Semantic Feature Augmentation in Few-shot Learning

Zitian Chen[1],Yanwei Fu[1]*,[⋆]Yinda Zhang[2],Yu-Gang Jiang[1],
Xiangyang Xue[1],Leonid Sigal[3]

[1]Fudan University,China [2]Princeton University [3]University of British Columbia
{chenzt15,yanweifu,ygj,xyxue}@fudan.edu.cn,
yindaz@cs.princeton.edu, lsigal@cs.ubc.ca

Abstract. A fundamental problem with few-shot learning is the scarcity of data in training. A natural solution to alleviate this scarcity is to augment the existing images for each training class. However, directly augmenting samples in image space may not necessarily, nor sufficiently, explore the intra-class variation. To this end, we propose to directly synthesize instance features by leveraging the semantics of each class. Essentially, a novel auto-encoder network dual TriNet, is proposed for feature augmentation. The encoder TriNet projects multi-layer visual features of deep CNNs into the semantic space. In this space, data augmentation is induced, and the augmented instance representation is projected back into the image feature spaces by the decoder TriNet. Two data argumentation strategies in the semantic space are explored; notably these seemingly simple augmentations in semantic space result in complex augmented feature distributions in the image feature space, resulting in substantially better performance. The code and models of our paper will be published on: `https://github.com/tankche1/Semantic-Feature-Augmentation-in-Few-shot-Learning`.

1 Introduction

The success of recent machine learning (especially deep learning) greatly relies on the training process that operates on hundreds or thousands of labelled training instances of each class. However, in practice, it might be extremely expensive or infeasible to obtain many labelled samples, *e.g.* for rare objects or objects that may be hard to observe. In contrast, humans can recognize an object category easily after seeing only few training examples [1]. Inspired by such an ability, few-shot learning aims to build classifiers from few, or even a single, examples.

The major obstacle of learning good classifiers in a few-shot learning setting is the lack of training data. Thus a natural recipe for few-shot learning is to first augment the data. A number approaches for data augmentation have been explored. The dominant approach, adopted by previous work, is to bring in more images [2] for each category as training data. These additional augmented

[⋆] *Yanwei Fu is the corresponding author training images could be borrowed from unlabelled data [3] or other relevant categories [4,5,6,7] in an unsupervised or semi-supervised fashion. However, the augmented data that comes from related classes is often semantically noisy and can result in *negative transfer* which leads to reduced (instead of improved) performance. On the other hand, synthetic images rendered from virtual examples [8,9,10,11,12,13] are semantically correct but require careful domain adaptation to transfer the knowledge to the real image domain. To avoid the difficulty of generating the synthesized images directly, it is thus desirable to augment the samples in the feature space itself. For example, the state-of-the-art deep Convolutional Neural Networks (CNNs) stack multiple feature layers in a hierarchical structure; we hypothesize that feature augmentation can, in this case, be done in feature spaces produced by CNN layers.

Despite clear benefits, feature augmentation techniques have been relatively little explored. The few examples include [12,13,14]. Notably, [12] and [13] employed the feature patches (*e.g.* HOG) of the object parts are combined them to synthesize new feature representations. Dixit *et al.* [14], for the first time, considered the attributes-guided augmentation to synthesize sample features. Their work, however, utilizes and relies on the pre-defined semantic attributes.

A straightforward approach to augment image feature is to add random vector noise to the feature of each single training image. However, such simple augmentation procedure may not substantially inform/improve the decision boundary. Our key idea is to bring in the additional semantic knowledge, *e.g.* encapsulated by the semantic space pre-trained, using linguistic model such as google word2vec [15], from a huge repository of linguistic corpora. In such a semantic domain, similar concepts are supposed to have similar semantic feature, such that the overall space demonstrates semantic continuity over natural meaning, which is ideal for feature augmentation.

To leverage such semantic space, we propose a dual TriNet architecture ($g\left(\mathbf{x}\right) = g_{Dec} \circ g_{Enc}\left(\mathbf{x}\right)$) to learn the transformation between the image features at multiple layers and the semantic space. The dual TriNet is paired with the 18-layer residual net (ResNet-18) [16]; it has encoder TriNet ($g_{Enc}(\mathbf{x})$) and the decoder TriNet ($g_{Dec}(\mathbf{x})$). Specifically, given one training instance, we can use the ResNet-18 to extract the features at different layers. The $g_{Enc}(\mathbf{x})$ efficiently maps these features into the semantic space. In the semantic space, the projected instance features can be corrupted by adding Gaussian noise, or replaced by its nearest semantic word vectors. We assume that slight changes of feature values in the semantic space will allow us to maintain semantic information while spanning the potential class variability. The decoder TriNet ($g_{Dec}(\mathbf{x})$) is then adopted to map the semantic instance features back to multi-layer (ResNet-18) feature space. It is worth noting that Gaussian augmentations/perturbations in the semantic space ultimately result in highly non-Gaussian augmentations in the original feature space. This is the core benefit of the semantic space augmentation. By using three classical supervised classifiers, we show that the augmented instance features can boost the performance of few-shot classification.

Contributions. The contributions are several folds. First, we propose a simply and yet elegant deep learning architecture: ResNet-18+dual TriNet with an efficient end-to-end training for few-shot classification. Second, the proposed dual TriNet can effectively augment visual features produced by multiple layers of ResNet-18. Third, and interestingly, we show that we can utilize semantic spaces of various types, including semantic attribute space, semantic word vector space, or even subspace defined by the semantic relationship of classes. Finally, extensive experiments on four datasets have validated the efficacy of proposed approach in solving the few-shot image recognition tasks.

2 Related work

2.1 Few-Shot Learning

Few-shot learning is inspired by capabilities of human learning [17,18] of being able to learn about new concepts from very few examples. Generalizing and being able to recognize new classes with only one or few examples [19] is beyond the capability of typical machine learning algorithms, which often rely on hundreds or thousands of training examples. Hence, direct learning strategies (including gradient descent) are often ineffective and one needs to resort to transfer learning by leveraging source/auxiliary data for zero-shot recognition. In terms of one-shot learning, there are two categories of approaches:

Direct supervised learning-based approaches, do not use auxiliary data; and they can directly learn one-shot classifier via instance-based learning (such as K-nearest neighbor), non-parameteric methods [20,21,22], deep generative models [23,24], or Bayesian auto-encoders [25]. Compared with our work, these methods only employ a rich class of generative models to explain the observed data, rather than directly augmenting instance features as we propose.

Transfer learning-based approaches, are explored via the paradigm of learning to learn [1] or meta-learning [26]. Specifically, these approaches employ the knowledge from auxiliary data to recognize new categories with few examples by either sharing features [19,27,28,29,30,31], semantic attributes [32,33,34], or contextual information [35]. Recently, the ideas of learning metric spaces from source data to support one-shot learning were quite extensively explored, such as matching networks [36] and prototypical networks [37]. Generally, these approaches can be roughly categorized as meta-learning algorithms (including MAML [38], Meta-SGD [39], DEML+Meta-SGD [40],META-LEARN LSTM [41], Meta-Net [42]) and Metric-learning algorithms (including Matching Nets [36], PROTO-NET [37], RELATION NET [43] and MACO [44]). The [45] maintained external memory for continuous learning. MAML[46] can learn good initial neural network weights which can be easily fine-tuned for unseen categories. With respect to these works, our framework is orthogonal but potentially useful – it is useful to augment instance features of novel classes before applying such methods.

2.2 Augmenting training instances

The standard augmentation techniques can be directly applied in image domain, such as flipping, rotating, adding noise and randomly cropping images [2,47,48].

Recently, more advanced data augmentation techniques have been studied to train supervised classifiers. In particular, augmented training data can also be employed to alleviate the problem of instances scarcity and thus avoid overfitting in one-shot/few-shot learning settings. Previous approaches can be categorized into six classes of methods: (1) Learning one-shot models by utilizing the manifold information of large amount of unlabelled data in a semi-supervised or transductive setting [3]; (2) Adaptively learning the one-shot classifiers from off-shelf trained models [4,5,6]; (3) Borrowing examples from relevant categories [7] or semantic vocabularies [49,50] to augment the training set; (4) Synthesizing new labelled training data by rendering virtual examples [8,9,10,11,51] or composing synthesized representations [12,13,52,53,54,55] or distorting existing training examples [2]; (5) Generating new examples using Generative Adversarial Networks (GANs) [56,57,58,59,60,61,62,63]; (6) Attribute-guided augmentation (AGA) [14] to synthesize samples at desired values or strength.

Despite breadth of research, previous methods may suffer from several problems: (1) semi-supervised algorithms rely on the manifold assumption, which however can not be effectively validated in practice. (2) transfer learning may suffer from the *negative transfer* when the off-shell models or relevant categories are very different from one-shot classes; (3) rendering, composing new virtual examples or distorting existing training examples may require domain expertise; (4) GANs based approaches mostly focused on learning good generators to synthesize "realistic" images to "cheat" the discriminators. Synthesized images may not necessarily preserve the discriminative information. This is in contrast to our network structure, where the discriminative instances are directly synthesized in visual feature domain. The AGA [14] mainly employed the attributes of 3D depth or pose information for augmentation; in contrast, our methods can additionally utilize semantic information to augment data. Additionally, the proposed dual TriNet networks can effectively augment multi-layer features.

2.3 Embedding Network structures

Learning of visual-semantic embeddings has been explored in various ways, such as with neural networks, *e.g.*, siamese network [64,65], discriminative methods (*e.g.*, Support Vector Regressors (SVR) [32,66,67]), metric learning methods [36,68,69], or kernel embedding methods [27,70]. One of the most common embedding approaches is to project visual features and semantic entities into a common *new* space. However, when dealing with the feature space of different layers in deep CNNs, previous methods have to learn an individual visual semantic embedding for each layer. In contrast, the proposed Dual TriNet can effectively learn a single visual-semantic embedding for multi-layer feature spaces.

Ladder Networks [71] utilized the lateral connections as auto-encoders for semi-supervised learning tasks. The [72] fused different intermediate layers of different networks to improve the image classification performance. Deep Layer Aggregation [73] aggregated the layers and blocks across a network to better fuse the information across layers. Rather than learn a specific aggregation node to merge different layers, our dual TriNet directly transforms, rescales and concatenates the features of different layers in an encoder-decoder structure.

3 Dual TriNet Network for Semantic Data Augmentation

3.1 Problem setup

We formulate few-shot learning task in a transfer learning scenario. Assume the source dataset $D_s = \left\{\mathbf{I}_i^s, z_i^s, \mathbf{u}_{z_i}^s\right\}_{i=1}^{N_s}$ of N_s samples. $\mathbf{I}_i^s$ indicates the raw image i. $z_i^s \in \mathcal{C}_s$ is a class label where $\mathcal{C}_s$ is the source class set; $\mathbf{u}_{z_i}^s$ is the semantic vector of the instance i in terms of its class label. The semantic vector $\mathbf{u}_{z_i}^s$ can be either semantic attribute [32], semantic word vector [15] or any subspace inferred by semantic relationship of classes.

We consider the target dataset $D_t = \left\{\mathbf{I}_i^t, z_i^t, \mathbf{u}_{z_i}^t\right\}$ and each class $z_i^t \in \mathcal{C}_t$ ($\mathcal{C}_s \bigcap \mathcal{C}_t = \emptyset$) with the total class label set $\mathcal{C} = \mathcal{C}_s \cup \mathcal{C}_t$. In particular, in few-shot learning setting, each class is only given a small number of training instances. The standard setting for the transfer-based few-shot learning would first employ the source dataset D_s to train an $L-$layer deep neural network $\left\{f_l\left(\mathbf{I}_i\right)\right\}_{l=1}^{L}$, where $f_l\left(\mathbf{I}_i\right)$ is the $l-th$ layer output feature vector. On the target dataset D_t, we use the trained network to extract the $l-th$ level feature $f_l\left(\mathbf{I}_i^t\right)$. These target dataset features are used to train classifiers in supervised manner with few examples and then applied at test time. Generally, different layer features may be used for various one-shot learning tasks. For example, as in [2], the features of fully connected layers can be used for one-shot image classification; and the output features of fully convolutional layers may be preferred for one-shot image segmentation tasks [74,75,76].

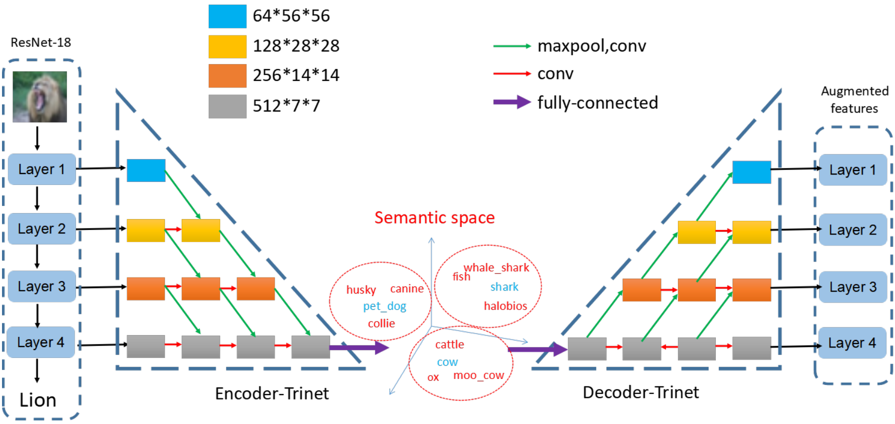


Fig. 1. Overview of our framework. We extract image features by ResNet-18 and augment features by dual TriNet. Encoder TriNet projects features to the semantic space. After augmenting data in semantic space, we use the decoder TriNet to obtain the corresponding augmented features. Both real and augmented data are used to train the classification model. Note that: (1) the small green arrow indicates the maxpooling with 2 × 2, and following by a "conv" layer which is the sequence Conv-BN-ReLU.

3.2 Overview

Objective. We seek to directly augment the training instance features of each target classes. Given one training instance $\mathbf{I}_i^t$ from the target dataset, feature extractor network can output the instance feature $\{f_l(\mathbf{I}_i)\}$ $(l = 1, \cdots, L)$; and the augmentation network $g(\mathbf{x})$ can generate a set of synthesized features $g(\{f_l(\mathbf{I}_i)\})$. Such synthesized features are able to be used as additional training instances for one-shot learning tasks. As illustrated in Fig. 1, we use the ResNet-18 [16] and propose a Dual TriNet network as the feature extractor network and augmentation network respectively. The whole architecture is trained in an end-to-end manner by combining the loss function of both networks,

$$\{\Omega, \Theta\} = \underset{\Omega,\Theta}{\operatorname{argmin}} J_1(\Omega) + \lambda \cdot J_2(\Theta) \tag{1}$$

where $J_1(\Omega)$ and $J_2(\Theta)$ are the loss functions for ResNet-18 [16] and dual TriNet network respectively; Ω and Θ represent corresponding parameters. Cross Entropy loss is used for $J_1(\Omega)$ as in [16]. Eq.(1) is optimized using source dataset.

Feature extractor network. We train from scratch ResNet-18 [16] to convert the raw images into image feature vectors. ResNet-18 has 4 sequential residual layers, *i.e.*, layer1, layer2, layer3 and layer4 as illustrated in Fig. 1. Each residual layer outputs corresponding feature map $f_l(\mathbf{I}_i) \quad l = 1, \ldots 4$. If we take each feature map as one image feature space, ResNet-18 actually learns the Multi-level Image Feature (M-IF) representation.

Augmentation network. We propose an encoder-decoder TriNet structure – dual TriNet ($g(\mathbf{x}) = g_{Dec} \circ g_{Enc}(\mathbf{x})$). As illustrated in Fig. 1, our dual TriNet can be divided into encoder-TriNet $g_{Enc}(\mathbf{x})$ and decoder-TriNet sub-network $g_{Dec}(\mathbf{x})$. The encoder-TriNet maps visual feature space to a semantic space. This is where augmentation takes place. The decoder-TriNet projects the augmented semantic space representation back to the feature space. Since ResNet-18 has four layers, the visual feature spaces produced by different layers can use the same encoder-decoder TriNet for data augmentation.

3.3 Dual TriNet Network

The dual TriNet is paired with ResNet-18. Feature representations obtained from different layers of such a deep CNNs architecture, are hierarchical, going from *general* (bottom layers) to more *specific* (top layers) [77]. For instance, the features produced by the first few layers are similar to Gabor filters [48] and thus agnostic to the tasks; in contrast, the high-level layers are specific to a particular task, *e.g.*, image classification. The feature spaces produced by ResNet-18 have different levels of abstract semantic information. Thus a natural question is whether we can augment features at different layers? Directly learning an encoder-decoder for each layer will not fully exploit the relationship of different layers, and thus may not effectively learn the mapping between feature spaces and semantic space. To this end we propose the dual TriNet network.

Dual TriNet learns the mapping between the Multi-level Image Feature (M-IF) spaces and the Semantic (Sem) space. The semantic space can be either semantic attribute space, or semantic word vector space introduced in Sec. 3.1. Semantic attributes can be pre-defined by human experts [14]. Semantic word vector $\mathbf{u}_{z_i}^s$ is the projection of each vocabulary entity $w_i \in \mathcal{W}$, where vocabulary $\mathcal{W}$ is learned by word2vec [15] on large-scale corpus. Furthermore, the subspace $\mathbf{u}_{z_i}^s$ can be spanned by Singular Value Decomposition (SVD) of the semantic relationship of classes. Specifically, we can use $\left\{\mathbf{u}_{z_i}^s;\mathbf{u}_{z_j}^t\right\}_{z_i\in\mathcal{C}_s,z_j\in\mathcal{C}_t}$ to compute the semantic relationship $\mathbf{M}$ of classes by cosine similarity. We decompose $\mathbf{M} = \mathbf{U\Sigma V}$ by SVD algorithm. The $\mathbf{U}$ is a unitary matrix and defines a new semantic space. Each row of $\mathbf{U}$ is taken as a new semantic vector of one class.

Encoder TriNet is composed of four layers corresponding to each layer of ResNet-18. It aims to learn the function $\hat{\mathbf{u}}_{z_i} = g_{Enc}\left(\{f_l\left(\mathbf{I}_i\right)\}\right)$ to map all layer features $\left(\{f_l\left(\mathbf{I}_i\right)\}\right)$ as close to the semantic vector $\hat{\mathbf{u}}_{z_i}$ of instance i as possible. The structure of subnetwork is designed similar to the tower of Hanoi as shown in Fig. 1. Such a structure can efficiently exploit the differences and complementarity of information encoded in multiple layers. The encoder TriNet is trained to match the four layers of ResNet-18 by merging and combining the outputs of different layers. The decoder TriNet has exactly the same architecture to project the features from semantic space $\hat{\mathbf{u}}_{z_i}$ to the feature space $\hat{f}_l\left(\mathbf{I}_i\right) = g_{Dec}\left(g_{Enc}\left(\{f_l\left(\mathbf{I}_i\right)\}\right)\right)$. We learn TriNet by optimizing the following loss:

$$J_2\left(\Theta\right) = \mathbb{E}_{\mathbf{I}_i\in D_s}\left[\sum_{l=1}^{4}\left(f_l\left(\mathbf{I}_i\right) - \hat{f}_l\left(\mathbf{I}_i\right)\right)^2 + \left(\hat{\mathbf{u}}_{z_i} - \mathbf{u}_{z_i}\right)^2\right] + \lambda P\left(\Theta\right) \tag{2}$$

where Θ indicates the parameter set of dual TriNet network and $P\left(\cdot\right)$ is the L_2-regularization term. The dual TriNet is trained on D_s and tested on D_t.

3.4 Feature Augmentation by Dual TriNet

With the learned dual TriNet, we have two ways of augmenting the features of training instances.

Semantic Gaussian (SG). A natural way of feature augmentation is via sampling instances from Gaussian distribution. Specifically, for the feature set $\{f_l\left(\mathbf{I}_i^t\right)\}(l = 1,\cdots,L)$ extracted by ResNet-18, the encoder TriNet can project the $\{f_l\left(\mathbf{I}_i^t\right)\}$ into $g_{Enc}\left(\{f_l\left(\mathbf{I}_i^t\right)\}\right)$ in the semantic space. In such a space, we assume that though $g_{Enc}\left(\{f_l\left(\mathbf{I}_i^t\right)\}\right)$ is corrupted by a random Gaussian noise, the semantic label does not change. This can be used to augment the data. Specifically, we sample the semantic vector $\mathbf{v}_{z_i}$ from the semantic Gaussian as follows,

$$\mathbf{v}_{z_i}^G \sim \mathcal{N}\left(g_{Enc}\left(\left\{f_l\left(\mathbf{I}_i^t\right)\right\}\right),\sigma\mathbf{E}\right) \tag{3}$$

where $\sigma \in \mathbb{R}$ is the variance of each dimension; $\mathbf{E}$ is the identity matrix; σ controls the deviation of noise added. To make the augmented feature vector $\{f_l\left(\mathbf{I}_i^t\right)\}$ still be representative enough of the class of z_i, we empirically set σ as 15% of the distance between $\{f_l\left(\mathbf{I}_i^t\right)\}$ and its nearest other class' instance

$\left\{ f_l\left(\mathbf{I}_j^t\right)\right\}$ $(z_i \neq z_j)$ as this gives the best performance. The decoder TriNet generates the virtual synthesized sample $g_{Dec}\left(\mathbf{v}_{z_i}^G\right)$ which is sharing the same class label z_i as the input image $\mathbf{I}_i^t$. By slightly corrupting the values of some dimensions of semantic vectors, we expect the sampled vectors $\mathbf{v}_{z_i}$ still have the same semantic meanings as $g_{Enc}\left(\left\{ f_l\left(\mathbf{I}_i^t\right)\right\}\right)$.

Semantic Neighbourhood (SN). Inspired by the recent vocabulary-informed learning [49], the large amount of vocabulary in the semantic word vector space (*e.g.*, word2vec [15]) can also be used for augmentation. Especially, the distribution of such vocabulary reflect the general semantic relationship in the linguistic corpora. For example, in word vector space, the vector of "truck" is closer to the vector of "car" than to the vector of "dog". Given the features $\left\{ f_l\left(\mathbf{I}_i^t\right)\right\}$ of training instance i, the augmented data $\mathbf{v}_{z_i}^N$ can be sampled from the neighborhood of $g_{Enc}\left(\left\{ f_l\left(\mathbf{I}_i^t\right)\right\}\right)$, *i.e.*,

$$\mathbf{v}_{z_i}^N \in Neigh\left(g_{Enc}\left(\left\{ f_l\left(\mathbf{I}_i^t\right)\right\}\right)\right) \tag{4}$$

$Neigh\left(g_{Enc}\left(\left\{ f_l\left(\mathbf{I}_i^t\right)\right\}\right)\right) \subseteq \mathcal{W}$ indicates the nearest neighborhood vocabulary set of $g_{Enc}\left(\left\{ f_l\left(\mathbf{I}_i^t\right)\right\}\right)$ and $\mathcal{W}$ indicate vocabulary set learned by word2vec [15] on large-scale corpus. These neighbors correspond to the most semantically similar examples to our training instance. The features of virtual synthesized samples can be decoded by $g_{Dec}\left(g_{Enc}\left(\left\{ f_l\left(\mathbf{I}_i^t\right)\right\}\right)\right)$.

There are several points we want to highlight. (1) For one training instance $\mathbf{I}_i^t$, we use as the Gaussian mean in Eq (3) or neighborhood center in Eq (4), the $g_{Enc}\left(\left\{ f_l\left(\mathbf{I}_i^t\right)\right\}\right)$ rather than its ground-truth word vector $\mathbf{u}_{z_i}$. This is due to the fact that $\mathbf{u}_{z_i}$ only represents the semantic center of class z_i, not the center for instance i. Experimentally, on *mini*ImageNet dataset, the results of augmenting features using $\mathbf{u}_{z_i}$, rather than $g_{Enc}\left(\left\{ f_l\left(\mathbf{I}_i^t\right)\right\}\right)$, will lead to $3\sim5\%$ performance drop on average on 1-shot/5-shot classification. (2) Semantic Gaussian noise added in Eq (3) or semantic neighborhood used in Eq (4) result in the synthesized training features that are highly nonlinear (non-Gaussian) for each class due to the non-linearity of our decoder TriNet $g_{Dec}\left(\mathbf{x}\right)$ and ResNet-18 $\left(\left\{ f_l\left(\mathbf{I}_i^t\right)\right\}\right)$. (3) Directly adding Gaussian noise to $\left\{ f_l\left(\mathbf{I}_i^t\right)\right\}$ is another naive way of augmenting features. However, in *mini*ImageNet dataset, such strategy does not give us any significant improvement in one-shot classification.

3.5 One-shot Classification

Given one training image $\mathbf{I}_i^t$, ResNet-18 can extract the feature $\left\{ f_l\left(\mathbf{I}_i^t\right)\right\},\quad l=1,\cdots,4$; and the dual TriNet can augment the corresponding layer features $g\left(\left\{ f_l\left(\mathbf{I}_i^t\right)\right\}\right)$. We thus use the augmented features for one-shot classification.

As previous works [2,16], the features produced by the final layer are utilized for one-shot classification tasks. The augmented $g\left(f_l\left(\mathbf{I}_i^t\right)\right)$ $(l=1,\cdots,4)$ can be taken as the $l-$th layer output of ResNet-18. Thus, by using $g\left(f_l\left(\mathbf{I}_i^t\right)\right)$ as the input of $l+1-$th layer, we can compute the final layer representation of augmented $g\left(f_l\left(\mathbf{I}_i^t\right)\right)$ $(l=1,\cdots,4)$. Finally, the final layer outputs of raw images and augmented features are used for one-shot classification. Particularly, three types of classical classifiers, *i.e.*, the K-nearest neighbors (KNN), Support Vector Machine (SVM) and Logistic Regression (LR), are utilized here, in order to show our augmentation framework can help various classifiers.

4 Experiments

4.1 Datasets

We conduct experiments on four datasets. Note that (1) on all datasets, ResNet-18 is only trained on the source dataset in the specified splits of previous works. (2) The same networks and parameter settings (including the size of input images) are used for all the datasets; and thus all images are resized to $224 * 224$.

***mini*ImageNet.** Originally proposed in [36], this dataset has 60,000 images from 100 classes. Thus each class has around 600 examples. To make our results more compared to previous works, we use the splits in [41] by using 64, 16, 20 classes as training validation, and testing set individually.

Cifar-100. It contains 60,000 images from 100 fine-grained categories, and 20 coarse-level categories [78]. The same data split as [79] to enable the comparison with previous methods. In particular, 64, 16, 20 classes have been used as training validation, and testing set respectively.

Caltech-UCSD Birds 200-2011 (CUB-200). It is a fine-grained dataset of totally 11788 images from 200 categories of birds [80]. As the split in [44], we use 100, 50, 50 classes for training, validation, and testing set. This dataset also provide the semantic attribute on a per-class level in 312 dimension vector.

Caltech-256. It has 30607 images from 256 classes [81]. As in [79], we split it into 150, 56, and 50 classes for training, validation, and testing respectively.

4.2 Network structures and Settings

The same ResNet-18 and dual Trinet are used for all four datasets.

Parameters. The dropout rate and learning rate of the auto-encoder network is set to 0.5 and $1e^{-3}$ respectively to prevent overfitting. The learning rate is divided by 2 every 10 epochs finished.The batch size is set to 64. The network is trained by Adam and usually converges in 300 epochs. To prevent randomness due to the small training set, all experiments are repeated as the experimental setups on each individual dataset. The accuracy results are reported with 95% confidence interval and are averaged over multiple test episodes, the same as previous works. Parts of training codes and baselines are attached as supplementary material. More source codes will be released upon acceptance.

Settings. We use the 100-dimensional semantic word vector extracted from the vocabulary dictionary released in [49]. The class name is projected into the space as the vector $\mathbf{u}_{z_i}^s$ or $\mathbf{u}_{z_i}^t$. The semantic attribute space is pre-defined by experts as [32,80]. In all experiments, given one training instance in one type of semantic space, the dual TriNet will augment 4 synthesized instances of each layer. We use the synthesized instances generated by all layers, unless otherwise specified.

4.3 Competitors and Classification models

Competitors. Previous methods are compared with the same source/target and training/testing settings as ours. These methods are Matching Nets [36], MAML [38], Meta-SGD [39], DEML+Meta-SGD [40], PROTO-NET [37], RELATION NET [43], META-LEARN LSTM [41], Meta-Net [42], and MACO [44].
Classification model. KNN, SVM and LR are used as the classification models to validate the effectiveness of our augmented data. The hyperparameters of classification models are cross-validated on the target dataset.

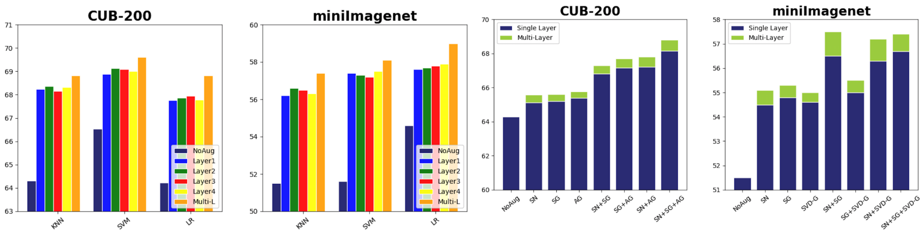
4.4 Experimental results on *mini*ImageNet and CUB-200 datasets

Methods	*mini*ImageNet (%)		CUB-200(%)	
	1-shot	5-shot	1-shot	5-shot
MAML [38]	48.70±1.84	63.11±0.92	38.43	59.15
Meta-SGD [39]	50.47±1.87	64.03±0.94	-	-
DEML+Meta-SGD [40]	**58.49**±0.91	71.28±0.69	-	-
META-LEARN LSTM [41]	43.44±0.77	60.60±0.71	40.43	49.65
Meta-Net [42]	49.21±0.96	-	-	-
Matching Nets [36]	43.56±0.84	55.31±0.73	49.34	59.31
PROTO-NET [37]	49.42±0.78	68.20±0.66	45.27	56.35
RELATION NET [43]	57.02±0.92	71.07±0.69	-	-
MACO [44]	41.09±0.32	58.32±0.21	60.76	74.96
ResNet-18	52.73±1.44	73.31±0.81	66.54±0.53	82.38±0.43
Ours: ResNet-18+Dual TriNet	58.12±1.37	**76.92**±0.69	**69.61**±0.46	**84.10**±0.35

Table 1. Results on *mini*ImageNet and CUB-200. The "±" indicates 95% confidence intervals over tasks. Note that "±" is not reported on CUB-200 in previous works.

Settings. On *mini*ImageNet dataset, we only have the semantic word space. Thus give one training instance, we can augment 16 and 16 synthesized instances for Semantic Gaussian (SG) and Semantic Neighbourhood (SN) respectively. On CUB-200 dataset, we use both the semantic word vector and semantic attribute spaces. Thus for one training instance, we augment 16 and 16 synthesized instances for SG and SN in semantic word vector space; and additionally, we generate 16 virtual instances (in all four layers) for Semantic Gaussian (SG) in semantic attribute space, which is denoted as Attribute Gaussian (AG).
Results. As in Tab. 1, the competitors can be divided into two categories: Meta-learning algorithms (including MAML, Meta-SGD, DEML+Meta-SGD, META-LEARN LSTM and Meta-Net) and Metric-learning algorithms (including Matching Nets, PROTO-NET, RELATION NET and MACO). We report the results of ResNet-18 (without data augmentation), to extract the features of training instances. The accuracy of our framework (ResNet-18+Dual TriNet) is also compared. The Dual TriNet synthesize the each layer features of ResNet-18 if given any training instance as in Sec. 3.4. We use SVM classifiers for ResNet-18 and ResNet-18+Dual TriNet in Tab. 1. In particular, we found that,

(1) Our framework can achieve the best performance. As shown in Tab. 1, the results of our framework, *i.e.*, ResNet-18+Dual TriNet can achieve the best performance and we can show a clear margin over all the other baselines on both datasets. This validates the effectiveness of our framework in solving the one-shot learning tasks. Note that greatly benefit from learning the residual, Resnet-18 is a very good feature extractor for one-shot learning tasks. Actually, the few-shot results of ResNet-18 on two datasets have almost beat all the other baselines. Note DEML+Meta-SGD [40] uses the ResNet-50 as the baseline models, and thus has better one-shot learning results than our ResNet-18. Nevertheless, with the augmented data by DualTriNet, we can observe a clear improvement over the ResNet-18 baseline. This further validates that our framework can efficiently solve one-shot learning with very good performance.



(a) Results of different layers/classifiers

(b) Results of different semantic spaces

Fig. 2. Ablation study on CUB-200 and *mini*ImageNet of one-shot learning. (a) Results of feature augmentation by different layers/classifiers. “NoAug”, “Layer1”, “Layer2”, “Layer3”, “Layer4” indicate the one-shot learning results without any augmentation, with the feature augmentation by using layer 1, layer 2, layer 3, layer 4 of ResNet-18. “Multi-L” denotes the performance of using all augmented instances of one-shot learning. The X-axis represents the different supervised classifiers. (b) Results of feature augmentation by different types of semantic spaces. “Single Layer” indicates the best one-shot performance augmented by using only single layer. “Multi-layer” represents the results of using synthesized instances from all layers.

(2) Our framework can effectively augment multiple layer features. We analyze the effectiveness of augmented features in each layer as shown in Fig. (2-a). On CUB-200 and *mini*ImageNet, we reports the results in 1-shot learning cases. We have several conclusions: (1) Only using the augmented features from one single layer (*e.g.*, Layer1 – Layer 4 in Fig. (2-a)) can also help improve the performance of one-shot learning results. This validates the effectiveness of our dual TriNet of synthesizing different layers features in a single framework. (2) The results of using synthesized instances from all layers (Multi-L) are even higher than those of each single layer. This indicates that the augmented features of different layers are intrinsically complementary to each other.

(3) Augmented features can boost the performance of different supervised classifiers. Our augmented features are not designed for one particular supervised classifiers. To show this point and as illustrated in Fig. (2-a), three classical supervised classifiers (*i.e.*, KNN, SVM and LR in the X-axis of Fig. (2-a)). It shows that our augmented features can boost the performance of three supervised classifiers on one-shot classification cases. This further shows the effectiveness of our augmentation framework.

(4) The augmented features by SG, SN and AG can also improve few-shot learning results. We compare different types of feature augmentation methods of various semantic spaces in Fig. (2-b). Specifically, we compare the SG and SN in semantic word vector space; and AG in semantic attribute space. On CUB-200 dataset, the augmented results by SG, SN and AG are better than those without argumentation. The accuracy of combining the synthesized instance features generated by any two of SG, SN and AG can be further improved over those of SG, SN or AG only. This means that the augmented feature instances of SG, SN and AG are complementary to each other. Finally, we observe that by combining augmented instances from all SG, SN and AG, the accuracy of one-shot learning is the highest one.

Methods	Caltech-256 (%)		CIFAR-100 (%)	
	1-shot	5-shot	1-shot	5-shot
DEML+Meta-SGD [40]	62.25±1.00	79.52±0.63	61.62±1.01	77.94±0.74
MAML [38]	45.59±0.77	54.61±0.73	49.28±0.90	58.30±0.80
Meta-SGD [39]	48.65±0.82	64.74±0.75	53.83±0.89	70.40±0.74
Matching Nets [36]	48.09±0.83	57.45±0.74	50.53±0.87	60.30±0.82
ResNet-18	60.13±0.71	78.79±0.54	59.65±0.78	76.75±0.73
ResNet-18+Dual TriNet	**63.77**±0.62	**80.53**±0.46	**63.41**±0.64	**78.43**±0.62

Table 2. Results on Caltech-256 and CIFAR-100 datasets. The "±" indicates 95% confidence intervals over tasks.

Settings. On Caltech-256 and CIFAR-100 dataset, we also use the semantic word vector space. For one training instance, we synthesize 16 and 16 instance features for SG and SN individually from all four layers of ResNet-18. On these two datasets, the results of competitors are implemented and reported in [40]. Our reported results are produced by using the augmented feature instances of all layers, both by SG and SN. The SVM classifier is used as the classification model.

(5) Even the semantic space inferred from semantic relationship of classes can also work well in our framework. To show this point, we again compare the results in Fig. (2-b). Particularly, we compute the similarity matrix of classes in *mini*ImageNet by using semantic word vector. The SVD is employed to decompose the similarity matrix; and the left singular vectors of SVD are spanned a new semantic space. Such a new space is thus utilized in learning the dual TriNet. We employ the Semantic Gaussian (SG) to augment the instance feature in the newly spanned space for one-shot classification; and the results are denoted as "SVD-G" . We report the results of such SVD-G augmentation in *mini*ImageNet dataset in Fig. (2-b). We highlight several interesting observations. (1) The results by SVD-G feature augmentation are still better than those without any augmentation. (2) The accuracy of SVD-G actually is slightly worse than that of SG, since the new spanned space is derived from the original semantic word space. (3) There is almost no complementary information in the augmented features between SVD-G and SG, still partly due to the new space spanned from the semantic word space. (4) The augmented features produced by SVD-G are also very complementary to those from SN as shown in the results of Fig. (2-b). This is due to the additional neighbourhood vocabulary information is not used in deriving the new semantic space. We have the similar experimental conclusion on CUB-200, with the results shown in the supplementary material.

4.5 Experimental results on Caltech-256 and CIFAR-100 datasets

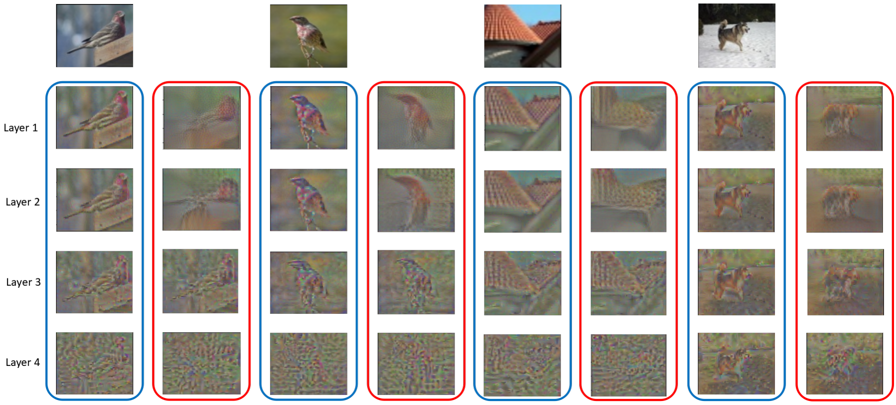


Fig. 3. Visualization of the original and augmented features.

Results. The results on Caltech-256 and CIFAR-100 are compared in Tab. 2. We found that (1) our method can still achieve the best results over the state-of-the-art algorithms, still thanks to the augmented feature instances of proposed framework. (2) The ResNet-18 is still a very strong baseline; and it can beat almost all the other baselines, except the DEML+Meta-SGD which uses ResNet-50 as the baseline structure. (3) There is a clearly improved margin of using our augmented instance features over using ResNet-18 only. This further validates the efficacy of proposed framework.

5 Further analysis

5.1 Dual TriNet structure

We propose the dual TriNet structure which intrinsically is derived from the encoder-decoder architecture. Thus we further analyze the other alternative network structures for feature augmentation. In particular, the alternative choices of augmentation network can be the auto-encoder [82] of each layer, or U-net

[83][1]. The results are compared in Tab. 3. We can show that our dual TriNet can best explore the complementary information of different layers, and thus our results are better than those without augmentation (ResNet-18), with U-net augmentation (ResNet-18+U-net) and with auto-encoder augmentation (ResNet-18+Auto-encoder). This validates that our dual TriNet can efficiently merge and exploit the information of multiple layers for feature augmentation.

5.2 Visualization

Using the technique in [84], we can visualize the image that can generate the augmented features $\hat{f}_l(\mathbf{I}_i) = g(f_l(\mathbf{I}_i))$ in ResNet-18. We firstly randomly generate an image $\mathbf{I}_{i_0}$. Then we optimize $\mathbf{I}_{i_0}$ by reducing the distance betwen $f_l(\mathbf{I}_{i_0})$ and $\hat{f}_l(\mathbf{I}_i)$ (both are the output of ResNet-18):

$$\mathbf{I}_{i_0} = \underset{\mathbf{I}_{i_0}}{\operatorname{argmin}} \frac{1}{2} \left\| f_l(\mathbf{I}_{i_0}) - \hat{f}_l(\mathbf{I}_i) \right\|_2^2 + \lambda \cdot R(\mathbf{I}_{i_0})$$

where $R(\cdot)$ is the Total Variation Regularizer for image smoothness; $\lambda = 1e-2$ When the difference is small enough, $\mathbf{I}_{i_0}$ should be the images that can generate the corresponding augmented feature.

Methods	MiniImagenet		CUB-200		Caltech-256		CIFAR-100	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
ResNet-18	52.73	73.31	66.54	82.38	60.13	78.79	59.65	76.75
ResNet-18+U-net	56.41	75.67	68.32	83.24	61.54	79.88	62.32	77.87
ResNet-18+Auto-encoder	56.80	75.27	68.56	83.24	62.41	79.77	61.76	76.98
ResNet-18+Dual-TriNet	**58.12**	**76.92**	**69.61**	**84.10**	**63.77**	**80.53**	**63.41**	**78.43**

Table 3. Results of using alternative augmentation networks.

By using SN and the visualization algorithm above, we visualize the original and augmented features in Fig. 3. The top row shows the input images of two birds, one roof and one dog. The blue circles and red circles indicate the visualization of original and augmented features of Layer 1 – Layer 4 respectively. The visualization of augmented features are similar, and yet different from that of original features. For example, the first two columns show that the visualization of augmented features actually slightly change the head pose of the bird. In the last two columns, the augmented features clearly visualize a dog which is similar have different appearance from the input image. This intuitively shows why our framework work.

6 Conclusions

This work purposes an end-to-end framework for feature augmentation. The proposed dual TriNet structure can efficiently and directly augment multi-layer

[1] The details of structures are in Supplementary.

visual features to boost the few-shot classification. We demonstrate our framework can efficiently solve the few-shot classification on four datasets. We mainly evaluate on classification tasks; it is also interesting and a future work of utilizing augmented features on the other related tasks, such as one-shot image/video segmentation [75,76]. Additionally, though dual TriNet is paired with ResNet-18 here, we can easily extend it for other feature extractor networks, such as, ResNet-50. Thus taken as another future work.

References

1. Thrun, S.: Learning To Learn: Introduction. Kluwer Academic Publishers (1996)
2. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. (2012)
3. Fu, Y., Hospedales, T.M., Xiang, T., Gong, S.: Transductive multi-view zero-shot learning. IEEE TPAMI (2015)
4. Wang, Y., Hebert, M.: Learning from small sample sets by combining unsupervised meta-training with cnns. In: NIPS. (2016)
5. Wang, Y., Hebert, M.: Learning to learn: model regression networks for easy small sample learning. In: ECCV. (2016)
6. Li, Z., Hoiem, D.: Learning without forgetting. In: ECCV. (2016)
7. Lim, J., Salakhutdinov, R., Torralba, A.: Transfer learning by borrowing examples for multiclass object detection. In: NIPS. (2011)
8. Movshovitz-Attias, Y.: Dataset curation through renders and ontology matching. In: Ph.D. thesis, CMU. (2015)
9. Park, D., Ramanan, D.: Articulated pose estimation with tiny synthetic videos. In: CVPR. (2015)
10. Movshovitz-Attias, Y., Yu, Q., Stumpe, M., Shet, V., Arnoud, S., Yatziv, L.: Ontological supervision for fine grained classification of street view storefronts. In: CVPR. (2015)
11. Dosovitskiy, A., Springenberg, J., Brox, T.: Learning to generate chairs with convolutional neural networks. In: CVPR. (2015)
12. Zhu, X., Vondrick, C., Fowlkes, C., Ramanan, D.: Do we need more training data? In: IJCV. (2016)
13. Opelt, A., Pinz, A., Zisserman, A.: Incremental learning of object detectors using a visual shape alphabet. In: IEEE Conference on Computer Vision and Pattern Recognition. Volume 1. (2006) 3–10
14. Dixit, M., Kwitt, R., Niethammer, M., Vasconcelos, N.: Aga: Attribute guided augmentation. In: CVPR. (2017)
15. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Neural Information Processing Systems. (2013)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. (2015)
17. Jankowski, Norbert, Duch, Wodzislaw, Grabczewski, Krzyszto: Meta-learning in computational intelligence. In: Springer Science & Business Media. (2011)
18. Lake, B.M., Salakhutdinov, R.: One-shot learning by inverting a compositional causal process. In: NIPS. (2013)
19. Bart, E., Ullman, S.: Cross-generalization: learning novel classes from a single example by feature replacement. In: CVPR. (2005)

20. Fei-Fei, L., Fergus, R., Perona, P.: A bayesian approach to unsupervised one-shot learning of object categories. In: IEEE International Conference on Computer Vision. (2003)
21. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE TPAMI (2006)
22. Tommasi, T., Caputo, B.: The more you know, the less you learn: from knowledge transfer to one-shot learning of object categories. In: British Machine Vision Conference. (2009)
23. Rezende, D.J., Mohamed, S., Danihelka, I., Gregor, K., Wierstra, D.: One-shot generalization in deep generative models. In: ICML. (2016)
24. Santoro, Bartunov, S., Botvinick, M., Wierstra, D., Lillicrap, T.: One-shot learning with memory-augmented neural networks. In: arx. (2016)
25. Kingma, D., Welling, M.: Auto-encoding variational bayes. In: ICLR. (2014)
26. JVilalta, R., Drissi, Y.: A perspective view and survey of meta-learning. Artificial intelligence review (2002)
27. Hertz, T., Hillel, A., Weinshall, D.: Learning a kernel function for classification with small training samples. In: ICML. (2016)
28. Fleuret, F., Blanchard, G.: Pattern recognition from one example by chopping. In: NIPS. (2005)
29. Amit, Y., Fink, M., S., N., U.: Uncovering shared structures in multiclass classification. In: ICML. (2007)
30. Wolf, L., Martin, I.: Robust boosting for learning from few examples. In: CVPR. (2005)
31. Torralba, A., Murphy, K., Freeman, W.: sharing visual features for multiclass and multiview object detection. In: IEEE TPAMI. (2007)
32. Lampert, C.H., Nickisch, H., Harmeling, S.: Attribute-based classification for zero-shot visual object categorization. IEEE TPAMI (2013)
33. Rohrbach, M., Ebert, S., Schiele, B.: Transfer learning in a transductive setting. In: NIPS. (2013)
34. Rohrbach, M., Stark, M., Szarvas, G., Gurevych, I., Schiele, B.: What helps where – and why? semantic relatedness for knowledge transfer. In: CVPR. (2010)
35. Torralba, A., Murphy, K.P., Freeman, W.T.: Using the forest to see the trees: Exploiting context for visual object detection and localization. Commun. ACM (2010)
36. Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., Wierstra, D.: Matching networks for one shot learning. In: NIPS. (2016)
37. Snell, J., Swersky, K., Zemeln, R.S.: Prototypical networks for few-shot learning. In: NIPS. (2017)
38. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML. (2017)
39. Li, Z., Zhou, F., Chen, F., Li, H.: Meta-sgd: Learning to learn quickly for few shot learning. In: arxiv:1707.09835. (2017)
40. Zhou, F., Wu, B., Li, Z.: Deep meta-learning: Learning to learn in the concept space. In: arxiv:1802.03596. (2018)
41. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning. In: ICLR. (2017)
42. Munkhdalai, T., Yu, H.: Meta networks. In: ICML. (2017)
43. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M.: Learning to compare: Relation network for few-shot learning. In: CVPR. (2018)

44. Hilliard, N., Phillips, L., Howland, S., Yankov, A., Corley, C.D., Hodas, N.O.: Few-Shot Learning with Metric-Agnostic Conditional Embeddings. ArXiv e-prints (February 2018)
45. zhongwen xu, linchao zhu, Yang, Y.: Few-shot object recognition from machine-labeled web images. In: arxiv. (2016)
46. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of the 34th International Conference on Machine Learning. (2017) 1126–1135
47. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: Delving deep into convolutional nets. In: BMVC. (2014)
48. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: ECCV. (2014)
49. Fu, Y., Sigal, L.: Semi-supervised vocabulary-informed learning. In: CVPR. (2016)
50. Ba, J.L., Swersky, K., Fidler, S., Salakhutdinov, R.: Predicting deep zero-shot convolutional neural networks using textual descriptions. In: ICCV. (2015)
51. Su, H., Qi, C.R., Li, Y., Guibas, L.J.: render for cnn viewpoint estimation in images using cnns trained with rendered 3d model views. In: ICCV. (2015)
52. Charalambous, C.C., Bharath, A.A.: A data augmentation methodology for training machine/deep learning gait recognition algorithms. In: BMVC. (2016)
53. Rogez, G., Schmid, C.: mocap-guided data augmentation for 3d pose estimation in the wild. In: NIPS. (2016)
54. Peng, X., Sun, B., Ali, K., Saenko, K.: Learning deep object detectors from 3d models. In: ICCV. (2015)
55. Hariharan, B., Girshick, R.: Low-shot visual recognition by shrinking and hallucinating features. In: ICCV. (2017)
56. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV. (2017)
57. Zhu, J.Y., Krähenbühl, P., Shechtman, E., Efros, A.A.: Generative visual manipulation on the natural image manifold. In: ECCV. (2016)
58. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., DavidWarde-Farley, Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS. (2014)
59. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text-to-image synthesis. In: ICML. (2016)
60. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: arxiv. (2016)
61. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z.: Least squares generative adversarial networks. In: arxiv. (2017)
62. Durugkar, I., Gemp, I., Mahadevan, S.: Generative multi-adversarial networks. In: ICLR. (2017)
63. Huang, X., Li, Y., Poursaeed, O., Hopcroft, J., Belongie, S.: Stacked generative adversarial networks. In: CVPR. (2017)
64. Bromley, J., Bentz, J., Bottou, L., Guyon, I., LeCun, Y., Moore, C., Sackinger, E., Shah, R.: Signature verification using a siamese time delay neural network. In: IJCAI. (1993)
65. Koch, G., Zemel, R., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition. In: ICML – Deep Learning Workshok. (2015)
66. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: CVPR. (2009)
67. Kienzle, W., Chellapilla, K.: Personalized handwriting recognition via biased regularization. In: ICML. (2006)

68. Quattoni, A., Collins, M., Darrell, T.: Transfer learning for image classification with sparse prototype representations. In: IEEE Conference on Computer Vision and Pattern Recognition. (2008) 1–8
69. Fink, M.: Object classification from a single example utilizing class relevance metrics. In: NIPS. (2005)
70. Wolf, L., Hassner, T., Taigman, Y.: The one-shot similarity kernel. In: ICCV. (2009)
71. Rasmus, A., Valpola, H., Honkala, M., Berglund, M., Raiko, T.: Semi-supervised learning with ladder networks. In: NIPS. (2015)
72. Wang, J., Wei, Z., Zhang, T., Zeng, W.: Deeply-fused nets. In: arxiv:1505.05641. (2016)
73. Yu, F., Wang, D., Shelhamer, E., Darrell, T.: Deep layer aggregation. In: CVPR. (2018)
74. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR. (2015)
75. Shaban, A., Bansal, S., Liu, Z., Essa, I., Boots, B.: One-shot learning for semantic segmentation. In: BMVC. (2017)
76. Caelles, S., Maninis, K.K., Pont-Tuset, J., Leal-Taixe, L., Cremers, D., Gool, L.V.: One-shot video object segmentation. In: CVPR. (2017)
77. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: NIPS. (2014)
78. Krizhevsky, A.: Learning multiple layers of features from tiny images. (2009)
79. Zhou, F., Wu, B., Li, Z.: Deep Meta-Learning: Learning to Learn in the Concept Space. ArXiv e-prints (February 2018)
80. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology (2011)
81. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. (2007)
82. Hinton, G.E., Salakhutdinov, R.R.: reducing the dimensionality of data with neural networks. (2006)
83. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI. (2015)
84. Mahendran, A., Vedaldi, A.: Understanding Deep Image Representations by Inverting Them. ArXiv e-prints (November 2014)

A Results on CUB-200

Method	Shots	R-18	Layer	Data Augmentation						
				SN	SG	SD	SN+SG	SG+SD	SN+SD	SN+SG+SD
KNN	1	64.30	S.	65.12	65.21	65.38	66.82	65.50	67.21	67.23
			M.	65.58	65.61	65.78	67.29	65.77	67.82	67.91
	5	77.66	S.	78.34	78.42	78.62	79.01	78.66	79.12	79.36
			M.	79.01	78.96	79.04	79.51	79.09	79.56	79.71
SVR	1	66.54	S.	67.63	67.49	67.69	68.23	67.60	68.41	68.56
			M.	68.10	68.03	68.22	68.71	67.98	68.89	69.01
	5	82.38	S.	83.01	83.07	83.02	83.59	83.11	83.42	83.44
			M.	83.47	83.51	83.60	83.82	83.49	83.99	84.10
LR	1	64.22	S.	65.29	65.33	65.43	66.59	65.46	66.89	67.01
			M.	65.71	65.92	65.89	67.12	65.74	67.63	67.55
	5	82.51	S.	83.37	83.31	83.60	83.61	83.59	83.62	83.69
			M.	83.82	83.83	83.90	84.21	83.71	84.23	84.17

Table 4. The classification accuracy of one-shot learning on Caltech-UCSD Birds in 5-way. Note that: "S." and "M." indicates the single and multiple layers respectively. "SD" is short for "SVD-G". "R-18" is short for "ResNet-18".

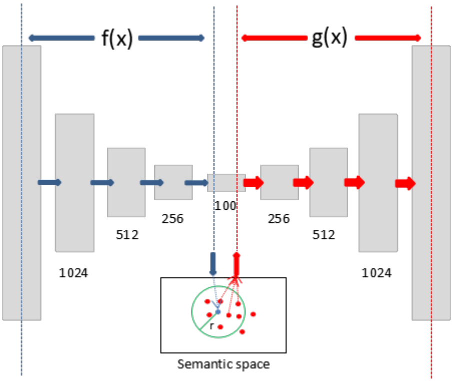


Fig. 4. The structure of auto-encoder.

B Auto-encoder Structure

The structure of auto-encoder is shown in Fig. 4. We trained an auto-encoder for each residual layer in Resnet-18, *i.e.*, we have 4 auto-encoder. The encoder projects visual space into semantic space and the decoder project the semantic space into visual space. Both encoder and decoder are constructed only by fully-connected layers and Relu layers.

C U-net Structure

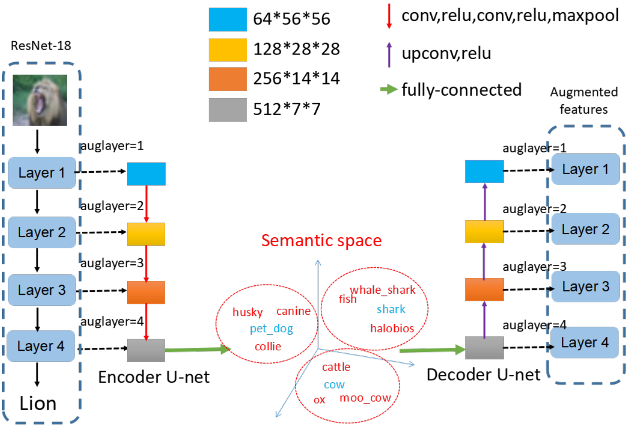


Fig. 5. The structure of U-net.

The structre of U-net is shown in Fig. 5. The U-net is paired with Resnet-18. The encoder U-net is composed of four layers corresponding to each layer of ResNet-18. It aims to learn the function $\hat{\mathbf{u}}_{z_i} = g_{Enc}(f_l(\mathbf{I}_i))$ to feature $(f_l(\mathbf{I}_i))$ as close to the semantic vector $\hat{\mathbf{u}}_{z_i}$ of instance i as possible. The decoder U-net has exactly the same architecture to project the features from semantic space $\hat{\mathbf{u}}_{z_i}$ to the feature space $\hat{f}_l(\mathbf{I}_i) = g_{Dec}(g_{Enc}(f_l(\mathbf{I}_i)))$.

D Resnet-18 Structure

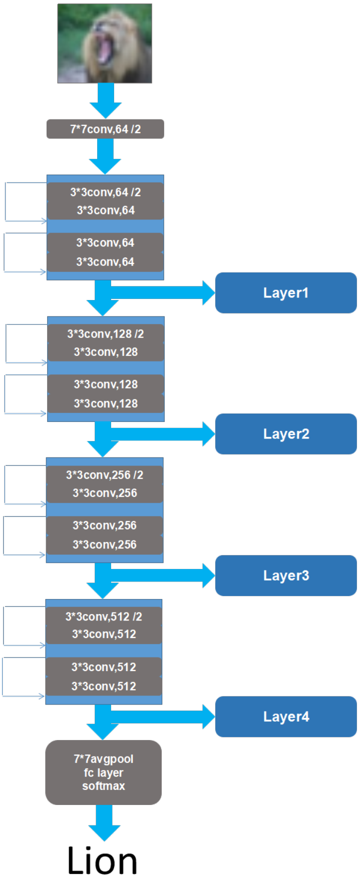


Fig. 6. The structure of Resnet-18.