Zero-Shot Learning from Adversarial Feature Residual to Compact Visual Feature

Bo Liu,[1,2] **Qiulei Dong,**[1,2,3]* **Zhanyi Hu**[1,2,3]
[1]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
[3]Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences
{bo.liu, qldong, huzy}@nlpr.ia.ac.cn

Abstract

Recently, many zero-shot learning (ZSL) methods focused on learning discriminative object features in an embedding feature space, however, the distributions of the unseen-class features learned by these methods are prone to be partly overlapped, resulting in inaccurate object recognition. Addressing this problem, we propose a novel adversarial network to synthesize compact semantic visual features for ZSL, consisting of a residual generator, a prototype predictor, and a discriminator. The residual generator is to generate the visual feature residual, which is integrated with a visual prototype predicted via the prototype predictor for synthesizing the visual feature. The discriminator is to distinguish the synthetic visual features from the real ones extracted from an existing categorization CNN. Since the generated residuals are generally numerically much smaller than the distances among all the prototypes, the distributions of the unseen-class features synthesized by the proposed network are less overlapped. In addition, considering that the visual features from categorization CNNs are generally inconsistent with their semantic features, a simple feature selection strategy is introduced for extracting more compact semantic visual features. Extensive experimental results on six benchmark datasets demonstrate that our method could achieve a significantly better performance than existing state-of-the-art methods by $\sim$1.2-13.2% in most cases.

Introduction

In recent years, zero-shot learning (ZSL) has attracted more and more attention in pattern recognition and machine learning. Given a set of labeled seen-class data as well as the semantic relationship between seen and unseen classes, ZSL aims to recognize unseen-class instances. Most existing ZSL methods focused on learning discriminative object features in an embedding feature space where the semantic relationship between seen and unseen classes is preserved, and they could be roughly divided into two categories: visual-to-semantic methods and semantic-to-visual methods.

The visual-to-semantic methods (Frome et al. 2013; Akata et al. 2015a; Xian et al. 2016) aim to build a projection function from visual features to semantic features. The visual features are generally extracted from the input images by CNNs (Convolutional Neural Networks), while the semantic features describe semantic attributes of object class, e.g. class-level attribute and text description. The projection function is trained on seen-class data by making the projected visual features closer to the semantic feature of their correct class. However, the distributions of the projected visual features of unseen classes by these methods are prone to be partly overlapped, leading to inaccurate object recognition.

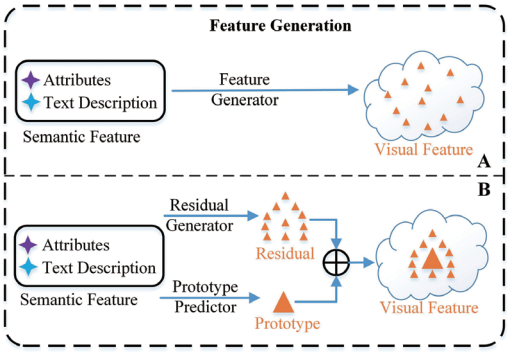

Figure 1: Comparison of our method with existing GAN-based ZSL methods. A: Existing GAN-based methods generate visual features conditioned on their semantic feature. B: Our method generates visual feature residuals conditioned on their semantic feature, and then synthesizes visual features by combining the residuals and a visual prototype predicted from its semantic feature.

In comparison to these visual-to-semantic methods, the semantic-to-visual methods (Zhu et al. 2018; Xian et al. 2018b; Li et al. 2019) have significantly improved the ZSL performance recently. Most of semantic-to-visual methods aim to generate visual features of unseen classes conditioned on their semantic features via generative adversarial network (GAN) as illustrated in Figure 1 A, and then train a classifier with the synthetic visual features and their corresponding labels for classifying real visual features of unseen classes.

*Corresponding author: Qiulei Dong

Despite their success, these GAN-based methods are still limited by the problem that the distributions of the synthetic unseen-class visual features are partly overlapped. Figure 2 A provides an example for illustrating this problem. As shown in Figure 2 A, different color points represent the visual features belonging to different unseen classes, which are generated by an existing GAN-based ZSL method (e.g. (Xian et al. 2018b)). Obviously, there is some overlap between different color points, indicating that the distributions of the synthetic visual features are partly overlapped. This overlap is probably because the GAN used to generate the unseen-class visual features is trained only on the seen-class data.

In addition, the fidelity of synthetic visual features of unseen classes by these GAN-based methods is also limited by the inconsistency between semantic features and visual features. The semantic-visual inconsistency refers to the fact that even if two classes have very similar semantic attributes, e.g. both elephants and tigers have the 'tail' attribute, their visual features could be very different, e.g. the visual features of an elephant's tail and a tiger's tail are quite different. As illustrated in Figure 2 C, due to this inconsistency, even though GAN can learn an accurate semantic-to-visual generative relationship (illustrated by straight line for convenience) on seen-class data, the distribution of synthetic unseen-class visual features (illustrated by bigger ellipse) according to the learned generative relationship is different from the distribution of real unseen-class visual features.

Addressing these two problems, we propose a novel adversarial network to learn compact semantic visual features for ZSL by integrating the visual prototype and the visual feature residual, which consists of a residual generator, a prototype predictor, and a discriminator. Here, the visual prototype represents general visual features of each class and the visual feature residual represents the feature deviation of each sample from its prototype. The residual generator is employed to generate the visual feature residual conditioned on semantic feature, and then the visual feature is synthesized by combining the residual with the class-level visual prototype predicted from its semantic feature by the prototype predictor, as illustrated in Figure 1 B. After the synthetic visual features are synthesized, the discriminator tries to distinguish the synthetic visual features from the real ones extracted from an existing categorization CNN. Since the visual prototypes are explicitly predicted for both seen and unseen classes and most of the residuals are generally numerically much smaller than the distances among prototypes of all classes, the synthetic visual features by the proposed method are less overlapped as shown in Figure 2 B. To alleviate the semantic-visual inconsistency problem, we propose a simple feature selection strategy which is able to adaptively select some semantically consistent feature dimensions from the original visual feature.

In summary, our contributions are three-fold:

- We propose a novel adversarial network to learn compact semantic visual features for ZSL, which are synthesized by integrating the generated feature residuals and predicted visual prototypes. The distributions of the synthetic visual features by the proposed method are less overlapped. To our best knowledge, this is the first work to utilize adversarial feature residual for ZSL.
- We propose a simple feature selection strategy that is able to adaptively select semantically consistent visual feature elements from the original visual feature, alleviating the semantic-visual inconsistency problem to some extent.
- Extensive experimental results demonstrate that the proposed method can outperform existing state-of-the-art methods with a significant improvement on six benchmark datasets.

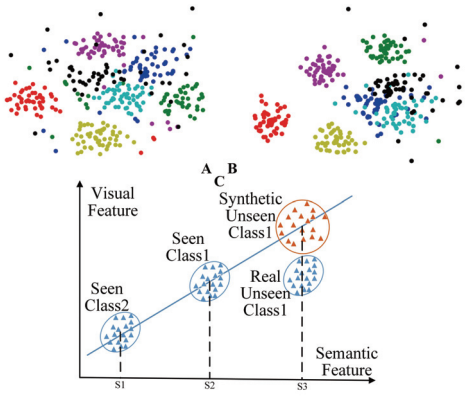

Figure 2: A: t-SNE visualization of synthetic unseen-class visual features by an existing GAN-based ZSL method, different color points represent the visual features belonging to different unseen classes. B: t-SNE visualization of synthetic unseen-class visual features by our method. C: A ellipse represents a distribution of the visual features belonging to a class, the distribution of the synthetic visual features of the unseen class (bigger ellipse) according to the semantic-visual generative relationship (straight line) learned on two seen classes is different from the real counterpart.

Related Work

Zero-Shot Learning. Lampert et al. (Lampert, Nickisch, and Harmeling 2013) proposed a two-step attribute-based classification method, where a probabilistic classifier was firstly learned for predicting probability of each attribute for each image, then the image was classified by a Bayesian classifier based on probabilities of attributes. Frome et al. (Frome et al. 2013) proposed an end-to-end visual-to-semantic projection method. In this method, visual features extracted by an categorization CNN were projected into a semantic feature space by a linear function which was trained to make the projected visual features closer to the semantic feature of their correct category. Following this work, a lot of methods have devoted themselves to improve it by replacing its loss function (Akata et al. 2015b; 2015a; Romera-Paredes and Torr 2015; Norouzi et al. 2013) or using nonlinear projection function (Socher et al. 2013; Xian et al. 2016). Recently, some works proposed to learn a semantic-to-visual mapping, they (Changpinyo, Chao, and Sha 2017; Zhang, Xiang, and Gong 2017) used semantic

features to predict visual features by a transformation function, or they (Zhu et al. 2018; Xian et al. 2018b) trained a GAN (Goodfellow et al. 2014) to generate visual features of unseen classes conditioned on their semantic features. Except for these, some works which leveraged mutually visual-semantic reconstruction (Kodirov, Xiang, and Gong 2017) or projected semantic features to parameter space (Changpinyo et al. 2016) were also proposed.

Visual Prototype Prediction. Visual features have a clustered structure in feature space, so it is feasible and beneficial to adopt a visual prototype to represent a class. Only a few works have applied visual prototype prediction to ZSL. Changpinyo et al. (Changpinyo, Chao, and Sha 2017) proposed a visual exemplar prediction method, where they trained a prediction function from semantic embeddings to visual exemplars and then the predicted exemplars were applied to other methods as visual training data or ideal semantic embeddings.

Visual Feature Generation. With the development of GAN, some works (Zhu et al. 2018; Xian et al. 2018b; Li et al. 2019; Paul, Krishnan, and Munjal 2019) have applied GAN to ZSL problem. In these methods, they all employed GAN to generate visual features of unseen classes conditioned on semantic features and then used the synthetic visual features to train a classifier for unseen classes. What they differ in is the way to restrict synthetic visual features and the choice of visual features and semantic features. Zhu et al. (Zhu et al. 2018) proposed to restrict synthetic visual features by adding a visual pivot regularization and they employed local visual features extracted from semantic regions of objects. By adding a classification penalty on synthetic visual features and using visual features from deeper CNN, Xian et al. (Xian et al. 2018b) proposed a feature generation network. Li et al. (Li et al. 2019) restricted synthetic visual features by adding multiple visual souls regularization. Different from them, we employ GAN to generate visual feature residual instead of visual feature.

Methodology

The definition of ZSL is as follows. Let $S = \{(x_n, y_n, e) \mid x_n \in X^s, y_n \in Y^s, e \in E, n = 1, 2, \cdots, N\}$ be a training dataset, where $x_n \in \mathbf{R^v}$ is the visual feature of the n-th labeled image in the training dataset, y_n is the class label of x_n, which belongs to seen-class set Y^s, N is the number of samples, and $e \in \mathbf{R^s}$ is the semantic feature of a class in the total class set Y which not only includes the seen-class set Y^s but also includes the unseen-class set Y^u. Note that the unseen-class set Y^u is disjoint with the seen-class set Y^s. Let X represents the test image set, conventional ZSL is to learn a mapping $f : X \to Y^u$, while generalized ZSL is to learn a mapping $f : X \to Y$.

To tackle the ZSL problem, we propose a novel network to synthesize compact semantic visual features of unseen classes with adversarial feature residual, called AFRNet, and then train a classifier with these synthetic visual features and their corresponding labels for feature classification, the overall pipeline is shown in Figure 3. As shown in the feature generation phase, the AFRNet consists of three modules: residual generator, prototype predictor, and discriminator. The residual generator is used to generate the visual feature residual, and then the visual feature is synthesized by integrating the residual and the visual prototype predicted by the prototype predictor, the real visual features are extracted by an feature extractor (implemented by an existing categorization CNN). The discriminator tries to distinguish the synthetic visual features from the real visual features. Further, by applying a feature selection strategy to visual prototypes and real visual features, we could learn the compact semantic visual features. After the AFRNet is trained, as shown in the classification phase of Figure 3, the synthetic unseen-class visual features are used to train a classifier for feature classification.

In the following, firstly we introduce a semantically compact prototype predictor for predicting visual prototypes from semantic features. We then describe how to learn compact semantic visual feature from adversarial feature residual. Next, the employed classifier is introduced. Finally, the comparison of the proposed method to some related works is given.

Semantically Compact Prototype Predictor

Predicting Visual Prototype. Here, our goal is to learn a prediction function with a set of training data belonging to seen classes for predicting the visual prototypes of unseen classes from their semantic features. We use the mean vector of the visual feature vectors of each seen class as the visual prototype of each class. Suppose we have N_c visual features for class c in the training data, then the visual prototype p_c for class c is computed by $\frac{1}{N_c}\sum_{i=1}^{N_c} x_c^i$, where $x_c^i \in \mathbf{R^v}$ represents the i-th visual feature belonging to class c, v is the dimensionality of a visual feature. We next denote semantic feature of class c by $e_c \in \mathbf{R^s}$, where s is the dimensionality of a semantic feature. After obtaining C pairs of visual prototype and semantic feature $\{(p_c, e_c) \mid c = 1, 2, \cdots, C\}$, for each dimension of visual prototype, we will train an individual SVR (Smola and Schölkopf 2004) with RBF kernel, the SVR used to predict the j-th dimension of the visual prototype is as follows:

$$\begin{aligned} \min_{w^j, \beta^j, \bar{\beta^j}, \delta} & \frac{1}{2}\left\|w^j\right\|^2 + \alpha\left(\frac{1}{C}\sum_{c=1}^{C}\left(\beta_c^j + \bar{\beta_c^j}\right)\right) \\ s.t. & (w^j)^T\Phi^j\left(e_c\right) - p_c^j \le \delta + \beta_c^j \\ & p_c^j - (w^j)^T\Phi^j\left(e_c\right) \le \delta + \bar{\beta_c^j} \\ & \beta_c^j \ge 0, \bar{\beta_c^j} \ge 0,\ c = 1, 2, \cdots, C \end{aligned} \tag{1}$$

where $\Phi^j\left(e_c\right)$ is the implicit semantic feature of class c in kernel space, w^j is trainable linear weight of the SVR. p_c^j is the j-th dimension of the visual prototype of class c, δ is the margin, indicating any error less than it will not be counted, α is a penalty parameter. Note that each SVR takes s-dimension semantic feature as input and output 1-dimension visual feature. Hence, v SVRs can be trained independently so that they could be optimized parallelly to better capture relationship between semantic feature and every dimensions of visual prototype. Considering that the

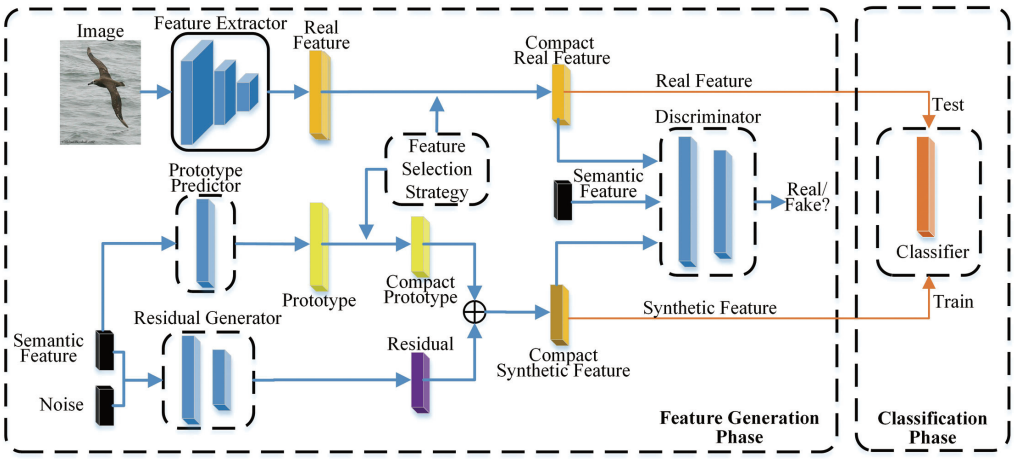

Figure 3: Pipeline of the proposed method.

number of seen classes is probably smaller than the dimensionality of semantic feature, to avoid overfitting, the dimensionality of semantic feature will be firstly reduced before being fed into SVR. With the trained SVRs, given semantic features of seen classes and unseen classes, we predict visual prototypes of both of them.

Predicting Compact Semantic Visual Prototype with Feature Selection Strategy. The predicted visual prototypes can achieve a high performance only if visual features are consistent with semantic features. However, as mentioned before, visual features and semantic features are not inherently matching, so that it is inevitable that the predicted visual prototypes have error compared to the real prototypes. This error was ignored by previous methods, however, we argue that visual prototype dimension with smaller error is one that is better consistent with semantic feature. Hence, we propose a simple yet effective feature selection strategy which selects the $Top\text{-}K$ visual prototype dimensions with relatively smaller prediction errors to build a compact semantic visual prototype as:

$$[j_1, \cdots, j_k, \cdots, j_v] = argsort\left[\sum_{c=1}^{C}(\Gamma^j(e_c) - p_c^j)^2\right] \quad (2)$$
$$p'_c = p_c\left[j_1, \cdots, j_k\right],\ c = 1, 2, \cdots, C$$

where e_c is the semantic feature of class c, $\Gamma^j()$ is the SVR used to predict the j-th dimension of visual prototype, p_c^j is the j-th dimension of visual prototype of class c, and $[j_1, \ldots, j_k, \cdots, j_v]$ is the index of visual prototype dimensions which rank in ascending order according to their prediction error, p'_c is the $Top\text{-}K$ dimensions of p_c with smallest error, K is a parameter which we fixed at $\frac{v}{2}$ in our experiments.

Compact Visual Feature Learning from Adversarial Feature Residual

Here, we describe how to learn compact semantic visual feature from adversarial feature residual in the proposed AFRNet. We employ the residual generator to generate visual feature residual, and then synthesize the compact semantic visual feature by combining the residual and the compact semantic visual prototype predicted by the aforementioned prototype predictor. The discriminator tries to distinguish the synthetic visual features from the real ones. After the adversarial training, we finally can synthesize compact semantic visual features. In this section, we begin with the general visual feature generation method as it is the basis of the proposed method, and then explain the proposed AFRNet in detail.

The general visual feature generation method employs the conditional WGAN (Arjovsky, Chintala, and Bottou 2017) to generate visual features conditioned on their corresponding semantic features as:

$$\begin{aligned}\min_G \max_D V = & E\left[D\left(x, e\left(y\right)\right)\right] - E\left[D\left(\hat{x}, e\left(y\right)\right)\right] \\ & -\lambda E\left[\left(\left\|\nabla_{\bar{x}} D\left(\bar{x}, e\left(y\right)\right)\right\|_2 - 1\right)^2\right]\end{aligned} \quad (3)$$

where G and D represent generator and discriminator respectively, which are both implemented by multi-layer perceptrons, x is real visual feature, $\hat{x} = G\left(z, e\left(y\right)\right)$ is synthetic visual feature conditionally generated from noise z and semantic feature $e\left(y\right)$ by generator G, $\bar{x} = \zeta x + \left(1-\zeta\right)\hat{x}$ with $\zeta \sim U\left(0,1\right)$ is used to estimate gradient. The objective of WGAN is to minimize Wasserstein distance which is implemented by the first two terms in Equation (3), and the last term is the gradient penalty of the discriminator, whose weight is controlled by a hyperparameter λ.

In the proposed AFRNet, the generator in conditional WGAN generates visual feature residuals instead of visual features. Specifically, given semantic feature e_y and its corresponding compact semantic visual prototype p'_y, we employ conditional WGAN to generate visual feature residual r_y conditioned on e_y, then the compact semantic visual feature is synthesized by combining the residual r_y with the compact semantic visual prototype p'_y. The proposed

method can be formalized as:

$$\begin{aligned}\min_G \max_D V_r =& E\left[D\left(x, e(y)\right)\right] \\ &- E\left[D\left(r_y + p'_y, e(y)\right)\right] \\ &- \lambda E\left[\left(\left\|\nabla_{\bar{x}_r} D\left(\bar{x}_r, e\left(y\right)\right)\right\|_2 - 1\right)^2\right]\end{aligned} \tag{4}$$

where p'_y is predicted by semantically compact prototype predictor, $r_y = G\left(z, e\left(y\right)\right)$ is generated by residual generator conditioned on its semantic feature. $(r_y + p'_y)$ is the synthetic compact semantic visual feature. $\bar{x}_r = \zeta x + (1-\zeta)\left(r_y + p'_y\right)$ with $\zeta \sim U\left(0, 1\right)$ is used to estimate gradient. The rest is similar to Equation (3).

After obtaining the synthetic visual features, the discriminator tries to distinguish the synthetic visual features from the real visual features. As the adversarial training goes, we end up with a powerful AFRNet which can synthesize compact semantic visual features that not only have less overlap but also are more consistent with semantic features.

Classification

Once the adversarial network has been trained, lots of visual features of unseen classes could be synthesized, associating with their labels. Then, ZSL is converted into a supervised classification problem. We could employ a naive softmax classifier as:

$$\min_\theta L\left(\theta\right) = -\frac{1}{N}\sum_{i=1}^{N} log\ p\left(y_i \mid x_i; \theta\right) \tag{5}$$

where θ is trainable linear transformation weight and $p(y_i \mid x_i; \theta) = \prod_{l=1}^{C}(\frac{exp(\theta_l^T x_i)}{\sum_{j=1}^{C} exp(\theta_j^T x_i)})^{1(y_i=l)}$. In testing phase, given a visual feature x, it is classified by:

$$f\left(x\right) = \arg\max_y p\left(y \mid x; \theta\right) \tag{6}$$

Comparison to Related Works

Here, we compare the proposed method with the related works. Different from (Zhu et al. 2018; Xian et al. 2018b; Li et al. 2019; Paul, Krishnan, and Munjal 2019), which all used GAN to generate visual features, we employ GAN to generate visual feature residuals, and then synthesize visual features by integrating the residuals and predicted visual prototypes. Since visual prototypes are explicitly predicted for both seen and unseen classes and the generated residuals are generally numerically much smaller than the distances among all the prototypes, the synthetic visual features are expected to be less overlapped for both seen and unseen classes.

Changpinyo et al. (Changpinyo, Chao, and Sha 2017) proposed to predict visual exemplars of unseen classes from semantic embeddings, and used the whole predicted visual exemplar either as visual training data or as ideal semantic embedding. Different from them, the proposed method adaptively selects some visual feature dimensions from the whole predicted visual prototype to build a compact semantic visual prototype, and applies this visual prototype to synthesize visual feature by being combined with visual feature residual.

Table 1: Statistics of six ZSL datasets. Att = Attributes, TF = TF-IDF feature, SCS = SCS-split, SCE = SCE-split, PS = PS-split, S = Seen classes, U = Unseen classes, P = Part feature, R = Res101 feature.

Dataset	Image	Att	TF	Feat	SCS		SCE		PS	
					S	U	S	U	S	U
CUB	11,788	-	7,551	P	150	50	160	40	-	-
NAB	49,562	-	13,217	P	323	81	323	81	-	-
APY	15,339	64	-	R	-	-	-	-	20	12
AWA1	37,475	85	-	R	-	-	-	-	40	10
AWA2	37,322	85	-	R	-	-	-	-	40	10
SUN	14,340	102	-	R	-	-	-	-	645	72

Experimental Results

Experimental Setup

Datasets. The proposed method is evaluated on the following six public datasets: Caltech USCD Birds-2011 (CUB) (Wah et al. 2011), North America Birds (NAB) (Van Horn et al. 2015), APascal-aYahoo (APY) (Farhadi et al. 2009), Animals with Attributes (AWA1) (Lampert, Nickisch, and Harmeling 2013), renewed Animals with Attributes (AWA2) (Xian et al. 2018a) and SUN attributes (SUN) (Patterson and Hays 2012). These datasets are of different scales and their statistics are summarized in Table 1. Note that CUB and NAB are two fine-grained datasets.

Visual and Semantic Feature. In order to make fair comparison, for APY, AWA1, AWA2 and SUN, as done in (Xian et al. 2018a), we use the 2048-D global features extracted by ResNet-101 (He et al. 2016) which is pre-trained on ImageNet1000 as the visual features, and attributes as the semantic features. For the CUB and NAB, as done in (Elhoseiny et al. 2017; Zhu et al. 2018; Ji et al. 2018), we use features which are merged with local features of several semantic regions of objects as the visual features, and Term Frequency-Inverse Document Frequency (TF-IDF) features as the semantic features. The TF-IDF features are commonly used for text description, which could represent the semantic feature of class-level text description. Specifically, the visual feature is 3584-D and 3072-D for CUB and NAB respectively, and we call these visual features extracted from local regions 'part feature'. Following the previous works, we also transform the original TF-IDF feature to 200-D and 400-D via linear PCA operation for CUB and NAB respectively. The statistics of visual features and semantic features are reported in Table 1.

Evaluation Protocol. As most methods did, we evaluate the proposed method by computing average per-class Top-1 accuracy (ACC). In the conventional ZSL setting, we compute ACC of unseen classes. In the generalized ZSL setting, we compute ACCs of both seen classes and unseen classes and compute harmonic mean of seen and unseen accuracy. In addition, data split has a huge impact on performance. As suggested by (Elhoseiny et al. 2017), on CUB and NAB, we evaluate the proposed method via SCS-split and SCE-split. Note that SCE-split is harder than SCS-split as the parent categories of unseen classes are exclusive to those of seen classes in SCE-split. Also note that few unseen classes in SCS-split have been seen by the pre-trained ImageNet1000 model, we use the same ImageNet1000 model

as the other methods for fair comparison. Following (Xian et al. 2018a), we evaluate the proposed method on the APY, AWA1, AWA2 and SUN datasets with PS-split. The PS-split where unseen classes presented in ImageNet1000 are replaced with other classes is an improved version of the original SS-split. The detailed split information is reported in Table 1.

Comparison Methods. For comparison, we cite the results (reported in the corresponding papers) of fourteen existing methods on the APY, AWA1, AWA2 and SUN datasets, including DAP (Lampert, Nickisch, and Harmeling 2013), DEVISE (Frome et al. 2013), ALE (Akata et al. 2015a), ES-ZSL (Romera-Paredes and Torr 2015), LATEM (Xian et al. 2016), SSE (Zhang and Saligrama 2015), SYNC (Changpinyo et al. 2016), SAE (Kodirov, Xiang, and Gong 2017), DEM (Zhang, Xiang, and Gong 2017), GAZSL (Zhu et al. 2018), f-CLSWGAN (Xian et al. 2018b), SR-GAN (Ye et al. 2019), SABR (Paul, Krishnan, and Munjal 2019), LiGAN (Li et al. 2019). Similarly, we also list the results of seven existing methods on the fine-grained CUB and NAB datasets, including WAC-kernel (Elhoseiny, Elgammal, and Saleh 2016), ESZSL (Romera-Paredes and Torr 2015), ZSLNS (Qiao et al. 2016), SYNC (Changpinyo et al. 2016), ZSLPP (Elhoseiny et al. 2017), GAZSL (Zhu et al. 2018), SGA-DET (Ji et al. 2018).

Implementation Details. In the proposed method, prototype prediction is implemented by SVR with RBF kernel. The generator and discriminator are both three-layer MLP with ReLU activation, which both employ 4096 units in hidden layer. Hyper-parameters in WGAN are set as they are suggested by the author.

Performance in Conventional ZSL Setting

Since most state-of-the-art methods have been evaluated on APY, AWA1, AWA2 and SUN in the conventional ZSL setting, we first evaluate the proposed method on these datasets with PS-split and then compare it with fourteen state-of-the-art methods. All these methods are tested with Res101 features and attributes. Results are reported in Table 2. From Table 2, we can easily find out that the proposed method significantly outperforms all the existing methods. Specifically, the proposed method achieves an improvement about 12.2% on APY, 4.4% on AWA1, 8.0% on AWA2, 1.2% on SUN. The reason why the improvement on SUN is smaller than the others is probably that the visual prototypes on SUN are harder to be predicted due to the fact that it has more classes and less per-class images.

For a more detailed evaluation, we also test the proposed method on two fine-grained datasets, CUB and NAB. We then compare it with seven recent state-of-the-art methods. All these methods are tested using part features and TF-IDF features. To make evaluation more challenging, we evaluate these methods with both easier SCS-split and harder SCE-split. Table 3 shows us the results. As shown in Table 3, the proposed method outperforms all the competitors with a significant performance gain. Specifically, on CUB, we achieve performance gain 6.6% and 9.6% under SCS-split and SCE-split. Significantly, the accuracy under SCE-split (20.5%) is about 90% higher than that of previous state-of-the-art (10.9%) on CUB. We visualize features of 10 unseen classes from CUB as shown in Figure 2 B, both the accuracy gain and the visualization indicate that the AFRNet can generate visual features with less overlap. The gain on NAB is slightly smaller than on CUB, which is 3.4% and 3.1% under SCS-split and SCE-split, this is probably because NAB is a larger dataset with 404 categories, which means inter-class difference on NAB is relatively subtle.

Table 2: Comparative results (Top-1 accuracy) in the conventional ZSL setting on APY, AWA1, AWA2 and SUN.

Method	APY	AWA1	AWA2	SUN
DAP	33.8	44.1	46.1	39.9
DEVISE	39.8	54.2	59.7	56.5
ALE	39.7	59.9	62.5	58.1
ESZSL	38.3	58.2	58.6	54.5
LATEM	35.2	55.1	55.8	55.3
SSE	34.0	60.1	61.0	51.5
SYNC	23.9	54.0	46.6	56.3
SAE	8.3	53.0	54.1	40.3
DEM	35.0	68.4	67.1	61.9
GAZSL	41.1	68.2	-	61.3
f-CSLWGAN	40.5	68.2	-	60.8
SR-GAN	44.0	72.0	-	62.3
SABR	-	-	65.2	62.8
LiGAN	43.1	70.6	-	61.7
AFRNet(Ours)	**56.2**	**76.4**	**75.1**	**64.0**

Table 3: Comparative results (Top-1 accuracy) in the conventional ZSL setting on the fine-grained CUB and NAB datasets.

Method	CUB		NAB	
	SCS	SCE	SCS	SCE
WAC-kernel	33.5	7.7	11.4	6.0
ESZSL	28.5	7.4	24.3	6.3
ZSLNS	29.1	7.3	24.5	6.8
SynC	28.0	8.6	18.4	3.8
ZSLPP	37.2	9.7	30.3	8.1
GAZSL	43.7	10.3	35.6	8.6
SGA-DET	42.9	10.9	39.4	9.7
AFRNet(Ours)	**50.3**	**20.5**	**42.8**	**12.8**

Performance in Generalized ZSL Setting

We evaluate the proposed method on APY, AWA1, AWA2 and SUN with PS-split in the generalized ZSL setting. Then, we conduct comparison with fourteen state-of-the-art methods. All these methods are evaluated using Res101 features and attributes. Results are reported in Table 4. Similar to results in the conventional ZSL setting, the proposed method achieves a significantly better performance than previous methods: 58.9% vs 45.7% on APY, 68.8% vs 62.3% on AWA1, 70.1% vs 46.9% on AWA2. On SUB, the proposed method only reaches the state-of-the-art performance probably because visual prototypes are harder to be predicted on SUB. In addition, accuracy of seen classes and that of unseen classes are better balanced in the proposed method than other methods, which informs us the proposed method has

Table 4: Comparative results in the generalized ZSL setting on APY, AWA1, AWA2 and SUN. U = Top-1 accuracy of unseen classes, S = Top-1 accuracy of seen classes, H = Harmonic mean of unseen and seen classes accuracy.

Method	APY			AWA1			AWA2			SUN		
	U	S	H	U	S	H	U	S	H	U	S	H
DAP	4.8	78.3	9.0	0.0	88.7	0.0	0.0	84.7	0.0	4.2	25.1	7.2
DEVISE	4.9	76.9	9.2	13.4	68.7	22.4	17.1	74.7	27.8	16.9	27.4	20.9
ALE	4.6	73.7	8.7	16.8	76.1	27.5	14.0	81.8	23.9	21.8	33.1	26.3
ESZSL	2.4	70.1	4.6	6.6	75.6	12.1	5.9	77.8	11.0	11.0	27.9	15.8
LATEM	0.1	73.0	0.2	7.3	71.7	13.3	11.5	77.3	20.0	14.7	28.8	19.5
SSE	0.2	78.9	0.4	7.0	80.5	12.9	8.1	82.5	14.8	2.1	36.4	4.0
SYNC	7.4	66.3	13.3	8.9	87.3	16.2	10.1	90.5	18.0	7.9	43.3	13.4
SAE	0.4	80.9	0.9	1.8	77.1	3.5	1.1	82.2	2.2	8.8	18.0	11.8
DEM	11.1	75.1	19.4	30.5	86.4	45.1	30.5	86.4	45.1	20.5	34.3	25.6
GAZSL	14.2	78.6	24.0	19.2	86.5	31.4	-	-	-	21.7	34.5	26.7
f-CLSWGAN	32.9	61.7	42.9	57.9	61.4	59.6	-	-	-	42.6	36.6	39.4
SR-GAN	22.3	78.4	34.8	41.5	83.1	55.3	-	-	-	22.1	38.3	27.4
SABR	-	-	-	-	-	-	30.3	93.9	46.9	50.7	35.1	**41.5**
LiGAN	34.3	68.2	45.7	52.6	76.3	62.3	-	-	-	42.9	37.8	40.2
AFRNet(Ours)	48.4	75.1	**58.9**	68.2	69.4	**68.8**	66.7	73.8	**70.1**	46.6	37.6	**41.5**

Table 5: Results with/without AFRNet-style feature generation method.

Method	CUB		NAB	
	SCS	SCE	SCS	SCE
AFRNet-non	41.6	9.1	37.1	5.7
AFRNet	48.7	18.6	41.5	12.7

Table 6: Results with/without feature selection strategy. 1NN and AFR refer to 1NN classifier and AFRNet.

Method	CUB				NAB			
	SCS		SCE		SCS		SCE	
	1NN	AFR	1NN	AFR	1NN	AFR	1NN	AFR
w/o	44.3	48.7	16.3	18.6	35.2	41.5	9.4	12.7
w	48.7	50.3	18.2	20.5	38.2	42.8	9.8	12.8

a better generalization to unseen classes. From Table 2 to Table 4, we also note that all the GAN-based methods consistently outperform other methods. Among all the GAN-based methods, the proposed method achieves the best performance, this indicates that the AFRNet method and feature selection strategy are effective to improve ZSL, which we will detailedly analysis in the Ablation Study section.

Ablation Study

Effect of Feature Generation Method. To prove benefit of the feature generation method proposed in the AFRNet, comparison experiments are conducted on CUB and NAB under both SCS-split and SCE-split using AFRNet-style feature generation method (AFRNet method) and non-AFRNet-style method (AFRNet-non method). The results are reported in Table 5. It is obvious that AFRNet method achieves a significant performance gain against AFRNet-non method: 7.1% and 9.5% under SCS-split and SCE-split on CUB; 4.4% and 7.0% under SCS-split and SCE-split on NAB. Notably, the accuracy of AFRNet method is more than 100% higher than that of AFRNet-non method under SCE-split on both CUB and NAB. This gain indicates that the AFRNet method can generate visual features with less overlap as shown in Figure 2 B, and these features could be used to train a more generalizable classifier. Note that both AFRNet method and AFRNet-non method in Table 5 have not employed feature selection strategy.

Effect of Feature Selection Strategy. To demonstrate benefit of the feature selection strategy (FSS), we conduct evaluation on both CUB and NAB under both SCS-split and SCE-split using both naive 1NN classifier and AFRNet. Results with or without FSS are reported in Table 6. We note that methods with FSS achieve better performance than methods without it whatever the evaluation settings are. This tells us that FSS is able to select visual feature dimensions that are better consistent with semantic features.

Conclusion

We propose a novel adversarial network called AFRNet to learn compact semantic visual features for ZSL. Unlike existing feature generation methods, the proposed AFRNet generates visual feature residual, and then synthesizes the visual feature by integrating the residual with the predicted visual prototype. Consequently, the synthetic visual features are less overlapped and classifier trained on these features is more generalizable. In addition, on the basis of existing prototype prediction method, we propose a novel feature selection strategy which can adaptively select semantically consistent visual feature elements from the original visual feature. The proposed method is proved to outperform existing state-of-the-art methods with a significant improvement by extensive experimental results on six benchmarks datasets.

Acknowledgments

This work was supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (XDB32070100) and National Natural Science Foundation of China (U1805264, 61421004, 61573359). We thank the anonymous reviewers so much for their helpful comments and suggestions.

References

Akata, Z.; Perronnin, F.; Harchaoui, Z.; and Schmid, C. 2015a. Label-embedding for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38(7):1425–1438.

Akata, Z.; Reed, S.; Walter, D.; Lee, H.; and Schiele, B. 2015b. Evaluation of output embeddings for fine-grained image classification. In *CVPR*, 2927–2936.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875*.

Changpinyo, S.; Chao, W.-L.; Gong, B.; and Sha, F. 2016. Synthesized classifiers for zero-shot learning. In *CVPR*, 5327–5336.

Changpinyo, S.; Chao, W.-L.; and Sha, F. 2017. Predicting visual exemplars of unseen classes for zero-shot learning. In *ICCV*, 3476–3485.

Elhoseiny, M.; Zhu, Y.; Zhang, H.; and Elgammal, A. 2017. Link the head to the " beak": Zero shot learning from noisy text description at part precision. In *CVPR*, 6288–6297.

Elhoseiny, M.; Elgammal, A.; and Saleh, B. 2016. Write a classifier: Predicting visual classifiers from unstructured text. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(12):2539–2553.

Farhadi, A.; Endres, I.; Hoiem, D.; and Forsyth, D. 2009. Describing objects by their attributes. In *CVPR*, 1778–1785.

Frome, A.; Corrado, G. S.; Shlens, J.; Bengio, S.; Dean, J.; Mikolov, T.; et al. 2013. Devise: A deep visual-semantic embedding model. In *NIPS*, 2121–2129.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NIPS*, 2672–2680.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Ji, Z.; Fu, Y.; Guo, J.; Pang, Y.; Zhang, Z. M.; et al. 2018. Stacked semantics-guided attention model for fine-grained zero-shot learning. In *NIPS*, 5995–6004.

Kodirov, E.; Xiang, T.; and Gong, S. 2017. Semantic autoencoder for zero-shot learning. In *CVPR*, 3174–3183.

Lampert, C. H.; Nickisch, H.; and Harmeling, S. 2013. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36(3):453–465.

Li, J.; Jin, M.; Lu, K.; Ding, Z.; Zhu, L.; and Huang, Z. 2019. Leveraging the invariant side of generative zero-shot learning. *arXiv preprint arXiv:1904.04092*.

Norouzi, M.; Mikolov, T.; Bengio, S.; Singer, Y.; Shlens, J.; Frome, A.; Corrado, G. S.; and Dean, J. 2013. Zero-shot learning by convex combination of semantic embeddings. *arXiv preprint arXiv:1312.5650*.

Patterson, G., and Hays, J. 2012. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, 2751–2758.

Paul, A.; Krishnan, N. C.; and Munjal, P. 2019. Semantically aligned bias reducing zero shot learning. In *CVPR*, 7056–7065.

Qiao, R.; Liu, L.; Shen, C.; and Van Den Hengel, A. 2016. Less is more: zero-shot learning from online textual documents with noise suppression. In *CVPR*, 2249–2257.

Romera-Paredes, B., and Torr, P. 2015. An embarrassingly simple approach to zero-shot learning. In *ICML*, 2152–2161.

Smola, A. J., and Schölkopf, B. 2004. A tutorial on support vector regression. *Statistics and computing* 14(3):199–222.

Socher, R.; Ganjoo, M.; Manning, C. D.; and Ng, A. 2013. Zero-shot learning through cross-modal transfer. In *NIPS*, 935–943.

Van Horn, G.; Branson, S.; Farrell, R.; Haber, S.; Barry, J.; Ipeirotis, P.; Perona, P.; and Belongie, S. 2015. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *CVPR*, 595–604.

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

Xian, Y.; Akata, Z.; Sharma, G.; Nguyen, Q.; Hein, M.; and Schiele, B. 2016. Latent embeddings for zero-shot classification. In *CVPR*, 69–77.

Xian, Y.; Lampert, C. H.; Schiele, B.; and Akata, Z. 2018a. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Xian, Y.; Lorenz, T.; Schiele, B.; and Akata, Z. 2018b. Feature generating networks for zero-shot learning. In *CVPR*, 5542–5551.

Ye, Z.; Lyu, F.; Li, L.; Fu, Q.; Ren, J.; and Hu, F. 2019. Srgan: Semantic rectifying generative adversarial network for zero-shot learning. In *ICME*, 85–90.

Zhang, Z., and Saligrama, V. 2015. Zero-shot learning via semantic similarity embedding. In *CVPR*, 4166–4174.

Zhang, L.; Xiang, T.; and Gong, S. 2017. Learning a deep embedding model for zero-shot learning. In *CVPR*, 2021–2030.

Zhu, Y.; Elhoseiny, M.; Liu, B.; Peng, X.; and Elgammal, A. 2018. A generative adversarial approach for zero-shot learning from noisy texts. In *CVPR*, 1004–1013.