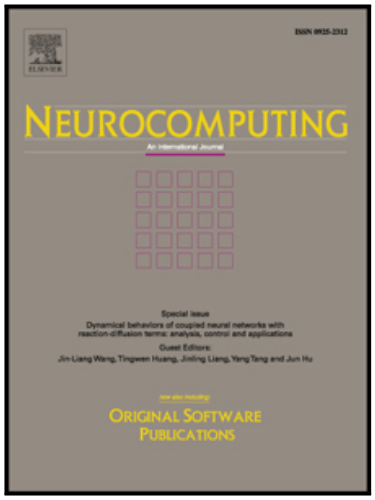
Communicated by Dr Li Sheng

Accepted Manuscript



Adversarial Unseen Visual Feature Synthesis for Zero-shot Learning

Haofeng Zhang, Yang Long, Li Liu, Ling Shao

PII: S0925-2312(18)31232-3
DOI: https://doi.org/10.1016/j.neucom.2018.10.043
Reference: NEUCOM 20065

To appear in: *Neurocomputing*

Received date: 3 June 2018
Revised date: 4 September 2018
Accepted date: 21 October 2018

Please cite this article as: Haofeng Zhang, Yang Long, Li Liu, Ling Shao, Adversarial Unseen Visual Feature Synthesis for Zero-shot Learning, *Neurocomputing* (2018), doi: https://doi.org/10.1016/j.neucom.2018.10.043

Adversarial Unseen Visual Feature Synthesis for Zero-shot Learning

Haofeng Zhang[a,*], Yang Long[b], Li Liu[c], Ling Shao[c,d]

[a]*School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China*
[b]*Open Lab, Newcastle University, Newcastle upon Tyne, UK*
[c]*Inception Institute of Artificial Intelligence (IIAI), Abu Dhabi, United Arab Emirates*
[d]*School of Computing Sciences, University of East Anglia, Norwich, UK*

Abstract

Due to the extreme imbalance of training data between seen classes and unseen classes, most existing methods fail to achieve satisfactory results in the challenging task of Zero-shot Learning (ZSL). To avoid the need for labelled data of unseen classes, in this paper, we investigate how to synthesize visual features for ZSL problem. The key challenge is how to capture the realistic feature distribution of unseen classes without training samples. To this end, we propose a hybrid model consists of Random Attribute Selection (RAS) and conditional Generative Adversarial Network (cGAN). RAS aims to learn the realistic generation of attributes by their correlations in nature. To improve the discrimination for the large number of classes, we add a reconstruction loss in the generative network, which can solve the domain shift problem and significantly improve the classification accuracy. Extensive experiments on four benchmarks demonstrate that our method can outperform all the state-of-the-art methods. Qualitative results show that, compared to conventional generative models, our method can capture more realistic distribution and remarkably improve the variability of the synthesized data.

Keywords: Zero Shot Learning, Generative Adversary Network, Random

*Corresponding author
Email addresses: zhanghf@njust.edu.cn (Haofeng Zhang), yang.long@ieee.org (Yang Long), liuli1213@gmail.com (Li Liu), ling.shao@ieee.org (Ling Shao)

1. Introduction

Conventional image classification relies on supervised learning with sufficient training samples for each category. Due to the fast increase of new concepts, collecting high-quality data for each of them is infeasible. Towards intelligent image classification, Zero-shot Learning (ZSL) [1, 2, 3, 4, 5, 6, 7] aims to learn a classification model with limited training classes, but in the hope of transferring to novel unseen classes. Conventional zero-shot learning methods rely on projecting visual features into semantic embedding space in order to infer the class labels through pre-defined human knowledge, such as attributes. However, since the projection is learned based on the seen classes only, the learnt models often suffer from the severe domain shift problem, *i.e.* the classification has strong bias towards seen classes. Therefore, despite recent zeal on ZSL, most of existing work is based on the unrealistic assumption that all of test images come from unseen classes. How to classify images from both seen and unseen classes remains challenging, which is known as Generalised Zero-shot Learning (GZSL).

A recent survey [8] shows that most of state-of-the-art ZSL approaches suffer from sever performance degradation. The first proposal of GZSL [9] considers to use anomaly detection to first differentiate seen and unseen classes, and then apply conventional ZSL approaches. Recently, a promising solution is to generate unseen visual data from semantic attributes so as to convert ZSL into a conventional supervised classification. In this way, seen and unseen classes are trained together and the bias is mitigated. Long *et al.* studied a embedding framework from attributes to visual features with visual-semantic structure preservation [10]. However, their approach requires expensive instance-level attributes. Y. Guo *et al.* estimated the the probability distribution of unseen classes by using the knowledge from seen classes and the class attributes [11], and then synthesised samples based on the distribution. This method needs to

assume a certain distribution for the unseen data, *e.g* the Gaussian distribution, which is distinctive to the reality and leads to un reliable classifiers.

Therefore, the core issue is: *How can we capture the realistic distribution of unseen images?* Such a problem is similar to human imagination. Given a semantic description, humans can imagine how does the object looks like. Similarly, Generative Adversarial network (GAN) [12] trains a generative network and a discriminative network, where discriminative network intends to classify real data from synthesised data, while generative network tries to generate fake data to cheat the discriminative network. Inspired by GAN, we consider to generate unseen data from semantic attributes. To address the problem of domain shift, we extend the conditional GAN with a reconstruction loss and a classification loss. In addition, to solve the problem that the generated data has the same distribution with the initialised noises, we embed a policy called Random Attribute Selection (RAS) to process the conditional class attribute during synthesising unseen new data. RAS selects maximal correlated attribute entries randomly according to the attributes correlation matrix, and cuts down all the left entries.

Without losing the generality, we carry out our experiments for both ZSL and GZSL on four benchmarks. Detailed analysis on synthesised data distribution and the importance of reconstruction item are also performed to convince the effectiveness of our method. It is worthwhile to list the contributions of our method:

a) We propose a novel method for zero-shot learning, which construct a conditional generative network to synthesis unseen class features from attributes. Hereafter, these features can be used to train a conventional supervised classifier for image recognition.

b) Reconstruction loss are added to the generative network to solve the domain shift problem. With this constraint, the synthesised features are much more accurate than those generated without it.

c) To solve the problem that the synthesised features have the same distribution

as the input noises, we propose a strategy called random attribute selection, which is used to choose the most correlated attribute entries randomly to reconstruct the unseen class features. This strategy can generate more similar features as real ones.

d) The experiments on four popular datasets for both ZSL and GZSL show that our method is more effective than the state-of-the-art methods.

The rest of this paper is organized as follows. In Section *'Related Work'*, we give a brief review of recent zero-shot learning methods and generative adversarial network. The details of our method for common attributes annotation and projection models are both described in Section *'Methodology'*. Section *'Experiments'* reports the experimental results on ZSL and GZSL, and analysis the distribution of synthesised data and the importance of reconstruction item in detail. Finally, we conclude this paper and discuss the probable future works in Section *'Conclusion'*.

2. Related Works

Zero-shot Learning Since visual attribute learning has been proposed, many researchers [13, 14, 15] conduct their work on how to find the intermediate attribute classifiers for zero-shot learning. Compatibility learning is the most popular framework, which learns linear or non-linear mapping functions using only seen data and attributes, and apply on unseen data. Direct Attribute Prediction (DAP) [16] is one of the earliest compatibility frameworks, which learns probabilistic attribute classifiers and estimate the label by integrating the ranks of the learnt classifiers. Label Embedding (ALE) [13], Structured Joint Embedding (SJE) [17], and Deep Visual-Semantic Embedding (DeViSE) [18] employ bilinear compatibility function to project features into semantic embedding space, where the features and attributes belongs to same class have maximal correlation, otherwise have minimal correlation. Latent Space Encoding (LSE) [19] exploits an encoder-decoder to connect the semantic relations of different modalities. In addition, Z. Ji *et al.* proposed a method called Manifold

regularized Cross-Modal Embedding (MCME) [20] to preserve the locally visual structure in the embedding process by formulating the manifold constraint for intrinsic structure of the visual features as well as aligning pairwise consistency. There are also some non-linear compatibility learning frameworks [21], which extends linear models into non-linear ones to improve the recognition accuracy.

Since it is not available to obtain the distribution of unseen classes in compatibility learning, transductive learning related methods [22, 23, 24, 25] were proposed to use the unseen data in training process to solve the domain shift problem. Though this type of methods can greatly improve the classification accuracy, the setting of it violates the original purpose that the unseen data is strictly not accessible during training.

Synthetic learning is a novel type method, which synthesis pseudo features from semantic attributes, and training classifiers using conventional algorithms such as Decision Tree (DT), Support Vector Machine (SVM). Unseen Visual Data Synthesis (UVDS) [26] and Adversarial Sample Synthesis (ASS) [27, 28, 29, 30] are partial typical methods of this type. Our method also belongs to this type.

ZSL related methods often rely on the intermediate attributes, which represent the semantic embeddings of both seen and unseen classes. Conventional attributes [31] are high dimensional, and usually annotated by experts with real values, Demire *et al.* [32] turn to use Word2Vec [33] to generate attributes based on the dataset *'Wikipedia'*. Another semantic attribute representation is based on similarity, which can be annotated by humans [34] or the textual descriptions [32].

Generative Adversarial Network GAN is a very interesting learning method, which can generate synthesised samples with noise input. GAN was first proposed by I. Goodfellow *et al.*[12], till now there are a large quantity of impressive progresses have been achieved, *e.g.* image generation [35], text generation [36], image editing [37] and conditional image generation such as text2image [38]. GAN's success depends on the variants of adversarial loss which tries to make the generated data to be indistinguishable from real images or features. As

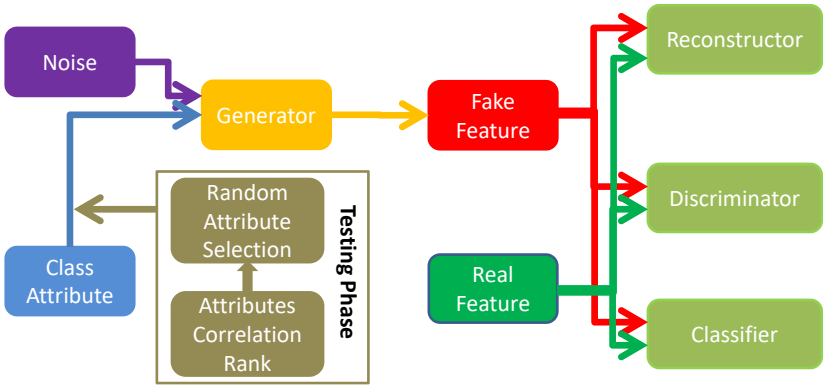


Figure 1: Illustration of our network structure for zero-shot learning. The training phase do not have the process of Random Attribute Selection (RAS), while the testing phase have it.

we know that GAN often fall into collapse, to handle this problem and make its training more stable, many training strategies have been proposed, such as Wasserstein GAN [39], least square GAN [40]. Furthermore, Cycle GAN [41] and Dual GAN [42] have been developed to address the problem of unpaired images training, which also described as unsupervised GAN.

3. Methodology

3.1. Notations

Let $\boldsymbol{Y} = \{y_1, \cdots, y_s\}$ and $\boldsymbol{Z} = \{z_1, \cdots, z_u\}$ denote a set of s seen and u unseen class labels, and they are disjoint $\boldsymbol{Y} \cap \boldsymbol{Z} = \emptyset$. Similarly, let $\boldsymbol{A_Y} = \{\boldsymbol{a_{y1}}, \cdots, \boldsymbol{a_{ys}}\} \in \mathbb{R}^{l \times s}$ and $\boldsymbol{A_Z} = \{\boldsymbol{a_{z1}}, \cdots, \boldsymbol{a_{zu}}\} \in \mathbb{R}^{l \times u}$ denote the corresponding s seen and u unseen class level attributes respectively. Given the training data in 3-tuple of N seen samples: $(\boldsymbol{x}_1, \boldsymbol{a}_1, \boldsymbol{y}_1), \cdots, (\boldsymbol{x}_N, \boldsymbol{a}_N, \boldsymbol{y}_N) \subseteq \boldsymbol{X}_s \times \boldsymbol{A_Y} \times \boldsymbol{Y}$, where $\boldsymbol{X}_s$ is d-dimensional features extracted from N seen images. When testing, the preliminary knowledge is u pairs of attributes and labels:$(\hat{\boldsymbol{a}}_1, \hat{\boldsymbol{z}}_1), \cdots, (\hat{\boldsymbol{a}}_u, \hat{\boldsymbol{z}}_u) \subseteq \boldsymbol{A_Z} \times \boldsymbol{Z}$. Zero-shot Learning aims to learn a classification function f: $\boldsymbol{X}_u \rightarrow \boldsymbol{Z}$ to predict the label of the input image from unseen classes, where $\boldsymbol{x}_i \in \boldsymbol{X}_u$ is totally unavailable during training.

3.2. Conditional GAN for ZSL

In this subsection, we will introduce our proposed generative model. As shown in Figure 1, our proposed model contains four parts: 1) the generative network G; 2) the discriminative network D; 3) the classification network C; 4) and the reconstruction network R.

The generative network G generates feature $\hat{\boldsymbol{x}}$ through sampling from a learned distribution $\mathrm{p}(\hat{\boldsymbol{x}}|\boldsymbol{z}, \boldsymbol{a}_c)$, where $\boldsymbol{a}_c$ is the class attribute of category c, and $\boldsymbol{z}$ is the randomly generated noise. The function of network G and D is the same as those in the conventional GAN. The network G intends to learn the real data distribution via the gradients computed by the discriminative network D, which learns to distinguish between *'real'* and *'fake'* samples. The function of network C is to calculate the posterior probability $\mathrm{p}(c|\boldsymbol{x})$. The function of network R is to preserve the structure of the generated samples by using the ℓ_2 loss.

In the training dataset, we have the prior knowledge that each feature belongs to a certain class among total K categories, so it is easy to train the classification network C with a standard full connection network. Taking the training sample $\boldsymbol{x}$ as input and K dimensional vector as output, the classifier C turns into computing class probabilities with a softmax function. Each entry of the output vector $\mathrm{p}(c|\boldsymbol{x})$ stands for the probability of each category of the input feature $\boldsymbol{x}$. In the training phase, the classifier C intent to minimise the softmax loss,

$$\mathcal{L}_C = -\mathbb{E}_{\boldsymbol{x}\sim \mathrm{p}_{data}(\boldsymbol{x})}[\mathrm{p}(c|\boldsymbol{x})]. \tag{1}$$

The classification network (Classifier) contains four full connection layers, which can be seen in Figure 2. The purpose of the discriminative network D is to distinguish real training data from synthesized feature, while the generative network G tries to deceive the discriminator D. Concretely, the network D

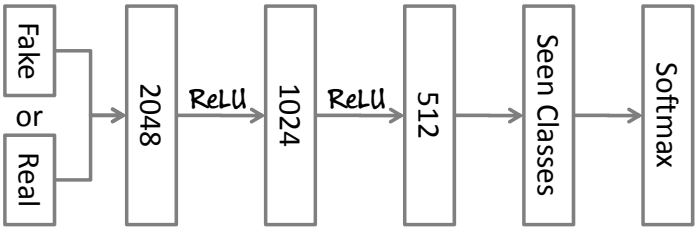


Figure 2: The architecture of classification network (Classifier).

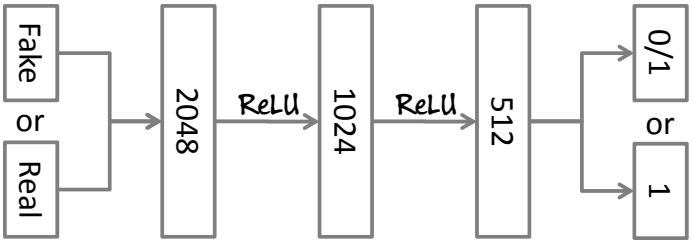


Figure 3: The architecture of discrimination network (Discriminator). The input of discriminator can be fake image or real image. The output is encouraged to generate 1 for the real image as input and 0 for the fake image as input respectively when training the discriminator. While training the generator, the output is only constrained to be 1.

should minimize the following loss function,

$$\begin{aligned}\mathcal{L}_D = &- \mathbb{E}_{\boldsymbol{x}\sim \mathrm{p}_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x})] \\ &- \mathbb{E}_{\boldsymbol{z}\sim \mathrm{p}_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{a}_c, \boldsymbol{z})))].\end{aligned} \tag{2}$$

The discrimination network (Discriminator) contains four full connection layers, which can be seen in Figure 3. The generative network G should have three objectives. Firstly, it should fool the discriminator D and make D recognise the synthesised feature as the real one, thus, the generator G should minimise the following loss function,

$$\mathcal{L}_{GD} = -\mathbb{E}_{\boldsymbol{z}\sim \mathrm{p}_z(\boldsymbol{z})}[\log D(G(\boldsymbol{a}_c, \boldsymbol{z}))]. \tag{3}$$

Secondly, the generated feature $\hat{\boldsymbol{x}}$ should also cheats the classifier and obtains

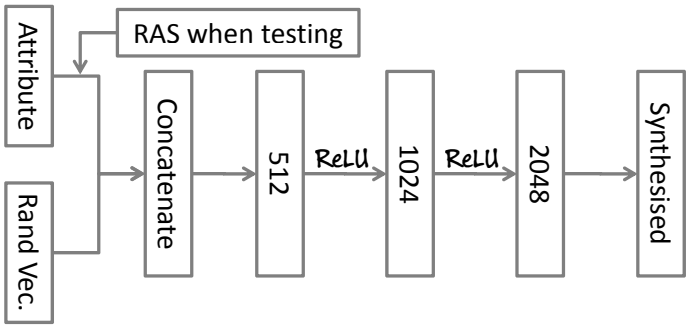


Figure 4: The architecture of generation network (Generator).

highest probability with its corresponding class, hence we try to minimise the classification loss with the following formulation,

$$\mathcal{L}_{GC} = -\mathbb{E}_{\boldsymbol{z}\sim \mathrm{p}_z(\boldsymbol{z})}[\mathrm{p}(c|G(\boldsymbol{a}_c, \boldsymbol{z}))]. \tag{4}$$

Finally, the synthesised feature should preserve the structure of the domain distribution and can reconstruct the original feature x, so we use the ℓ_2 loss to keep the constraint, and minimise the following function,

$$\mathcal{L}_{GR} = \|\boldsymbol{x} - G(\boldsymbol{a}_c, \boldsymbol{z})\|_F^2, \tag{5}$$

where, $\|\cdot\|_F$ denotes the Frobenius norm. We combine the above three constraints, and optimise them simultaneously with Equation 6,

$$G = \min_G \mathcal{L}_G = \min_G (\mathcal{L}_{GC} + \alpha\mathcal{L}_{GD} + \beta\mathcal{L}_{GR}), \tag{6}$$

where, α and β are the balance parameters to control the importance of the last two items. The generation network (Generator) also contains four full connection layers, which can be seen in Figure 4. The total network of the generator G, the discriminator D, and the classifier C can be iteratively optimized with Stochastic Gradient Descent (SGD). The total training procedure can be found in Algorithm 1.

Algorithm 1 Adversarial training of feature synthesis.

Input:

Training image set $\boldsymbol{X}_S$, Corresponding attributes $\boldsymbol{A}_Y$, Class labels $\boldsymbol{Y}$. Hyper-parameters: α, β, iteration times T;

Output:

Parameters of the Generator.

1: **for** each $i \in [1, T]$ **do**

2: Fix the Discriminator D and the Generator G, train the Classifier C with Equation 1;

3: Fix the Classifier C and the Generator G, train the Discriminator D with Equation 2;

4: Fix the Discriminator D and the Classifier C, train the Generator G with Equation 6;

5: **end for**

6: **return** the Parameters of the Generator;

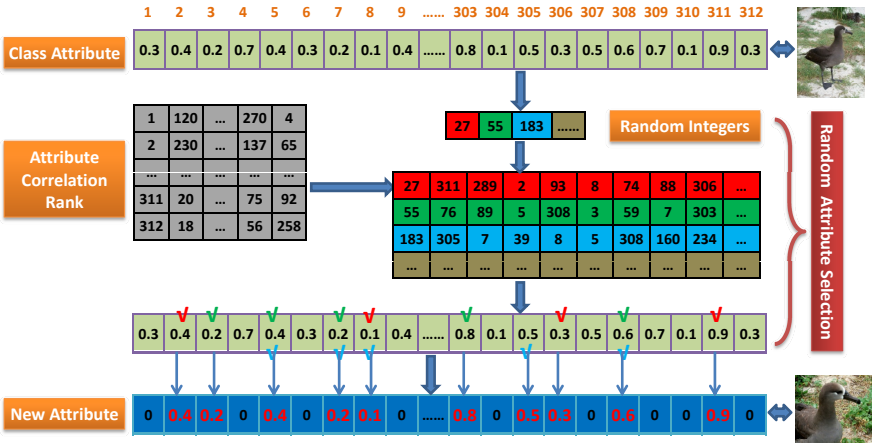


Figure 5: Illustration of the process of Random Attribute Selection (RAS) in our method.

3.3. Random Attribute Selection

Conventional GAN usually utilize noise as input to generate samples, which often causes the synthesised samples also have the same distribution as the input noise. For example, if the input noise follows Gaussian distribution, the output synthesized samples usually obey the same Gaussian distribution. However, in most realistic scenarios, we often cannot obtain what distribution the data should follow previously, thus, it is unreasonable to employ a fixed function to postulate the distribution of the data.

Generally speaking, each entry of class attribute vector has its realistic visual or semantic meaning, *e.g.*, the 23^{th} entry of the attribute for AWA [43] is *'paws'*, which represents for whether the animal has paws. Therefore, if we keep parts of the entries of a class attribute vector and set the left to 0, we will get a new attribute vector but corresponding to the same original class, which means that we create a new image which keep part of its original visual content, but still belongs to the same original class, *e.g.* in dataset CUB [44], if we keep the head and body related entries of a class attribute vector, and remove the foot related ones, it represents that the corresponding synthesized feature only have the head and body parts, but still belongs to the original bird type.

In real scenarios, the entries of a class attribute often have correlations with each other, *e.g.* in dataset AWA [43], the attribute unit *'domestic'* often has great relationship with the unit *'ground'*. Thus, we compute the attribute correlation using $\boldsymbol{R} = \boldsymbol{A}^T\boldsymbol{A}$ and sort each row of $\boldsymbol{R}$ in descending order. We randomly generate k_1 integers, and find the corresponding rows in $\boldsymbol{R}$. Furthermore, the top k_2 positions of the found rows are extracted as the kept values, which are exploited to reserve the corresponding entries of original class attribute, and set the left to zero. The total process of Random Attribute Selection (RAS) is illustrated in Figure 5.

During training, the process of RAS is not included in the total network, because the processed attributes can not well match with the features, which will result in bad reconstruction. While in testing, we attach the RAS into the total generative network. Although RAS can introduce randomness, we still

retain the random noise z as input, since the number of RAS is limited, while z is infinite, which can bring in much more diversity of synthesised features.

When testing for GZSL, we combine generated synthesised features $\boldsymbol{X}_F$ with the train seen set $\boldsymbol{X}_S$ as total dataset, in which we find the nearest feature of unseen data with Equation 7, and assign the corresponding label to the unseen data as its category.

$$c_i = \underset{\boldsymbol{x}_j \in \boldsymbol{X}_S \cup \boldsymbol{X}_F}{\arg\max} \frac{\boldsymbol{x}_i^T \boldsymbol{x}_j}{\|\boldsymbol{x}_i\|_2 \|\boldsymbol{x}_j\|_2}, \tag{7}$$

where, $\boldsymbol{x}_i \in \boldsymbol{X}_U$ is the test feature from unseen set, and $\|\cdot\|_2$ denotes the ℓ_2 norm.

4. Experiments

In this section, we will first give a brief review of the selected datasets for evaluation our method, then detailed experiments will be carried out to show the performance of our method both on the assessment of unseen classes accuracy of Zero-shot Learning (ZSL) and harmonic accuracy of Generalised Zero-shot Learning (GZSL), finally we will analysis the influence of RAS with t-SNE [45] in detail.

4.1. Datasets and Settings

In our experiments, we evaluate our zero-shot learning method on four popular datasets. The dataset split follows the setting of [8], and are listed as following.

(1) SUN (SUN attributes) [46] SUN is a fine-grained and medium-sized dataset, which contains 14,340 images from 717 types of scene. Among the total number of 717 classes, 1,440 samples of 72 classes are used as unseen testing data, and the left 645 classes are divided into two parts, including 10,320 seen training samples and 2,580 seen testing samples.

(2) CUB (Caltech-UCSD-Birds 200-2011) [44] CUB is also a fine-grained and medium-sized dataset, which was composed with 11,788 images

from 200 different categories of birds. In our experiments, 50 of the total 200 classes are set as the unseen training set, including 2,967 images, and the remains are set as the seen training set, which contains 7,057 seen training images and 1,764 seen testing images.

(3) AWA (Animals with Attributes) [43] AWA is a coarse-grained and medium-scale dataset, which contains 30,475 images coming from 50 categories. The literature [8] proposed a split strategy that 40 classes are used for training, in which 19,832 images are set as seen train set and 4,958 images are set as seen test set, and 10 left classes of 5,685 images are used for testing, we also follow this setting.

(4) aPY (Attribute Pascal and Yahoo) [16] aPY is a coarse-grained and small-scale dataset, which has 15,339 image instances from 32 classes. Among all the 32 classes, 20 Pascal classes of 7,415 images are utilised for training and the left 12 Yahoo classes are utilised for testing in our experiments. For the purpose of GZSL, the 20 Pascal classes are also divided into seen training set of 5,932 images and seen test set of 1,483 images.

In our experiments, we use the features extracted from pre-trained ResNet[47] model on ImageNet [48], and each feature has 2,048 dimensions. When training, we set the balance parameters $\alpha = 1$ and $\beta = 5$. During testing, suppose the attribute dimension is l, the parameters of RAS are set as $k_1 = l/10$ and $k_2 = l/15$, the total number of selected entries will be less than $\ell^2/150$.

Moreover, to balance the number of each seen class in the dataset, we choose the quantity of each synthesised class equals to the average number of each seen class in the training set, e.g. in dataset SUN, we synthesise 16 features for each unseen classes. The synthesised number of unseen classes on four datasets are listed in Table 1.

4.2. Zero-shot Learning (ZSL)

Image classification accuracy on single label usually evaluated with top-1 accuracy, *i.e.* if the predicted label is same as the real label, then we say the prediction is correct. In some conventional evaluating methods [14], the zero-

Table 1: Synthesised number of each class on four popular datasets.

Dataset	training features	training classes	synthesised features of each class
SUN	10,320	645	16
CUB	7,057	150	47
AWA	19,832	40	496
aPY	5,932	20	297

Table 2: Results of Our Method on four popular datasets SUN, CUB, AWA, and aPY. Our method outperform other 12 methods on three datasets except CUB. SAE*: Implemented by us according to the algorithm described in its original paper.

Method	**SUN**	**CUB**	**AWA**	**aPY**
DAP	39.9	40.0	44.1	33.8
IAP	19.4	24.0	35.9	36.6
CONSE	38.8	34.3	45.6	26.9
CMT	39.9	34.6	39.5	28.0
SSE	51.5	43.9	60.1	34.0
LATEM	55.3	49.3	55.1	35.2
ALE	58.1	54.9	59.9	39.7
DEVISE	56.5	52.0	54.2	39.8
SJE	53.7	53.9	65.6	32.9
ESZSL	54.5	53.9	58.2	38.3
SYNC	56.3	**55.6**	54.0	23.9
SAE*	53.4	42.0	58.1	32.9
Ours	**61.7**	52.6	**67.4**	**40.1**

shot learning accuracy is averaged for all images, which will lead to the bad situation that high performance on densely populated classes is encouraged, *e.g.* one of unseen aPY classes *'person'*, whose number accounts for 64% of the total unseen samples will play more important role than other classes. But we are interested in achieving high performance in all classes, even in sparsely populated classes, hence we choose to use the average of each class accuracy [8], which can be described as following,

$$acc_{\mathcal{S}} = \frac{1}{\|\mathcal{S}\|} \sum_{c=1}^{\|\mathcal{S}\|} \frac{\#\text{ correct predictions in } c}{\#\text{ samples in } c}, \tag{8}$$

where, $\|\mathcal{S}\|$ is the number of test classes $\mathcal{S}$. In zero-shot learning, we set $\mathcal{S} = \boldsymbol{Z}$, and the search space is $\boldsymbol{Z}$.

We compare our algorithm with 12 recently proposed baseline methods, including DAP [16], IAP [16], CONSE [49], CMT [50], SSE [14], LATEM [51], ALE [13], DEVISE [18], SJE [17], ESZSL [52], SYNC [9], and SAE [53], and record the results in Table 2, in which SAE is implemented by us according to the algorithm described in its original paper [53].

From the Table 2, we can find that our method outperforms all the 12 state-of-the-art methods on dataset SUN, AWA and aPY, and achieve the fifth place on dataset CUB. Concretely, the result on dataset SUN exceeds the best competitor ALE 3.6%, and surpasses 1.8% over SJE on dataset AWA, the smallest winner is on the dataset aPY, just obtains 0.2% promotion. On dataset CUB, our method is not the best performer, and lower than the best algorithm SYNC 3%. Although it is the fact that our algorithm cannot win on all dataset for ZSL, this does not indicate that the effectiveness of our method is bad, because it is not reasonable to search the unseen feature on unseen classes only. Instead, the more practical way is to find the feature on all the seen and unseen classes, and this searching method is named as GZSL, which will be described in the following subsection.

Table 3: The results of Generalized Zero-Shot Learning on four popular attribute datasets. For unseen test accuracy and harmonic mean accuracy, our method outperforms all the other 12 methods. CMT*: CMT with novelty detection. SAE*: Implemented by us according to the algorithm described in original paper.

Method	**SUN**			**CUB**			**AWA**			**aPY**		
	ts	**tr**	**H**	**ts**	**tr**	**H**	**ts**	**tr**	**H**	**ts**	**tr**	**H**
DAP	4.2	25.1	7.2	1.7	67.9	3.3	0.0	**88.7**	0.0	4.8	78.3	9.0
IAP	1.0	37.8	1.8	0.2	**72.8**	0.4	2.1	78.2	4.1	5.7	65.6	10.4
CONSE	6.8	39.9	11.6	1.6	72.2	3.1	0.4	88.6	0.8	0.0	**91.2**	0.0
CMT	8.1	21.8	11.8	7.2	49.8	12.6	0.9	87.6	1.8	1.4	85.2	2.8
CMT*	8.7	28.0	13.3	4.7	60.1	8.7	8.4	86.9	15.3	10.9	74.2	19.0
SSE	2.1	36.4	4.0	8.5	46.9	14.4	7.0	80.5	12.9	0.2	78.9	0.4
LATEM	14.7	28.8	19.5	15.2	57.3	24.0	7.3	71.7	13.3	0.1	73.0	0.2
ALE	21.8	33.1	26.3	23.7	62.8	34.4	16.8	76.1	27.5	4.6	73.7	8.7
DEVISE	16.9	27.4	20.9	23.8	53.0	32.8	13.4	68.7	22.4	4.9	76.9	9.2
SJE	14.7	30.5	19.8	23.5	59.2	33.6	11.3	74.6	19.6	3.7	55.7	6.9
ESZSL	11.0	27.9	15.8	12.6	63.8	21.0	6.6	75.6	12.1	2.4	70.1	4.6
SYNC	7.9	**43.3**	13.4	11.5	70.9	19.8	8.9	87.3	16.2	7.4	66.3	13.3
SAE*	17.1	28.1	21.3	17.4	50.7	25.9	11.0	83.8	19.5	6.7	59.6	12.1
Ours	**41.2**	26.7	**32.4**	**31.5**	40.2	**35.3**	**38.7**	74.6	**51.0**	**27.5**	70.6	**39.6**

4.3. Generalised Zero-shot Learning (GZSL)

In real world application, we do not know whether a new image belongs to a seen class or an unseen class. Hence, in generalised zero-shot learning, the search space for evaluating a novel image is expanded to both test classes and train classes, which is more realistic. Furthermore, to get rid of the unbalance situation of seen test and unseen test, we avoid to utilise the arithmetic mean, and turn to use the harmonic mean computed from training and testing accuracy, following the setting of [8],

$$H = \frac{2 \times acc_{tr} \times acc_{ts}}{acc_{tr} + acc_{ts}}, \tag{9}$$

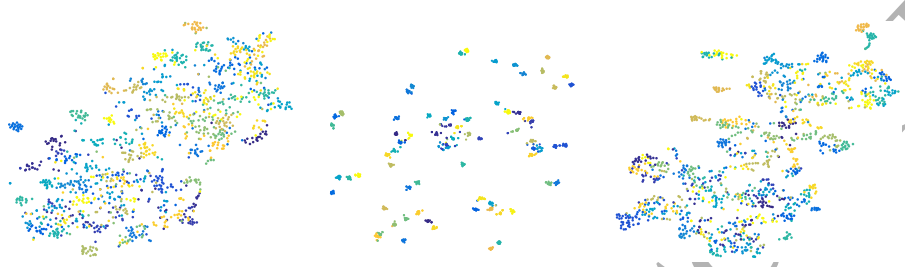
where, acc_{tr} and acc_{ts} are accuracy of the test seen features and test unseen features respectively on all classes. acc_{tr} and acc_{ts} are computed using the Equation 8, and the search space is set as $\boldsymbol{Y} \cup \boldsymbol{Z}$. $\mathcal{S} = \boldsymbol{Y}$ and $\mathcal{S} = \boldsymbol{Z}$ are executed when calculating acc_{tr} and acc_{ts} respectively.

We compute the harmonic accuracy H and corresponding train accuracy tr and test accuracy ts of our algorithm on above mentioned all four datasets, and record all the results in Table 3. We also implemented the algorithm of SAE according its original description, and cite the other results of current competitive algorithms from [8], which are also listed in Table 3.

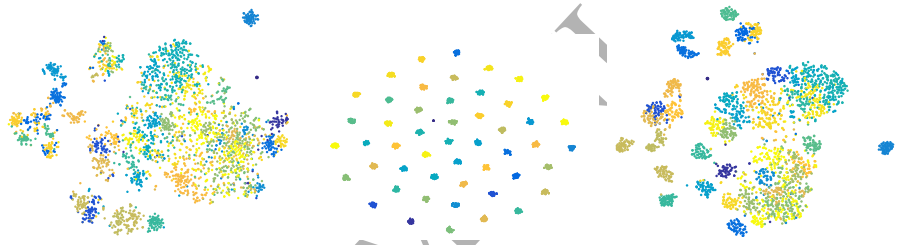
From Table 3, we can discover that our algorithm can achieve best performance on both ts and H among all the listed methods. For the test accuracy ts, our algorithm can exceed current best methods 19.4% on SUN, 7.7% on CUB, 21.9% on AWA, and 16.6% on aPY respectively. For the harmonic accuracy H, our method also outperforms all the methods listed in Table 3, and obtains 6.1%, 0.9%, 23.5%, and 20.6% improvement on dataset SUN, CUB, AWA, and aPY respectively. The biggest gap between our method and the best competitor lies on AWA, and achieves more than 20%, which demonstrate the effectiveness of our method.

Although conventional algorithms such as SYNC, DAP, IAP, and CONSE, have high train accuracies tr, their corresponding test accuracies are extremely low, *e.g.* IAP achieves 72.8% on CUB, DAP gets 88.7% on AWA and CONSE obtains 91.2% on aPY, but their relevant ts is zero or approximate zero, which makes the harmonic accuracies to be zero too. SYNC has the largest tr on dataset SUN, which is 16.6% higher than our method, but has 33.3% lower result for tr, which also lead to about 20% lower for the harmonic accuracy.

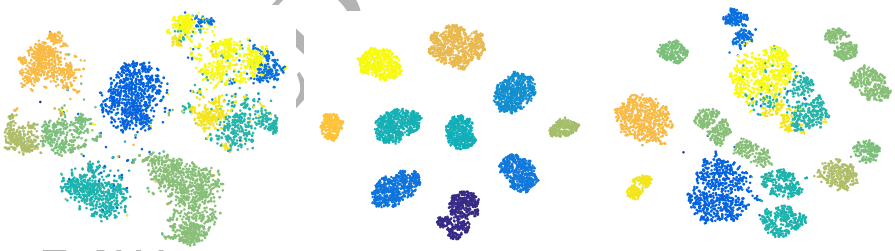
The results of high tr but low ts clearly reveals that those methods such as DAP, IAP and CONSE over-fit on certain datasets. Those methods train a very suitable classifier for training data, but are very terrible for testing data. Oppositely, our method obtain balanced results on both ts and tr, which lead to high values on H for all four datasets, which strongly indicate the effectiveness of our method.



(a) SUN



(b) CUB



(c) AWA

Figure 6: The synthesised features of SUN, CUB, and AWA shown using t-SNE. The first column shows the real features, the second column demonstrate the fake features generated using cGAN, and the synthesised features of our method are illustrated in the third column.

Table 4: The results of conventional GAN with RAS but without reconstruction network.

Dataset	ZSL	GZSL		
		ts	tr	H
SUN	36.7	6.7	30.4	11.0
CUB	20.0	0.8	48.5	1.6
AWA	43.5	5.7	83.4	10.7
aPY	15.9	1.9	85.8	3.72

4.4. Detailed Analysis

Distribution Analysis of Synthesised Data The purpose of feature or image synthesis is to obtain the real or approximate real distribution of original unseen dataset, so it is necessary to check whether our method can obtain the realistic distribution. For the sake of demonstrating the effectiveness of our method, we draw the distributions of the original unseen data, the synthesised data generated with conditional GAN, and the synthesised data generated with our method, and show these figures in Figure 6. In Figure 6, for better comparison, we set the synthesised feature number of each unseen class equals to the number of corresponding unseen class in the test set.

As we known that the synthesised data with input from noise of a fixed distribution have the same distribution of the noise, which can be convinced in the second column of Figure 6. In our experiments, Gaussian noise is exploited as the input, and the results shown in the second column of Figure 6 also abbey the Gaussian distribution, which are obviously very different from that shown in the first column. The third column shows the results of our method, which are more realistic and reasonable than the second column according to the first column.

Influence Analysis of Kept Dimension of Attribute There is only one parameter for our proposed RAS, the kept dimension of attribute. Here, we take AWA as an example to analyse the influence of different kept dimension to the final performance. The result are illustrated in Figure 7, from which we

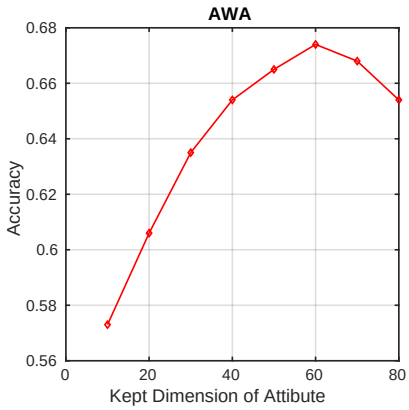


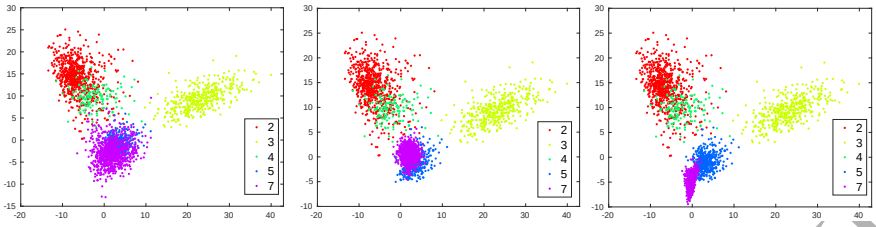
Figure 7: The accuracy of ZSL under different kept dimension of attribute when RAS applied.

can find that the best performance appears when the kept dimension equals 60. This phenomenon is also consistent with the above generated data distribution that the best result emerges when the proposed RAS is applied. In addition, the results in Table 2 and Table 3 are computed when the kept dimension is equal to 80% of the attribute dimension.

Importance Analysis of Reconstruction Item Traditional conditional GAN usually do not have the reconstruction item, following we will discuss the importance of this item in our algorithm. We remove the reconstruction item and compute the results of both ZSL and GZSL on all four datasets, and record them in Table 4.

For the test accuracy ts of ZSL, the results are much lower than the results with reconstruction item, and about the half of the value listed in Table 2. For the accuracy of GZSL, tr obtains higher performance than that with reconstruction item, but ts and H are much lower than that recorded in Table 2. In addition, ts and H are more important than tr. These compared results indicate that the the conditional GAN without reconstruction item may cause domain shift problem, which make the synthesised features shift compared to real unseen data.

To verify the domain shift problem, we choose partial classes of the seen



(a) real features (b) with reconstruction item (c) without reconstruction item

Figure 8: Scatter maps of real features and synthesised features. Purple points are synthesised features, the left points are the seen training features.

training data and one class of the synthesised data, and draw the scatter map in Figure 8 after dimensionality reduction with PCA. In this figure, the synthesised class is drawn in purple, the seen classes are in other colours. From the figure, we can discover that the synthesised features with reconstruction item are much more similar with real features than those without reconstruction item, and have approximately the same position with real data, while the synthesised data without reconstruction item are much lower than the real data, which also indicate that our method with reconstruction item can solve the domain shift problem.

5. Conclusion

In this paper, we propose a novel algorithm which trains a conditional GAN to synthesis unseen features from attributes. our method add a reconstruction item loss when training the network, which can resolve the problem of domain shift. During testing, we utilise a policy of Random Attribute Selection to choose the class attribute entries randomly, which can synthesis much realistic features of unseen classes. Experiments on four popular dataset for both ZSL and GZSL show that our proposed method can outperform all the state-of-the-art methods. We also draw the scatter maps of synthesised features, and discover that our algorithm with reconstruction item is much better than conventional

GAN without RAS.

6. Acknowledgements

This work was supported in part by National Natural Science Foundation of China (No.61872187, No.61603190, No.61773215) and the Major Special Project of Core Electronic Devices, High-end Generic Chips and Basic Software (No.2015ZX01041101).

References

References

[1] Z. Fu, T. Xiang, E. Kodirov, S. Gong, Zero-shot learning on semantic class prototype graph, IEEE transactions on pattern analysis and machine intelligence 40 (8) (2018) 2009–2022.

[2] R. Qiao, L. Liu, C. Shen, A. van den Hengel, Less is more: zero-shot learning from online textual documents with noise suppression, in: CVPR, 2016, pp. 2249–2257.

[3] Y. Long, L. Liu, L. Shao, Attribute embedding with visual-semantic ambiguity removal for zero-shot learning, in: BMVC, 2016.

[4] Y. Yu, Z. Ji, J. Guo, Y. Pang, Zero-shot learning with regularized cross-modality ranking, Neurocomputing 259 (October) (2017) 14–20.

[5] H. Zhang, Y. Long, L. Shao, Zero-shot hashing with orthogonal projection for image retrieval, Pattern Recognition Letters`doi:https://doi.org/10.1016/j.patrec.2018.04.011`.

[6] X. Li, M. Fang, J. Wu, Zero-shot classification by transferring knowledge and preserving data structure, Neurocomputing 238 (May) (2017) 76–83.

[7] S. Pachori, A. Deshpande, S. Raman, Hashing in the zero shot framework with domain adaptation, Neurocomputing 275 (January) (2018) 2137–2149.

[8] Y. Xian, B. Schiele, Z. Akata, Zero-shot learning-the good, the bad and the ugly, in: CVPR, 2017.

[9] S. Changpinyo, W.-L. Chao, B. Gong, F. Sha, Synthesized classifiers for zero-shot learning, in: CVPR, 2016, pp. 5327–5336.

[10] Y. Yang, Y. Luo, W. Chen, F. Shen, J. Shao, H. T. Shen, Zero-shot hashing via transferring supervised knowledge, in: ACM MM, ACM, 2016, pp. 1286–1295.

[11] Y. Guo, G. Ding, J. Han, Y. Gao, Synthesizing samples for zero-shot learning, in: IJCAI, 2017, pp. 1774–1780.

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

[13] Z. Akata, F. Perronnin, Z. Harchaoui, C. Schmid, Label-embedding for image classification, IEEE TPAMI 38 (7) (2016) 1425–1438.

[14] Z. Zhang, V. Saligrama, Zero-shot learning via semantic similarity embedding, in: ICCV, 2015, pp. 4166–4174.

[15] M. Liu, D. Zhang, S. Chen, Attribute relation learning for zero-shot classification, Neurocomputing 139 (September) (2014) 34–46.

[16] C. H. Lampert, H. Nickisch, S. Harmeling, Attribute-based classification for zero-shot visual object categorization, IEEE TPAMI 36 (3) (2014) 453–465.

[17] Z. Akata, S. Reed, D. Walter, H. Lee, B. Schiele, Evaluation of output embeddings for fine-grained image classification, in: CVPR, 2015, pp. 2927–2936.

[18] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al., Devise: A deep visual-semantic embedding model, in: NIPS, 2013, pp. 2121–2129.

[19] Y. Yu, Z. Ji, J. Guo, Z. Zhang, Zero-shot learning via latent space encoding, IEEE transactions on cybernetics PP (99) (2018) 1–12. doi:10.1109/TCYB.2018.2850750.

[20] Z. Ji, Y. Yu, Y. Pang, J. Guo, Z. Zhang, Manifold regularized cross-modal embedding for zero-shot learning, Information Sciences 378 (2017) 48–58.

[21] H. Zhang, Y. Long, W. Yang, L. Shao, Dual-verification network for zero-shot learning, Information Sciences 470 (2018) 43–57.

[22] Y. Fu, T. M. Hospedales, T. Xiang, Z. Fu, S. Gong, Transductive multi-view embedding for zero-shot recognition and annotation, in: ECCV, Springer, 2014, pp. 584–599.

[23] Y. Guo, G. Ding, X. Jin, J. Wang, Transductive zero-shot recognition via shared model space learning, in: AAAI, 2016, pp. 3494–5000.

[24] E. Kodirov, T. Xiang, Z. Fu, S. Gong, Unsupervised domain adaptation for zero-shot learning, in: ICCV, 2015, pp. 2452–2460.

[25] Y. Yu, Z. Ji, J. Guo, Y. Pang, Transductive zero-shot learning with adaptive structural embedding, IEEE transactions on neural networks and learning systems 9 (29) (2017) 4116–4127.

[26] Y. Long, L. Liu, L. Shao, F. Shen, G. Ding, J. Han, From zero-shot learning to conventional supervised classification: Unseen visual data synthesis, in: CVPR, 2017.

[27] Y. Xian, T. Lorenz, B. Schiele, Z. Akata, Feature generating networks for zero-shot learning, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018.

[28] V. K. Verma, G. Arora, A. Mishra, P. Rai, Generalized zero-shot learning via synthesized examples, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[29] L. Chen, H. Zhang, J. Xiao, W. Liu, S.-F. Chang, Zero-shot visual recognition using semantics-preserving adversarial embedding network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, 2018.

[30] Y. Zhu, M. Elhoseiny, B. Liu, X. Peng, A. Elgammal, A generative adversarial approach for zero-shot learning from noisy texts, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[31] C. H. Lampert, H. Nickisch, S. Harmeling, Learning to detect unseen object classes by between-class attribute transfer, in: CVPR, IEEE, 2009, pp. 951–958.

[32] B. Demirel, R. G. Cinbis, N. I. Cinbis, Attributes2classname: A discriminative model for attribute-based unsupervised zero-shot learning, in: CVPR, 2017.

[33] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: ICLR, 2013.

[34] Y. Long, L. Shao, Describing unseen classes by exemplars: Zero-shot learning using grouped simile ensemble, in: WACV, IEEE, 2017, pp. 907–915.

[35] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, H. Lee, Learning what and where to draw, in: NIPS, 2016, pp. 217–225.

[36] X. Liang, Z. Hu, H. Zhang, C. Gan, E. P. Xing, Recurrent topic-transition gan for visual paragraph generation, in: ICCV, 2017.

[37] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, A. A. Efros, Generative visual manipulation on the natural image manifold, in: ECCV, 2016, pp. 597–613.

[38] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, H. Lee, Generative adversarial text to image synthesis, in: ICML, 2016.

[39] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein gan, arXiv preprint arXiv:1701.07875.

[40] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, S. P. Smolley, Least squares generative adversarial networks, in: ICCV, 2017.

[41] J.-Y. Zhu, T. Park, P. Isola, A. A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: ICCV, 2017.

[42] Z. Yi, H. Zhang, P. T. Gong, et al., Dualgan: Unsupervised dual learning for image-to-image translation, in: ICCV, 2017.

[43] A. Farhadi, I. Endres, D. Hoiem, D. Forsyth, Describing objects by their attributes, in: CVPR, IEEE, 2009, pp. 1778–1785.

[44] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, P. Perona, Caltech-ucsd birds 200.

[45] L. v. d. Maaten, G. Hinton, Visualizing data using t-sne, JMLR 9 (Nov) (2008) 2579–2605.

[46] G. Patterson, C. Xu, H. Su, J. Hays, The sun attribute database: Beyond categories for deeper scene understanding, IJCV 108 (1-2) (2014) 59–81.

[47] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: CVPR, 2016, pp. 770–778.

[48] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: CVPR, IEEE, 2009, pp. 248–255.

[49] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, J. Dean, Zero-shot learning by convex combination of semantic embeddings, in: ICLR, 2014.

[50] R. Socher, M. Ganjoo, C. D. Manning, A. Ng, Zero-shot learning through cross-modal transfer, in: NIPS, 2013, pp. 935–943.

[51] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, B. Schiele, Latent embeddings for zero-shot classification, in: CVPR, 2016, pp. 69–77.

[52] B. Romera-Paredes, P. Torr, An embarrassingly simple approach to zero-shot learning, in: ICML, 2015, pp. 2152–2161.

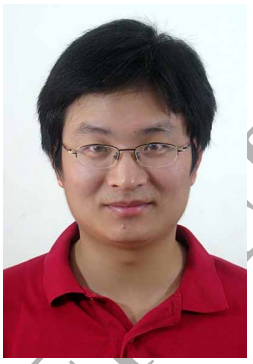
[53] E. Kodirov, T. Xiang, S. Gong, Semantic autoencoder for zero-shot learning, in: CVPR, 2017.

Dr. Haofeng Zhang currently is an Associate Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology. He received the B.Eng. and the Ph.D. degrees from the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China, in 2003 and 2007, respectively. His research interests include computer vision, deep learning, zero-shot learning and mobile robotics.

Dr. Yang Long is currently a Research Fellow with OpenLab, School of Computing, Newcastle University. He received his Ph. D. degree in Computer Vision and Machine Learning from the Department of Electronic and Electrical Engineering, the University of Sheffield, UK, in 2017. He received the M.Sc. degree from the same institution, in 2014. His research interests include Artificial Intelligence, Machine Learning, Computer Vision, Deep Learning, Zero-shot Learning, with focus on Transparent AI for Healthcare Data Science. He has authored/co-authored papers in refereed journals/conferences such as IEEE TPAMI, TIP, CVPR, AAAI and ACM MM, and holds 1 Chinese patent. He is also a regular reviewer for leading journals and conferences. He is a member of the British Computer Society, ACM, and IEEE.

Dr. Li Liu joined the Inception Institute of Artificial Intelligence(IIAI), Abu Dhabi, UAE, as the director of research in computer vision in 2017. He received the B.Eng. degree in electronic information engineering from Xi'an Jiaotong University, Xi'an, China, in 2011, and the Ph.D. degree from the Department of Electronic and Electrical Engineering, The University of Sheffield, Sheffield, U.K., in 2014. His research interests include computer vision, machine learning, multimedia, and data mining.

Professor Ling Shao joined the University of East Anglia as Chair in Computer Vision and Machine Learning in 2016. He received the B.Eng. degree in Electronic and Information Engineering from the University of Science and Technology of China (USTC), the M.Sc. degree in Medical Image Analysis and the Ph.D. (D.Phil.) degree in Computer Vision under the supervision of Sir Michael Brady at the Robotics Research Group from the University of Oxford. Previously, he was a Professor (2014-2016) with Northumbria University, a Senior Lecturer (2009-2014) with the Department of Electronic and Electrical Engineering at the University of Sheffield and a Senior Scientist (2005-2009) with Philips Research, The Netherlands. His research interests include Computer Vision, Deep Learning/Machine Learning, Multimedia, and Image/Video Processing. He is an Associate Editor of IEEE Transactions on Image Processing, IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Circuits and Systems for Video Technology, and several other journals.



570