed in Ravi and Larochelle, where both short-term knowledge within a task and long-term knowledge common among all the tasks are learned. MAML (Finn, Abbeel, and Levine 2017), LEO (Rusu et al. 2019) and MT-net (Lee and Choi 2018) all differentiate through gradient update steps to optimize performance after fine-tuning. While MAML operates directly in high dimensional parameter space, LEO performs meta-learning within a low-dimensional latent space. Different from MAML that assumes a fixed model, MT-net chooses a subset of its weights to fine-tune. Franceschi et al. propose a method based on bi-level programming that unifies gradient-based hyper-parameter optimization and meta-learning.

Generation Based Approaches

Sample synthesis using the generative models has recently emerged as a popular direction for few-shot learning (Zhu et al. 2017; Goodfellow et al. 2014). How to synthesize new samples based on a few examples remains an interesting open problem. AGA (Dixit et al. 2017) and FATTEN (Liu et al. 2018) are attribute-guided (w.r.t. pose and depth) augmentation methods in feature space by leveraging a corpus with attribute annotations. Hariharan and Girshick tries to transfer transformations from a pair of examples from a known category to a "seed" example of a novel class. Finding specific generation targets requires a carefully designed pipeline with heuristic steps. Δ-encoder (Schwartz et al. 2018) also tries to extract intra-class deformations between image pairs sampled from the same class. Wang et al. proposes to generate samples by adding random noises to support features. Different from previous methods, MetaGAN (Zhang et al. 2018) generates fake samples that need to be discriminated by the classifier instead of augmentation, which sharpens the decision boundaries of novel categories.

Our proposed DTN shares a philosophical similarity with image hallucination (Hariharan and Girshick 2017) and Δ-encoder (Schwartz et al. 2018) with distinct differences in the following aspects. The first difference is that DTN does not require to set specific target points for the generator. More specifically, Δ-encoder takes a pair of images X_a^1 and X_a^2 from the same class and learns to infer the diversity between them by reconstructing X_a^2. The image hallucination method collects quadruples $(X_a^1, X_a^2, X_b^1, X_b^2)$ for training based on clustering and traversal; each quadruple contains two image pairs from two classes a and b; a generation network is trained to predict a sample X_a^2 from the quadruple when the rest three (X_a^1, X_b^1, X_b^2) are given as input. Then, synthesized samples are used to train a linear classifier. The input of the generator in DTN is also a triplet (X_a^1, X_b^1, X_b^2) as Hariharan and Girshick, but the generated sample $\hat{X}_a^2$ is used directly to construct the meta-classifier, and the generator is optimized by minimizing the meta-classification loss instead of setting specific generation targets. Secondly, DTN integrates feature extraction, feature generation, and meta-learning into a single network and enjoys the simplicity and effectiveness of end-to-end training, while Hariharan and Girshick; Schwartz et al. are stage-wise methods.

More recent work based on sample generation and data augmentation are IDeMe-Net (Zitian Chen 2019) and SalNet (Zhang, Zhang, and Koniusz 2019). The former utilizes an additional deformation sub-network with a large number of parameters to synthesize diverse deformed images, the latter needs to pre-train a saliency network on the MSRA-B dataset. In contrast to these approaches, our method is based on a simple diversity transfer generator that learns a better proxy of each category with fewer parameters and faster convergence speed. Besides, our method can be regarded as an instance of compositional learning (Yuille 2011) in the latent feature space.
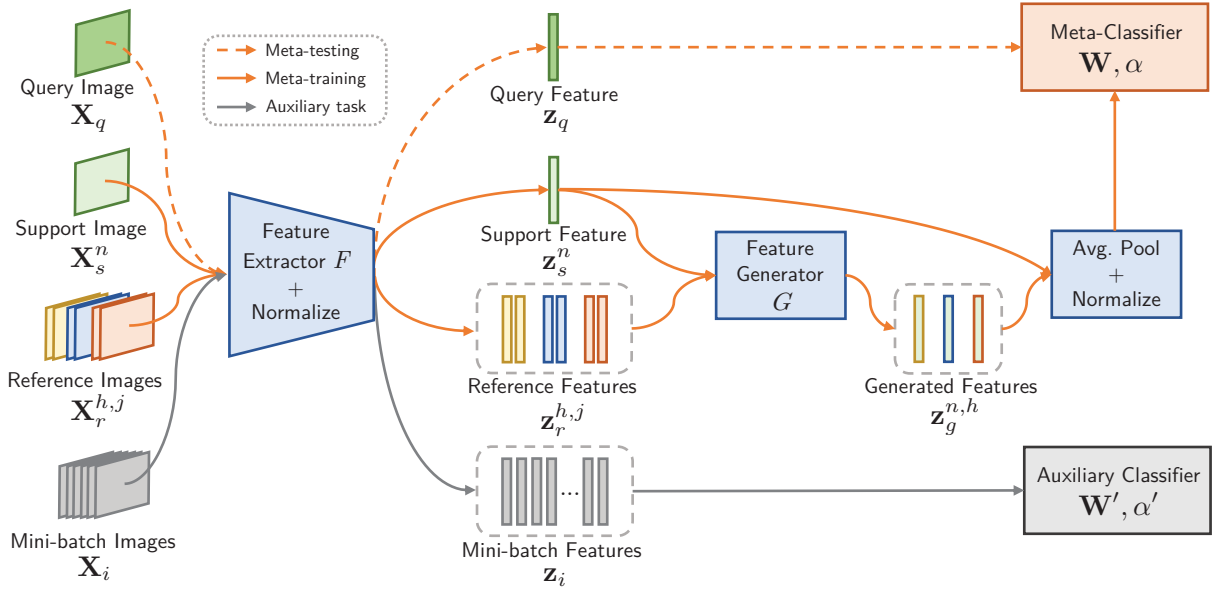
Figure 1: Illustration of the proposed diversity transfer network. The branch indicated by orange arrows is the meta-task, which is trained in a meta-learning way. The orange solid arrows indicate the process of meta-training, while the orange dashed arrows indicate the process of meta-testing. During meta-training, the features of the support image and reference images from the feature extractor are fed into the feature generator to generate new features. The parameters of the meta-classifier are formed by the averaged proxies of the support features and generated features. Then the query image is fed to evaluate the performance of the meta-classifier during meta-testing. The branch indicated by grey arrows is the auxiliary task aimed to accelerate convergence and improve the generalization ability.

Method

Problem Definition

Different from the conventional classification task, where the training set $\mathcal{D}_{train}$ and the testing set $\mathcal{D}_{test}$ consist of samples from the same classes, few-shot learning aims to address the problem where the label spaces are disjoint between $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$. We follow the standard N-way K-shot classification scenario defined in Vinyals et al. to study the few-shot learning problem. An N-way K-shot task is termed as an episode. An episode is formed by N classes sampled from the training/testing set firstly. Then K images sampled from each of the N classes constitute the support set $\{(\mathbf{X}_s^{n,k}, Y_s^{n,k})\}$, where $n \in [1, 2, ..., N]$ and $k \in [1, 2, ..., K]$. For the sake of simplicity, we take N-way 1-shot (i.e., $K = 1$) classification for example in the following sections, and the support set will be simplified to $\{(\mathbf{X}_s^n, Y_s^n)\}$. The query sample $(\mathbf{X}_q, Y_q)$ is sampled from the rest images of the N classes. The goal is to classify the query into one of the N classes correctly based only on the support set and the prior meta-knowledge learned from the training set $\mathcal{D}_{train}$.

An Overview of Diversity Transfer Network

The overall structure of the Diversity Transfer Network (DTN) is shown in Fig. 1. DTN contains four modules and is organized into two task branches. The task branch indicated by orange arrows is the meta-task, which is trained in a meta-learning way. The input for the meta-task consists of

the following three parts: support images $\mathbf{X}_s^n$, a query image $\mathbf{X}_q$ and reference images $\mathbf{X}_r^{h,j}$, where $n \in [1, 2, ..., N]$, $h \in [1, 2, ..., H]$, $j \in [1, 2]$. All images are mapped to L_2-normalized feature vectors $\mathbf{z} = F(\mathbf{X})$ by a feature extractor F, where $\mathbf{z} \in \mathbb{R}^C$. $\mathbf{z}_r^{h,1}$ and $\mathbf{z}_r^{h,2}$ are feature vectors of two reference images. They come from the same category and make up a reference pair. The diversity of the pair is transferred to the support feature $\mathbf{z}_s^n$ to generated a new feature by the feature generator G. The generated feature $\mathbf{z}_g^{n,h}$ is supposed to belong to the same category with $\mathbf{z}_s^n$. For each support feature, there are H samples generated based on it. Since a meta-task is an N-way 1-shot image classification task, the meta-classifier is an N-way classifier consisting of a weight matrix $\mathbf{W}$ and a trainable temperature α. The values in the $\mathbf{W}$ are determined by the proxies formed by support features and features generated by them. The meta-classifier is differentiable, so the feature extractor F and feature generator G can be updated by standard back-propagation according to the loss function defined by the cosine similarity between the query and the proxies. The task branch indicated by grey arrows in Fig. 1 is the auxiliary task, aiming to accelerate and stabilize the training of DTN. It is a conventional classification task over all categories of the training set $\mathcal{D}_{train}$.

Feature Generation via Diversity Transfer

Each image $\mathbf{X}$ is mapped to a feature vector $\mathbf{z} = F(\mathbf{X})$ by the feature extractor F. $\mathbf{z}_q$, $\mathbf{z}_s^n$ and $\{\mathbf{z}_r^{h,1}, \mathbf{z}_r^{h,2}\}$ are feature vectors of the query image $\mathbf{X}_q$, the support image $\mathbf{X}_s^n$ and
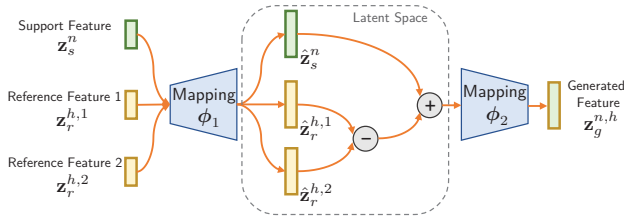
Figure 2: Feature generator in DTN. The three input features are mapped into a latent space by the mapping function ϕ_1. Then the diversity (i.e., offset) between the reference features is added with the support feature in this space. Then it is mapped by the ϕ_2 to keep the same size as inputs. The output is a generated feature which is supposed to be a sample belonging to category n.

the reference images pair $\{\mathbf{X}_r^{h,1}, \mathbf{X}_r^{h,2}\}$ respectively, where $n \in [1, 2, ..., N]$, $h \in [1, 2, ..., H]$. For a specific support feature $\mathbf{z}_s^n$, during both meta-training and meta-testing phase, the reference image pairs $\{\mathbf{X}_r^{h,1}, \mathbf{X}_r^{h,2}\}$ are always sampled from the training set $\mathcal{D}_{train}$ (seen, known). Specifically, we first randomly sample H classes from the whole training classes $\mathcal{D}_{train}$ with replacement. For each sampled class, we then randomly sample two different images $\mathbf{X}_r^{h,1}$ and $\mathbf{X}_r^{h,2}$ to form a reference pair. We do not sample any images from $\mathcal{D}_{test}$ (unseen, novel) during the whole process. The conventional few-shot evaluation setting, termed as N-way K-shot setting, requires to get a N-way classifier with the support of only K samples for each novel class and the prior meta-knowledge from the whole training set $\mathcal{D}_{train}$. Therefore, our sampling method strictly complies with the few-shot learning protocol.

As shown in Fig. 2, the feature generator G of DTN consists of two mapping functions ϕ_1 and ϕ_2. Three input features are firstly mapped into a latent space $\hat{\mathbf{z}} = \phi_1(\mathbf{z})$, $\hat{\mathbf{z}} \in \mathbb{R}^{C'}$. The elementwise difference $\hat{\mathbf{z}}_r^{h,1} - \hat{\mathbf{z}}_r^{h,2}$ measures the diversity between the two reference features. It is applied to the support feature by a simple linear combination $\hat{\mathbf{z}}_s^n + (\hat{\mathbf{z}}_r^{h,1} - \hat{\mathbf{z}}_r^{h,2})$. After mapping it by ϕ_2, we get a feature $\mathbf{z}_g^{n,h}$ which has the same size of the input $\mathbf{z}_s^n$ and should belong to the same category with the support feature $\mathbf{z}_s^n$. More specifically:

$$
\begin{aligned}
\mathbf{z}_g^{n,h} &= G(\mathbf{z}_s^n, \mathbf{z}_r^{h,1}, \mathbf{z}_r^{h,2}) \\
&= \phi_2(\phi_1(\mathbf{z}_s^n) + \phi_1(\mathbf{z}_r^{h,1}) - \phi_1(\mathbf{z}_r^{h,2})).
\end{aligned}
\tag{1}
$$

Given H different reference pairs for a single support feature $\mathbf{z}_s^n$, there will be H generated features that enrich the diversity of category n. They are helpful to construct a more robust classifier for unseen categories. When $K > 1$, each of the K support samples is taken as a "seed" and H samples are generated based on it. Therefore, there will be K support samples and $K \times H$ generated samples for each novel category.

Meta-Learning Based on Averaged Proxies
The meta-task branch of DTN is shown in Fig. 1 indicated by orange arrows. The orange solid arrows and dashed ar-

rows indicate the process of meta-training and meta-testing, respectively. Each image $\mathbf{X}$ is mapped to a feature vector $\mathbf{z} = F(\mathbf{X})$. Similar to Qiao et al.; Gidaris and Komodakis; Qi, Brown, and Lowe, all the features here are L_2-normalized vectors (i.e., $\|\mathbf{z}\|_2 = 1$). The support feature $\mathbf{z}_s^n$ and all the H reference feature pairs $\{\mathbf{z}_r^{h,1}, \mathbf{z}_r^{h,2}\}(h \in [1, 2, ..., H])$ are fed into the generator G to generate H new features $\mathbf{z}_g^{n,h}$ (H is set to 3 in Fig. 1 for example). So we get $H + 1$ features for the n-th category. The meta-task is an N-way classification task, therefore the meta-classifier is represented by a matrix $\mathbf{W} \in \mathbb{R}^{N \times C}$, in which each row $\mathbf{w}_n \in \mathbb{R}^C$ can be viewed as a proxy (Movshovitz-Attias et al. 2017) of the n-th category. After obtaining all the $H + 1$ features for category n, the n-th row $\mathbf{w}_n$ of $\mathbf{W}$, termed as averaged proxy, is the L_2-normalized average of those features:

$$
\hat{\mathbf{w}}_n = \frac{1}{H+1} \left(\mathbf{z}_s^n + \sum_{h=1}^H \mathbf{z}_g^{n,h} \right),
\tag{2}
$$

$$
\mathbf{w}_n = \frac{\hat{\mathbf{w}}_n}{\|\hat{\mathbf{w}}_n\|}.
\tag{3}
$$

All the averaged proxies are also L_2-normalized vectors, so that the meta-classifier essentially becomes a cosine-similarity based classification model. After constructing the meta-classifier, the L_2-normalized query feature $\mathbf{z}_q$ is fed into it for evaluation. The prediction $\mathbf{p} = \mathbf{z}_q \mathbf{W}^\top$ is the combination of n classification scores $p_n = \mathbf{z}_q \mathbf{w}_n^\top$ of each category. To further increase stability and robustness when dealing with a large number of categories, we adopt a learnable temperature α in our meta-task loss as Qi, Brown, and Lowe, where α is updated by back-propagation during training. The meta-task loss $\mathcal{L}$ can be defined as follow:

$$
\mathcal{L} = -\log \frac{\exp\left(\alpha \mathbf{z}_q \mathbf{w}_{Y_q}^\top\right)}{\sum_{n=1}^N \exp\left(\alpha \mathbf{z}_q \mathbf{w}_n^\top\right)}.
\tag{4}
$$

Organized Auxiliary Task Co-training
In order to accelerate the convergence of training and get better generalization ability, the meta-learning network in DTN is jointly trained with an auxiliary task. The auxiliary task is a conventional classification for all N' categories in $\mathcal{D}_{train}$. It shares the same feature extractor F with the meta-task branch. Different from the meta-classifier $\mathbf{W} \in \mathbb{R}^{N \times C}$, which consists of the averaged proxies, the auxiliary classifier $\mathbf{W}' \in \mathbb{R}^{N' \times C}$ after the feature extractor F are randomly initialized and updated via back-propagation. The mini-batch $\{(\mathbf{X}_i, Y_i)\}$ is randomly sampled from the training set $\mathcal{D}_{train}$, where $i \in [1, 2, ..., B]$, and B is the batch size. The auxiliary task loss $\mathcal{L}'$ has the same form as the meta-task loss $\mathcal{L}$:

$$
\mathcal{L}' = -\log \frac{\exp\left(\alpha' \mathbf{z}_i \mathbf{w}_{Y_i}'^\top\right)}{\sum_{n=1}^{N'} \exp\left(\alpha' \mathbf{z}_i \mathbf{w}_n'^\top\right)},
\tag{5}
$$

where $\mathbf{z}_i$ is one of the training features in the mini-batch, $\mathbf{w}_n' \in \mathbb{R}^C$ is the n-th row of $\mathbf{W}'$, and α' is learnable.

In TADAM (Oreshkin, Rodríguez López, and Lacoste 2018), the auxiliary task is sampled with a probability that is annealed exponentially. We observe some positive effects

Table 1: Few-shot images classification accuracies on *mini*ImageNet. '-': not reported.

Methods	Ref.	Backbone	5-way 1-shot	5-way 5-shot
Matching Network (Vinyals et al.)	NeurIPS'16	64-64-64-64	$43.56\% \pm 0.84\%$	$55.31\% \pm 0.73\%$
Meta-Learn LSTM (Ravi and Larochelle)	ICLR'17	64-64-64-64	$43.44\% \pm 0.77\%$	$60.60\% \pm 0.71\%$
MAML (Finn, Abbeel, and Levine)	ICML'17	32-32-32-32	$48.70\% \pm 1.84\%$	$63.11\% \pm 0.92\%$
Prototypical Network (Snell, Swersky, and Zemel)	NeurIPS'17	64-64-64-64	$49.42\% \pm 0.78\%$	$68.20\% \pm 0.66\%$
Relation Network (Sung et al.)	CVPR'18	64-96-128-256	$50.44\% \pm 0.82\%$	$65.32\% \pm 0.70\%$
MT-net (Lee and Choi)	ICML'18	64-64-64-64	$51.70\% \pm 1.84\%$	-
MetaGAN◆ (Zhang et al.)	NeurIPS'18	64-96-128-256	$52.71\% \pm 0.64\%$	$68.63\% \pm 0.67\%$
Qiao et al.	CVPR'18	64-64-64-64	$54.53\% \pm 0.40\%$	$67.87\% \pm 0.20\%$
Gidaris and Komodakis	CVPR'18	64-64-64-64	$56.20\% \pm 0.86\%$	$73.00\% \pm 0.64\%$
DTN (Ours)◆		64-64-64-64	$\mathbf{57.89\%} \pm 0.84\%$	$\mathbf{73.28\%} \pm 0.65\%$
Gidaris and Komodakis	CVPR'18	ResNet-12	$55.45\% \pm 0.89\%$	$70.13\% \pm 0.68\%$
adaResnet (Munkhdalai et al.)	ICML'18	ResNet-12	$56.88\% \pm 0.62\%$	$71.94\% \pm 0.57\%$
TADAM (Oreshkin, Rodríguez López, and Lacoste)	NeurIPS'18	ResNet-12	$58.50\% \pm 0.30\%$	$76.70\% \pm 0.30\%$
Qiao et al.	CVPR'18	WRN-28-10	$59.60\% \pm 0.41\%$	$73.74\% \pm 0.19\%$
STANet (Yan, Zhang, and He)	AAAI'19	ResNet-12	$58.35\% \pm 0.57\%$	$71.07\% \pm 0.39\%$
TPN (Liu et al.)	ICLR'19	ResNet-12	59.46%	75.65%
LEO (Rusu et al.)	ICLR'19	WRN-28-10	$61.76\% \pm 0.08\%$	$77.59\% \pm 0.12\%$
Δ-encoder (Schwartz et al.)◆	NeurIPS'18	VGG-16	59.9%	69.7%
IDeMe-Net (Zitian Chen)◆	CVPR'19	ResNet-18♠	$59.14\% \pm 0.86\%$	$74.63\% \pm 0.74\%$
SalNet Intra-class Hal. (Zhang, Zhang, and Koniusz)◆	CVPR'19	ResNet-101♣	$62.22\% \pm 0.87\%$	$77.95\% \pm 0.65\%$
Deep DTN (Ours)◆		ResNet-12	$\mathbf{63.45\%} \pm 0.86\%$	$\mathbf{77.91\%} \pm 0.62\%$

◆ Generation based approaches   ♠ Using a deformation sub-network   ♣ Using a saliency network pre-trained on MSRA-B

from this training strategy compared with naïve meta-training and multi-stage training in our DTN.

However, the inadequacy of this approach is: the randomness in both the *frequency* and the *order* of the two tasks affects the final result to some extent, and the distribution of auxiliary tasks are *unpredictable* rather than annealed exponentially, especially when the number of training epochs is not very large. Another problem brought by the randomness is that it is hard to determine the training schedule, e.g., the learning rate, the number of training epochs, etc., since the permutation of auxiliary tasks and meta-tasks varies according to the random seed. We empirically find that the stochastic auxiliary task co-training strategy used in TADAM results in a large fluctuation in the meta-classification accuracy (over 4%, see Table 3 for details) when using different random seeds. This randomness makes the choice of hyperparameters as well as the training schedule more difficult.

Therefore, we propose the OAT (Organized Auxiliary task co-Training) strategy, which organizes auxiliary tasks and meta-tasks in a more orderly and more reasonable manner. More specifically, there are two kinds of training epochs: the auxiliary training epoch $\mathcal{A}$ and the meta-training epoch $\mathcal{M}$. We select T training epochs to form one training unit $\mathcal{U}$, the i-th training unit $\mathcal{U}_i$ has γ_i meta-training epochs, and $(T - \gamma_i)$ auxiliary training epochs. The array of γ_i is denoted as $\boldsymbol{\gamma} = \{\gamma_i\}_{i=1}^{L}$, where L is the total number of training units. Then the total number of training epochs is $T \times L$, and the whole training sequence $\mathcal{S}$ can be expressed as follow:

$$\mathcal{S} = \sum_{i=1}^{L} \Big((T - \gamma_i)\, \mathcal{A} + \gamma_i \mathcal{M} \Big). \quad (6)$$

By changing T and $\boldsymbol{\gamma}$, we can obtain different training sequences arranged in different frequency and order, which is proven to be more manageable and effective compared with

Table 2: The **5**-way **1**-shot/**5**-way **5**-shot images classification accuracies on CIFAR100 and CUB. '-': not reported.

Methods	CIFAR100	CUB
Nearest neighbor	56.1%/68.3%	52.4%/66.0%
Meta-Learn LSTM (Ravi and Larochelle)	-	40.4%/49.7%
Matching Network (Vinyals et al.)	50.5%/60.3%	49.3%/59.3%
MAML (Finn, Abbeel, and Levine)	49.3%/58.3%	38.4%/59.1%
Δ-encoder (Schwartz et al.)	66.7%/79.8%	69.8%/82.6%
Deep DTN (Ours)	**71.5%/82.8%**	**72.0%/85.1%**

the training strategy used in TADAM. Intuitively, we would like to gradually add harder few-shot classification tasks into a series of simpler auxiliary classification tasks. Therefore the setting of T and $\boldsymbol{\gamma}$ is quite simple and straightforward. We choose $T = 5$ and $\boldsymbol{\gamma} = [0, 0, 1, 1, 2, 2]$ for training DTN, though a more careful scheduling may achieve better performance. Therefore the whole training sequence $\mathcal{S}$ is organized as follow:

$$\mathcal{S} = 5\mathcal{A} \times 2 + (4\mathcal{A} + 1\mathcal{M}) \times 2 + (3\mathcal{A} + 2\mathcal{M}) \times 2. \quad (7)$$

Initially, the auxiliary tasks could be considered as a simpler curriculum(Bengio et al. 2009), later they bring regularization effects to meta-tasks. Ablation studies show that compared with the training strategy used in TADAM, DTN trained by OAT obtains better and more robust results with a faster convergence speed.

Experiments

Implementation Details

Dataset. The proposed method is evaluated on multiple datasets: *mini*ImageNet, CIFAR100 and CUB. The *mini*ImageNet dataset has been widely used by few-shot learning since it is firstly proposed by Vinyals et al.. There are 64, 16 and 20 classes for training, validation, and testing
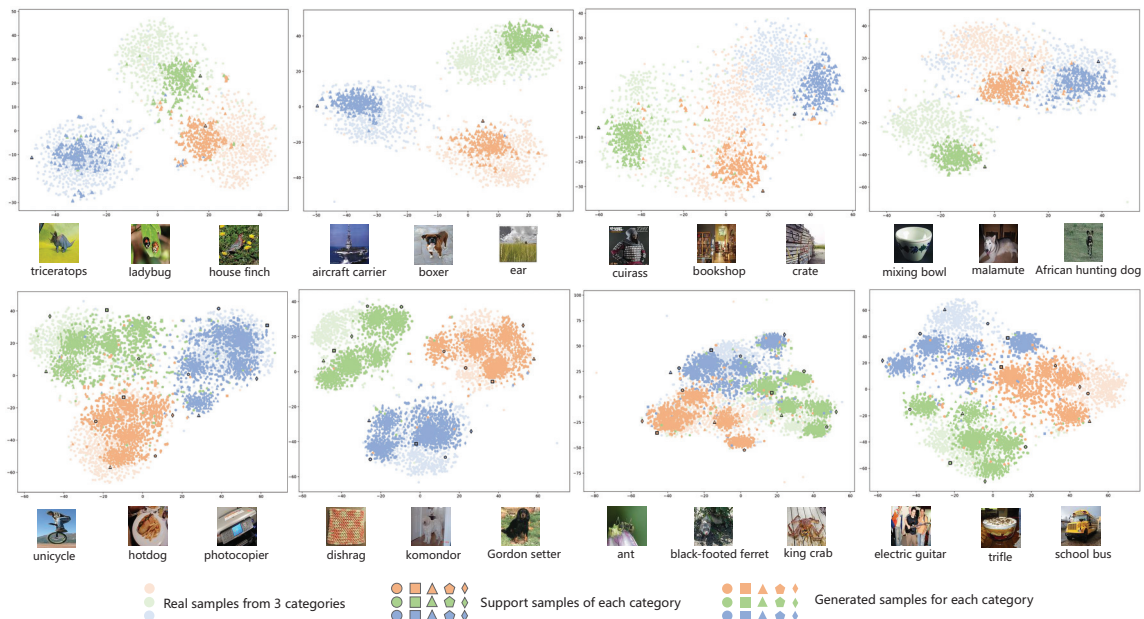
Figure 3: Visualization of generated samples, support samples, and real samples. The light dots indicate real samples, the shapes (circle, square, triangle, pentagon and diamond) with black border indicate support samples which are also real samples, and the shapes without border indicate generated samples. There are 64 generated samples for each support sample. The top row shows the results of 3-way 1-shot learning and the bottom row shows the results of 3-way 5-shot learning. The data in the left two columns are from the training set and the data in the right two columns are from the testing set.

respectively. The hyper-parameters are optimized on the validation set. After that, it will be merged into the training set for the final results. The CIFAR100 dataset (Krizhevsky and Hinton 2009) contains 6000 images of 100 classes. We use 64, 16, and 20 classes for training, validation, and testing, respectively. The CUB dataset (Wah et al. 2011) is a fine-grained dataset from 200 categories of birds. It is divided into training, validation, and testing sets with 100, 50, and 50 categories respectively. The splits of CIFAR100 and CUB follow Schwartz et al..

Architectures. The feature extractor for DTN is a CNN with 4 convolutional modules. Each module contains a 3×3 convolutional layer with 64 channels followed by a batch normalization(BN) layer, a ReLU non-linearity layer, and a 2×2 max-pooling layer. The structure of feature extractor is the same as those in former methods, e.g., Vinyals et al.; Snell, Swersky, and Zemel for fair comparisons. Many other works also use deeper networks for feature extraction to achieve better accuracy, e.g., Munkhdalai et al.; Oreshkin, Rodríguez López, and Lacoste. To make a comparison with them, we also implement our algorithm with ResNet-12 architecture(He et al. 2015). The output of the feature extractor is a 1024-dimensional vector. The mapping function ϕ_1 in the feature generator G is a fully-connected (FC) layer with 2048 units followed by a leaky ReLU activation $(\max(x, 0.2x))$ layer, and a dropout layer with 30% dropout rate. The mapping function ϕ_2 has the same settings with ϕ_1 except that the number of units of the FC layer is 1024.

Results

Quantitative Results. Table 1 provides comparative results on the *mini*ImageNet dataset. All these results are reported with 95% confidence intervals following the setting in Vinyals et al.. Under the 4-CONV feature extractor setting, our approach significantly outperforms the previous state-of-the-art works, especially in the 5-way 1-shot task. As for the comparisons with models using deep feature extractor, deep DTN also surpasses other alternatives in the 5-way 1-shot scenario and achieves very competitive results under the 5-way 5-shot setting. The results confirm that our feature generation method is extremely useful to address the problem of learning with scarce data, i.e., the 5-way 1-shot scenario. DTN is also one of the simplest and lightweight feature generation methods which learns to enrich intra-class diversity, and does not rely on any extra information from other datasets, such as the salient object information in Zhang, Zhang, and Koniusz.

Table 2 shows that DTN also gets large improvements on the CIFAR100 and CUB datasets compared with existing state-of-the-arts in both 5-way 1-shot task and 5-way 5-shot task, which confirms DTN is generally useful for different few-shot learning scenarios.

Visualization Results. To better understand the results, Fig. 3 shows tSNE (Maaten and Hinton 2008) visualizations of generated samples, support samples and real samples. It can be seen that our method can greatly enrich the diversity of an unseen class with only a single or a few support examples given. Most of the generated samples fit the distribution of real samples, which means that the category information

Table 3: Ablation studies on the fluctuation of the results obtained by AT and OAT on *mini*ImageNet. AT: auxiliary task co-training strategy used in TADAM. OAT: organized auxiliary task co-training. The representation of the training sequence follows the notation introduced in the previous section, and the training sequence of AT is completely determined by the random seed. The results show that compared with the AT strategy(over 4% fluctuation), the model trained by OAT (less than 1.5% fluctuation) obtains better and more robust results.

Random seed	Training sequence of AT (number of total training epochs = 30)	Results of AT		Training sequence of OAT (number of total training epochs = 30)	Results of OAT	
		5-way 1-shot	5-way 5-shot		5-way 1-shot	5-way 5-shot
Seed #1	$13\mathcal{A}$-$1\mathcal{M}$-$2\mathcal{A}$-$1\mathcal{M}$-$1\mathcal{A}$-$1\mathcal{M}$-$2\mathcal{A}$-$1\mathcal{M}$-$2\mathcal{A}$-$1\mathcal{M}$-$2\mathcal{A}$-$1\mathcal{M}$-$5\mathcal{A}$	58.60%	72.78%		62.78%	77.58%
Seed #2	$11\mathcal{A}$-$1\mathcal{M}$-$7\mathcal{A}$-$1\mathcal{M}$-$7\mathcal{A}$-$3\mathcal{M}$	60.71%	74.10%		62.19%	76.82%
Seed #3	$13\mathcal{A}$-$1\mathcal{M}$-$9\mathcal{A}$-$1\mathcal{M}$-$4\mathcal{A}$-$2\mathcal{M}$	61.61%	75.52%	$10\mathcal{A}$-$4\mathcal{A}$-$1\mathcal{M}$-$4\mathcal{A}$-$1\mathcal{M}$-$3\mathcal{A}$-$2\mathcal{M}$-$3\mathcal{A}$-$2\mathcal{M}$	63.45%	77.91%
Seed #4	$18\mathcal{A}$-$1\mathcal{M}$-$2\mathcal{A}$-$1\mathcal{M}$-$1\mathcal{A}$-$1\mathcal{M}$-$2\mathcal{A}$-$2\mathcal{M}$-$1\mathcal{A}$-$1\mathcal{M}$	60.97%	75.77%		62.48%	77.22%
Seed #5	$14\mathcal{A}$-$1\mathcal{M}$-$4\mathcal{A}$-$1\mathcal{M}$-$3\mathcal{A}$-$2\mathcal{M}$-$3\mathcal{A}$-$2\mathcal{M}$	62.37%	77.45%		63.17%	77.47%

Table 4: Ablation studies for different feature generators on *mini*ImageNet.

	Methods	5-way 1-shot	5-way 5-shot
1	Gaussian noise generator	$60.14\% \pm 1.24\%$	$75.57\% \pm 0.96\%$
2	Δ-encoder[†]	$60.38\% \pm 1.12\%$	$73.44\% \pm 0.92\%$
3	DTN w/ two-stage training	$61.95\% \pm 0.85\%$	$76.52\% \pm 0.65\%$
4	**DTN w/ OAT (Ours)**	$\mathbf{63.45\% \pm 0.86\%}$	$\mathbf{77.91\% \pm 0.62\%}$

[†] Our reimplementation, which outperforms the original.

Table 5: Ablation studies for different training strategies on *mini*ImageNet. AT: auxiliary task co-training strategy used in TADAM . OAT: organized auxiliary task co-training.

	Methods	5-way 1-shot	5-way 5-shot
1	DTN w/ naïve meta-training	$59.81\% \pm 1.36\%$	$74.97\% \pm 1.01\%$
2	DTN w/ two-stage training	$61.95\% \pm 0.85\%$	$76.52\% \pm 0.65\%$
3	DTN w/ AT	$62.65\% \pm 0.86\%$	$77.12\% \pm 0.65\%$
4	**DTN w/ OAT (Ours)**	$\mathbf{63.45\% \pm 0.86\%}$	$\mathbf{77.91\% \pm 0.62\%}$

Table 6: Ablation studies for DTN trained with different number of generated features on *mini*ImageNet. Numbers in the "( )" are difference in meta-classification accuracies compared with the result with 64 generated features.

Number of generated features	5-way 1-shot	5-way 5-shot
2	$60.71\% (-2.74\%)$	$75.56\% (-2.35\%)$
4	$61.18\% (-2.27\%)$	$76.17\% (-1.74\%)$
16	$61.87\% (-1.58\%)$	$76.76\% (-1.15\%)$
32	$62.58\% (-0.87\%)$	$77.14\% (-0.77\%)$
64	$\mathbf{63.45\%} (-0.00\%)$	$\mathbf{77.91\%} (-0.00\%)$
96	$63.09\% (-0.36\%)$	$77.45\% (-0.46\%)$
128	$63.26\% (-0.19\%)$	$77.99\% (+0.08\%)$

of each support sample is well preserved by the generated sample, and they are close to the center of the real distribution even when the support sample lies on the edge. From the diagrams of 3-way 5-shot learning, it can be seen that generated features from 5 support samples can cover the major distribution of the real samples, which facilitates to build a more robust classifier for unseen classes.

Ablation Study

In this section, we study the impact of the feature generator, the training strategy and the number of generated features. We conduct the following ablation studies using models with deep feature extractor on *mini*ImageNet. All the results are summarized in Table 3, Table 4, Table 5 and Table 6.

First, we make a comparison between different feature generators. For the sake of fairness, we use exactly the same meta-classifier (cosine-similarity based classifiers) and the same training strategy (two-stage training), only the feature generators are different. All models are trained until convergence. Experiments show that diversity transfer generator outperforms Gaussian noise seeded generator (by 1.81% in 5-way 1-shot, 0.95% in 5-way 5-shot. Table 4, Row 1 and Row 3) and Δ-encoder (by 1.57% in 5-way 1-shot, 3.08% in 5-way 5-shot. Table 4, Row 2 and Row 3).

Second, we study the effects of different training strategies. Obviously, OAT (Table 5, Row 4) surpasses the naïve meta-

training (Table 5, Row 1) and two-stage training (Table 5, Row 2). As mentioned before, a large fluctuation was observed (e.g., from 58.60% to 62.37% for 5-way 1-shot, from 72.78% to 77.45% for 5-way 5-shot, see Table 3 for details) in the meta-classification accuracy if the DTN is trained with auxiliary task sampled via a probability in 30 epochs. The result becomes better and more stable if we increase the total number of training epochs to 60 (Table 5, Row 3), but this is still worse than the result obtained via DTN trained with OAT in only 30 epochs (Table 5, Row 4). A comparison in the result's fluctuation between two training strategies is detailed in Table 3.

Finally, in Table 6 we study the impact on the number of generated features. The results gradually become better as the number of generated features increases. No improvement was observed when the number of generated features exceeds 64. We attribute this to the fact that 64 generated features have been well fitted to the real sample distribution.

Conclusion and Future Work

In this work, we propose a novel generative model, Diversity Transfer Network (DTN), for few-shot image recognition. It learns transferable diversity from the known categories and augments the unseen category with the sample generation. DTN achieves competitive performance on three benchmarks. We believe that the proposed generative method can be utilized in various problems challenged by the scarcity of supervision information, e.g., semi-supervised learning, active learning and imitation learning. These interesting research directions will be explored in the future.

Acknowledgment

References

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *International Conference on Machine Learning (ICML)*.

Bloom, P. 2000. *How children learn the meanings of words*, volume 377. Citeseer.

Dixit, M.; Kwitt, R.; Niethammer, M.; and Vasconcelos, N. 2017. Aga: Attribute-guided augmentation. In *Computer Vision and Pattern Recognition (CVPR)*.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning (ICML)*.

Franceschi, L.; Frasconi, P.; Salzo, S.; Grazzi, R.; and Pontil, M. 2018. Bilevel programming for hyperparameter optimization and meta-learning. In *International conference on machine learning (ICML)*.

Gidaris, S., and Komodakis, N. 2018. Dynamic few-shot visual learning without forgetting. In *Computer Vision and Pattern Recognition (CVPR)*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Neural Information Processing Systems (NIPS)*.

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

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 770–778.

Krizhevsky, A., and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical report, Citeseer.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Neural Information Processing Systems (NIPS)*.

Lee, Y., and Choi, S. 2018. Gradient-based meta-learning with learned layerwise metric and subspace. In *International conference on machine learning (ICML)*.

Liu, B.; Wang, X.; Dixit, M.; Kwitt, R.; and Vasconcelos, N. 2018. Feature space transfer for data augmentation. In *Computer Vision and Pattern Recognition (CVPR)*.

Liu, Y.; Lee, J.; Park, M.; Kim, S.; Yang, E.; Hwang, S. J.; and Yang, Y. 2019. Learning to propagate labels: Transductive propagation network for few-shot learning. In *International Conference on Learning Representations (ICLR)*.

Maaten, L. v. d., and Hinton, G. 2008. Visualizing data using t-sne. *Journal of machine learning research* 9(Nov):2579–2605.

Movshovitz-Attias, Y.; Toshev, A.; Leung, T. K.; Ioffe, S.; and Singh, S. 2017. No fuss distance metric learning using proxies. In *International Conference on Computer Vision (ICCV)*.

Munkhdalai, T., and Yu, H. 2017. Meta networks. In *International conference on machine learning (ICML)*.

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

Oreshkin, B.; Rodríguez López, P.; and Lacoste, A. 2018. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Neural Information Processing Systems (NIPS)*.

Qi, H.; Brown, M.; and Lowe, D. G. 2018. Low-shot learning with imprinted weights. In *Computer Vision and Pattern Recognition (CVPR)*.

Qiao, S.; Liu, C.; Shen, W.; and Yuille, A. L. 2018. Few-shot image recognition by predicting parameters from activations. In *Computer Vision and Pattern Recognition (CVPR)*.

Ravi, S., and Larochelle, H. 2017. Optimization as a model for few-shot learning. In *International Conference on Learning Representations (ICLR)*.

Rusu, A. A.; Rao, D.; Sygnowski, J.; Vinyals, O.; Pascanu, R.; Osindero, S.; and Hadsell, R. 2019. Meta-learning with latent embedding optimization. In *International Conference on Learning Representations (ICLR)*.

Schwartz, E.; Karlinsky, L.; Shtok, J.; Harary, S.; Marder, M.; Kumar, A.; Feris, R.; Giryes, R.; and Bronstein, A. 2018. Delta-encoder: an effective sample synthesis method for few-shot object recognition. In *Neural Information Processing Systems (NIPS)*.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *CoRR* abs/1409.1556.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. In *Neural Information Processing Systems (NIPS)*.

Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P. H.; and Hospedales, T. M. 2018. Learning to compare: Relation network for few-shot learning. In *Computer Vision and Pattern Recognition (CVPR)*.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.; et al. 2016. Matching networks for one shot learning. In *Neural Information Processing Systems (NIPS)*.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The caltech-ucsd birds-200-2011 dataset.

Wang, Y.-X.; Girshick, R.; Hebert, M.; and Hariharan, B. 2018. Low-shot learning from imaginary data. In *Computer Vision and Pattern Recognition (CVPR)*.

Yan, S.; Zhang, S.; and He, X. 2019. A dual attention network with semantic embedding for few-shot learning. In *Association for the Advance of Artificial Intelligence (AAAI)*.

Yuille, A. L. 2011. Towards a theory of compositional learning and encoding of objects. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 1448–1455. IEEE.

Zhang, R.; Che, T.; Ghahramani, Z.; Bengio, Y.; and Song, Y. 2018. Metagan: An adversarial approach to few-shot learning. In *Neural Information Processing Systems (NIPS)*.

Zhang, H.; Zhang, J.; and Koniusz, P. 2019. Few-shot learning via saliency-guided hallucination of samples. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *International Conference on Computer Vision (ICCV)*.

Zitian Chen, Yanwei Fu, Y.-X. W. L. M. W. L. M. H. 2019. Image deformation meta-networks for one-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.