Neurocomputing 391 (2020) 74–82

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A novel dataset-specific feature extractor for zero-shot learning

Yuxuan Luo[a,b], Xizhao Wang[a,*], Weipeng Cao[a]

[a] College of Computer Science and Software Engineering, Shenzhen University, 518060, China
[b] The Guangdong Key Laboratory of Intelligent Information Processing, Shenzhen University, 518060, China

ARTICLE INFO

Article history:
Received 29 November 2019
Revised 11 January 2020
Accepted 20 January 2020
Available online 27 January 2020

Communicated by Dr. Nianyin Zeng

Keywords:
Zero shot learning
Feature extractor
Residual networks
Label tree

ABSTRACT

Most of the existing Zero-Shot Learning(ZSL) algorithms adopt pre-trained neural networks as their feature extractors. Since these pre-trained models are not specially designed for ZSL tasks, it is difficult to guarantee the stability and generalization ability of the ZSL algorithms due to the feature mismatch. To alleviate this problem, we propose a novel dataset-specific feature extractor for ZSL according to an attribute-based label tree. Specifically, an attribute-based label tree is firstly built via K-means clustering and then the information extracted from the label tree is used to fine-tune the parameters of the pre-trained models in order to make the extracted features more suitable for the current ZSL task. The experimental results on three typical ZSL datasets show that our approach can effectively improve the predictive accuracy of the existing ZSL algorithms and significantly accelerate their convergence rate. Additionally we explain the experimental phenomena from the perspective of feature visualization, which experimentally show that the features extracted by our method are much more separable than those of the original pre-trained models.

1. Introduction

Deep learning has achieved many breakthroughs in various fields in recent years, such as image classification [9], speech recognition [10], natural language processing [16], and image segemantation [30,32]. Most of the deep learning algorithms use the supervised learning mechanism, which requires not only a large number of training samples but also the accurately annotated labels. In addition, the label types in training set and testing set should be consistent in traditional deep learning. These premises are difficult to be satisfied in many practical scenarios. In real-life applications, the cost of collecting enough correctly labeled training data is often very high, and sometimes it is not feasible because the testing samples may contain labels that have never appeared in the training samples. Zero-Shot Learning (ZSL) is an advanced technique that aims to solve this problem, which was proposed by Lampert et al. in 2009 [13]. In ZSL, each class belongs to one type, i.e. the seen class or the unseen class. In ZSL, we consider the class that can collect enough training samples as the seen class, and the class that cannot collect enough training samples (or even no samples) as the unseen class. The research of ZSL is to use the information of the samples of the seen classes to assist the prediction of the samples of the unseen classes. In recent years, many achievements in ZSL have been acquired [14,24].

However, there are still many fundamental problems in ZSL that have not been solved. One of them is that there is no unified and effective way to build the connection between the seen and the unseen classes, which is the major obstacle of the wide application of ZSL algorithms. To alleviate this problem, three typical methods have been proposed in recent years, that is, traditional ZSL [13,14,20,24] and Generalized ZSL [13,20,28]. The strategies they used are a.) from attributes to features; b.) from features to attributes; and c.) from features and attributes to a third space. Here the attributes refer to the vector descriptions of the classes and the features refer to the vectors extracted from the raw images. Besides, there are some methods that use the auxiliary information (e.g., word2vec, label names, etc.) to help the model to learn the association or knowledge transfer between the seen and the unseen classes. Many advanced techniques such as GAN-based [28,29] and VAE-based [11,20] algorithms have been proposed to overcome the difficulty of transferring knowledge from the seen classes to the unseen classes.

Although the above mentioned methods can use the seen class information to help predict the labels of the unseen classes in some specific scenarios, their performance is subject to a

* Corresponding author.
E-mail addresses: luoyuxuan2018@email.szu.edu.cn (Y. Luo), xizhaowang@ieee.org (X. Wang), caoweipeng123@gmail.com (W. Cao).

https://doi.org/10.1016/j.neucom.2020.01.069
0925-2312/

common constraint, that is, the quality of the extracted data features must be very high. In other words, the performance of these algorithms depends heavily on the performance of their feature extractors.

In general, there are two ways to get the feature extractor. The first way is to train directly on the dataset using common feature extraction techniques. But this method is less feasible in ZSL because there are only a few samples in each class of the dataset. Another way is to use pre-trained models (e.g., the pre-trained residual networks) as their feature extractors, which is adopted by most current ZSL algorithms. Compared with the former method, the latter method can effectively improve the performance of the ZSL classifier with the help of the rich features of the pre-trained model. Up to now, most of the pre-trained models for ZSL are trained based on the Residual Network (ResNet), which has been confirmed to be a very effective feature extractor.

However, since these pre-trained models are not trained specifically for the current ZSL task, the stability and the generalization ability of the ZSL algorithms are difficult to be guaranteed. Sometimes the accuracy of the ZSL model can be improved and sometimes it can be decreased. In other words, the ZSL algorithms using the pre-training models as their feature extractors ignore the intrinsic connections between the datasets used for the pre-training models and the training dataset used for the current task.

Some latest researches have shown that the degree of correlation between the ancillary dataset and the training dataset has a significant impact on the performance of the final model [33]. The pre-trained models used in ZSL can also be regarded as the special ancillary datasets. Therefore, one can infer that, if we use the dataset-specific feature extractor to extract features for ZSL, the model may be able to have better performance than that using the general pre-trained models.

Inspired by this idea, we propose a novel dataset-specific feature extractor for ZSL according to an attribute-based label tree in this paper, which is built upon the pre-trained ResNet and uses the information of the seen classes and attributes to fine-tune its parameters to better serve the current task. The advantages of Dataset-Specific Feature Extractor according to Attribute-based Label Tree(ALT-DSFE) include that (1) the pre-trained ResNet can bring rich auxiliary features to the current task; (2) the dataset-specific design can make the model extract the most favorable features from the pre-trained ResNet and make the classification easier. Specifically, ALT-DSFE builds an attribute-based label tree via K-means clustering and uses the label tree to fine-tune the parameters of pre-trained ResNet. ALT-DSFE fully exploits the intrinsic connection between the pre-trained model and the current task and provides specific features for the ZSL model. The experimental results on the three benchmark datasets show that the our approach proposed in this paper can not only effectively improve the prediction accuracy of the existing ZSL models such as CADA-VAE [20] and can significantly improve their convergence speed. Additionally, we explained the experimental phenomena from the perspective of the feature visualization.

The contributions of this paper are as follows.

(1) The concept of Attribute-based Label Tree (ALT) is defined and a novel ALT based Dataset-Specific Feature Extractor (ALT-DSFE) is proposed for ZSL in this paper. In ALT-DSFE, in addition to introducing the pre-trained ResNet to improve the diversity and richness of the features used for ZSL, we also use the information of the seen classes and attributes to guide the fine-tuning process of the pre-trained model. In this way, the pre-trained model is transformed into a feature extractor related to the current task, which can better extract effective features for ZSL. Extensive experimental results show that the proposed ALT-DSFE can effectively improve the accuracy of the existing ZSL algorithms.
(2) ALT-DSFE provides a unified framework to extract specific features for different ZSL tasks, and this framework can be easily embedded into most of the existing ZSL algorithms. For example, one can use ALT-DSFE to initialize the trainable feature extractors of the GAN-based and VAE-based ZSL algorithms to accelerate their convergence rate.
(3) We provide a visual way to explain the experimental phenomena in this paper, which show that the features extracted by our proposed method are much more separable than that of the original pre-trained ResNets. This is very helpful for the ZSL classifier to make the correct decision.

The remaining of this paper is organized as follows. We introduce the related works in Section 2. The details of the proposed ALT-DSFE are given in Section 3. Section 4 describes the experimental settings, the experimental results, and the corresponding analysis. In Section 5, we conclude this study.

2. Preliminaries

In this section, we introduce the related works include Zero-Shot Learning (ZSL), Convolutional Neural Network (CNN), and CADA-VAE model.

2.1. Zero-shot learning (ZSL).

Many ZSL methods imitate the human reasoning process in the real world, that is, humans need the description of the unseen classes and use the knowledge they have known so that humans can recognize when they first see the new classes. Each class in ZSL has an attribute to describe this class. The attribute, which is built from the whole dataset (including both the seen and unseen classes), is a numeric vector. Each component of the vector, which ranges from 0 to 1, corresponding to the class description represents the degree of the class has this trait or not.

The basic hypothesis of the traditional ZSL is that all testing data come from unseen classes and the goal of the ZSL model is to classify the testing data (i.e. the unseen classes) as correctly as possible. The carrier of knowledge transfer between the seen classes and the unseen classes has many forms in ZSL, such as semantic attributes and word vectors. Using the same carrier to build the connection between the seen classes and the unseen classes could provide a way to transfer the information of the seen classes to the unseen classes.

The main indicator for evaluating a traditional ZSL algorithm is the performance of the trained classifier on the unseen classes. However, in many real-life applications, the coming samples may belong to both the unseen classes and the seen classes. Inspired by this observation, Changpinyo et al. [4] extended the traditional ZSL to a more general ZSL algorithm named Generalized-ZSL, which does not restrict the testing data must be the unseen classes. Besides, they proposed a new metric H (as follows), the harmonic means of the accuracy of the model on the seen and unseen classes, to measure the performance of the model.

$$H = \frac{2 * acc_{seen} * acc_{unseen}}{acc_{seen} + acc_{unseen}} \tag{1}$$

2.2. Convolutional Neural Network (CNN).

The unique sparse-connectivity structure and parameter-sharing strategy adopted by Convolutional Neural Network (CNN) make it very good at extracting local features from data. Therefore, CNN and its variant algorithms have been widely utilized in the computer vision field in recent years [9,22,23]. Usually, in a typical

CNN based deep learning algorithm, CNN units are used as the feature extractor and the fully-connected layer is used as the classifier. The quality of the extracted feature has a direct impact on the performance of the classifier. Specifically, if the feature extractor is able to extract sufficiently good features from the training data, then even with a simple classifier for decision making, the final model performance is often acceptable. In other words, it is difficult to make accurate predictions based on low-quality features no matter how complex the classifier is. Up to now, many CNN based feature extractors have been proposed such as VGG [22], Inception [23], and ResNet [9]. These feature extractors can help the deep neural network extract the effective features from the original data, and then complete the classification or regression tasks more efficiently.

Take the ResNet [9] as an example, the unique residual structure allows it can be extended to a neural network having much more hidden layers than before (e.g., sometimes the number of the hidden layers is more than one thousand [8]). Such a deep architecture provides the ability to learn the rich patterns effectively from large-scale datasets. In recent years, it has become fashionable to train pre-trained models based on the dataset ImageNet and ResNet, and thus there are many pre-trained models available [15,21]. These pre-trained ResNet models can be used as the feature extractors in many image processing tasks.

2.3. CADA-VAE Model.

CADA-VAE is the latest ZSL algorithm, which was proposed by Schonfeld et al. [20] in 2019. In CADA-VAE, the authors built two VAE models to reconstruct the features and attributes respectively. Through making a cross-connection within the two VAE models, the model could learn the shared cross-modal latent representations of attributes and images features to enhance the representation of both attributes and features in the latent space. Then the authors aligned the distribution of features and attributes in the third space (i.e., the latent space). In this way, they could generate the most similar features to the raw images by only using the class attributes. The cross alignment and cross-modal representation make this algorithm achieve state-of-the-art performance on many ZSL datasets. This model also provides a way to minimize the semantic gap between the features and the attributes.

Like most of the ZSL algroithms, the input of the CADA-VAE is the features extracted by ResNet-101 and the class attributes. Therefore, one can infer that the qualities of the features and the class attributes have a significant impact on the performance of the CADA-VAE model.

As we mentioned in Section 1, although the pre-trained models trained with ResNet on large-scale datasets have rich information, they may ignore the intrinsic connection between the datasets used for training the pre-trained models and the current ZSL dataset. Therefore, the qualities of the features and the class attributes are hard to be guaranteed. To solve this problem, we design a novel dataset-specific feature extractor named ALT-DSFE for ZSL, which is extention of the pre-trained ResNet with the information of the seen classes and attributes. ALT-DSFE can exploit the inner relationship between the datasets used for training the pre-trained models and the current ZSL dataset to some extent. And then ALT-DSFE provides specific features for different ZSL tasks. Next, we present the details of ALT-DSFE.

3. Dataset-specific feature extractor according to ALT (ALT-DSFE)

As we mentioned above, most of the existing ZSL models use the pre-trained CNNs as their feature extractors. These pre-trained CNNs are generally obtained by using CNN based deep learning algorithms to train on the large-scale data sets such as ImageNet [6]. These training datasets almost have no direct connection to the datasets used in ZSL. What's worse, there is no other strategy adopted in the ZSL algorithms to fine-tune the parameters of these pre-trained models in the training process of ZSL to adapt to the current task. As a consequence, the feature mismatch problem always makes it difficult for these feature extractors (i.e., the pre-trained models) to be used sufficiently, and the low quality of extracted features makes it difficult for ZSL algorithms to train a model with good generalization ability.

To alleviate this problem, we propose a novel dataset-specific feature extractor according to an attribute-based label tree (ALT-DSFE) in this section, which can utilize the information of the seen classes and attributes to fine-tune the parameters of the pre-trained models. In this way, the extracted features could be much more suitable for the current ZSL task and help to improve the training efficiency of the ZSL algorithm and the accuracy of the model. Next, we present the details of the proposed ALT-DSFE.

To make a better introduction to the proposed ALT-DSFE, we first make a review of the concept of the Label Tree (LT) and describe a new concept named Attribute-based Label Tree(ALT).

Label Tree (LT) and Attribute-based Label Tree(ALT). Label Tree (LT) is a tree structure constructed by a hierarchical class label where a high-level class label contains multiple low-level class labels. For example, ImageNet [6] dataset has a hierarchical label tree, which is built according to the WordNet database [17]. However, the WordNet database is organized by humans, which means that the structure of the WordNet database is built subjectively.

In ZSL, benchmark datasets have specifical attributes to describe each class with the same criterion. All these attributes are numeric vectors, and intuitively one can use the K-means technique to group them and generate newly high-level labels. The center of the super-classes can be chosen as the clustering center of basic classes. In this way, a hierarchical label tree can be organized in a relatively objective way. We call it Attribute-based Label Tree(ALT).

To satisfy the setting requirement of ZSL algorithms, we input the seen classes' attributes into a revised K-means model to generate the high-level labels. If the number of the generated label's level is more than two, the centroid of the higher level label would be chosen as the means of the base-level vectors. In transductive ZSL, we can use the seen and unseen classes' attributes to build the hierarchical label tree. In this way, we can transfer more knowledge from the seen classes to the unseen classes.

Next, we introduce the details of the proposed ALT-DSFE.

Dataset-Specific Feature Extractor According to ALT (ALT-DSFE). In our method, we use the pre-trained ResNet as the base feature extractor. Different from existing methods that only use one classifier to predict the label, our method uses multiple classifiers to predict a label tree. It is worth noting that if the label tree has more than one-level, the number of classifiers in our method will be set as the same number of level. For example, given a dataset with N level labels, we use N classifiers to do the prediction. In this way, we can use the information of the seen classes and attributes to fine-tune the parameters of the pre-trained model. We call this method ATL based Datasets-Specific Feature Extractor (ALT-DSFE). The structure of the proposed ALT-DSFE is shown as Fig. 1.

As shown in Fig. 1, the most obvious feature of ALT-DSFE is that there are multiple fully-connected layers connected to the output layer of the feature extractor.

It is noted that the conventional models usually only have one classifier and their fine-tuning process is very slow. In our method, we use mutiple classifiers to speed up the fine-tuning process by optimizing their joint loss at the same time. For example, suppose

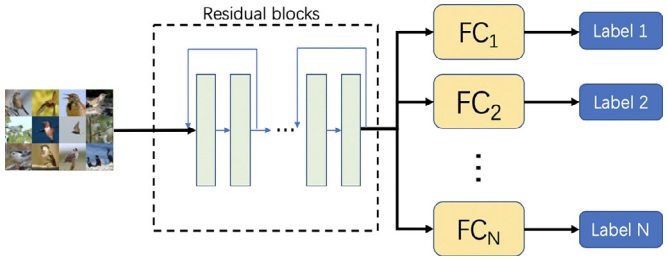


Fig. 1. The structure of ALT-DSFE.

Table 1
Datasets introduction.

	Seen	Unseen	Total	Attributes
CUB	150	50	200	312
SUN	645	72	717	102
AWA2	40	10	50	85

Note: The values in the column Seen, Unseen, and Total refer to the number of the seen classes, unseen classes, and total classes, respectively. The values in the column Attributes refers to the dimension of the vector that describes each class.

there are N classifiers, the loss function can be expressed as

$$L = \lambda_1 * L_1 + \lambda_2 * L_2 + \ldots + \lambda_n * L_N \tag{2}$$

$$L_i = \sum -(y_i^m \log target_i^m + (1 - y_i^m) \log\left(1 - target_i^m\right))(1 \le i \le N) \tag{3}$$

where L_i is the loss function of different level classifiers, λ are used to control the influence of different higher-level labels, m is the number of samples in each seen class, and i is the level value of the label.

The learning process of the proposed algorithm is shown in Algorithm 1. Once the training process is done, we can obtain the dataset-specific feature extractor.

4. Experimental results and analysis

In this section, we give the details of our experimental settings and results.

4.1. Experimental datasets

In our experiments, the training datasets should have both the raw images and the attributes information of the classes. Thus we chose three typical ZSL datasets that satisfy this requirement, that is, CUB-200-2011 (CUB) [25], SUN attribute (SUN) [19], and Animal with attribute 2 (AwA2) [27], as our experimental datasets. The details of the three datasets are shown in Table 1.

4.2. Experimental settings and results

In our study, we used the pre-trained ResNet-101 excluding the last pooling layer as the base feature extractor of the proposed ALT-DSFE. To demonstrate the scalability of our algorithm, we chose a state-of-the-art model CADA-VAE [20] as the base classifier and the learning rate of the classifiers is ten times to the learning rate of the ResNet-50 network. The parameter λ used in our method is set to 1. The total losses of all classifiers are optimized together to get the final feature extractor.

Once the training process is done, we can input the learned features to $CADA-VAE$ [20] and evaluate the performance of the model. The details of the experimental results are shown in Tabel 2.

Algorithm 1 ALT-DSFE ALGORITHM.

Input: Learning rate α, influence controller λ, training seen classes $\{(x^h|(x^h) \in (X_{seen})\}$, unseen classes X_{unseen}; the class attributes $A = a_1, a_2, ..., a_n$; the number of the clusters in each level label K; the number of the levels N.

Output: The parameters of all level classifiers θ_R, noval features F($F_{seen}, F_{unseen} \in F$).

Step 1: K-means clustering

Randomly initialize k sample $a_k (1 \le k \le n)$ as the clusters centroids $c_1, c_2, ..., c_K$
Initialize Flag=True.
if $N == 1$ **then** break
else
  for $level = 2, ..., N$ **do**
    while Flag == True **do**
      Initialize the clustering sets $S^{(i)}$ as empty sets.$(1 \le i \le K_{level})$
      for $l = 1, 2, ..., n$ **do**
        Calculate $L2$ distance between the classes attribute vector a_l and the clusters centroid vector $c_j (1 \le j \le K_{level})$
        $d_{lj} = \sqrt{(a_l - cj)^2}$ $(1 \le l \le n, 1 \le j \le K_{level})$
        Put the nearest a_l to the sets Z_j
        $Z_j = argmin_{k \in \{1,2,...,K\}} d_{ji}$
      for $k = 1, 2, ..., K$ **do**
        Calculate the new centroid c'_k
        $c'_k = \frac{1}{|S^{(k)}|} \sum_{a \in S^{(i)}} a$
        if $c'_i \ne c_i$ **then**
          Update the centroid with c'_i
          $c_i = c'_i$
        else
          $Flag = False$
    Initialize the centroid set c_i as the input of next level label.
  Return all level labels

Step 2: Constructing ALT

Generate label tree data $S = \{(x^h, y_1^h, y_2^h, ..., y_N^h)|(x_h, y_1^h, y_2^h, ..., y_N^h) \in (X_{seen}, Y_1, Y_2, ..., Y_N)\}$
Initialize $\theta_R, \theta_{C_1}, \theta_{C_2}, ..., \theta_{C_N}$
while not done **do**
  for $1 \le m \le N$ **do**
    $L_m = -(y_m^h \lg\left(\theta_{C_m}(\theta_R(x^h))\right) + (1 - y_m^h) \lg\left(1 - \theta_{C_m}(\theta_R(x^h))\right)$
  $L = L_1 + \sum_{m \in \{2,N\}} \lambda_m * L_m$
  $(\theta_R, \theta_{C_1}, \theta_{C_2}, ..., \theta_{C_N}) = (\theta_R, \theta_{C_1}, \theta_{C_2}, ..., \theta_{C_N}) - \alpha * \nabla L$
Input X_{seen}, X_{unseen} to θ_R generating the noval features F.
$F = \theta_R(X_{seen}, X_{unseen})$
Return θ_R, F

From Table 2, we can observe that compared with other ZSL algorithms, our method can significantly improve the prediction accuracy of the model. For example, on the dataset *CUB*, our method improves the synthetical accuracy (i.e., H) of the state-of-the-art CADA-VAE model from 52.4% to 64.2%. Similar observations can also be found in the dataset *AwA2*. For the dataset *SUN*, although our model does not achieve the highest accuracy, the gap with the current state-of-the-art CADA-VAE model is very small.

These experimental results imply that our method can provide ZSL classifier with higher quality features, which proves that the feature extractor proposed in this paper is effective.

Besides, we studied the effect of the proposed algorithm on the efficiency of model training. In the experiment, we used the same input data and the same hyper-parameters for the algorithm

Table 2
The details of the experimental results.

	CUB			SUN			AWA2		
	acc_{seen}	acc_{unseen}	H	acc_{seen}	acc_{unseen}	H	acc_{seen}	acc_{unseen}	H
LATEM [26]	57.3	15.2	24.0	28.8	14.7	19.5	77.3	11.5	20.0
CVAE [18]	-	-	34.5	-	-	26.7	-	-	51.2
SP-AEN [5]	34.7	70.6	46.6	24.9	38.6	30.3	23.3	90.9	37.1
PSRZSL [2]	24.6	54.3	33.9	20.8	37.2	26.7	20.7	73.8	32.3
DeViSE [7]	53.0	23.8	32.8	27.4	16.9	20.9	74.7	17.1	27.8
ALE [1]	62.8	23.7	34.4	33.1	21.8	26.3	81.8	14.0	23.9
SYNC [4]	70.9	11.5	19.8	43.3	7.9	13.4	90.5	10.0	18.0
SE [12]	53.3	41.5	46.7	30.5	40.9	34.9	68.1	58.3	62.8
f-CLSWGAN [28]	57.7	43.7	49.7	36.6	42.6	39.4	68.9	52.1	59.4
CADA-VAE [20]	53.5	51.6	52.4	35.7	47.2	**40.6**	75.0	55.8	63.9
Ours	68.0	60.1	**64.2**	37.7	42.8	40.1	78.5	55.6	**65.1**

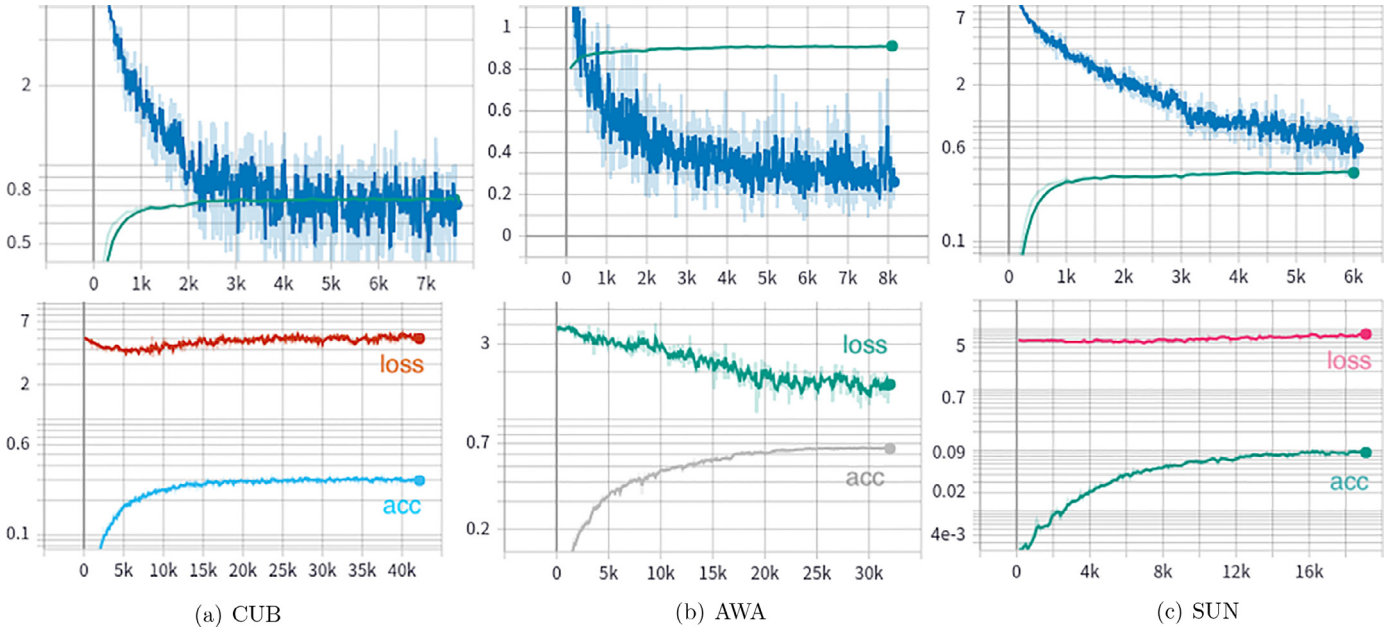


Fig. 2. The changing curves of the loss value and the accuracy of the model on the three datasets. On the top figures, green lines refer to the accuracy of the model, the blue lines refer to the loss change during the learning process of the model with the multi-level label training strategy. The meanings of the curves in the bottom figure are shown in the figures.

Table 3
The effect of the multi-level label and the one-level label training strategies on the performance of the model.

DATASETS	Multi-level	One-level
CUB	74.5	44.9
SUN	46.3	10.1
AWA2	91.2	66.2

Note: Multi-level and One-level refer to that we use the information extracted from the multi-level label and the base label to fine-tune the pre-trained ResNet respectively.

but only with different level labels. Then we marked the changing curve of the loss value and the accuracy of the model with a multi-level label training strategy and that of the model with a one-level label training strategy. The experimental results are shown in Table 3 and Fig. 2.

From Table 3, we can observe that the model with a multi-level label training strategy has significantly better performance than the model with a one-level label training strategy on all the three datasets. For example, the accuracy of the model with a multi-level label training strategy is more than 30% than that of the model with a one-level label training strategy on the dataset *SUN*.

In addition, Fig. 2 shows that the proposed multi-level labels training strategy can accelerate the convergence rate of the algorithm and greatly reduce the training time. For example, multi-level labels feature extractor only needs about 6000 iterations to converge, but one-level labels feature extractor needs at least 15,000 iterations. Moreover, it can be seen from Fig. 2 that the loss value of the multi-level labels feature extractor decreases significantly faster than that of the one-level labels feature extractor, which means that our method can effectively accelerate the training efficiency of the model.

In conclusion, the experimental results on three benchmark datasets show that our proposed feature extractor is effective. Next, we explain the above experimental phenomena from a visual point of view.

4.3. Explanation to the experimental results

The above experimental results reflect the effectiveness of the proposed method from the perspective of the predictive accuracy

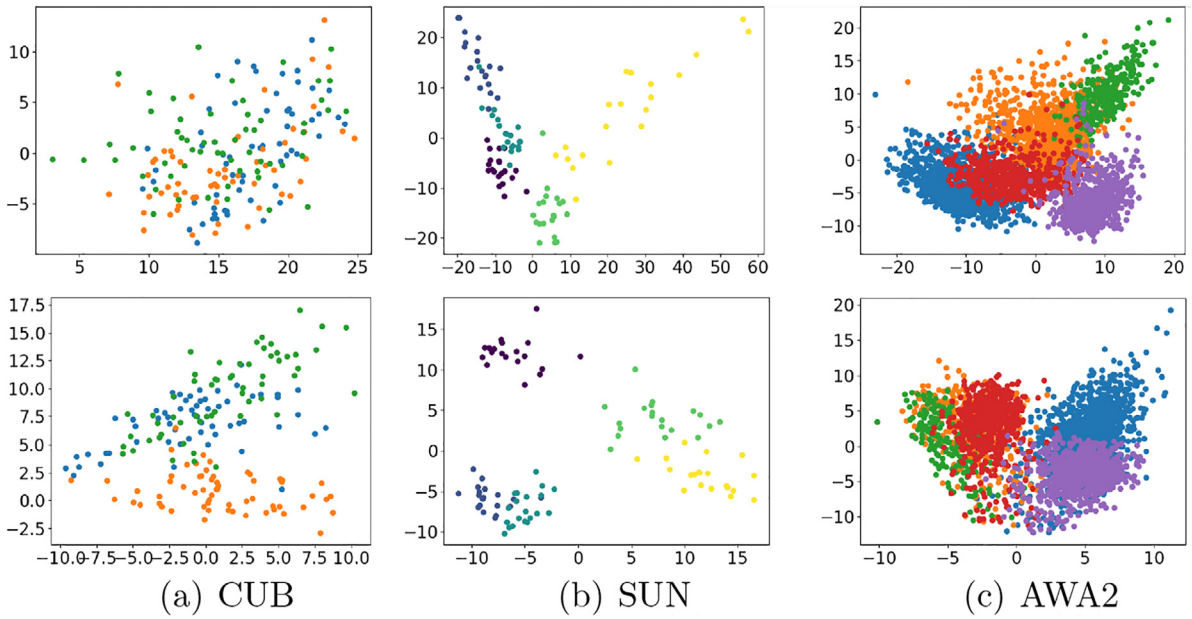
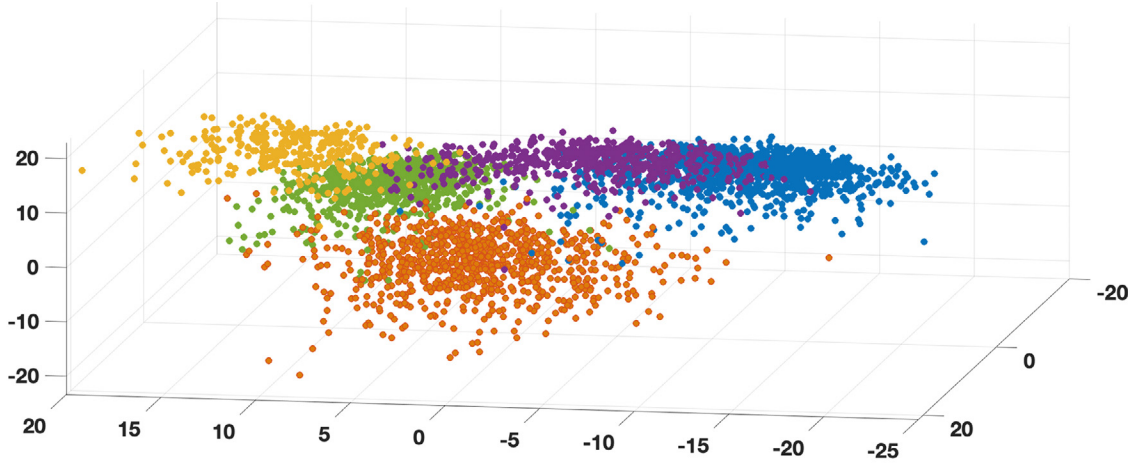
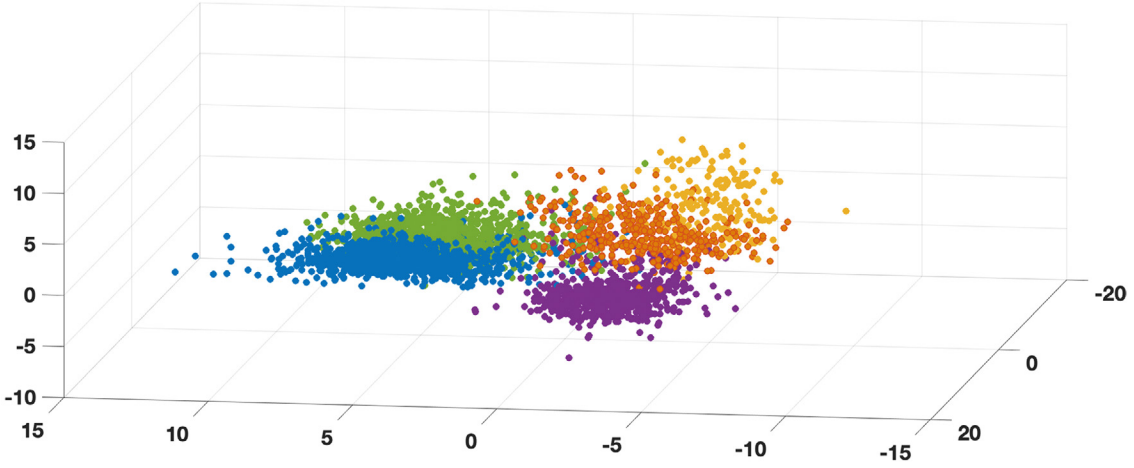


Fig. 3. The results of feature visualization of three datasets.



(a) Original features



(b) Features extracted by our mehtod

Fig. 4. *AWA2* features visualized on 3D.

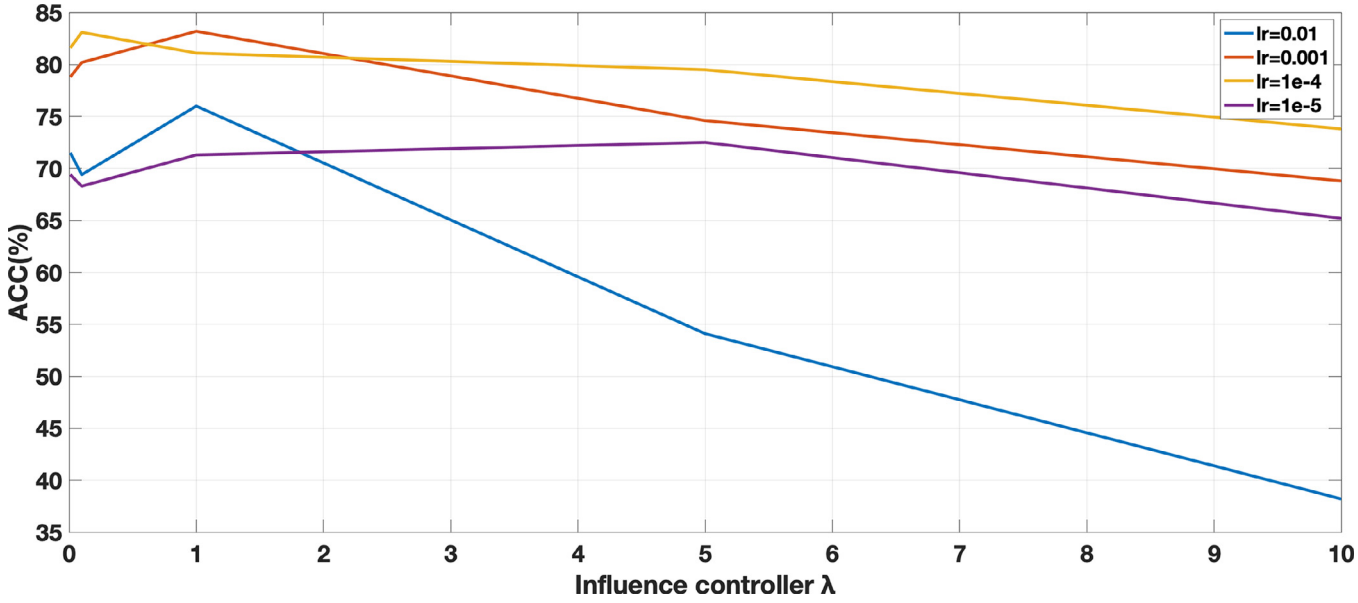


Fig. 5. Sensitivity of the hyper parameters to the performance of model.

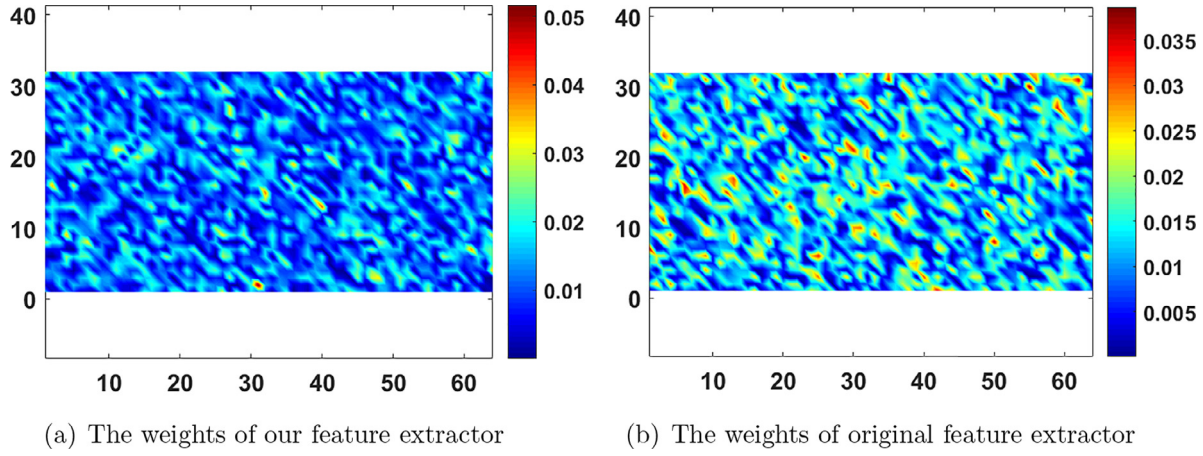


Fig. 6. The weights comparison of our feature extractor and the original one.

and convergence rate of the model. In this section, we directly study the effect of our method on the change of data features in a visual way to explain the above experimental results. Specifically, we visualized the original features and the extracted features by our method through the dimensionality reduction method such as PCA. The experimental results are shown in Fig. 3.

From Fig. 3, the first line figures are the distribution of the original features, the second line figures are the distribution of the features extracted by our method. We can observe that the features at the first line are more separable than that of the second line. For example, the original features of *CUB* are mixed, while the features extracted by our method have an explicit classification boundary. Similar phenomena can be observed from the experimental results on the dataset *SUN*.

In the case of two-dimensional visualization, the experimental results on the dataset *AwA*2 are not obvious enough. Therefore, we use a three-dimensional visualization method to optimize it and the experimental results are shown in Fig. 4. From Fig. 4, we can clearly observe that the feature extracted by our method are more separable than the original features.

The parameter value with better performance among several trials will be given. In this paper, we conducted extra experiments to test the sensitivity of the influence controller λ and learning rate on the performance of the proposed method. From our experiments result in Fig. 5, we found that, when the parameter learning rate was set to 0.01 and λ was set to 1, the model could get the highest accuracy.

Through the above analysis, we can give a speculative explanation for the experimental results in this paper. That is, the proposed feature extractor can make the data features more separable, which helps the classifier to make the decision faster and better. It validated that the convergence speed and accuracy of the model have been greatly improved.

4.4. Remark

(1) The validity of the proposed algorithm is verified from Bartlett theory.

Inspired by the Bartlett theory, that is, the smaller the norm of weights, the better the generalization ability of the model

for those feed-forward neural networks with the same network complexity [3], we did an extra experiment to test whether the norm of the learned weights is smaller than that of the original weights. The experimental results are shown in Fig. 6 from which we can observe that our features weight have more values close or equal to zero. These results mean that our model has a greater ability in generalization than the original one.

(2) Our method vs transfer learning.

Our feature extractor is trained by using a pre-trained model and the training process of our model indeed uses the idea of transfer learning, parameter-transfer. It's confirmed that using our feature extractor can greatly reduce training time and computational cost.

(3) Our method vs ensemble learning.

Our method is totally different from ensemble learning. Ensemble learning is to use multiple classifiers to make predictions, and by combining these predictions, to determine a final label. In our approach, the multiple classifiers are used to predict multiple level of class labels rather than obtaining a final label.

(4) Why ResNet? Many variants of CNN, such as VGG and Inception, can be the feature extractor in ZSL. A considerable number of references verify algorithmically and experimentally that features extracted by ResNets are really more effective than those by other CNN variant models. Mathematically and logically, the essential explanation to this reason is still unclear so far.

5. Conclusions

In this work, we proposed a novel feature extraction method named ALT-DSFE for ZSL. ALT-DSFE uses the information extracted from the Attribute-based Label Tree (ALT) to fine-tune the parameters of the pre-trained ResNet model and then provides the ZSL classifier with dataset-specific features. Compared with the traditional feature extraction methods for ZSL (i.e. directly using the pre-trained models as the feature extractor), ALT-DSFE can provide ZSL with features that are closely related to the current task, which helps the classifier to make the prediction better. The experimental results on three benchmark datasets show that ALT-DSFE can not only effectively improve the predictive accuracy of the ZSL model, but also significantly accelerate the convergence rate of the model. We analyzed the experimental phenomena from the perspective of visualization, which experimentally show that our method can make the features more separable than the pre-trained ResNets. It implicitly explains why the proposed feature extractor can improve the classification ability of the ZSL models.

Although ALT-DSFE provides an efficient way to extract datasets-specific features for ZSL, many issues are remained to be further studied. For example, we still cannot explain the effectiveness of ALT-DSFE theoretically. In the future, we will further explore this issue and test the ALT-DSFE's sensitivity to different types of classifiers such as SVM [31].

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Yuxuan Luo: Methodology, Software, Validation, Writing - original draft, Writing - review & editing, Visualization. **Xizhao Wang:** Conceptualization, Writing - review & editing, Supervision, Project administration, Funding acquisition. **Weipeng Cao:** Validation, Writing - original draft, Writing - review & editing, Visualization.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (grants nos. 61976141 and 61732011) and in part by Basic Research Project of Knowledge Innovation Program in ShenZhen (grant nos. JCYJ20180305125850156).

References

[1] Z. Akata, F. Perronnin, Z. Harchaoui, C. Schmid, Label-embedding for image classification, IEEE Trans. Pattern Anal. Mach. Intell. 38 (7) (2015) 1425–1438.
[2] Y. Annadani, S. Biswas, Preserving semantic relations for zero-shot learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7603–7612.
[3] P.L. Bartlett, The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network, IEEE Trans. Inf. Theory 44 (2) (1998) 525–536, doi:10.1109/18.661502.
[4] S. Changpinyo, W.-L. Chao, B. Gong, F. Sha, Synthesized classifiers for zero-shot learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5327–5336.
[5] L. Chen, H. Zhang, J. Xiao, W. Liu, S. Chang, Zero-shot visual recognition using semantics-preserving adversarial embedding network, CoRR abs/1712.01928 (2017).
[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, Ieee, 2009, pp. 248–255.
[7] A. Frome, G.S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, T. Mikolov, Devise: a deep visual-semantic embedding model, in: Advances in Neural Information Processing Systems, 2013, pp. 2121–2129.
[8] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in: European conference on computer vision, Springer, Cham, 2016, pp. 630–645.
[9] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[10] G. Hinton, L. Deng, D. Yu, G.E. Dahl, A.R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups, IEEE Signal Process. Mag. 29 (6) (2012) 82–97.
[11] E. Kodirov, T. Xiang, S. Gong, Semantic autoencoder for zero-shot learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3174–3183.
[12] V. Kumar Verma, G. Arora, A. Mishra, P. Rai, Generalized zero-shot learning via synthesized examples, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4281–4289.
[13] C.H. Lampert, H. Nickisch, S. Harmeling, Learning to detect unseen object classes by between-class attribute transfer, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 951–958.
[14] Y. Li, J. Zhang, J. Zhang, K. Huang, Discriminative learning of latent features for zero-shot recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7463–7471.
[15] B. Lim, S. Son, H. Kim, S. Nah, K. Mu Lee, Enhanced deep residual networks for single image super-resolution, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2017.
[16] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, S. Khudanpur, Recurrent neural network based language model, Eleventh annual conference of the international speech communication association, 2010.
[17] G.A. Miller, WordNet: An Electronic Lexical Database, MIT press, 1998.
[18] A. Mishra, S.K. Reddy, A. Mittal, H.A. Murthy, A generative model for zero shot learning using conditional variational autoencoders, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2018.
[19] G. Patterson, J. Hays, Sun attribute database: Discovering, annotating, and recognizing scene attributes, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 2751–2758.
[20] E. Schonfeld, S. Ebrahimi, S. Sinha, T. Darrell, Z. Akata, Generalized zero-and few-shot learning via aligned variational autoencoders, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 8247–8255.
[21] M. Simon, E. Rodner, J. Denzler, Imagenet pre-trained models with batch normalization, CoRR (2016) abs/1612.01452.
[22] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, International Conference on Learning Representations (ICLR), 2015.
[23] C. Szegedy, S. Ioffe, V. Vanhoucke, A.A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017.
[24] X. Wang, Y. Ye, A. Gupta, Zero-shot recognition via semantic embeddings and knowledge graphs, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6857–6866.
[25] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, P. Perona, Caltech-UCSD Birds 200, Technical Report, California Institute of Technology, 2010.
[26] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, B. Schiele, Latent embeddings for zero-shot classification, Computer Vision & Pattern Recognition, 2016.

[27] Y. Xian, C.H. Lampert, B. Schiele, Z. Akata, Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly, IEEE Trans Pattern Anal Mach Intell (2018).
[28] 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, pp. 5542–5551.
[29] Y. Xian, S. Sharma, B. Schiele, Z. Akata, f-VAEGAN-D2: A feature generating framework for any-shot learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 10275–10284.
[30] N. Zeng, Z. Wang, H. Zhang, K. Kim, Y. Li, X. Liu, An improved particle filter with a novel hybrid proposal distribution for quantitative analysis of gold immunochromatographic strips, IEEE Trans. Nanotechnol. 18 (2019) 819–829, doi:10.1109/TNANO.2019.2932271.
[31] N. Zeng, H. Qiu, Z. Wang, W. Liu, H. Zhang, Y. Li, A new switching-delayed-pso-based optimized svm algorithm for diagnosis of alzheimer's disease, Neurocomputing 320 (2018) 195–202.
[32] N. Zeng, Z. Wang, H. Zhang, W. Liu, F.E. Alsaadi, Deep belief networks for quantitative analysis of a gold immunochromatographic strip, Cognit. Comput. 8 (4) (2016) 684–692.
[33] L. Zhao, X. Wang, Seemingly unrelated extreme learning machine, Neurocomputing 355 (2019) 134–142.

Yuxuan Luo received his bachelor degree in Information Management and Information System in June 2017 from Guangdong Pharmaceutical University, Guangzhou, China. He is currently a master student major in Software Engineering at College of Computer Science and Software Engineering, Shenzhen University. His research interests include machine learning and deep learning.

Xizhao Wang was a Professor and the Dean with the School of Mathematics and Computer Sciences, Hebei University, before 2014. Since 2014, he has been working as a Professor with Big Data Institute, ShenZhen University, ShenZhen, China.His main research interests include uncertainty modeling and machine learning for big data. He has edited more than special issues and authored and co-authored three monographs, two textbooks, and more than 200 peer-reviewed research papers. As a Principle Investigator or co-Principle Investigator, he has completed more than 30 research projects. He has supervised more than 100 M.phil. and Ph.D. students. Prof.Wang is the previous Board of Governorsmember of the IEEE Systems, Man, and Cybernetics (SMC) Society, the Chair of the IEEE SMC Technical Committee on Computational Intelligence, the Chief Editor of Machine Learning and Cybernetics Journal, and an Associate Editors for a couple of journals in the related areas. He was the recipient of the IEEE SMCS Outstanding Contribution Award in 2004 and the recipient of IEEE SMCS Best Associate Editor Award in 2006. He is the general Co-Chair of the 2002–2017 International Conferences on Machine Learning and Cybernetics, cosponsored by the IEEE SMCS. He was a Distinguished Lecturer of the IEEE SMCS.

Weipeng Cao received his PhD degree in Computer Science in June 2019 from Shenzhen University, Shenzhen, China. He is currently an associate researcher at College of Computer Science and Software Engineering, Shenzhen University. Since 2017 he is serving as a visiting scholar at the School of Engineering and Computer Science, University of the Pacific, California, USA. His research interests include machine learning and deep learning.