Pattern Recognition 80 (2018) 129–142

Contents lists available at ScienceDirect

Pattern Recognition

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

Attribute-Based Synthetic Network (ABS-Net): Learning more from pseudo feature representations

Jiang Lu[a,b], Jin Li[a], Ziang Yan[a], Fenghua Mei[b], Changshui Zhang[a,*]

[a] Department of Automation, Tsinghua University, State Key Laboratory of Intelligent Technologies and Systems, Tsinghua National Laboratory for Information Science and Technology (TNList), Beijing, China
[b] China Marine Development and Research Center (CMDRC), Beijing, China

ARTICLE INFO

Article history:
Received 20 September 2017
Revised 30 November 2017
Accepted 4 March 2018
Available online 6 March 2018

Keywords:
Pseudo feature representation
Zero-shot learning
Supervised learning
Data augmentation
Attribute learning

ABSTRACT

In large-scale visual recognition tasks, researchers are usually faced with some challenging problems, such as the extreme imbalance in the number of training data between classes or the lack of annotated data for some classes. In this paper, we propose a novel neural network architecture that automatically synthesizes pseudo feature representations for the classes in lack of annotated images. With the supply of semantic attributes for classes, the proposed *Attribute-Based Synthetic Network (ABS-Net)* can be applied to zero-shot learning (ZSL) scenario and conventional supervised learning (CSL) scenario as well. For ZSL tasks, the pseudo feature representations can be viewed as annotated feature-level instances for novel concepts, which facilitates the construction of unseen class predictor. For CSL tasks, the pseudo feature representations can be viewed as products of data augmentation on training set, which enriches the interpretation capacity of CSL systems. We demonstrate the effectiveness of the proposed ABS-Net in ZSL and CSL settings on a synthetic colored MNIST dataset (C-MNIST). For several popular ZSL benchmark datasets, our architecture also shows competitive results on zero-shot recognition task, especially leading to tremendous improvement to state-of-the-art mAP on zero-shot retrieval task.

1. Introduction

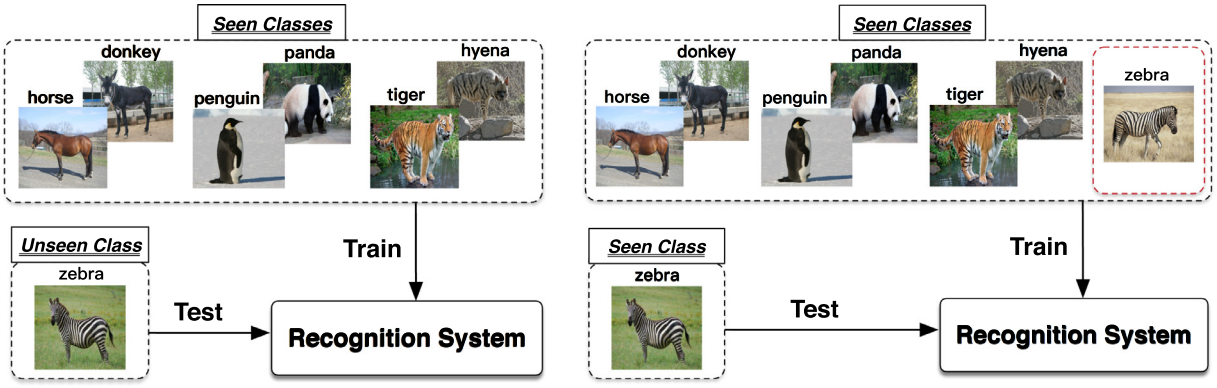
Although large scale classification based on supervised learning has achieved major successes in recent years by deep learning [1–5], the collection and annotation of huge amounts of training data for growing classes are time consuming and expensive, which consequently raises a dilemma of the lack in training data for some classes, forming the 'long-tail' phenomenon in the distribution for the number of training data between classes. The extreme imbalance in training data size between classes or the lack of annotated data for some classes are forcing us to develop more efficient learning paradigms [6–8].

As we all known that collecting semantic attributes or distinctive descriptions could be more easier than collecting massive real images for newly defined classes. As a practicable solution, zero-shot learning (ZSL) has gained growing attention recently [9–12], which aims to recognize previously unseen classes without labeled training data. For ZSL tasks, some intermediate semantic properties, such as attributes [9,13] or category hierarchies [14], are usually revealed and shared for seen and unseen classes, acting as side information by which the unseen classes could be inferred rationally. Explicitly, the dataset of seen classes is well labeled with categories and attributes tags in ZSL settings, whereas the unseen classes are faced with absence of training data but presence of their attribute descriptions. As illustrated in Fig. 1, the purpose of ZSL for visual recognition is to predict for each novel image which of unseen classes it belongs to. Due to the disjoint and unrelated classes for ZSL, the gaps between the distribution of training data (seen classes) and test data (unseen classes) are usually considered to exist [11,15,16]. In contrast, conventional supervised learning (CSL) [17] aims to predict the class labels of novel images for seen classes.

It is generally known that the process to recognize novel concepts for most of people is abstractively from individuality to generality, and then from generality to individuality. More specifically, assuming the fundamental understandings about what features represent the attributes *horselike/stripe/black and white* are obtained from some prepared images, like *horse, donkey, tiger, hyena, penguin, panda* etc, one can surmise roughly what a zebra looks like if told zebra has above attributes. Inspired by humans' behaviors of recognizing a novelty, in this paper we present a novel

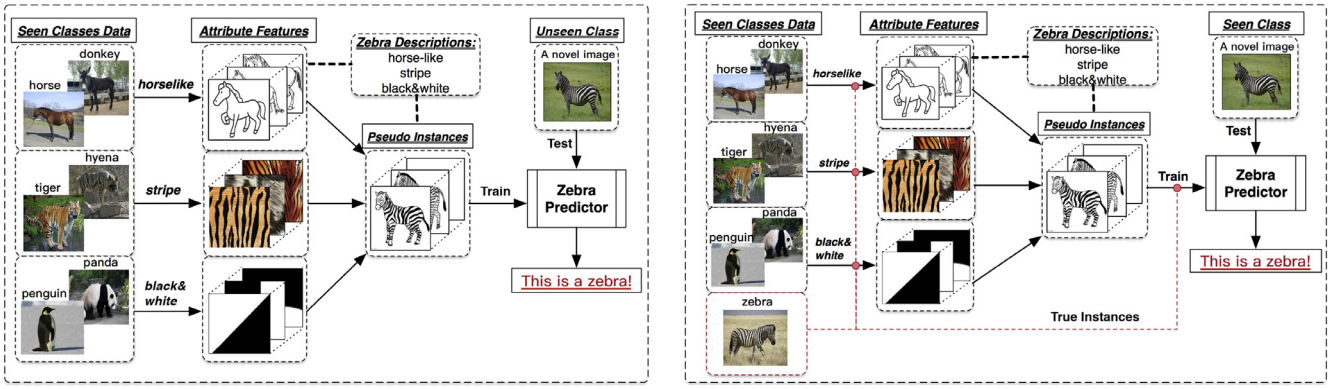
* Corresponding author.
E-mail addresses: lu-j13@mails.tsinghua.edu.cn (J. Lu), lijin14@mails.tsinghua.edu.cn (J. Li), yza15@mails.tsinghua.edu.cn (Z. Yan), mei_fh@21cn.com (F. Mei), zcs@mail.tsinghua.edu.cn (C. Zhang).

https://doi.org/10.1016/j.patcog.2018.03.006
0031-3203/



(a) Scenario of zero-shot learning. (b) Scenario of conventional supervised learning.

Fig. 1. Comparison of zero-shot learning and conventional supervised learning.



(a) Illustration of our proposed method for ZSL task. (b) Illustration of our proposed method for CSL task.

Fig. 2. Illustration of our proposed method. We capture some understandings from prepared dataset about what features represent *horselike/stripe/black and white* respectively. These attribute features associated with key descriptions of zebra will be combined into lots of synthetic pseudo instances about zebras' appearance, which can be used for the training of zebra predictor.

neural network architecture that automatically synthesize pseudo feature representations for the classes in lack of annotated images, called *Attribute-Based Synthetic Network (ABS-Net)*. Given the semantic attribute descriptions, our ABS-Net can be applied to ZSL tasks and CSL tasks as well. As illustrated in Fig. 2, our intuition is to firstly learn some credible feature representations for each attribute by utilizing prepared dataset of seen classes, and then summarize these attribute representations into an combined representation, according to the specified attribute descriptions of each unseen class (ZSL) or seen class (CSL). Finally, the combined representation can be viewed as a so-called pseudo feature-level instance of the unseen class (ZSL) or seen class (CSL), which will offer valuable guidance for the training of overall recognition system.

Leveraging the Convolutional Neural Network (CNN) based image features [3], we firstly train a Joint Attribute Feature Extractor (JAFE) in which each fundamental unit is put in charge of the extraction of one attribute feature. Regardless of labels of seen classes, we extract all possible feature vectors for every attribute tag by JAFE, then store these attribute features into a Fragment Repository based on a confidence factor filter. According to the attribute descriptions of one specified class (unseen class for ZSL and seen class for CSL), a probability based sampling strategy is exploited to select some attribute features from Fragment Repository to synthesize combined vectors. This strategy allows us access to lots of synthetic feature vectors for the specified class, called pseudo feature representations, which fills the gaps between training domain and test domain for ZSL and achieves data augmentation in feature level as well for CSL from another perspective. Taking these pseudo feature representations as inputs, a multi-way predictor for some specified classes can be learned. During test time, a novel image goes through the JAFE to generate its combined feature vector over all attributes, followed by above well-trained predictor to perform the ultimate inference. Experimental results on a synthetic colored MNIST dataset (C-MNIST) demonstrate the effectiveness of our ABS-Net in ZSL and CSL settings. Furthermore, its performance on several ZSL benchmark datasets show improvements to state-of-the-art results, especially for zero-shot retrieval mAP (i.e. mean average precision).

We conclude our contributions as follows. First, we develop an concise architecture ABS-Net to deal with ZSL tasks, which fills the gaps between seen and unseen concepts and achieves competitive results on several challenging ZSL benchmark datasets. Second, our ABS-Net realizes inherently the data augmentation in feature level, which can be easily extended to CSL scenario.

2. Related work

2.1. Zero-shot learning

Some ZSL methods are based on a two-stage recognition: attribute prediction and classification inference. [18–20] regarded each unseen class as a binary vector, i.e. signature, where each entry delegates the presence or absence of one attribute. For a

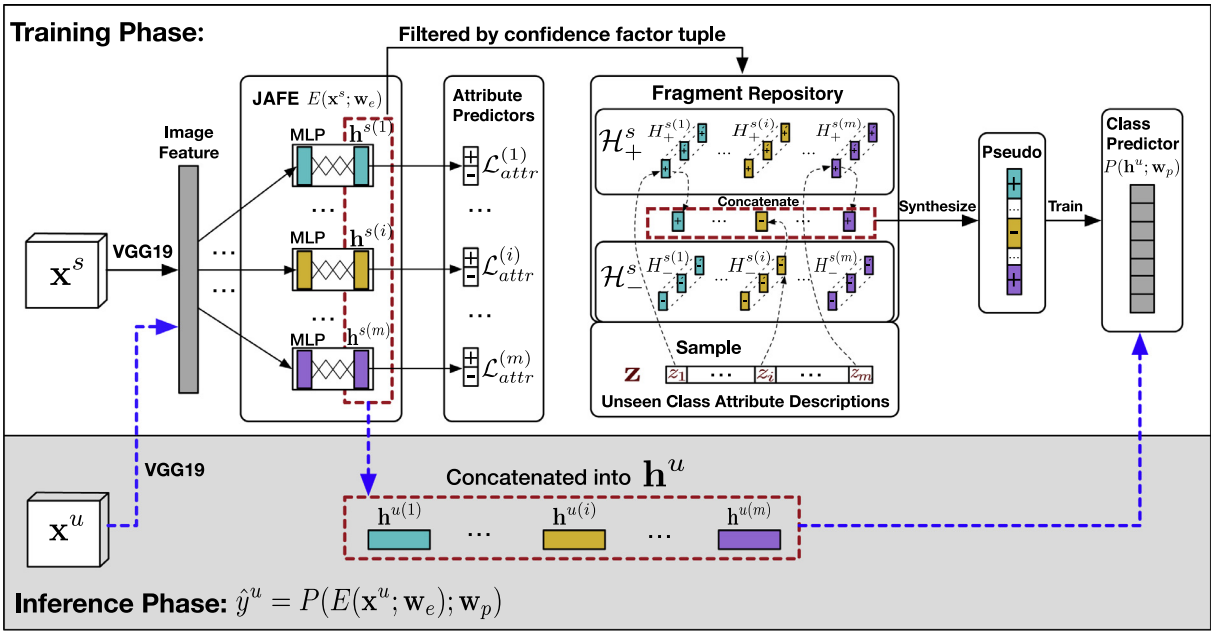


Fig. 3. Framework of our ABS-Net. The bars in different colors denote respectively different attribute features. The bars with "+" are sufficiently positive feature vectors on the presence of its corresponding attribute, and it is opposite for the ones with "−". The JAFE, attribute predictors and class predictor need to be trained from scarcity. The Fragment Repository can be constructed based on the well-trained JAFE and attribute predictors. Pseudo representations of unseen classes, which guide the formation of class predictor, are generated via a probability based sampling. Best viewed in color.

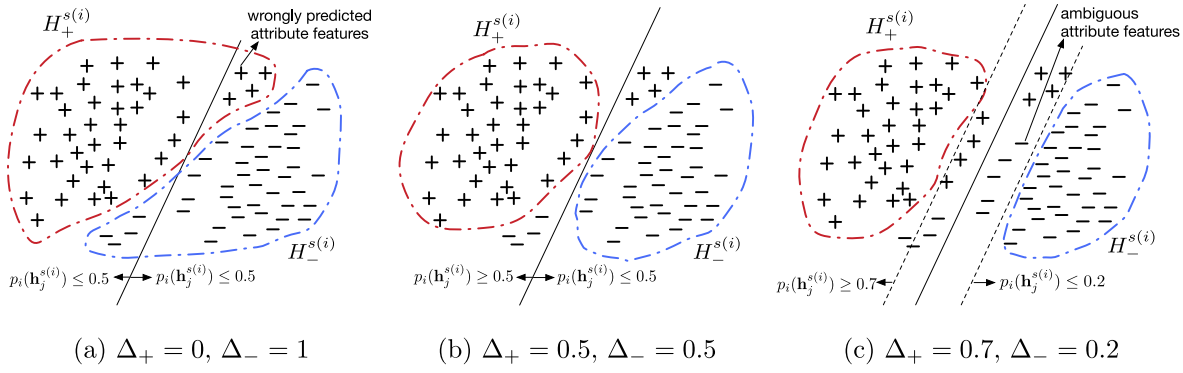


Fig. 4. Illustration of Fragment Repository for the ith attribute under three different setttings of Δ_+ and Δ_-. The point notated as " + " represents the attribute feature $\mathbf{h}_j^{s(i)}$ with positive attribute tag, and the point notated as " − " represents the attribute feature $\mathbf{h}_j^{s(i)}$ with negative attribute tag.

new image, its attributes are first predicted, and then it will be mapped to an unseen class whose signature is most similar to the predicted attributes. Explicitly, [20] proposed a probabilistic Directed Attribute Prediction (DAP) framework where attribute probabilistic classifiers are learned firstly and then a MAP step is performed for seeking the most promising unseen class. [21] proposed an Author-Topic model to describe the attribute-specified distributions of image features. [22] constructed a Bayesian Network (BN) based unified model to capture the object-dependent and object-independent attribute relationships. [23] proposed a weighted version of DAP based on observation probability of attributes. [24] developed a max-margin multi-label classification formulation (M3L) for attribute prediction.

Other ZSL methods are based on view of embedding. Their key idea is to encode both images and labels into a common space and then learn a discriminative compatibility/similarity function, where image embedding can be feature vector or other related representations, and label embedding can be particular coding, e.g. attributes, or available text corpus, e.g. English Wikipedia. [10,25] utilized label embedding to construct a common space where the compatibility between images and labels can be measured. [26–28] leveraged CNN based features [2,4] as image embedding to learn compatibility function. [15] built better embedding by alleviating domain shift problem which was considered to exist between source domain (seen classes) and target domain (unseen classes). [16] embedded source and target domain data into semantic space, i.e. mixture proportions of seen classes. [29] learned a collection of linear models to construct overall nonlinear latent embedding models. [11] presented a general probabilistic model embedding both domains to a joint latent space. [12] suggested to better control semantic embedding of images by metric learning [30–32]. Such works essentially translated zero-shot tasks into sub-task associations of semantic embedding and similarity measurement, exhibiting excessive reliance on the capability of common embedding spaces.

Moreover, few related approaches follow different perspectives, such as semantic transfer [33], co-occurrence statistics of visual concepts [34], random forest [35] and semantic manifold structure [36]. Different from these aforementioned methods, our ABS-Net

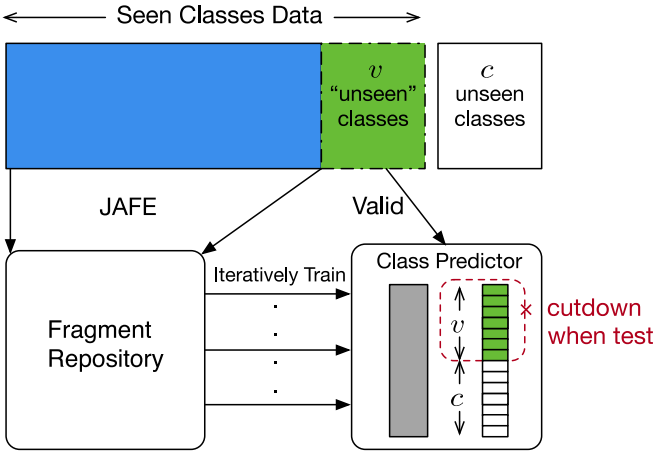


Fig. 5. Illustration for training and validation strategy for class predictor. Best viewed in color.

proposes to synthesize pseudo feature representations to enrich our comprehension for novelties.

2.2. Conventional supervised learning

In order to obtain robust generalization ability of deep model, many visual recognition tasks of CSL scenario adopted various forms of data augmentation, such as generating image translations and horizontal reflections [2,5], altering the intensities of the RGB channels [2,5] and scale jittering [3,4,18] etc. Different from these operations on original images, our method actually realizes the feature augmentation for the classes in lack of training data, which is called feature-level data augmentation by us.

3. Methodology

3.1. Overview

Formulation. Suppose we represent each image by a d-dimensional visual feature vector $\mathbf{x}$ and represent each attribute description by an m-dimensional binary vector $\mathbf{a}$ where m is the number of attributes. Let c represents the number of unseen classes, and let T represents the label set of unseen classes, i.e. $|T| = c$. Moreover, let $\mathbf{S} = \left\{(\mathbf{x}_i^s, y_i^s, \mathbf{a}_i)\right\}_{i=1}^{N_s}$ represents the labeled dataset of seen classes with N_s images, and $\mathbf{U} = \{(\mathbf{x}_i^u, y_i^u)\}_{i=1}^{N_u}$ represents the unlabeled dataset of unseen classes with N_u images, where y denotes the class label. Besides, the prepared class-level attribute descriptions for all unseen classes are preprocessed into the probabilistic form where its values represent the possibilities of presence for attributes, which are denoted by $\mathbf{Z} = \left\{\mathbf{z}^t, t \in T\right\}$, where $\mathbf{z}^t \in \mathbb{R}^m$ denotes the attribute descriptions of unseen class t. In this way our goal for ZSL is to utilize the prepared $\mathbf{S}$ and $\mathbf{Z}$ to predict y^u for unseen class image $\mathbf{x}^u$. Concerning on our method, let $E(\mathbf{x}; \mathbf{w}_e)$ be a function parameterized by $\mathbf{w}_e$ which maps $\mathbf{x}$ to a combined hidden representation $\mathbf{h}$ concatenated by m sub-vectors $\mathbf{h}^{(i)}, i = 1, \ldots, m$, where $\mathbf{h}^{(i)}$ denote the ith sub-vector of $\mathbf{h}$. For clarity, we denote by $\mathbf{h}^s$ the combined hidden representation of seen classes, and $\mathbf{h}^u$ the pseudo hidden representation of unseen classes. Thus $\mathbf{h}^{s(i)}$ and $\mathbf{h}^{u(i)}$ denote respectively the ith sub-vector of $\mathbf{h}^s$ and $\mathbf{h}^u$. Let $P(\mathbf{h}; \mathbf{w}_p)$ be a classification function parameterized by $\mathbf{w}_p$ which maps a combined hidden representation $\mathbf{h}$ to the unseen class prediction $\hat{y}$. It's remarkable that function E is learned via dataset $\mathbf{S}$ but P via synthetic pseudo feature representations of unseen classes.

Table 1
Configuration of the shared encoder. No padding after Conv. operation and all strides are (1, 1).

Units	Size or Value
Input	(3, 28, 28)
Conv.1+Relu	(32, 3, 3)
Conv.2+Relu	(32, 3, 3)
Pooling	(2, 2)
Dropout	$p = .25$
Flatten	–

Like previous works [11,12,16], we utilize pre-trained CNN model [3] to obtain image features. Our ABS-Net is depicted in Fig. 3. The JAFE, i.e. $E(\mathbf{x}; \mathbf{w}_e)$, aims to extract all attribute features and then form a combined feature representation. The class predictor, i.e. $P(\mathbf{h}; \mathbf{w}_p)$, is adjustable with specified tasks, such as prediction on seen classes (i.e. CSL) instead of unseen classes. Based on dataset $\mathbf{S}$, we adopt a joint training strategy for JAFE to reduce time cost. Due to absence of training images for unseen classes, we employ a confidence factor based filter to construct the Fragment Repository for all attributes. Then, we regard attribute descriptions of unseen classes as sampling probabilities to generate pseudo feature representations from Fragment Repository, which allows us access to the training of unseen class predictor.

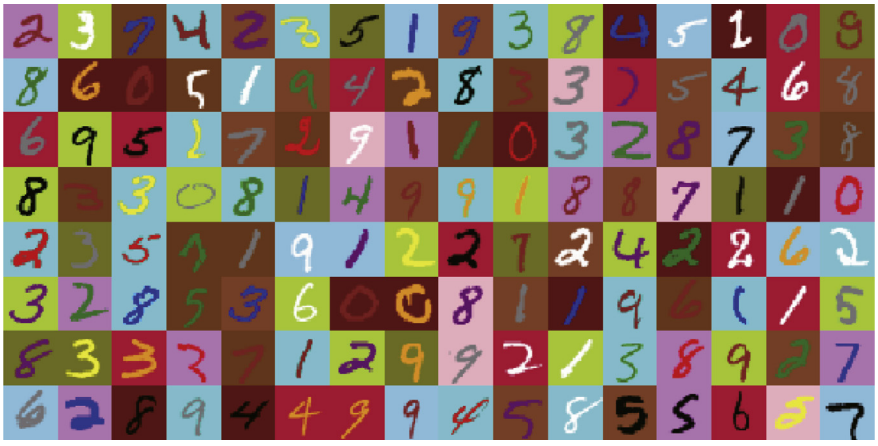
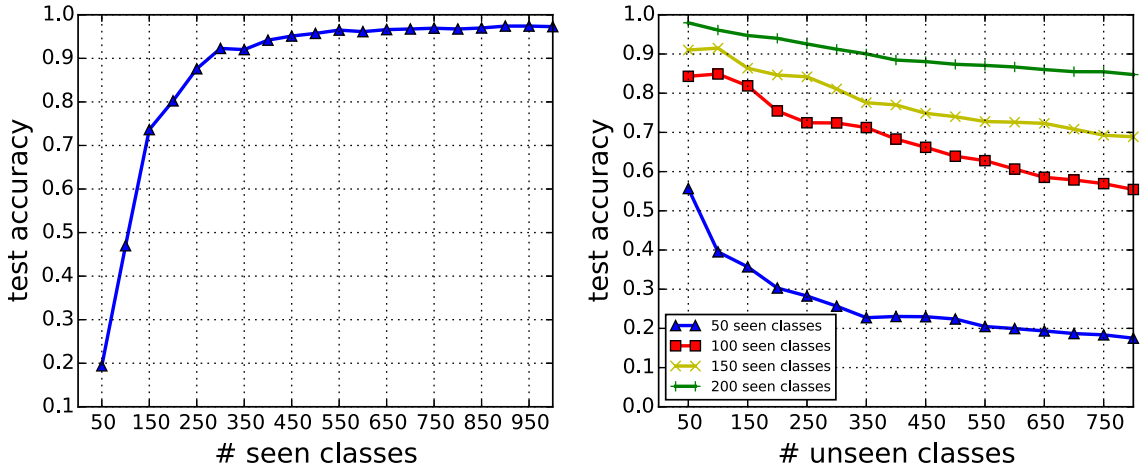


Fig. 6. Some examples of C-MNIST. The images from same class own the same digit, b-color and f-color. The size per class is almost 70k/1k=70. Best viewed in color.



(a) Prediction on all 1k classes. (b) Prediction on unseen classes

Fig. 7. Zero-shot recognition on C-MNIST by ABS-Net.

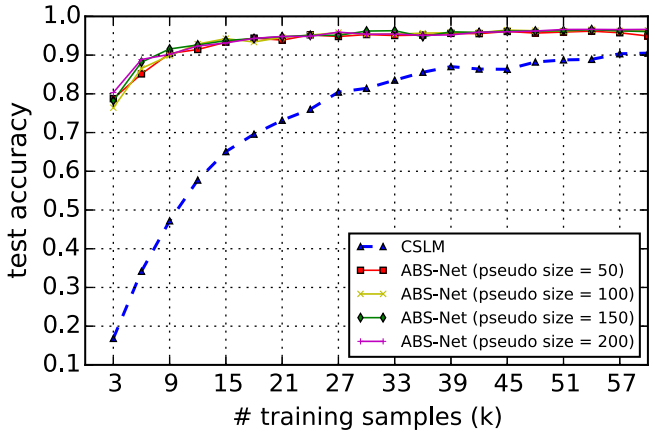


Fig. 8. Fully supervised recognition comparison between our proposed ABS-Net and conventional supervised learning model (CSLM) on C-MNIST.

3.2. Joint attribute feature extractor (JAFE)

In this section we detail our JAFE framework. Given the labeled dataset **S**, we can train our JAFE regardless of their class-level labels. In our paper, we construct our JAFE by single-layer perceptron (SLP) or multi-layer perceptron (MLP) group, which of course can be replaced by other feature extraction model. Explicitly, each unit in JAFE is a SLP or MLP, assigned to deal with the extraction of one specified attribute feature, i.e. $\mathbf{h}^{(i)}$ as notated in Section 3.1. Then the combined feature representation is defined as follows:

$$E(\mathbf{x};\mathbf{w}_e)=\mathbf{h}=\mathcal{C}(\mathbf{h}^{(1)},\ldots,\mathbf{h}^{(m)}), \tag{1}$$

where $\mathcal{C}$ denotes concatenation operation on sub-vectors. Of course, we can set different feature dimensions for different attributes, but here we let all $\mathbf{h}^{(i)}$, $i=1,\ldots,m$, are l-dimensional vectors for simplicity, thus $\mathbf{h}\in\mathbb{R}^{ml}$.

Given one image of seen class $\mathbf{x}^s$, we firstly process its attribute description **a** into a binary vector based on the threshold which was obtained by an simple average operation on all given continuous attribute tags for one class. Then we utilize the Logistic Regression to predict the presence or absence of one attribute. We represent the probability that $\mathbf{x}^s$ is positive about the ith attribute by $p_i(\mathbf{h}^{s(i)})$, which can be expressed as follows:

$$p_i(\mathbf{h}^{s(i)})=\frac{1}{1+\exp(-\boldsymbol{\theta}^{(i)\mathrm{T}}\mathbf{h}^{s(i)})}, \tag{2}$$

where $\boldsymbol{\theta}^{(i)}$ is the weight of the Logistic Regression module connected with the ith unit of JAFE. We minimize the negative binomial cross-entropy for the ith attribute prediction:

$$\mathcal{L}_{attr}^{(i)}=a^{(i)}\log p_i(\mathbf{h}^{s(i)})+(1-a^{(i)})\log(1-p_i(\mathbf{h}^{s(i)})), \tag{3}$$

where $a^{(i)}\in\{0,1\}$ is ground truth label of the i-th attribute tag in **a**. Since $\mathbf{h}^{s(i)}$ is obtained from $E(\mathbf{x}^s;\mathbf{w}_e)$, as shown in Eq. (1), the parameter $\mathbf{w}_e$ can be optimized by the following joint loss function:

$$\mathcal{L}_{joint}=\sum_{i=1}^{m}\alpha_i\mathcal{L}_{attr}^{(i)}, \tag{4}$$

where α_i, $i=1,\ldots,m$, achieve a weight assignment between all attribute predictions. Especially, the training loss for JAFE and attribute predictors is formulated in one-sample case, as shown in Eqs. (3) and (4), but it can be extended to a mini-batch version just by an average operation, which can be optimized by gradient back-propagation algorithm.

3.3. Fragment repository of attribute features

Just as humans often summarize the characteristics of one certain attribute from some different things, our approach also follow this inspiration to build a Fragment Repository for some credible attribute-level feature vectors, which lays the foundation for subsequent pseudo feature representations generation.

Note that the construction of Fragment Repository is implemented after the training phase of JAFE and attribute predictor. For one image of seen class $\mathbf{x}^s$ and its binary attribute vector **a**, the well-trained JAFE and attribute predictors lead to attribute features $\{\mathbf{h}^{s(i)}\}_{i=1}^m$ associated with their corresponding positive estimations $\{p_i(\mathbf{h}^{s(i)})\}_{i=1}^m$ as Eqs. (1) and (2). These attribute features $\{\mathbf{h}^{s(i)}\}_{i=1}^m$ represent the captured understandings about attributes from image $\mathbf{x}^s$, and its positive estimations $\{p_i(\mathbf{h}^{s(i)})\}_{i=1}^m$ represent the confidence level that the corresponding attributes exist in image $\mathbf{x}^s$. For the ith attribute, according to its binary tag, we can divide its Fragment Repository into positive subset $H_+^{s(i)}$ and negative subset $H_-^{s(i)}$, corresponding to its presence and absence in images

Table 2
Zero-shot recognition accuracy (%) on C-MNIST by ABS-Net. Test on all classes.

# seen classes	50	100	150	200	250	300	350	400	450	500
test acc	19.41	46.98	73.64	80.24	87.61	92.28	92.01	94.18	95.12	95.73
# seen classes	550	600	650	700	750	800	850	900	950	1000
test acc	96.49	96.13	96.60	96.74	96.92	96.74	96.97	**97.41**	97.40	97.28

Table 3
Zero-shot recognition accuracy (%) on C-MNIST by ABS-Net. Test on unseen classes.

# unseen classes	50	100	150	200	250	300	350	400
# seen classes=50	**55.68**	39.52	35.70	30.30	28.26	25.67	22.73	23.04
# seen classes=100	84.31	**84.90**	81.87	75.47	72.46	72.41	71.24	68.30
# seen classes=150	91.02	**91.50**	86.38	84.64	84.18	81.08	77.57	77.01
# seen classes=200	**97.96**	96.11	94.72	94.00	92.56	91.22	90.03	88.45
# unseen classes	450	500	550	600	650	700	750	800
# seen classes=50	22.98	22.39	20.48	19.95	19.35	18.68	18.35	17.50
# seen classes=100	66.20	63.92	62.79	60.65	58.54	57.87	56.93	55.44
# seen classes=150	74.87	74.01	72.78	72.60	72.30	70.83	69.29	68.87
# seen classes=200	88.06	87.37	87.09	86.69	86.03	85.49	85.47	84.75

Table 4
Fully supervised recognition results on C-MNIST by CSLM and ABS-Net.

training samples (k)		3	6	9	12	15	18	21	24	27	30
	CSLM	16.84	34.17	47.20	57.73	65.08	69.58	73.17	76.02	80.45	81.42
ABS-Net	pseudo size=50	78.76	85.14	90.41	91.45	93.34	94.39	93.87	95.22	94.80	95.30
	pseudo size=100	76.42	86.56	89.90	92.86	94.25	93.39	94.59	95.16	95.83	95.41
	pseudo size=150	78.26	88.17	91.58	92.63	93.75	94.28	94.76	95.11	95.11	96.21
	pseudo size=200	80.32	88.87	90.19	92.34	93.14	94.32	94.82	94.84	95.93	95.38
training samples (k)		33	36	39	42	45	48	51	54	57	60
	CSLM	83.53	85.57	87.03	86.40	86.33	88.21	88.76	88.90	90.35	90.51
ABS-Net	pseudo size=50	95.06	95.18	95.61	95.54	96.10	95.69	95.95	**96.18**	95.77	94.93
	pseudo size=100	95.39	95.68	95.79	95.81	96.46	96.29	96.21	**96.54**	96.24	96.42
	pseudo size=150	96.30	94.78	96.01	95.85	96.22	96.21	96.31	**96.56**	96.26	96.06
	pseudo size=200	95.45	95.10	95.22	95.91	96.01	96.22	96.62	**96.64**	96.59	**96.64**

respectively, which can be expressed as follows:

$$H_+^{s(i)} = \{\mathbf{h}_j^{s(i)} | a_j^{(i)} = 1, j = 1, \ldots, N_s\}, \tag{5}$$

$$H_-^{s(i)} = \{\mathbf{h}_j^{s(i)} | a_j^{(i)} = 0, j = 1, \ldots, N_s\}, \tag{6}$$

where $\mathbf{h}_j^{s(i)}$ is the ith attribute feature of the j th image in dataset $\mathbf{S}$ extracted by JAFE, and $a_j^{(i)}$ is ground truth label of the ith attribute tag of the jth image. In this way, we can construct the total Fragment Repository for all attributes, which are denoted by $\mathcal{H}_+^s = \{H_+^{s(i)}\}_{i=1}^m$ and $\mathcal{H}_-^s = \{H_-^{s(i)}\}_{i=1}^m$.

Intuitively, the positive subset $H_+^{s(i)}$ should cover all positive vectors and negative subset $H_-^{s(i)}$ should cover all negative vectors for the ith attribute, as shown in Eqs. (5) and (6). However, the JAFE and attribute predictors can not predict all attribute tags correctly, and these wrongly predicted attribute features would make no sense for enriching our understandings about attributes. Therefore, we construct $H_+^{s(i)}$ and $H_-^{s(i)}$ according to the true label of attribute features as well as its positive estimation $p_i(\mathbf{h}^{s(i)})$. Furthermore, our intuition tells us that the ambiguous attribute features may not be applicable for synthesizing pseudo combined representations, since their uncertainty will affect our judgments. In this way we introduce a confidence factor tuple (Δ_+, Δ_-) to filter out the ambiguous attribute features and guarantee the truth-reliability of understanding about attributes, where $\Delta_+, \Delta_- \in [0, 1]$. Explicitly, the Fragment Repository for the ith attribute is reconstructed as follows:

$$H_+^{s(i)} = \{\mathbf{h}_j^{s(i)} | a_j^{(i)} = 1, p_i(\mathbf{h}_j^{s(i)}) \geq \Delta_+, j = 1, \ldots, N_s\}, \tag{7}$$

$$H_-^{s(i)} = \{\mathbf{h}_j^{s(i)} | a_j^{(i)} = 0, p_i(\mathbf{h}_j^{s(i)}) \leq \Delta_-, j = 1, \ldots, N_s\}. \tag{8}$$

For example, if we set $\Delta_+ = 0.7$ and $\Delta_- = 0.2$, the above operation means that for ith attribute we just take into account the positive vectors whose positive probability exceeds 0.7 and the negative vectors whose negative probability exceeds 0.8 (as illustrated in Fig. 4(c)).

In fact, if we set $\Delta_+ = 0$ and $\Delta_- = 1$, Eqs. (7) and (8) degenerate into Eqs. (5) and (6) (as illustrated in Fig. 4(a)). If we set $\Delta_+ = 0.5$ and $\Delta_- = 0.5$, all elements in Fragment Repository are predicted correctly by JAFE and attribute predictors (as illustrated in Fig. 4(b)). When $\Delta_+ > 0.5$ and $\Delta_- < 0.5$, some ambiguous attribute features near decision boundary are discarded for Fragment Repository (as illustrated in Fig. 4(c)).

3.4. Synthesizing pseudo feature representations

Back to the example of zebra (Fig. 2), suppose that all relevant attributes of zebra are *horselike/stripe/black and white*, the combined feature vector for zebra can be expressed as below:

$$\mathbf{h}_{zebra}^u = \mathcal{C}\left(\mathbf{h}_{zebra}^{u(horselike)}, \mathbf{h}_{zebra}^{u(stripe)}, \mathbf{h}_{zebra}^{u(black\&white)}\right), \tag{9}$$

where $\mathbf{h}_{zebra}^{u(horselike)}$, $\mathbf{h}_{zebra}^{u(stripe)}$ and $\mathbf{h}_{zebra}^{u(black\&white)}$ represent respectively the attribute features of *horselike/stripe/black and white* for unseen class zebra. But for ZSL tasks we have no access to these true attribute features for zebra. Therefore we select $\mathbf{h}_{horse}^{s(horselike)}$, $\mathbf{h}_{donkey}^{s(horselike)}$ from the constructed Fragment Repository to replace $\mathbf{h}_{zebra}^{u(horselike)}$ (similar for $\mathbf{h}_{zebra}^{u(stripe)}$ and $\mathbf{h}_{zebra}^{u(black\&white)}$). In this way we can get some pseudo feature representations, such as:

$$\mathbf{h}_{zebra}^u \approx \mathcal{C}\left(\mathbf{h}_{horse}^{s(horselike)}, \mathbf{h}_{tiger}^{s(stripe)}, \mathbf{h}_{penguin}^{s(black\&white)}\right), \tag{10}$$

Table 5
Statistics of the three benchmark datasets.

Dateset	# Images	# Attributes	# Seen/Unseen classes
aP&Y	15,339	64 (image-level)	20 / 12
CUB-200-2011	11,788	312 (image-level)	150 / 50
SUN Attribute	14,340	102 (image-level)	707 / 10

$$\mathbf{h}^u_{zebra} \approx \mathcal{C}\left(\mathbf{h}^{s(horselike)}_{donkey}, \mathbf{h}^{s(stripe)}_{hyena}, \mathbf{h}^{s(black\&white)}_{panda}\right). \tag{11}$$

These pseudo feature representations can be regarded as annotated feature-level instances for zebra, which facilitates the construction of zebra predictor without any zebra images.

In fact, the attribute descriptions for one object are ususlly nondeterministic but probabilistic. As described in Section 3.1, the class-level attribute description $\mathbf{Z} = \{\mathbf{z}^t, t \in T\}$ acts as side information of our ABS-Net to provide direction for synthesizing pseudo representations, where each entry has been preprocessed into a probabilistic form by averaging all provided descriptions from multiple people. In details, the certain entry z^t_i of $\mathbf{z}^t$ in $\mathbf{Z}$ is viewed as the probability of sampling an attribute feature from $H^{s(i)}_+$ instead $H^{s(i)}_-$ with respect to the i th attribute for unseen class t, where $t \in T$, which naturally suggests the synthesizing algorithm in Algorithm 1. Consequently, for arbitrary unseen class $t \in T$, the corresponding pseudo feature-level training dataset H^t can be obtained, which also constitutes the complete pseudo training dataset $\mathcal{H}^u = \{H^t, t \in T\}$ with supervision information.

Note that the above analysis or algorithm can be extended to CSL scenario without any modifications except for changing the unseen classes into some seen classes.

3.5. Inference

On the basis of pseudo feature-level training dataset $\mathcal{H}^u = \{H^t, t \in T\}$, we can optionally choose one suitable classifier as our unseen class predictor, namely $P(\mathbf{h}^u; \mathbf{w}_p)$ notated in Section 3.1. In this paper we use a c-way Softmax classifier to make class prediction :

$$\hat{y}^u = P(\mathbf{h}^u; \mathbf{w}_p) = \arg\max_{t \in T} Softmax(\mathbf{h}^u; \mathbf{w}_p) = \arg\max_{t \in T} \frac{e^{\mathbf{w}^T_{pt}\mathbf{h}^u}}{\sum_{k \in T} e^{\mathbf{w}^T_{pk}\mathbf{h}^u}}, \tag{12}$$

where $\mathbf{w}_p = \{\mathbf{w}_{pk}, k \in T\}$ is the weights of this Softmax classifier.

In this way, the inference process can be summarized as follows: one test image is firstly embedded into a 4096-dimentional feature vector $\mathbf{x}^u$ by CNN, and then go through the well-trained JAFE E to form a combined feature representation by concatenating all individual attribute features, as shown in Eq. (1). Finally, this combined feature representation is sent to class predictor P to make inference, which is formulated as follows:

$$\hat{y}^u = P(E(\mathbf{x}^u; \mathbf{w}_e); \mathbf{w}_p). \tag{13}$$

For our ABS-Net model, the part that needs to be trained consists of three modules, i.e. JAFE, attribute predictor and class predictor. As emphasized in Section 3.1, the JAFE and attribute predictor are designed to construct the Fragment Repository of attribute features, which is the basis for generating pseudo feature representations and constructing class predictor. Therefore, a two-step training strategy is adopted for the optimization of our ABS-Net. Firstly, the JAFE and attribute predictor are jointly trained with all supervised data from seen classes. Then the class predictor is trained upon the Fragment Repository generated from the well-trained JAFE and attribute predictor.

3.6. Training and validation

The joint training process of JAFE and attribute predictor can be advanced by optimizing the loss function as shown is Eq. (4). But for class predictor, its training strategy is somewhat complex. Due to absence of training images for unseen classes, as depicted in Fig. 5, we use some seen classes as our validation dataset. More clearly, we add randomly v seen classes into the set of "unseen" classes, then our class predictor is to deal with a ZSL task with $c + v$ unseen classes, whose training process can be supervised by its performance on the v seen classes. Once well trained, the weights of predictor for original c unseen classes will be extracted alone to construct a cutdown predictor just upon the c unseen classes.

In summary, once the JAFE and attribute predictor are well trained, the Fragment Repository will be constructed based on confidence factor tuple and then fixed as a repository where a large number of attribute feature vectors are deposited. Based on the repository, many combinations of attribute feature vectors can be synthesized, which are all regarded as pseudo feature representations for unseen classes. In order to search for best performance on validate classes, we iteratively perform Algorithm 1 to generate the new pseudo training dataset $\mathcal{H}^u$ for unseen classes from the fixed Fragment Repository. Explicitly, we use the current $\mathcal{H}^u$ to optimize $\mathbf{w}_p$ for few epochs and then generate a new $\mathcal{H}^u$ by Algorithm 1 to play a repeat until the occurrence of the best validation performance, which intends to avoid overfitting on one random pseudo dataset $\mathcal{H}^u$.

4. Experimental results and analysis

We test our method on a synthetic colored MNIST dataset named C-MNIST and other three benchmark image datasets for ZSL recognition, i.e. aPascal & aYahoo (aP&Y) [18], Caltech-UCSD Birds-200-2011 (CUB-200-2011) [37] and SUN attribute database [38]. Our codes[1] are written in Python and the neural network model are implemented with Tensorflow [39] embedded into Keras. We run our codes on a NVIDIA TITAN X (Pascal) GPU.

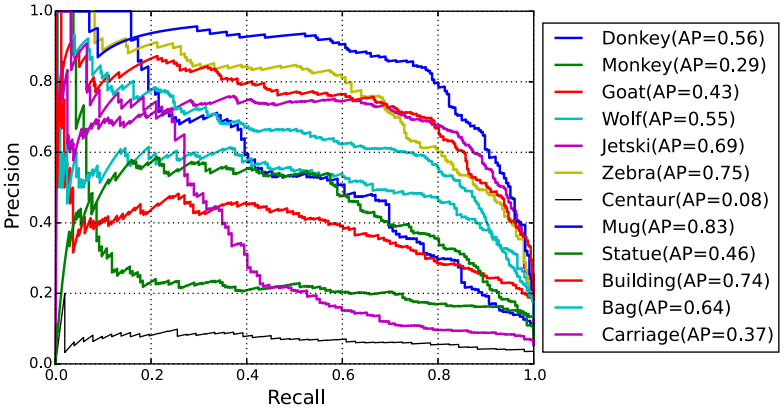
4.1. Colored MNIST dataset (C-MNIST)

In order to verify the validity and extensibility of our ABS-Net model, we build a task-specific toy dataset based on MNIST. In details, for original gray images of MNIST, we add randomly 10 different colors into their backgrounds and other 10 different colors into their strokes (also called foregrounds), thus resulting in a new C-MNIST which consists of 70k colored RGB digital images with resolution of 28×28 (60k for training and 10k for testing) from 1k possible combinations (10 digits $\times$ 10 b-colors $\times$ 10 f-colors). Some examples from C-MNIST are shown in Fig. 6. These 1k combinations are regarded as 1k different classes to perform experiments.

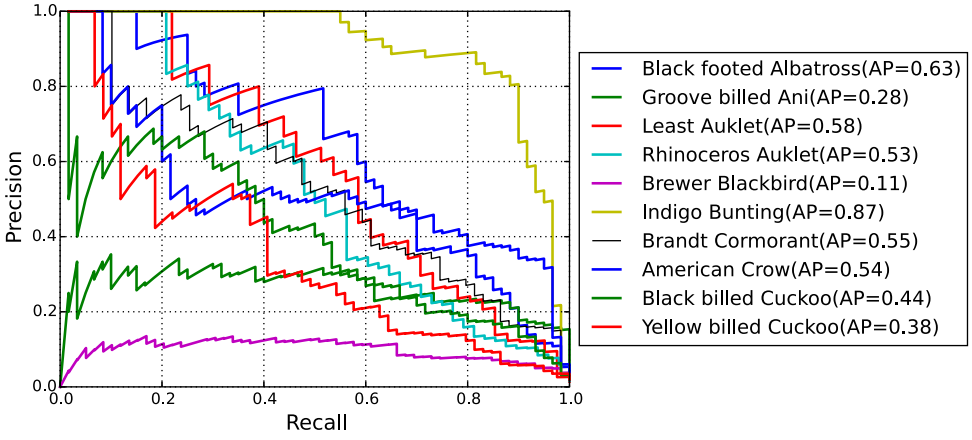
Based on C-MNIST, two types of tasks will be discussed in this section, i.e. zero-shot recognition and fully supervised recognition.

Experimental settings. We use a simple CNN architecture as shared encoder whose detailed configuration is shown in Table 1. We use SLP as JAFE unit where the number of neurons is 32 and the nonlinear functions are *tanh*. Thus we have 3 SLP units for the whole JAFE, which are assigned to deal with digits, b-colors and f-colors, respectively. Unlike the binary selection for attributes as discussed in Section 3.2, we here use three 10-way Softmax classifiers as our attribute predictors for three attributes. Some trials indicate that the acquisition for understandings about digit is more difficult

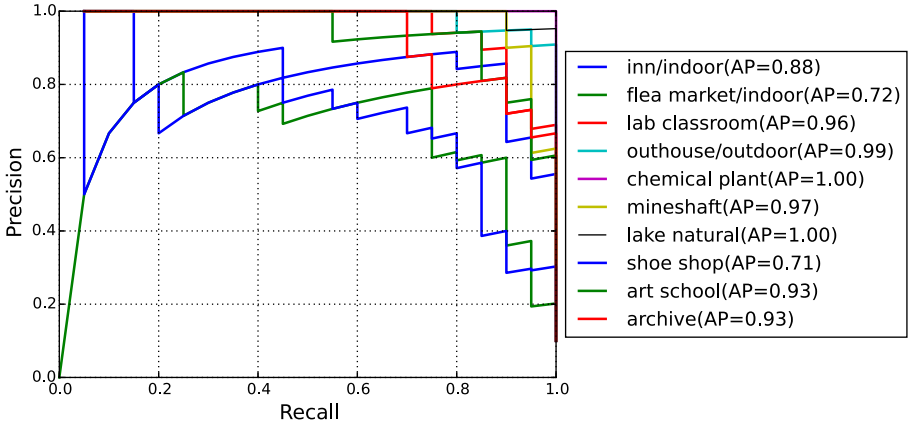
[1] Our source codes will soon be released on https://github.com/LuJiangTHU.



(a) aP&Y



(b) CUB-200-2011



(c) SUN Attribute

Fig. 9. Precision-Recall curves for unseen classes in three benchmark datasets. For CUB-200-2011, we show the first 10 classes from 50 test classes. Best viewed in color.

than b-color and f-color, hence we let $\alpha_d = 1$ but $\alpha_b = \alpha_f = 0.1$ in Eq. (4). For simplicity, we construct our Fragment Repository without consideration about confidence margins, but instead we just select attribute feature into our Fragment Repository which has a highest score for its attribute prediction. The whole model is optimized by Adam [40] with mini-batch size 32.

Zero-shot recognition. For zero-shot recognition on C-MNIST, two groups of experiments are designed to certify the capacity of ABS-Net on ZSL: (1) We increase incrementally the number of seen classes from 50 to 1000 to recognize test images containing all classes; (2) We randomly select respectively 50, 100, 150 and 200 classes as seen classes to make predictions on the rest unseen classes whose number are increased incrementally from 50 to 800. For both settings, we train our JAFE for 10 epochs and then play 5

Table 6
Zero-shot recognition accuracy (%) comparison (mean ± std) on aP&Y, CUB-200-2011, and SUN Attribute. Here we list most popular ZSL methods with their performances, which are cited from the original papers [11,12,16].

Feature	Method	aP&Y	CUB-200-2011	SUN attribute
Non-Deep	Farhadi et al. [18]	32.50	–	–
	Mahajan et al. [41]	37.93	–	–
	Wang and Ji [22]	45.05	–	–
	Akata et al. [10]	–	27.30	–
	Mensink et al. [34]	–	14.40	–
	Lampert et al. [20]	19.10	–	52.50
	Jayaraman and Grauman [35]	26.02 ± 0.05	–	56.18 ± 0.27
	Romera-Paredes and Torr [25]	27.27 ± 1.62	–	65.75 ± 0.51
AlexNet	Akata et al. [26]	–	40.30	–
	Bucher et al. [12]	46.14 ± 0.91	41.98 ± 0.67	75.48 ± 0.43
VGG-19	Lampert et al. [20]	38.16	–	72.00
	Romera-Paredes and Torr [25]	24.22 ± 2.89	–	82.10 ± 0.32
	Zhang and Saligrama [16]	46.23 ± 0.53	30.41 ± 0.20	82.50 ± 1.32
	Zhang Saligrama [11]	50.35 ± 2.97	42.11 ± 0.55	83.83 ± 0.29
	Bucher et al. [12]	53.15 ± 0.88	43.29 ± 0.38	84.41 ± 0.71
	Our ABS-Net	**54.62 ± 1.12**	**44.67 ± 0.98**	**84.67 ± 0.47**

Table 7
Zero-shot retrieval mAP (%) comparison on three benchmark datasets. The results of other methods are cited from [11,12].

Method	aP&Y	CUB	SUN	Ave.
Zhang and Saligrama [16]	15.43	4.69	58.94	26.35
Zhang and Saligrama [11]	38.30	29.15	80.01	49.15
Bucher et al. [12]	36.92	25.33	52.68	38.31
Our ABS-Net	**53.12**	**50.50**	**90.74**	**64.79**

synthesizing iterations to train class predictor. For each synthesizing iteration we generate 100 pseudo representations per class (i.e. pseudo size = 100) and train predictor for 1 epoch. The results of (1) and (2) are depicted respectively in Fig. 7(a) and Fig. 7(b) (accuracies are detailed in Table 2 and 3). As shown in Fig. 7(a), the test accuracy on all classes has reached 92.28% just with 300 seen classes, which is improved steadily until convergence with the number of seen classes increasing. This phenomenon also can be concluded in Fig. 7(b). That is, with the number of seen classes from 50 to 200, the performance of ABS-Net on a certain number of unseen classes gradually become better, because the newly added classes have enriched the Fragment Repository of ABS-Net model. However, the curves drop gradually as the number of unseen classes increasing, as shown in Fig. 7(b). For example, in the case of 200 seen classes, the ABS-Net performance decreased from 97.96% to 84.75% with unseen classes increasing from 50 to 800. To summary, the effectiveness of our ABS-Net is demonstrated by these experimental results.

Fully supervised recognition. For analyzing the extensibility in supervised recognition scenario of our ABS-Net, we make a comparison with conventional supervised learning model (CSLM). To explain, the so-called fully supervised recognition here is to use some training images containing 1k classes to recognize the test images which are also from these 1k classes. For ABS-Net, we use the same configuration with the above setting (1) except that here we own some training data for all classes. In addition to no training for JAFE and no pseudo representation synthesis, the compared model CSLM also has the same configuration with ABS-Net (i.e. CNN+SLP+Softmax), which however adopts a naive end-to-end training strategy based on available training data for all classes. Like above experiments, for ABS-Net we train JAFE for 10 epochs and then played 5 synthesizing iterations, but we train CSLM for 30 epochs to get the optimum. For a more detailed comparison, we select 4 different pseudo sizes to actualize our ABS-Net model on this fully supervised task. We show the effect of varying number of training images on test accuracy in Fig. 8. As we see, even with few training images, our ABS-Net can also achieve good performance on recognition for all classes, far exceeding the results of CSLM in the same training dataset size. Moreover, no matter how much pseudo size or training dataset size is, the results of ABS-Net are always superior to CSLM (detailed in Table 4). Therefore, we believe that our ABS-Net realizes a feature-level data augmentation for supervised learning and the synthetic pseudo representations enrich our understandings with respect to test classes.

4.2. Benchmark comparison

We test out method on other three benchmark image datasets for ZSL, i.e. aP&Y, CUB-200-2011 and SUN Attribute. The statistics of each dataset have been summarized in Table 5. More specifically, each image of aP&Y, CUB-200-2011 and SUN Attribute has its own image-level attribute notations. For seen classes we utilize their image-level attribute notations to train JAFE, but for unseen classes we take the means of their image-level attribute notations as their class-level attribute descriptions. To make comparisons with previous works, we use the same seen/unseen splits as [18] (aP&Y), [26] (CUB-200-2011) and [35] (SUN Attribute).

Experimental settings. Like previous work [11,12,16], we utilize Keras with "VGG19" pre-trained model [3] without fine-tuning on these benchmark datasets to extract the 4096-dimensional CNN feature vector on the first fully connected layer for each image. Moreover, we randomly select respectively 3, 10 and 30 classes from the seen classes set of aP&Y, CUB-200-2011 and SUN Attribute to validate our ABS-Net model. It is worth mentioning that the network configuration and the hyperparameter settings for different datasets are somewhat different, but their JAFE units are all two-layer MLPs. For aP&Y, CUB-200-2011 and SUN Attribute, the number of neurons for each two-layer JAFE unit are 64-12, 64-6 and 48-6, respectively. The confidence factor tuple (Δ_+, Δ_-) for three benchmark datasets are respectively set to (0.6, 0.4), (0.55, 0.45) and (0.65, 0.35). The class predictors for different datasets are all Softmax classifiers, which are designed to match their corresponding numbers of target classes, i.e. 12+3, 50+10 and 10+30 respectively. The whole model is optimized by Adam [40] or RMSprop [42] with mini-batch size 256 for JAFE but 64 for class predictor.

Zero-shot recognition. The task of zero-shot recognition consists in identifying for a novel image which of unseen classes it belongs to. We summarize our comparison with state-of-the-art ZSL methods

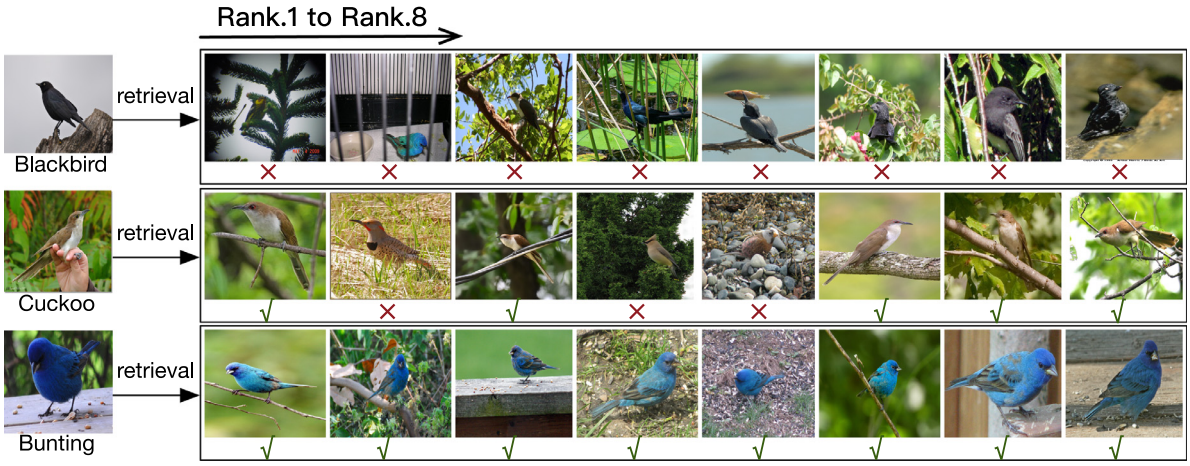


Fig. 10. Top-8 zero-shot retrieval results by our ABS-Net for class *"Brewer Blackbird"*, *"Black billed Cuckoo"* and *"Indigo Bunting"* (from top to down) in CUB-200-2011.

in Table 6, where the results of previous methods on three benchmark datasets have been listed and the '–' indicates that no experiments have been performed on this dataset in original paper. For comparison we also reported average results over 3 trials. As we see, our proposed method outperforms slightly state-of-the-art performance in all three datasets.

Zero-shot retrieval. The task of zero-shot retrieval is to search some images related to the specified attribute descriptions of unseen classes. Here we use above well-trained ABS-Net model to rank all test images along each unseen class based on their final Softmax scores output by the class predictor. Table 7 presents the comparative results for mAP in three benchmark datasets. Note that our ABS-Net significantly and consistently outperforms state-of-the-art ZSL methods by 14.82%, 21.35% and 10.73% on three benchmark datasets respectively and 15.64% on average, which are tremendous improvements for zero-shot retrieval tasks. We believe it benefits from the good tradeoff of our ABS-Net model in class prediction, namely one misclassified image still owns high score for its correct class. Fig. 9 shows precision-recall curves and AP values for all unseen classes of three benchmark datasets. Compared with the curves in [12], our method obviously has a larger area under the curves for arbitrary dataset. For intuition, we select 3 unseen classes of CUB-200-2011 in different difficulty levels to visualize our zero-shot retrieval results in Fig. 10 with top-8 returns. Two most frequent failure modalities are: (1) retrieved images contains cluttered background, and (2) a visually similar category is confused with the target category. These failure modalities are still some difficult challenges for ZSL as well as fine-grained recognition.

4.3. Ablation study

To demonstrate how the JAFE and the Fragment Repository filtered by confidence factor tuple in ABS-Net contribute to the performance, we conduct a series of abalation experiments on the above mentioned three benchmark datasets for ZSL. Concretely, we compared four model variants about the JAFE and the Fragment Repository with the complete ABS-Net whose performance has been reported in Section 4.2. We detailed these variants as follows.

ABS-No-JAFE-Net. The JAFE module is removed from ABS-Net, and for each image which owns m attributes we randomly divide its original 4096-dimensional ImageNet pre-trained VGG19 based feature into m parts. The number of feature points in each part approximately equals to 4096/m, and these feature points are concatenated into a feature sub-vector of approximately 4096/m dimensions. These feature sub-vectors are used as attribute feature vectors corresponding to the attribute tags of images in random order, which are then stored into the Fragment Repository for synthesizing pseudo representations for unseen classes. Because no JAFE performed on original CNN features, there are naturally no positive or negative estimations for attribute feature vectors. Therefore we adopted Eq. (5) and (6) to construct the Fragment Respository. Note that many random trials have been acted upon division of original CNN features and their correspondence with attribute tags, we report the average performance of the best three trials.

ABS-No-FG-Net. The Fragment Repository is removed from ABS-Net, which means that we don not distinguish between positive and negative attribute features extracted by the JAFE. Instead, we just gruffly randomly sample the attribute feature from $H_+^{s(i)} \cup H_-^{s(i)}$ regardless of value of unseen class attribute description $\mathbf{z}^t$.

ABS-Net($\Delta_+ = 0, \Delta_- = 1$). The Fragment Repository is constructed according to Eqs. (5) and(6), which means we do not care about the evaluations by attribute predictors, as shown in Fig. 4(a).

ABS-Net($\Delta_+ = 0.5, \Delta_- = 0.5$). The Fragment Repository is constructed according to Eqs. (7) and (8) regardless of these ambiguous attribute features, as shown in Fig. 4(b).

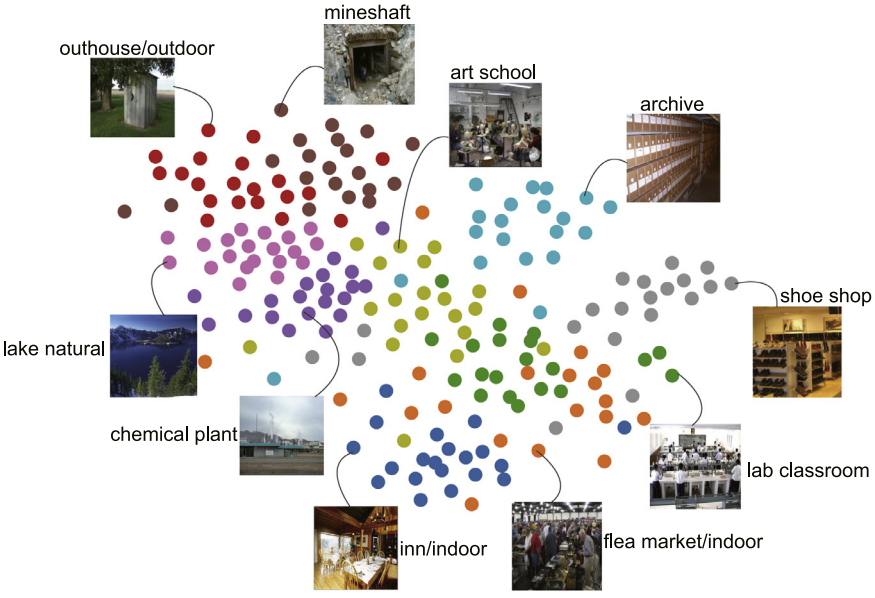
With the results in Table 8, the comparisons between different model variants are available to show each component's contribution. the JAFE and Fragment Repository can be regarded as the basic essentials of the ABS-Net because of the significant improvements they lead to. Meanwhile, the importance of confidence factor tuple is also demonstrated.

4.4. Visualization for feature representations

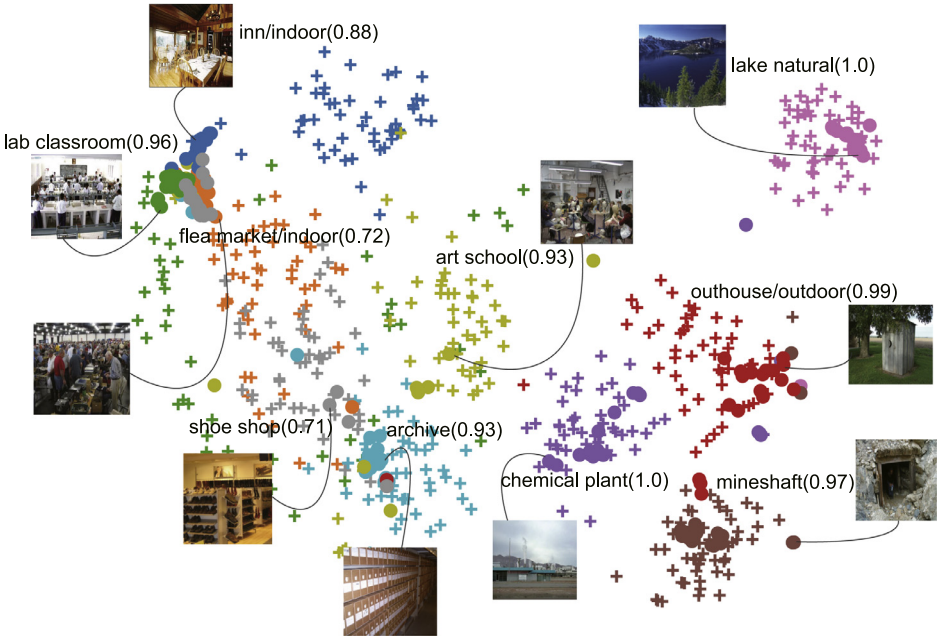
For better comprehension of our ABS-Net, we visualize the original features extracted by ImageNet pre-trained VGG19 model and the learned pseudo feature representations as well as the real feature representations extracted by JAFE on SUN Attribute using t-SNE [43] in Fig. 11 (a) and Fig. 11(b), respectively. On the one hand, as depicted in Fig. 11(a), the 10 unseen classes of SUN Attribute are hard to distinguish in original CNN feature space. But once these original CNN features are embeded into a common attribute feature space via JAFE, most of unseen classes can be separated from others, like *"lake natural"*, *"mineshaft"* and *"chemical*

Table 8
Zero-shot recognition accuracy (%) comparison (mean ± std) between different model variants.

Model	aP&Y	CUB	SUN
ABS-No-JAFE-Net	9.21 ± 2.15	8.64 ± 0.85	24.33 ± 1.31
ABS-No-FG-Net	14.86 ± 1.17	10.58 ± 1.93	30.67 ± 2.66
ABS-Net($\Delta_+ = 0, \Delta_- = 1$)	39.26 ± 0.64	28.84 ± 1.12	60.67 ± 1.03
ABS-Net($\Delta_+ = 0.5, \Delta_- = 0.5$)	46.74 ± 1.02	38.23 ± 0.52	76.50 ± 0.82
ABS-Net($\Delta_+ \geq 0.5, \Delta_- \leq 0.5$)	**54.62 ± 1.12**	**44.67 ± 0.98**	**84.67 ± 0.47**



(a) Original features based on ImageNet pre-trained VGG19 model.



(b) Real features extracted by JAFE (point) and the learned pseudo feature representations (plus).

Fig. 11. *t*-SNE visualization for the original features, real and pseudo feature representations of 10 unseen classes in SUN Attribute. The 10 unseen classes are indicated with 10 different colors respectively. For (b), the solid points denote the real feature representations but plus signs denote the pseudo ones. The number in brackets is the retrieval AP (%). Best viewed in color.

Algorithm 1: Synthesizing pseudo feature representations

Input: $\mathcal{H}_+^s = \{H_+^{s(i)}\}_{i=1}^m$, $\mathcal{H}_-^s = \{H_-^{s(i)}\}_{i=1}^m$, $\forall\ \mathbf{h}^{s(i)} \in H_+^{s(i)}$ or $H_-^{s(i)}$, $\mathbf{h}^{s(i)} \in \mathbb{R}^l$, unseen class label $t \in T$, its class-level attribute description $\mathbf{z}^t \in \mathbb{R}^m$
Initialize: pseudo size n, pseudo training dataset $H^t = \{\}$
for $k = 1$ ***to*** n **do**
  for $i = 1$ ***to*** m **do**
    Randomly generate $\epsilon \sim \mathbf{U}(0, 1)$;
    if $\epsilon \leq z_i^t$, randomly sample $\mathbf{h}_k^{u(i)} \sim H_+^{s(i)}$, $\mathbf{h}_k^{u(i)} \in \mathbb{R}^l$;
    else, randomly sample $\mathbf{h}_k^{u(i)} \sim H_-^{s(i)}$, $\mathbf{h}_k^{u(i)} \in \mathbb{R}^l$;
  end
  $\mathbf{h}_k^u \leftarrow \mathcal{C}(\mathbf{h}_k^{u(1)}, \ldots, \mathbf{h}_k^{u(m)})$, $\mathbf{h}_k^u \in \mathbb{R}^{ml}$;
  Add the $(\mathbf{h}_k^u, t)$ pair into set H^t;
end
Output: $H^t = \{(\mathbf{h}_k^u, t)\}_{k=1}^n$.

plant", etc, which can be concluded by observing the solid points in Fig. 11(b). This phenomenon demonstrate that the JAFE has enlarged the difference in feature representations between classes. On the other hand, as shown in Fig. 11(b), the real feature representations and the pseudo ones gather together nicely for some unseen classes that owns distinguishing characteristics compared to other classes, e.g. *"lake natural"*, *"mineshaft"* and *"chemical plant"*, which facilitates a high retrieval AP reasonably. However, for those classes that owns some common characteristics, e.g. *"inn/indoor"*, *"lab classroom"*, *"flea market/indoor"* and *"shoe shop"*, it is difficult to distinguish them by real feature representations since they are all indoor scenes. Hence, their gathering of pseudo and real feature representations are relatively poor, naturally leading to a relatively low retrieval AP. Figuratively, the synthetic pseudo representations can be approximately viewed as the results of adding slight disturbance upon real feature representations, realizing the feature-level data augmentation.

5. Conclusions

In this paper we develop a novel Attribute-Based Synthetic Network (ABS-Net) by generating pseudo feature representations. We summarize all accessible understandings related to attributes to construct our surmises for unseen classes, which also realizes inherently the data augmentation in feature level for conventional supervised learning scenario. Compared with most existing ZSL methods, the superiorities of our method are the simplicity for implementation and the extensibility for supervised recognition scene. Our method on three benchmark datasets shows competitive performance for zero-shot recognition task, and leads to significant improvement to state-of-the-art mAP for zero-shot retrieval task.

Conflict of interest

None declared.

Acknowledgments

This work is (jointly or partly) funded by the NSFC (Grant No. 61473167), Beijing Natural Science Foundation (Grant No. L172037) and the German Research Foundation (DFG) in Project Crossmodal Learning, NSFC 61621136008 / DFG TRR-169.

References

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, et al., Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252.
[2] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[3] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proceedings of the International Conference on Learning Representations, 2015.
[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[5] 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.
[6] E. Rodner, J. Denzler, Learning with few examples for binary and multiclass classification using regularization of randomized trees, Pattern Recognit. Lett. 32 (2) (2011) 244–251.
[7] Z. Sun, Q. Song, X. Zhu, H. Sun, B. Xu, Y. Zhou, A novel ensemble method for classifying imbalanced data, Pattern Recognit. 48 (5) (2015) 1623–1637.
[8] A. Shrivastava, V.M. Patel, R. Chellappa, Non-linear dictionary learning with partially labeled data, Pattern Recognit. 48 (11) (2015) 3283–3292.
[9] M. Liu, D. Zhang, S. Chen, Attribute relation learning for zero-shot classification, Neurocomputing 139 (2014) 34–46.
[10] Z. Akata, F. Perronnin, Z. Harchaoui, C. Schmid, Label-embedding for image classification, IEEE Trans. Pattern Anal. Mach. Intell. 38 (7) (2016) 1425–1438.
[11] Z. Zhang, V. Saligrama, Zero-shot learning via joint latent similarity embedding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 6034–6042.
[12] M. Bucher, S. Herbin, F. Jurie, Improving semantic embedding consistency by metric learning for zero-shot classiffication, in: Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 730–746.
[13] L. Liu, A. Wiliem, S. Chen, B.C. Lovell, What is the best way for extracting meaningful attributes from pictures? Pattern Recognit. 64 (2017) 314–326.
[14] M. Rohrbach, M. Stark, B. Schiele, Evaluating knowledge transfer and zero-shot learning in a large-scale setting, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 1641–1648.
[15] Y. Fu, T.M. Hospedales, T. Xiang, S. Gong, Transductive multi-view zero-shot learning, IEEE Trans. Pattern Anal. Mach. Intell. 37 (11) (2015) 2332–2345.
[16] Z. Zhang, V. Saligrama, Zero-shot learning via semantic similarity embedding, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4166–4174.
[17] T. Hastie, R. Tibshirani, J. Friedman, Overview of supervised learning, in: The Elements of Statistical Learning, Springer, 2009, pp. 9–41.
[18] A. Farhadi, I. Endres, D. Hoiem, D. Forsyth, Describing objects by their attributes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1778–1785.
[19] F.X. Yu, L. Cao, R.S. Feris, J.R. Smith, S.-F. Chang, Designing category-level attributes for discriminative visual recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 771–778.
[20] C.H. Lampert, H. Nickisch, S. Harmeling, Attribute-based classification for zero-shot visual object categorization, IEEE Trans. Pattern Anal. Mach. Intell. 36 (3) (2014) 453–465.
[21] X. Yu, Y. Aloimonos, Attribute-based transfer learning for object categorization with zero/one training example, in: Proceedings of the European Conference on Computer Vision, Springer, 2010, pp. 730–746.
[22] X. Wang, Q. Ji, A unified probabilistic approach modeling relationships between attributes and objects, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 2120–2127.
[23] M. Suzuki, H. Sato, S. Oyama, M. Kurihara, Transfer learning based on the observation probability of each attribute, in: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2014, pp. 3627–3631.
[24] B. Hariharan, S. Vishwanathan, M. Varma, Efficient max-margin multi-label classification with applications to zero-shot learning, Mach. Learn. 88 (1–2) (2012) 127–155.
[25] B. Romera-Paredes, P.H. Torr, An embarrassingly simple approach to zero-shot learning, in: Proceedings of the International Conference on Machine Learning, 2015, pp. 2152–2161.
[26] Z. Akata, S. Reed, D. Walter, H. Lee, B. Schiele, Evaluation of output embeddings for fine-grained image classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2927–2936.
[27] A. Frome, G.S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al., Devise: a deep visual-semantic embedding model, in: Proceedings of the Advances in Neural Information Processing Systems, 2013, pp. 2121–2129.
[28] J. Lei Ba, K. Swersky, S. Fidler, et al., Predicting deep zero-shot convolutional neural networks using textual descriptions, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4247–4255.
[29] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, B. Schiele, Latent embeddings for zero-shot classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 69–77.
[30] T. Mensink, J. Verbeek, F. Perronnin, G. Csurka, Distance-based image classification: generalizing to new classes at near-zero cost, IEEE Trans. Pattern Anal. Mach. Intell. 35 (11) (2013) 2624–2637.

[31] J. Wang, Z. Deng, K.-S. Choi, Y. Jiang, X. Luo, F.-L. Chung, et al., Distance metric learning for soft subspace clustering in composite kernel space, Pattern Recognit. 52 (2016) 113–134.
[32] B. Nguyen, C. Morell, B. De Baets, Supervised distance metric learning through maximization of the Jeffrey divergence, Pattern Recognit. 64 (2017) 215–225.
[33] M. Rohrbach, S. Ebert, B. Schiele, Transfer learning in a transductive setting, in: Proceedings of the Advances in Neural Information Processing Systems, 2013, pp. 46–54.
[34] T. Mensink, E. Gavves, C.G. Snoek, Costa: co-occurrence statistics for zero-shot classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2441–2448.
[35] D. Jayaraman, K. Grauman, Zero-shot recognition with unreliable attributes, in: Proceedings of the Advances in Neural Information Processing Systems, 2014, pp. 3464–3472.
[36] Z. Fu, T. Xiang, E. Kodirov, S. Gong, Zero-shot object recognition by semantic manifold distance, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2635–2644.
[37] Wah C., Branson S., Welinder P., Perona P., Belongie S. The Caltech-UCSD birds-200-2011 dataset. Computation & Neural Systems Technical Report, CNS-TR-2011-001, 2011.
[38] G. Patterson, J. Hays, Sun attribute database: discovering, annotating, and recognizing scene attributes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2751–2758.
[39] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, et al., Tensorflow: a system for large-scale machine learning., in: OSDI, 16, 2016, pp. 265–283.
[40] D. Kingma, J. Ba, Adam: a method for stochastic optimization, in: Proceedings of the International Conference on Learning Representations, 2015.
[41] D. Mahajan, S. Sellamanickam, V. Nair, A joint learning framework for attribute models and object descriptions, in: Proceedings of the IEEE International Conference on Computer Vision, 2011, pp. 1227–1234.
[42] T. Tieleman, G. Hinton, Lecture 6.5-RMSPROP: divide the gradient by a running average of its recent magnitude, COURSERA: Neural Netw. Mach. Learn. 4 (2) (2012) 26–31.
[43] L.v.d. Maaten, G. Hinton, Visualizing data using t-sne, J. Mach. Learn. Res. 9 (Nov) (2008) 2579–2605.

Jiang Lu received the B.S. degree from Tsinghua University, Beijing, China, in 2013, where he is currently working toward the Ph.D. degree in the Department of Automation. His research interests include machine learning, deep learning and computer vision.

Jin Li received the B.S. degree in 2014 and the M.S. degree in 2017, from the Department of Automation, Tsinghua University, Beijing, China. His research interests include machine learning and deep learning.

Ziang Yan received the B.S. degree from Tsinghua University, Beijing, China, in 2015, where he is currently working toward the Ph.D. degree in the Department of Automation. His research interests include machine learning, deep learning and computer vision.

Fenghua Mei received the B.S. degree in aero-engine from Northwestern Polytechnical University, Xi'an, China, in 1993, and the M.S. degree in computer science from China Marine Development and Research Center, Beijing, China, in 2000. He is currently a research director in China Marine Development and Research Center. His research interests lie in the area of avionics system and computer information system integration.

Changshui Zhang received the B.S. degree in mathematics from Peking University, Beijing, China, in 1986, and the Ph.D. degree from the Department of Automation, Tsinghua University, Beijing, in 1992. His currently research interests include artificial intelligence, image processing, pattern recognition, machine learning, and evolutionary computation.