Latent Fine-grained Features embedding model for Unsupervised Zero-shot Learning

Guiyu Tian\textsuperscript{1,*}, YiZheng Tao\textsuperscript{2} and Yang Xie\textsuperscript{3}

\textsuperscript{1,2,3} Institute of Computer Application, China Academy of Engineer Physics, Mian Yang, China
\textsuperscript{*}Corresponding author’s e-mail: guiyu@cqu.edu.cn

Abstract. Zero-shot learning (ZSL) aims to identify image categories that never appear in the training set by learning a mapping from visual information to the semantic space. For existing works in recent years, it has been paying much attention to user-defined-attributes-based supervised ZSL models and manual annotation-based unsupervised ZSL methods, whilst the important drawback of user-defined attributes is ignored. User-defined attributes usually need to be precisely annotated not only for seen classes but also for unseen classes. This procedure of collecting user-defined attributes is an error-prone and time-consuming work limiting the scalability of the methods to a great extent. Moreover, user-defined attributes are semantic embedding but they are not exhaustive. In this paper, we propose a manual annotation-free unsupervised ZSL method with a great scalability which is a benefit for the large-scale ZSL tasks, just using public word vectors of categories without any dedicated attributes annotation efforts. In addition, our approach can automatically extract latent fine-grained features to reduce visual information losses caused by the absence of user-defined attributes. Extensive experimental results show that our method outperforms the previous methods among the manual annotation-free unsupervised ZSL methods.

1. Introduction

Due to the existence of massive object categories in real life, zero-shot learning (ZSL) has attracted more and more attention in recent years. Unlike traditional image recognition methods that seek to predict a target whose category has already appeared in the training set (seen classes), ZSL enables identification of categories that are not seen before by transferring knowledge learned from seen classes to describe unseen classes. In order to achieve ZSL tasks, side information (user-defined attributes of categories \cite{1}, the word vectors of the class names \cite{2,4}, the textual descriptions of categories \cite{3}) are provided to describe seen and unseen classes, both of them sharing a common semantic space (side information).

In supervised ZSL, user-defined attributes as a common semantic space for seen and unseen classes. Indeed, user-defined attributes need to be precisely annotated not only for seen classes but also for unseen classes. Another branch of ZSL, at the training stage, training the model by using the user-defined attributes of all seen classes. But in the testing phase, predicting unseen classes by using a readily public source like word vectors of unseen classes instead of user-defined attributes. Specifically, user-defined attributes just only for seen classes, not for unseen classes, and motivated by \cite{4}, we simply refer to these as manual annotation-based unsupervised ZSL. To date, many works have made a great performance in ZSL. However, there are some problems existed in ZSL.
Firstly, in recent years, due to the good performance contributed by user-defined attributes, many works are focused on supervised ZSL [1] or manual annotation-based unsupervised ZSL [2] that both need user-defined attributes of seen classes at the training stage. These works need to manually collect fine-grained attributes information about classes, which is an error-prone and time-consuming job limiting the scalability of the methods to a great extent [4]. Moreover, user-defined attributes are semantic embedding but they are not exhaustive, especially for many large-scale datasets in real life, thus further making most existing works hardly cannot work well on large-scale ZSL tasks.

Secondly, without the user-defined attributes of seen and unseen classes, the ZSL methods will obtain a great scalability but a low recognition performance [2].

Motivated by those as above, we propose a latent fine-grained features (LFF) model aims to obtain a great scalability and recognition performance. We simply refer to our method as manual annotation-free unsupervised ZSL, and the reason as follows: as described in [4], B. Demirel et al proposed a method that uses the user-defined attributes and the image label of seen classes to achieve ZSL tasks, and they simply refer to their method as attribute-based unsupervised ZSL (third and fourth paragraphs in [4]). But in our paper, we name such method as manual annotation-based (the same meaning as "attributes-based") unsupervised ZSL. Thus, motivated by [4], we simply refer to our method as manual annotation-free unsupervised ZSL (our method doesn't need any user-defined attributes).

To sum up, we have made the following contributions:

1) a manual annotation-free unsupervised ZSL method that aims to use the combination of seen classes to recognize unseen classes is proposed to achieve large-scale ZSL tasks. Note that we train our model in an end-to-end manner, but the prediction is in a hybrid way. Extensive experimental results show that our method outperforms the state-of-the-art among the manual annotation-free unsupervised ZSL methods in two benchmark datasets.

2) a neural network is proposed to extract image fine-grained features for the recognition of tiny objects. Our experimental result demonstrates that our neural network will be more beneficial for the ZSL tasks comparing with VGG19 [11].

2. Related work

According to whether or not dedicated manually annotation efforts (user-defined attributes, expert knowledge, etc.) are required for seen or unseen classes, existing ZSL works can category as supervised ZSL, manual annotation-based unsupervised ZSL, and manual annotation-free unsupervised ZSL.

2.1. Supervised ZSL

Towards supervised ZSL (dedicated manually annotation efforts not only for seen classes but also for unseen classes), many works focus on this aspect [1]. Among these works, by training each attribute classifier individually, Lampert et al [1] propose direct (DAP) and indirect attribute prediction (IAP) to achieve ZSL tasks. Yan Li [12] propose a cascaded zooming network that joints user-defined attributes and discriminative latent features to achieve ZSL tasks. The latent features of unseen classes are obtained by the mapping relationship of user-defined attributes between seen and unseen classes, which cannot exactly reflect the true latent features relationship between seen and unseen classes.

2.2. Manual annotation-based unsupervised ZSL

In manual annotation-based unsupervised ZSL (dedicated manually annotation efforts just only for seen classes, not for unseen classes). B. Demirel et al [4] proposed a method that adopts word vectors of the combination of attributes name to achieve ZSL tasks.

2.3. Manual annotation-free unsupervised ZSL

In manual annotation-free unsupervised ZSL (without any artificial attributes annotation efforts), there are two very representative works [5, 6] in recent years. The DeVise [5] is based on a deep visual-semantic embedding model, where a hinge rank loss is used to train this model. ConSE [6] is a hybrid model, which is similar to our work. At the class-level, ConSE regards the class probability value of
seen classes as combine weights to predict unseen classes, which leads to a strong bias problem. Our method different from ConSE [6] in two major ways. First, instead of establishing the relationship between seen and unseen classes in the class-level, we model the deep relationship between seen and unseen classes in the fine-grained feature-level, which can alleviate the bias problem and get a better embedding word vector for unseen images. Second, a new neural network is proposed to extract the latent fine-grained features as embedding space, where seen and unseen classes are sharing.

3. ZSL Problem Statement
In our method, the training set (seen classes) is defined as \( S = (x_i^s, y_i^s) \), where \( x_i^s \) is the i-th image of the seen classes and \( y_i^s \) denotes the class label of image \( x_i^s \). The testing set (unseen classes) is defined as \( U = (x_j^u, y_j^u) \), where \( x_j^u \) is the j-th image of the unseen classes and \( y_j^u \) denotes the class label of image \( x_j^u \). In ZSL setting, S and U are disjoint, \( S \cap U = \emptyset \). In our method, b-dimension word vectors of class name from GloVE [7] are provided to build a shared semantic space (side information) for both seen and unseen classes, where \( w_i^s \) and \( w_j^u \) denote word vectors of i-th seen class name and j-th unseen class name respectively. In the testing phase, by calculating the word vector of image \( x_i^u \), we can predict the category of \( x_i^u \).

4. Our Method
Our method consists of FGNet and LFF, which are described in detail below.

4.1. Our Fine-grained Network (FGNet)
Motivated by paper [8], we propose FGNet adapted from VGG19 to extract image fine-grained visual features for ZSL tasks. As shown in Figure 1 left panel, conv in blue parts denotes a convolution layer with 64 kernels, 1*1 size, and 1 stride. As illustrated in Figure 1 right panel, the principle of passthrough layer is presented in detail, where a 4*4 convolution kernel is divided into four 2*2 convolution kernels instead of spatial convolution operation. The rest of the network structure is the same as VGG19.

![Figure 1](image1.png)

Our new design aims to concatenate the high-resolution features with the low-resolution features for the final recognition tasks. Specifically, high-resolution features denote fine-grained features, which refer to the output of the passthrough layer. Additionally, low-resolution features refer to the output of block5-maxpool, as shown in Figure 1.

4.2. Latent fine-grained features (LFF) Model For ZSL
Our model is illustrated in Figure 2, which joint softmax layer and latent features layer as the final output. In the training stage, we train the softmax layer and latent features layer together. But in ZSL
prediction, only using latent features layer to achieve ZSL tasks for unseen classes. The following will be presented in detail.

![Image](image.png)

**Figure 2** the illustrate of **FGNet**

### 4.2.1. ZSL Training.
Softmax layer aims to output class probability, which only for seen classes, not for unseen classes. We train this layer by a standard softmax loss, which is illustrated in equation (1), where $m$ denotes the number of seen classes and $z_j$ denotes the output of $j$-th neuron in softmax layer.

$$L_1 = -\frac{1}{n} \sum_{i=1}^{n} \log \frac{e^{x_i^s}}{\sum_{j=1}^{m} e^{x_j^s}}$$  

$$L_2 = \max\left(0, m + \|\text{lat}(x^s_i) - \text{lat}(x^s_j)\|_2^2 - \|\text{lat}(x^s_i) - \text{lat}(x^s_k)\|_2^2\right)$$  

$$L = L_1 + L_2$$  

(2)  

Latent features layer aims to output latent fine-grained features, which are learned to distinguish different categories by regulating inter/intra class distances. We train this layer by minimizing a triplet loss function, which is shown in equation (2), where image $x^s_i$ and $x^s_j$ belong to same classes and $x^s_k$ is from a different class. $\text{lat}(x^s_i)$ denotes the output of latent features layer about image $x^s_i$ and $m$ is a margin which is set to 1.0 in our experiment.

We train our model in three steps which will be presented in the following. **Step 1**, we initialize our convolution layer (black parts in **Figure 2** left panel) with VGG19 pre-trained weights on ImageNet, the rest are randomly initialized. **Step 2**, we train the softmax layer (the FGNet weights are not fixed) using equation (1). **Step 3**, frozen the FGNet weights, only train the latent features layer using equation (2). **Step 4**, fine-tuning the whole model (softmax layer and latent features layer together) in an end-to-end manner using equation (3).

### 4.2.2. ZSL Prediction.
ZSL prediction is performed in a hybrid way that aims to use the combination of seen classes to recognize unseen classes. There are four steps to predict unseen classes.

First, we model the deep relationship between seen and unseen classes in the fine-grained feature-level, thus we must know the latent features prototype of all seen classes. In our ZSL method, after our model was trained, we obtain the latent features prototype of all seen classes by equation (4), where $\text{lat}(y^s_j)$ denotes latent features prototype of seen class $y^s_j$ and $n$ denotes the number of images in category $y^s_j$. $p(\Delta(x^s_j, y^s_j))$ denotes probability value that image $x^s_j$ is predicted to be a true category $y^s_j$, where $\Delta$ is a classifier (softmax layer in our model). $\text{lat}(x^s_i)$ denotes the output of latent features layer about image $x^s_i$.

$$\Omega = k_{neighbors}(d(\text{lat}(x^u_i), \text{lat}(y^s_j)))$$  

(5)  

Second, in order to model the deep relationship between seen and unseen classes, we obtain $k$ seen classes that are closest to the test image $x^u_i$ of unseen classes by comparing the latent features between the test image $x^u_i$ and all seen classes. This procedure can be done by equation (5), where $d(x,y)$ is the Euclidean distance between $x$ and $y$. $\Omega$ is a set that stores index of $k$ seen classes.
vec(x^i) = \sum_{y \in Y^s} e^{d(lat(x^i), lat(y^s))}w^s_{y^s}

(6)

Third, transferring knowledge learned from seen classes to describe unseen classes. Specifically, we calculate embedding word vector of test image x^i by combining k seen classes obtained from the equation (5). This operation can be done by equation (6), where w^s_{y^s} denotes the word vector (GloVE) of the name of category y^s:

y(x^i) = \arg \max_{y \in Y^u} s(vec(x^i), w^j)

(7)

Finally, we make the final classification by using vec(x^i) obtained from the equation (6). This operation can be done by equation (7), where w^j indicates word vector (GloVE) of the j-th unseen class name and s(x, y) denotes the cosine similarity between x and y.

5. Experiments

We conduct our experiment on two benchmark datasets namely AwA2 [9] and aPaY [10]. AwA2 dataset contains 37322 images of 50 different animal classes, where 40 categories belong to seen classes and other 10 categories belong to unseen classes in our ZSL setting. APaY dataset contains 15339 images of 32 categories, where 12695 images of 20 categories belong to seen classes and other 2644 images of 12 categories belong to unseen classes in our ZSL setting. In addition, we adopt b dimension GloVE [7] word vector as our side information (word vector of class name) in all experiments. In our method, we set b=300. For those class names that contain multiple words, we adopt the average of the word vectors.

Table 1  Comparison with previous methods

| Manual annotation-free unsupervised ZSL method | AwA2 [9](%) | aPay [10](%) |
|-----------------------------------------------|------------|-------------|
| DeVise[5]-with-VGG19                          | 38.5       | 25.0        |
| ConSE[6]-with-VGG19                          | 38.6       | 22.0        |
| LFF (Ours)                                    | 48.0       | 31.1        |

Figure 3 The squared Euclidean distance for 10 unseen AwA2 classes

Compared with the previous methods among the manual annotation-free unsupervised ZSL methods, our method outperforms them on both benchmark datasets. As shown in Table 1, for AwA2 datasets, our method improves 9.5% and 9.4% comparing with DeVise and ConSE respectively. For aPaY datasets, our method outperforms 6.1% and 9.1% comparing with DeVise and ConSE respectively.

Furthermore, we analyze the effect of latent features layer for ZSL prediction. As shown in Figure 3, the squared Euclidean distance computed with 1000-dimension latent features (the output of latent features layer) for 10 unseen AwA2 classes. From the result, we can see that the squared Euclidean distances between different unseen classes are both larger than m (in equation (2) and equals to 1). This
result demonstrates that the latent features layer makes different classes with a certain distance, which is beneficial to ZSL prediction.

6. Conclusion
In this work, a manual annotation-free unsupervised ZSL method that aims to use the combination of seen classes to recognize unseen classes is proposed to achieve large-scale ZSL tasks without any user-defined attributes. Our proposed method is easily scalable to deal with any dataset, especially for the big dataset. Experimental results on two benchmark datasets demonstrate that our method outperforms the previous methods among the manual annotation-free unsupervised ZSL methods.

References
[1] C. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. IEEE Trans. Pattern Anal. Mach. Intell., 36(3):453–465, March 2014.
[2] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In Proc. IEEE Conf. Comput. Vis. Pattern Recog, pages 2927–2936, 2015.
[3] J. Lei Ba, K. Swersky, S. Fidler, et al. Predicting deep zeroshot convolutional neural networks using textual descriptions. In Proc. IEEE Int. Conf. on Computer Vision, pages 4247–4255, 2015.
[4] B. Demirel, R. G. Cinbis, and N. I. Cinbis. Attributes2classname: A discriminative model for attribute-based unsupervised zero-shot learning. In The IEEE International Conference on Computer Vision (ICCV), 2017.
[5] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In NIPS, pages 2121–2129, 2013.
[6] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. arXiv preprint arXiv:1312.5650, 2013.
[7] J. Pennington, R. Socher, and C. D. Manning. GloVe: Global vectors for word representation. Proc. of the Empirical Methods in Natural Language Processing, 12:1532–1543, 2014.
[8] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[J]. arXiv preprint, 2017.
[9] Xian Y, Lampert C H, Schiele B, et al. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly[J]. IEEE transactions on pattern analysis and machine intelligence, 2018.
[10] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In Proc. IEEE Conf. Comput. Vis. Pattern Recog., 2009.
[11] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.
[12] M. Elhoseiny, B. Saleh, and A. Elgammal. Write a classifier: Zero-shot learning using purely textual descriptions.In Proc. IEEE Int. Conf. on Computer Vision, pages 2584–2591, 2013.