Cross-Class Sample Synthesis for Zero-shot Learning

Jinlu Liu
liujinlu@ruc.edu.cn
Xirong Li
xirong@ruc.edu.cn
Gang Yang*
yanggang@ruc.edu.cn

School of Information
Renmin University of China
Beijing, China
*Corresponding author

Abstract

Zero-shot learning (ZSL) aims to recognize unseen classes which have no available training samples, through establishing an association with seen classes. Existing approaches mostly learn a comparability function to predict the class of an image. Different from previous approaches, we put forward a novel method, Cross-Class Sample Synthesis (CCSS), to directly synthesize samples of unseen classes from specific seen classes in the visual feature space. We adopt class-graph to measure inter-class similarity and propose class entropy to select classes as the synthesis source of target classes. An end-to-end network is constructed to realize sample synthesis from source classes to target classes. Specially, rule of attribute guiding cross-class transfer is built into the network, to which various samples of different source classes can be used to synthesize samples of each target class according. The synthesized samples are used as training data of unseen classes and it turns ZSL into a supervised learning problem. Experiments on five benchmark datasets efficiently demonstrate the advantage of our proposed method.

1 Introduction

Zero-shot learning (ZSL) is proposed to achieve object recognition of classes which have no available training data [18]. These classes are regarded as unseen classes in ZSL while classes with labeled samples are viewed as seen classes. The main idea of ZSL is establishing an association between seen and unseen classes to achieve unseen classes recognition when lacking labeled samples [12].

Associating unseen classes with seen classes via limited information is a tough task. Researchers usually use attribute, textual descriptions or word vectors as side information to construct a space, where seen and unseen classes are related [2, 16]. Attributes are most commonly used in zero-shot learning and also adopted in this paper to correlate different classes. There are two popular frameworks in zero-shot learning. Compatibility learning is a widely adopted framework which predicts the class by calculating compatibility scores for each image [1, 7, 20, 23]. Mixture of source classes is another framework which uses linear/non-linear combination of semantic embeddings to predict labels of unseen objects [3, 14, 17, 26]. These existing methods are at risk of overfitting to seen classes but performing

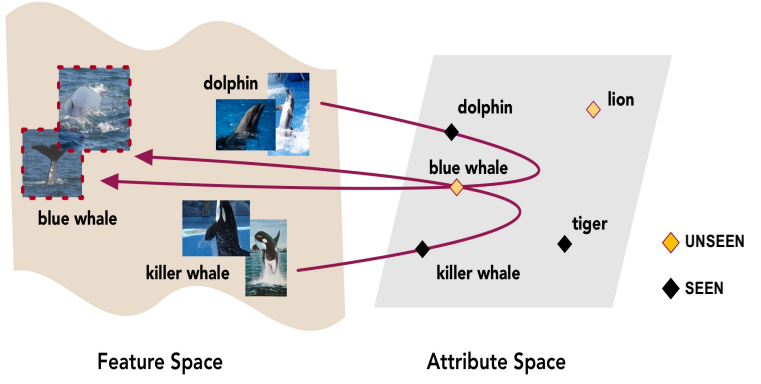


Figure 1: An intuitive illustration of sample synthesis. Samples of unseen class 'blue whale' can be synthesized from samples of seen classes 'dolphin' and 'killer whale' under the guidance of class attributes. Various samples of different source classes can be used to synthesize the same sample of target class due to that the rule of attribute guiding cross-class transfer has been automatically established at training stage.

poorly on unseen classes to an extent, due to the problem of lacking training data from unseen classes [4]. Under this consideration, if samples of unseen classes are synthesized, ZSL is simplified into a supervised learning problem which can be tackled by existing methods such as SVM.

Inspired by this idea, we propose a novel method, named Cross-Class Sample Synthesis (CCSS), to synthesize abundant samples of unseen classes in the visual feature space, which can effectively tackle the problem of lacking labeled samples. Different from existing synthesis methods, samples synthesized by our proposed method and real samples of unseen classes have not only close distribution but also similar visual features. Our proposed method is schematically illustrated in Figure 1 where samples of unseen classes (targets) are synthesized from samples of seen classes (sources) in feature space, under a guidance of class attributes. In the process of sample synthesis, there are two key points of decisive effect: selecting 'source-target' pairs of synthesis and setting rules of sample synthesis. As for the selection of source classes, we adopt the idea of class-graph to measure inter-class similarity and furthermore, class entropy is defined to determine the exact number of source classes. On the other hand, an end-to-end network is constructed to realize sample synthesis from source to target, which automatically establishes transfer rules across different classes when training. While after training, with seen classes similar to unseen classes put in as synthesis sources, the network will generate abundant samples that belong to unseen classes. In particular, once we have these labeled data, many supervised classifiers can be specifically trained to recognize unseen classes. The contributions of this paper are:

1. Different from previous synthesis methods projecting both unseen and seen classes into a latent embedding space to synthesize samples according to joint space distribution, our method directly synthesizes samples of each unseen class from certain sample or various samples of different seen classes. Cross-class transfer is guided by class attribute without finding a joint embedding space.

2. We directly synthesize samples of unseen classes, which can be used as labeled data to train a classifier. It specifically aims to tackle a key problem of zero-shot learning: lack of

labeled samples.

3. We conduct experiments on five benchmark datasets, adopting an unified evaluation criteria to give a comprehensive comparison with previous approaches. Experiment results are comparable and convincing to show the good performance of our proposed method.

2 Related Work

Most existing methods align image feature with attribute or word vector to correlate seen classes with unseen classes. CONSE [17] uses convex combination to map images into the semantic embedding space of class labels in vocabulary from existing image classifiers. As introduced in SYNC [3], linear projection is proposed to align semantic space with model space where virtual classifier of unseen classes can be formed by convex combination of semantic space coordinates. EXEM [4] projects visual feature and semantic representation into semantic embedding space by using structural constraint. Bilinear/non-linear compatibility function is commonly utilized to associate different domains. For example, DeViSE [7] is proposed as a linear embedding model mapping deep visual feature with word vector based on a ranking loss. [1] defines a bilinear form of compatibility function between input space and structured output space which recognizes unseen samples by finding the highest joint compatibility score in label yield.

SAE [11] also learns linear projection matrix between feature space and semantic space and it's like a linear auto-encoder with additional constrains. To some extent, our model is also similar to auto-encoder, but differently, our model is designed as an end-to-end structure which is better behaved than SAE.

Distinguished from the above approaches, methods to synthesize samples have emerged in recent years. UVDS [15] synthesizes visual features from semantic attributes during semantic-visual embedding. IBSC [25] synthesizes unseen classes samples through recombination of seen classes samples. Both UVDS and IBSC are inspired by the ability of human imagination. [21] proposes a deep generative neural network which has two stages including constructing-synthesizing and inverse interpreting. However, it uses orientation-invariant feature of synthetic aperture radar (SAR) dataset which is not a popular dataset in ZSL so that it is weakly comparable with existing benchmarks. [8] estimates probability distribution of unseen classes so as to synthesize data by random sampling. Although the synthesized data has a close distribution to real data in mathematical form, the synthesized data in visual feature space is unaccountable compared with original feature. Furthermore, experiment in [8] has a pre-trained problem mentioned in [24]. [24] emphasizes an ubiquitous problem in previous experiments that some test classes are included in 1K classes of ImageNet which are used to pre-train feature extractors [5]. Unlike these methods, we synthesize samples of unseen classes whose visual feature is similar to real data. Meanwhile, we conduct comprehensive experiments on both originally split and newly split benchmark datasets to measure the influence of the pre-trained problem.

3 Cross-Class Sample Synthesis

Our method, Cross-Class Sample Synthesis(CCSS), is proposed to synthesize samples of unseen classes to accomplish the task of unseen class recognition. In particular, synthesizing unseen classes refers to synthesizing samples of unseen classes. Based on an observa-

tion that different classes have specific locations in the feature space, sample synthesis can be achieved through transferring among different locations. Because of difference among classes reflected in attribute representation, the transfer across various classes is guided by class attribute. To synthesize high-quality samples via proper transferring, we perform the following methodology. First, select proper samples in source classes as bases of synthesis. Strategy based on inter-class similarity and class entropy is applied to choose appropriate 'source-target' pairs of synthesis. Second, we train an end-to-end sample synthesis network to build a rule of inter-class transfer. It is a symmetrical multilayer network that automatically establishes a rule of transferring between feature space and attribute space to synthesize samples from source to target at training stage. Then, selected source classes are input into the network, generating abundant synthesized samples of target classes. Specifically, during the training stage, both source classes and target classes are seen classes but when generating, source classes are seen classes while target classes are unseen classes. Given these synthesized samples, most traditional classifiers can be used to recognize unseen classes.

3.1 Notations

To clearly introduce our method, notations are given as follows. There are total T classes $\mathcal{C} = \{c_i\}_{i=1}^{T}$ in a dataset which is usually split into seen classes and unseen classes. Only labeled data set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ of seen classes is given as training data, where $x_i \in \mathbb{R}^{D_f}$. Attributes of all classes are denoted by $\mathcal{A} = \{a_i\}_{i=1}^{T}$ where $a_i \in \mathbb{R}^{D_a}$. D_f/D_a is the dimension of visual feature/attribute.

3.2 Synthesis Source Selection

To achieve high-quality target sample synthesis, selecting suitable samples in proper classes as source is of great importance. Hence selection strategy is designed on two levels: *class selection* and *sample selection*. As the main idea of proposed method is cross-class synthesis, we put more emphasis on class selection.

Class Selection It's obvious to see that synthesis from similar classes avoids more information loss than from other classes since similar classes are located more closely in attribute and feature space. Sample synthesis is realized by space transfer, and it's crucial to build inter-class relationship. In general, attribute is used to measure similarity among classes. We adopt the idea about class-graph [3], meaning that all classes form a weighted bipartite graph in attribute space where each node denotes a class in form of attribute vector and weights of edges represent inter-class relationship. Weights are measured by Eq. (1):

$$s_{ij} = \frac{exp(-d_{ij})}{\sum_{i,j=1}^{T} exp(-d_{ij})} \tag{1}$$

where d_{ij} is the Mahalanobis distance between classes c_i and c_j. Thus, inter-class similarity can be measured by weight value s_{ij}. It is a simple way to select source classes with larger similarity to be synthesis source of target classes at both training and generating stage.

Moreover, we conduct more accurate source class selection when training network. We observe that samples in the same class are likely to have large visual difference, resulting in that most similar samples of them belong to various $n_{sim}(\geq 1)$ classes. It can be viewed as intra-class dispersion which makes it unreasonable to choose the most similar class of each unseen class as synthesis source. Hence when selecting source classes, dispersion in target

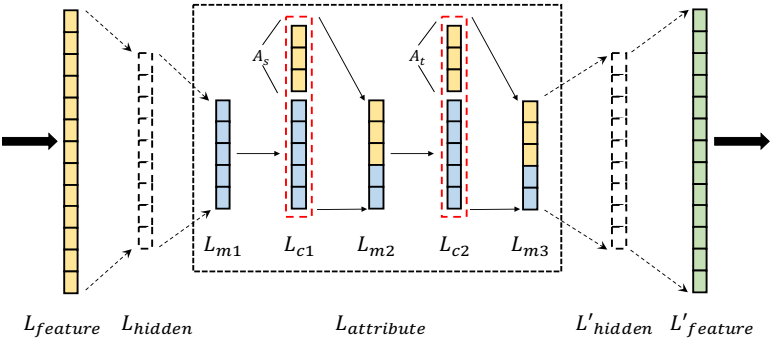


Figure 2: Sample Synthesis Network: an end-to-end symmetrical network where L_{hidden} and L'_{hidden} are multi hidden layers between feature layers $L_{feature}$/$L'_{feature}$ and attribute layers $L_{attribute}$. A_s/A_t is the attribute of source/target class, which is used to guide cross-class transfer.

classes should be taken into consideration. To measure the dispersion degree in a class, we compute the value of class entropy, ε_i, which is defined as Eq. (2):

$$\varepsilon_i = \sum_{j=1}^{n_{sim}} -p_{ij} \log_2 p_{ij} \tag{2}$$

where p_{ij} represents a percentage that in all samples of class c_i, how many of their most similar samples belong to class c_j. Target class of larger entropy value has larger intra-class dispersion thus, more classes are likely to be chosen as its synthesis source. Furthermore, we compute relative class entropy by:

$$\phi_i = \frac{\varepsilon_i}{\lambda} \tag{3}$$

where λ is the minimum value of all class entropies. ϕ_i represents the relative dispersion degree of class c_i compared with other classes in a dataset. Value of relative class entropy determines the exact number of selected source classes . For class with minimum dispersion, whose relevant class entropy equals to 1, the most similar class is chosen to be synthesis source. As for other classes, number of selected similar classes, measured by Eq. (1), is set to $\lfloor \phi_i \rfloor$ but not more than an upper limit. Obviously, class entropy can not be applied to unseen classes due to the lack of labeled data.

Sample Selection Since source classes have been chosen by the aforementioned strategy, source samples in these classes are selected by instance similarity which is measured by ℓ_2 distance in feature space. When preparing to train the network, we select samples in source classes with short ℓ_2 distance as sources for target samples.

3.3 Sample Synthesis Network

In our method, we design a deep network to synthesize target samples based on the idea about space transfer. The network automatically makes a set of rules that instruct samples of source classes to transfer to samples of target classes at training stage. After the network has been trained over on seen classes, put selected samples of seen classes into the net then we

will get abundant synthesized samples of unseen classes. As for sample synthesis, there is a key problem to be solved that the gap between two spaces causes information loss during space transferring, due to a large dimensionality difference between feature and attribute. To deal with it, we build a fully-connected network to realize space transfer for the reason that approach of full connection can bridge the gap by simplifying it to a regression problem. The network realizes sample synthesis by dimensionality reduction and dimensionality raising because an image can be represented in forms of different dimensions. First, the network makes an image feature (a sample of source class) regress to a lower dimension and then realizes inter-class transfer under a guidance of class attribute. Next, the feature is raised to original dimension so that we view the new feature as a sample of target class. Figure 2 displays the whole structure of the proposed network. It is a symmetrical network mainly constructed by layers of three types: feature layers, hidden layers and attribute layers. Feature layers include input layer $L_{feature}$ and output layer $L'_{feature}$ which respectively represent source image feature and target image feature. Hidden layers are constructed as transitional layers between feature layers and attribute layers. And attribute layer $L_{attribute}$ is made up of five layers: *medium layers* (L_{m1}, L_{m2}, L_{m3}) and *merged layers* (L_{c1}, L_{c2}).

Attribute is put into the network, determining transferring direction across different classes in terms of its class distinguishing ability. Even using one source sample, different target samples can be synthesized under the guidance of different class attributes. As shown in the middle part of Figure 2, attributes of source classes and target classes are put into merged layers guiding inter-class transfer when synthesizing samples. More detailedly, A_s/A_t is put in to concatenate a medium layer L_{m1}/L_{m2}, forming a merged layer L_{c1}/L_{c2}. During the regression from L_{c1} to L_{m2} and L_{c2} to L_{m3}, source samples have already had the characteristic of target classes and they will be synthesized as target samples after dimensionality raising.

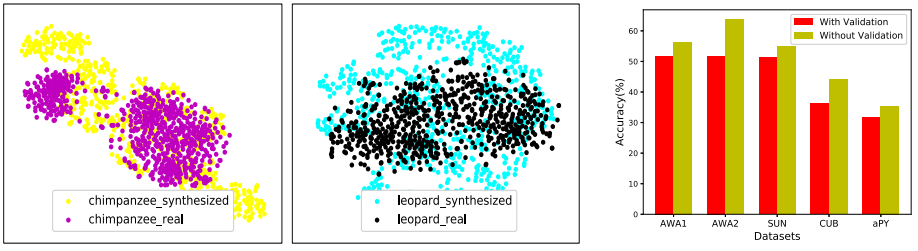
Regression between feature layers and attribute layers absolutely brings about much information loss of which hidden layers are set accordingly on account. Since practices have proven the good performance of multi layers in deep learning, it's of great importance to set appropriate quantity of hidden layers. We empirically design Eq. (4) to determine the quantity Q of hidden layers L_{hidden} and L'_{hidden}. D_f/D_a is the dimension of visual feature/attribute.

$$Q = \frac{\log_2 D_f - \lfloor \log_2 D_a - 1 \rfloor}{2} \tag{4}$$

As inter-class transfer is mainly realized in $L_{attribute}$ at lower dimensions, it's also essential to set proper dimensionality of medium layers so as to retain characteristic of samples in different classes. For purpose of retaining more information to avoid transferring deviation, we set D_m larger than D_a. Hence we design Eq. (5) to determine the dimensionality D_m of medium layers.

$$D_m = 2^{\lceil \log_2 D_a \rceil} \tag{5}$$

Rules of sample synthesis will be established in the network after training on seen classes then, we can attain variety of synthesized samples of unseen classes by putting selected samples of seen classes into the network. Furthermore, classification of unseen classes is simplified to a supervised learning problem which can be achieved by traditional classifiers.



(a) (b)

Figure 3: (a) t-SNE visualization of synthesized samples and real samples of unseen classes 'leopard' and 'chimpanzee' in AWA2. (b) Top-1 per-class accuracy of five originally split datasets when training network with/without validation set.

4 Experiments

4.1 Setup

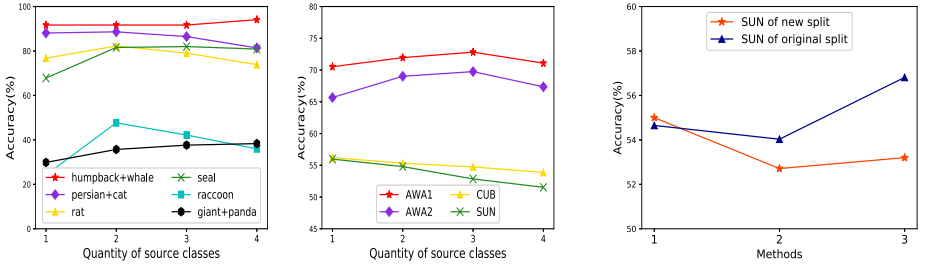
Datasets In order to make a clear comparison with existing methods, we experiment on five benchmark datasets: AWA1, AWA2, SUN, CUB and aPY. Dataset Animal with Attributes (AWA1) [13] contains total 30,475 images and 50 classes. The second is Animal with Attributes2 (AWA2) [24] which has been recently proposed, including the same 50 classes of AWA1. But AWA2 has 37,322 images in all which don't overlap with images in AWA1. Patterson and Hays [19] provide SUN dataset containing 717 classes and 14,340 images, which includes many indoor and outdoor objects. The fourth dataset is Caltech-USCD-Birds-200-2011 (CUB) [22] that has 200 kinds of birds. The last one is aPascal-aYahoo [6] of 32 classes and 15,339 images. Attribute dimensionalities of five datasets are: 85, 85, 102, 312 and 64.

Data Splits In this paper, we adopt two data splits in five datasets: originally used split and newly proposed split defined in [24]. AWA1 is originally split into 40 classes for training and 10 classes for testing [13]. AWA2 has the same split. Different from [10], original split of SUN contains 645 training classes and 72 test classes. Original split of CUB is in keeping with [1] that divides 200 classes into 150 training classes and 50 test classes. In aPY, it generally uses 20 aPascal classes as training data and 12 aYahoo classes as test data. Numbers of training and test classes in new split and original split are equal but classes are newly divided to guarantee that none of test classes in new split appears in ImageNet 1K [5].

Feature and Attribute 101-layered ResNet feature [9] is adopted to represent an image and continuous attribute is used to describe a class.

Evaluation Protocol We use average per-class top-1 accuracy as evaluation protocol so as to make a reasonable comparison with results in [24]. The accuracy is defined as follows where acc_i represents correct predictions in unseen class c_i and acc_{mean} represents average per-class accuracy of all T^u unseen classes.

$$acc_{mean} = \frac{1}{T^u} \sum_{i=1}^{T^u} acc_i \tag{6}$$



(a) (b)

Figure 4: (a) Left: Top-1 accuracy of different source class quantities. It is compared among six unseen classes in originally split dataset AWA2. Right: Per-class top-1 accuracy of different source class quantities, compared among four datasets: AWA1, AWA2, CUB and SUN. All datasets are originally split. (b) Accuracies of using three methods to select samples from source classes. Method 1: top-5 closest samples to source class center; Method 2: all samples in source class; Method 3: center of source class. In this section, the most similar class is used as source class.

4.2 Analysis

T-SNE Visualization As shown in Figure 3(a), real samples of unseen classes locate within spatial distribution of synthesized samples in feature space. Visualization of samples distribution shows that the synthesized samples can well express the characteristic of unseen samples. That is why classifiers trained on synthesized samples have good recognition ability of unseen classes.

Validation Set Seen classes are split into two parts in experiments: training classes and valid classes. According to it, we train the sample synthesis network with/without validation set separately. Results on five originally split datasets are shown in Figure 3(b). Valid classes are utilized to monitor the quality of the network when training but information of these classes are not saved in the net. As a result, if synthesis source of unseen classes are included in valid classes, the synthesized samples will have relatively poor quality. It has been proven in Figure 3(b) that results with validation are lower than results without validation. That is to say, when training, containing full and detailed information of seen classes as much as possible guarantees the quality of synthesized samples.

Quantity of Selected Source Classes Quantity of selected source classes straightforwardly affects the quality of synthesized classes. In experiments, the upper limit of source class quantity is set to 4. And Figure 4(a) shows the effects of different source class numbers, within a dataset (left) and among datasets (right). It can be clearly seen in Figure 4(a) (left) that in originally split AWA2, most classes are recognized more precisely if samples are synthesized from 2 to 3 source classes. Correspondingly, total accuracy of AWA2, the same as AWA1, is relative higher in the case of 2 to 3 source classes as shown in Figure 4(a) (right). By contrast, accuracies of other two datasets reach the maximum when merely using the most similar class as source. In conclusion, it has a great meaning to use class entropy to set correct number of source classes.

Quantity and Quality of Selected Source Samples We take several approaches to investigate the effect of source samples selection. In addition, the number of synthesized target samples is decided by the number of selected source samples, this experiment is also con-

Approaches	SUN		CUB		AWA1		AWA2		aPY	
	OS	NS	OS	NS	OS	NS	OS	NS	OS	NS
CONSE[17]	44.2	38.8	36.7	34.3	63.6	45.6	67.9	44.5	25.9	26.9
DEVISE[9]	57.5	56.5	53.2	52.0	72.9	54.2	68.6	59.7	35.4	39.8
SJE[1]	57.1	53.7	55.3	53.9	76.7	65.6	69.5	61.9	32.0	32.9
ESZSL[20]	57.3	54.5	55.1	53.9	74.7	58.2	75.6	58.6	34.4	38.3
SSE[26]	54.5	51.5	43.7	43.9	68.8	60.1	67.5	61.0	31.1	34.0
LATEM[23]	56.9	55.3	49.4	49.3	74.8	55.1	68.7	55.8	34.5	35.2
SAE[11]	42.4	40.3	33.4	33.3	80.6	53.0	80.7	54.1	8.3	8.3
SYNC[3]	59.1	56.3	54.1	55.6	72.2	54.0	71.2	46.6	39.7	23.9
CCSS	56.0	56.8	57.0	44.1	72.8	56.3	71.2	63.7	28.4	35.5

Table 1: Results Comparison: evaluated by average per-class top-1 accuracy in %. It is reported on five benchmark datasets SUN, CUB, AWA1, AWA2, aPY. 'OS' represents the original split and 'NS' represents the new split. The best is marked in red.

Target Class	Top-3 Most Similar Classes
building	bus, boat, aeroplane
monkey	person, cat, dog
centaur	person, horse, dog
bag	boat, sofa, train
wolf	cat, dog, cow
goat	cat, dog, cow

Table 2: Class similarity in originally split aPY: there exists large visual difference between target classes and their most similar source classes. Many source classes of different target classes are repetitive.

ducted to measure the effect of the number of synthesized samples. Results are shown in Figure 4(b). It shows that samples close to class center have stronger power to express the characteristic of a class. Even though a classifier is trained on little data, it still has a robust recognition ability if the samples can better characterize a class. That is to say, quality of selected samples (or synthesized samples) plays a more important role than quantity.

4.3 Benchmark Comparison

Table 1 shows the comprehensive comparison with existing state-of-the-art methods. In order to make an objective comparison, we use results based on an unified evaluation protocol, which are collected from [24]. All experiments use ResNet [8] as feature extractor, avoiding the pre-trained problem mentioned in the second section. We can see that the proposed method exceeds others in SUN of new split, CUB of original split, AWA2 of new split. Since the proposed method synthesizes rich samples which well characterize unseen classes, classifier trained on these data can achieve relatively higher accuracy. It's worth noting that SYNC performs best on 3/10 cases but compared with our proposed method, it performs worse than our method on 6/10 cases and achieves the same accuracy on 1/10 case as ours. As for SAE, it performs best on originally split AWA1 and AWA2 but has notably bad performance on the rest datasets especially on aPY. Both SAE and our method are similar to auto-encoder, due to the design of end-to-end network frame and precise training pairs selection, ours has stronger robustness than SAE. Overall, our method has a better performance than others.

Different from the methods presented in Table 1, our proposed method relies on the quality of synthesized samples which depends largely on similarity between source classes and target classes. Thus, lower class similarity results in a relatively poor performance especially on aPY. According to the relationship between source classes and target classes displayed in Table 2, two points should be highlighted about aPY: 1. it includes variety of classes from human landscape to natural biology, which has great differences among classes; 2. the most similar class is extremely different from target class visually, such as 'bag' and 'boat'. According to inter-class discrepancy in the dataset, samples which are synthesized via inter-class similarity can not express the characteristic of unseen classes accurately. That's why our method performs poorly on aPY especially in the case of original split.

Different results of AWA1 in two cases can be explained by the great similarity within unseen classes. For example, in case of new split, 'dolphin' and 'blue whale' are split into unseen classes which have quite similar visual features. Moreover, the top-2 most similar seen classes of them are totally identical: 'killer+whale' and 'humpback+whale'. To deal with it, we adjust source combination which contains no repetitive classes as synthesis sources to enhance the discrimination ability of synthesized samples.

5 Conclusion

We propose a novel method, Cross-Class Sample Synthesis (CCSS), to directly synthesize labeled samples of unseen classes. Selection strategy using inter-class similarity and class entropy is proposed to make reasonable source data selection. Moreover, an end-to-end sample synthesis network is trained to automatically build a transfer rule between feature space and attribute space, which realizes synthesis from source samples to target samples. Samples synthesized by the network primely express the characteristic of unseen classes so the task of unseen classes recognition is simplified to a supervised learning problem. We conduct experiments on five benchmark datasets where the comprehensive results effectively demonstrate the advantage of our proposed method.

6 Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 61773385, No. 61672523), the Fundamental Research Funds for the Central Universities and the Research Funds of Renmin University of China (No. 18XNLG19).

References

[1] Zeynep Akata, Scott E. Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. Evaluation of output embeddings for fine-grained image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2927–2936, 2015.

[2] Jimmy Lei Ba, Kevin Swersky, Sanja Fidler, and Ruslan Salakhutdinov. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *IEEE International Conference on Computer Vision*, pages 4247–4255, 2015.

[3] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Synthesized classifiers for zero-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5327–5336, 2016.

[4] Soravit Changpinyo, Wei-Lun Chao, and Fei Sha. Predicting visual exemplars of unseen classes for zero-shot learning. In *IEEE International Conference on Computer Vision*, pages 3496–3505, 2017.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[6] Ali Farhadi, Ian Endres, Derek Hoiem, and David A. Forsyth. Describing objects by their attributes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1785, 2009.

[7] Andrea Frome, Gregory S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems 26*, pages 2121–2129, 2013.

[8] Yuchen Guo, Guiguang Ding, Jungong Han, and Yue Gao. Synthesizing samples for zero-shot learning. In *Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 1774–1780, 2017.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[10] Dinesh Jayaraman and Kristen Grauman. Zero-shot recognition with unreliable attributes. *Neural Information Processing Systems*, pages 3464–3472, 2014.

[11] Elyor Kodirov, Tao Xiang, and Shaogang Gong. Semantic autoencoder for zero-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4447–4456, 2017.

[12] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 951–958, 2009.

[13] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, 2014.

[14] Xirong Li, Shuai Liao, Weiyu Lan, Xiaoyong Du, and Gang Yang. Zero-shot image tagging by hierarchical semantic embedding. In *SIGIR 2015: International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 879–882, 2015.

[15] Yang Long, Li Liu, Ling Shao, Fumin Shen, Guiguang Ding, and Jungong Han. From zero-shot learning to conventional supervised classification: Unseen visual data synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6165–6174, 2017.

[16] Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[17] Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S. Corrado, and Jeffrey Dean. Zero-shot learning by convex combination of semantic embeddings. In *International Conference on Learning Representations*, 2014.

[18] Mark M. Palatucci, Dean A. Pomerleau, Geoffrey E. Hinton, and Tom Mitchell. Zero-shot learning with semantic output codes. *Neural Information Processing Systems*, pages 1410–1418, 2009.

[19] Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2751–2758, 2012.

[20] Bernardino Romera-Paredes and Philip H. S. Torr. An embarrassingly simple approach to zero-shot learning. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2152–2161, 2015.

[21] Qian Song and Feng Xu. Zero-shot learning of sar target feature space with deep generative neural networks. *IEEE Geoscience and Remote Sensing Letters*, 14(12): 2245–2249, 2017.

[22] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. *Advances in Water Resources*, 2011.

[23] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh N. Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 69–77, 2016.

[24] Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning - a comprehensive evaluation of the good, the bad and the ugly. *arXiv preprint arXiv:1707.00600*, 2017.

[25] Gang Yang, Jinlu Liu, and Xirong Li. Imagination based sample construction for zero-shot learning. In *SIGIR 2018: International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 941–944, 2018.

[26] Ziming Zhang and Venkatesh Saligrama. Zero-shot learning via semantic similarity embedding. In *IEEE International Conference on Computer Vision*, pages 4166–4174, 2015.