Attentive Region Embedding Network for Zero-shot Learning

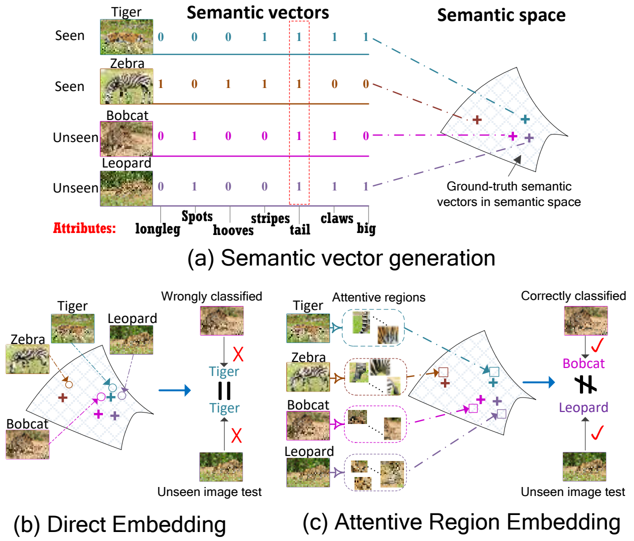
Guo-Sen Xie[1], Li Liu[1], Xiaobo Jin[3], Fan Zhu[1], Zheng Zhang[2], Jie Qin[1], Yazhou Yao[1], Ling Shao[1]
[1]Inception Institute of Artificial Intelligence, UAE
[2]University of Queensland, Australia
[3]Xi'an Jiaotong-Liverpool University, China

Abstract

Zero-shot learning (ZSL) aims to classify images from unseen categories, by merely utilizing seen class images as the training data. Existing works on ZSL mainly leverage the global features or learn the global regions, from which, to construct the embeddings to the semantic space. However, few of them study the discrimination power implied in local image regions (parts), which, in some sense, correspond to semantic attributes, have stronger discrimination than attributes, and can thus assist the semantic transfer between seen/unseen classes. In this paper, to discover (semantic) regions, we propose the attentive region embedding network (AREN), which is tailored to advance the ZSL task. Specifically, AREN is end-to-end trainable and consists of two network branches, i.e., the attentive region embedding (ARE) stream, and the attentive compressed second-order embedding (ACSE) stream. ARE is capable of discovering multiple part regions under the guidance of the attention and the compatibility loss. Moreover, a novel adaptive thresholding mechanism is proposed for suppressing redundant (such as background) attention regions. To further guarantee more stable semantic transfer from the perspective of second-order collaboration, ACSE is incorporated into the AREN. In the comprehensive evaluations on four benchmarks, our models achieve state-of-the-art performances under ZSL setting, and compelling results under generalized ZSL setting.

1. Introduction

Zero-shot learning (ZSL) [29, 1, 21, 46] is proposed for solving challenging classification tasks, wherein, the label spaces for the training set and test set are disjoint from each other, and there are no training samples (zero-shot) for test categories. Most recently, for traditional recognition systems [16, 23, 33, 32, 31, 47], two issues hinder their advancements, i.e., 1) annotating large-scale samples is both time consuming and expensive [52], and 2) new categories are constantly emerging [50], and some of them are difficult (or even dangerous) to be collected, e.g., the identified coffinfish in the deep-sea. In contrast, ZSL has the intrinsic advantages of tackling the image annotation and novel class recognition problems, which makes it a hot topic in recent years.



(a) Semantic vector generation

(b) Direct Embedding

(c) Attentive Region Embedding

Figure 1. (a) Toy example of generating semantic vectors; crosses indicate ground-truth semantic vectors in the semantic space. Four animals all have the attribute "tail" (shown in red dashed box), as such, "tail" is misleading the afterward classification. (b) Circles represent samples in the semantic space by direct embedding (DE) images to this space. For DE, Bobcat and Leopard are **wrongly recognized** as Tiger. (c) Squares are these samples in the semantic space embedded by ARE; in this case, by preserving the discrimination from the part level, the confused unseen images (Bobcat and Leopard) can be well **distinguished** and **correctly recognized**. Best viewed in color.

To transfer semantic knowledge for images from two disjoint category spaces, the semantic description of each category (for both seen/unseen classes), as the high-level side information, is key for accomplishing ZSL. Widely-used side information includes attributes [11], word vectors [39], sentences [36], and gaze [18], among which, attributes draw the greatest attention, and are adopted in this paper. A general scenario for ZSL is to find an embedding space based on seen images. Typically, the semantic space (e.g., the space in Fig. 1(a), which is spanned by quantized attributes, i.e., semantic vectors) [37, 13, 20, 1, 3, 36, 44, 39, 8, 42, 35, 27, 40], the image feature space [4, 26, 45, 26], and the latent intermedium space [41] usually serve as the embedding space. In that space, to further distinguish unseen images, nearest neighbour search is used to match the tested image representation with that of unseen class prototypes, i.e., semantic vector w.r.t. each unseen class.

Most leading ZSL methods, whether end-to-end convolutional neural network (CNN) based [27, 42, 22], or deep feature-based approaches [19, 51, 26, 53, 54, 34], emphasize on learning the embedding between the (learned) global image (or feature) and the counterpart semantic vector. However, all these methods are actually based on global projection of the whole image. After acquiring the semantic vectors w.r.t. seen/unseen classes, there exist two drawbacks for these global projection methods: 1) due to the subtle difference of the seen image (tiger) and unseen images (bobcat, leopard) in the global feature space, they are neighbors (circles in Fig. 1(b)) in the projected semantic space, where, it is hard to distinguish them; 2) the annotated ground-truth semantic vectors of bobcat, leopard and tiger are extremely similar (Fig. 1(a)). It is thus hard, by feeding the global features to the embedding model, to learn a desirable projection for matching the input similar images with their confused semantic vectors. In contrast, ARE can fit input images with their confused ground-truth semantic vectors pretty well (in Fig. 1(c), squares are near their ground-truth, i.e., crosses).

Since the high-level abstractions of some image regions can lead to the attribute concept [10], and in order to alleviate the above problems, we resort to the regions (parts) in the images. We observe that 1) besides the global image representation, properly discovered regions account for better knowledge transfer from seen to unseen class, and 2) some regions can capture local appearance differences for the same attribute concept, e.g., the region blocks of different *tails* are different in appearance. In this sense, part regions are more discriminative than the corresponding attribute. Therefore, projecting region representation into the semantic space can preserve more such local differences. In this way, tiger, bobcat, zebra, and leopard can be well recognized from each other (Fig. 1(c), squares). In term of discriminative feature learning, the part based feature has long been established as a powerful one [12, 25]. Motivated by the above observations, to facilitate the semantic transfer between seen/unseen images in the part level, we propose an end-to-end attentive region embedding network (AREN) (Fig. 2) for ZSL. To sum up, our contributions are:

1) An attention mechanism is leveraged to automatically discover semantic/discriminative regions (parts), without any part detection or annotation. Moreover, a novel adaptive thresholding mechanism is further proposed to suppress redundant attentive regions and introduce robustness, therefore leading to the attentive region embedding (ARE) subnet. This is the first attempt to introduce attention to ZSL/GZSL freely, without any part detection/annotation.

2) To capture second-order appearance differences collaboratively with different attentive regions, an attentive compressed second-order embedding (ACSE) is further incorporated into the AREN framework. This is the first time second-order statistics have been explored within ZSL/GZSL.

3) Integrating ARE and ACSE together yields the end-to-end AREN framework, which is trained with the guidance of a compatibility loss with frozen classifier weights (taken from the seen class attributes).

2. Related Works

(Generalized) Zero-shot Learning. As the pioneering work of ZSL, Lampert et al. [21] propose direct attribute prediction (DAP) model, which first learns the attribute classifiers, and then calculates the posterior of a test class for a given image. However, DAP neglects the associations between different attributes. To mitigate the unreliability of the individually learned attribute classifiers, a random forest solution [17] is advocated. As a whole, the leading methods for ZSL are the embedding based ones equipped with the compatibility loss, which can well associate the images and their attributes. Specifically, Akata et al. [1] proposed ALE, where a bilinear-style hinge loss is leveraged. LATEM [44] was then introduced to incorporate nonlinearity to the model. Other embedding based approaches include DEVICE [13], SJE [3], CMT [39], ESZSL [37], SAE [20], and DEM [52]; for a more detailed description of them, refer to [46]. Most of the methods mentioned above adopt deep features and emphasize the model itself, thus resulting in relatively inferior ZSL performances. Most recently, another branch of approaches, i.e., end-to-end trainable CNN models, have been proposed. The most representatives of these train CNN model by 1) alleviating prediction bias, i.e., QSFL [42], 2) gradually zooming global image objects, i.e., LDF [22], and 3) automatically learning the relations, i.e., RN [50]. However, none of them focus on the part (region) level for enhancing the semantic transfer in ZSL. By expanding the search label space to also consider seen classes during testing, ZSL becomes Generalized ZSL (GZSL). All ZSL methods can be adopted to solve GZSL task by obeying the data splits proposed in [46].

Attention. Attention [49], widely used and extensive-

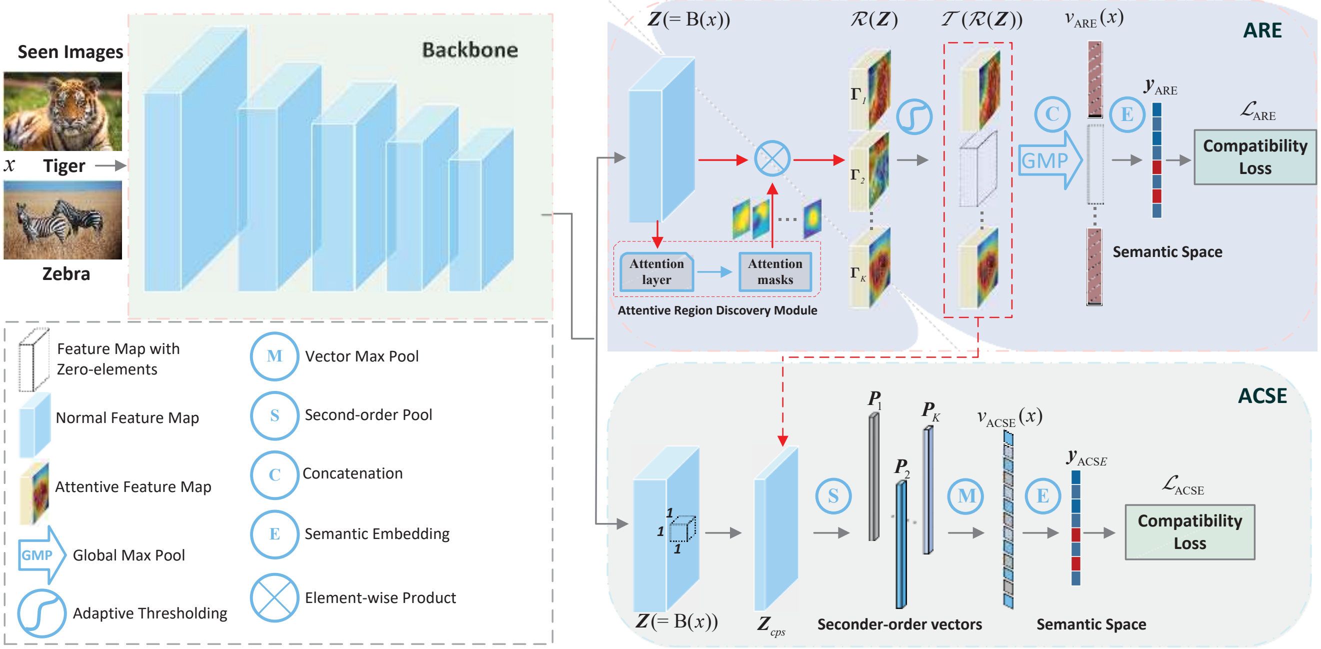


Figure 2. The architecture of the proposed Attentive Region Embedding Network (AREN) model, and the upper branch is ARE, meanwhile, the bottom branch is ACSE. For ARE: the input image x is first fed into the backbone net, thus generating the last convolutional feature map $\mathbf{Z}$ which undergoes the attentive region discovery module, and the K attentive feature maps $\Gamma_k, k = 1, 2, \cdots, K$ is produced. Then, AT is applied to them. Global max pooling, concatenation and embedding to semantic space are further conducted. For ACSE: $\mathbf{Z}$ is first compressed by 1×1 convolution, leading to $\boldsymbol{Z}_{cps}$. $\boldsymbol{Z}_{cps}$ and attentive feature maps after AT are used to construct the second-order vectors, then vector max pooling and ebmbedding to semantic space are leveraged. Finally, compatibility losses are used for ARE/ACSE training.

ly studied in recent years, has been successfully applied to various fields such as visual question answering [48] and semantic segmentation [9]. Inspired by the above achievements, an attention mechanism is incorporated into our AREN framework, with the guidance of associating images and their attributes, i.e., the function of the compatibility loss. In this way, the learned attention maps can capture multiple semantic regions that are useful for semantic transfer between seen/unseen images.

Transductive setting: In addition to seen images, utilizing the unseen images (without labels) during the training phase yields the transductive setting for ZSL/GZSL [14]. In this paper, our focus is inductive ZSL/GZSL, i.e., the most general setting in the realistic scenario.

3. Methodology

Notations. Suppose that the training set (seen classes) $\mathcal{S} = \{(x_i^s, y_i^s), i = 1, 2, \cdots, N_s\}$ is given, where $x_i^s \in \mathcal{X}^{\mathcal{S}}$ is the i-th data samples (totally N_s samples) with a corresponding seen class label as y_i^s, here $y_i^s \in \mathcal{Y}^{\mathcal{S}}$, and $\mathcal{Y}^{\mathcal{S}}$ is the label set of seen classes. In ZSL, given the testing set $\mathcal{U} = \{(x_i^u, y_i^u), i = 1, 2, \cdots, N_u\}$ from the unseen classes, where $x_i^u \in \mathcal{X}^{\mathcal{U}}$ and $y_i^u \in \mathcal{Y}^{\mathcal{U}}$ is the i-th unseen sample and its label, respectively; the seen and unseen label sets are disjoint, i.e., $\mathcal{Y}^{\mathcal{S}} \cap \mathcal{Y}^{\mathcal{U}} = \emptyset$. Moreover, we denote the semantic vector (class prototype) set w.r.t. seen/unseen classes as $\{a_i^s\}_{i=1}^{\mathcal{C}^s}$ and $\{a_i^u\}_{i=1}^{\mathcal{C}^u}$, herein, $a_i^s/a_i^u \in \mathbb{R}^Q$ is the semantic vector corresponding to the i-th seen/unseen class. $\mathcal{C}^s/\mathcal{C}^u$ is the category number for seen/unseen classes, and Q is the dimension of the semantic vector, and also the dimension of the semantic space. The difference for ZSL and GZSL [46] lies in that, for GZSL, while testing the seen/unseen image x_j^{test}, its predicted label set is $\mathcal{Y} = \mathcal{Y}^{\mathcal{S}} \cup \mathcal{Y}^{\mathcal{U}}$.

3.1. The Attentive Region Embedding Network

The Attentive Region Embedding Network (AREN) is illustrated in Fig. 2, which consists of two branches, i.e., the Attentive Region Embedding (ARE) and the Attentive Compressed Second-order Embedding (ACSE). In particular, ARE is the main body for capturing discriminative regions automatically, without any part-level annotation/detection. Meanwhile, ACSE is targeted at grasping more subtle semantic information by second-order inference. To achieve the ZSL, in both ARE and ACSE, the embedding to semantic space is leveraged, which is the commonly utilized strategy [42, 22, 52, 27] in the end-to-end deep network based ZSL framework.

3.1.1 Attentive Region Embedding

In Fig. 2 (upper stream), the last convolutional feature map $\boldsymbol{Z}$ of the backbone (e.g., ResNet101) for input image x, is fed into the ARE, which first undergoes the Attentive Region Discovery (ARD) module, followed by an adaptive thresholding (AT) procedure. In this way, the attention regions can be effectively focused and highlighted. The AT operation can further purify the generated attention regions by filtering out the ones with low attentive strength, thus generating some feature maps with all zero elements. Afterward, unlike the widely-used global average pooling, we leverage the global maximum pooling for these feature maps and then concatenate them, which leads to the representation $\boldsymbol{v}_{\text{ARE}}(x)$ for an image x in the region space. Specifically, we formulate $\boldsymbol{v}_{\text{ARE}}(x)$ as follows:

$$\boldsymbol{v}_{\text{ARE}}(x) = \mathcal{G}(\mathcal{T}(\mathcal{R}(\boldsymbol{Z}))),\ \boldsymbol{Z} = \text{B}(\text{x}), \tag{1}$$

where B, $\mathcal{R}$, $\mathcal{T}$, and $\mathcal{G}$ are the backbone network operation, the ARD operation, the AT operation, and the GMP/concatenation operation, respectively. The model parameters in Eqn. (1) are omitted to ease reading.

Due to our AT operation, some segments of the cascaded region vector $\boldsymbol{v}_{\text{ARE}}$ will be all zeros. As validated in the experiments of subsection 4.5, $\boldsymbol{v}_{\text{ARE}}$ can achieve an improved performance over its counterpart without an AT operation. Finally, to accomplish the ZSL/GZSL task, $\boldsymbol{v}_{\text{ARE}}$ is embedded into the semantic space. The projected representation $\boldsymbol{y}_{\text{ARE}}$ in the semantic space for x is defined as

$$\boldsymbol{y}_{\text{ARE}} = \mathcal{E}(\boldsymbol{v}_{\text{ARE}}). \tag{2}$$

The parameters in Eqn. (1) and (2) are jointly trained with the guidance of a cross-entropy-like compatibility function[22, 1, 42, 3, 27] (we will revisit the compatibility loss in detail in subsection 3.2).

Attentive Region Discovery Module: To discover the multiple attentive regions of an input image x, which serve as the bridge for semantic transfer from the part level, we leverage the attention mechanism to automatically learn to focus. With the supervision of high-level semantic attributes (from compatibility loss) in the topmost layer of the net, we hope that the discovered regions can match with the annotated semantic attributes. In this way, the yielded regions are essentially communicating between seen/unseen classes, e.g., a child has heard the description of "zebra" as looking like a "horse" with black-white stripes; then when she sees the picture of "zebra", by focusing on the black-white stripy regions, she can tell it's a "zebra". In this section, we will elaborate on the ARD module, i.e., mapping $\mathcal{R} : \boldsymbol{Z} \to \mathcal{R}(\boldsymbol{Z})$, where, $\boldsymbol{Z}$ is the last convolutional feature map of the backbone net. $\boldsymbol{Z}$ is a 3D tensor, and we suppose $\boldsymbol{Z} \in \mathbb{R}^{H\times W\times C}$, where C, H, and W are the size of the channel, height, and width, respectively. Let $z(h, w, c) \in \mathbb{R}$ be the response value in location (h, w) of the c-th channel from $\boldsymbol{Z}$. We further denote the *desirable* region number as K, where, *desirable* means that 1) the number of regions is discriminative for distinguishing seen/unseen images; and 2) some of these regions are matched with the semantic attributes, e.g., the region of leg matches the attribute "leg". Inspired by the application of attention models to various fields, such as image captioning [49], we employ the attention mechanism to the ZSL field as well, with the aim of grasping the semantic regions and further narrowing the semantic gap between seen/unseen images.

Specifically, by taking $\boldsymbol{Z}$ as input, we generate K 2-dimensional masks $\boldsymbol{M}_k \in \mathbb{R}^{H\times W}$, $(k = 1, 2, \cdots, K)$:

$$\boldsymbol{M}_k = \mathcal{M}_{\text{MaskGenerate}_k}(\boldsymbol{Z}), \tag{3}$$

where $\mathcal{M}_{\text{MaskGenerate}_k}(\cdot)$ is a mask generation operation which is implemented by convolution on $\boldsymbol{Z}$ followed by the Sigmoid thresholding. Thus, the value $m_k(h, w)$ in location (h, w) of $\boldsymbol{M}_k$ can reflect the strength that location (h, w) of $\boldsymbol{Z}$ falls into the k-th region. Furthermore, suppose the k-th attentive convolutional feature map is $\boldsymbol{\Gamma}_k \in \mathcal{R}(\boldsymbol{Z})$. In particular, $\boldsymbol{\Gamma}_k$ is obtained by

$$\boldsymbol{\Gamma}_k = \boldsymbol{O}_{\text{Reshape}}(\boldsymbol{M}_k) \otimes \boldsymbol{Z}. \tag{4}$$

In Eqn. (4), $\boldsymbol{O}_{\text{Reshape}}(\cdot)$ reshapes the input to be the same size as that of $\boldsymbol{Z}$, $\otimes$ indicates an element-wise product.

Adaptive Thresholding: After the ARD, the generated K attentive maps usually have redundancy, such as the background noises. To purify these maps, we propose the AT operation. AT takes these K attentive feature maps (in Eqn. (4)) as inputs, calculates the maximum value of each 2D mask map ($\boldsymbol{M}_k$), yielding the maximum value vector $\boldsymbol{m}_v \in \mathbb{R}^{K\times 1}$ w.r.t. these K mask maps. Then, the maximum value of $\boldsymbol{m}_v$ is achieved, denoted as AT_{max}:

$$AT_{\text{max}} = \max_{1\leqslant k\leqslant K} \boldsymbol{m}_v(k). \tag{5}$$

AT_{max} is the global maximum value of these K attention mask maps in Eqn. (3). An adaptive coefficient α $(0 \leqslant \alpha \leqslant 1)$ is introduced, based on which, we denote the final thresholding bound as $T_B = \alpha \times AT_{\text{max}}$. To this end, if the k-th value in $\boldsymbol{m}_v$ is less than T_B, the corresponding attentive feature map $\boldsymbol{\Gamma}_k$ will be set as all zero elements. Throughout the paper, for a given fixed K, there is only one parameter α to be tuned. Experimental evaluation shows improvements in performance by setting a proper value for α.

3.1.2 Attentive Compressed Second-order Embedding

In Fig. 2, after acquiring the last convolutional feature map $\boldsymbol{Z}$ and the K purified attentive feature maps (from the ARE), i.e., $\mathcal{T}(\boldsymbol{\Gamma}_k), k = 1, 2, \cdots, K$, with some of them having all zero-elements, we resort to second-order pooling [24] to alleviate the semantic gap between seen/unseen images. We first compress $\boldsymbol{Z}$ by a 1×1 convolution. The resulting compressed feature map $\boldsymbol{Z}_{\text{cps}}$ has N_{cps}(=20 throughout this paper) channels. In this way, the resulting compressed second-order representation will be compact and efficiently trainable.

For each $\mathcal{T}(\boldsymbol{\Gamma}_k)$, the second-order pooling with $\boldsymbol{Z}_{\text{cps}}$ yields the k-th second-order representation $\boldsymbol{P}_k$, which may be equal to all zero vector due to the attentive mechanism. $\boldsymbol{P}_k$ is formulated as:

$$\boldsymbol{P}_k = \boldsymbol{Z}_{\text{cps}} \circledcirc \mathcal{T}(\boldsymbol{\Gamma}_k), \tag{6}$$

where $\circledcirc$ is the seconder-order operation [24] between two input matrices. A vector maximum pooling is utilized to pool these K vectors, thus generating the final ACSE representation $\boldsymbol{v}_{\text{ACSE}}(x)$ for the input image x.

Similarly, $\boldsymbol{v}_{\text{ACSE}}(x)$ is embedded into the semantic space to achieve ZSL/GZSL. The projected representation $\boldsymbol{y}_{\text{ACSE}}$ implies the second-order statistics for better semantic transfer. To the best of our knowledge, this is the first time that second-order representation is incorporated into ZSL.

3.2. The Compatibility Loss

In this section, we discuss the problem of embedding to the semantic space, which is the most utilized strategy for making ZSL extendable. In general, the ZSL task formulates a mapping (prediction) function $f : \mathcal{X}^{\mathcal{S}} \mapsto \mathcal{Y}$ from the (seen) training set, as follows:

$$f(x, \boldsymbol{W}) = \arg\max_{y \in \mathcal{Y}} F(x, y; \boldsymbol{W}). \tag{7}$$

Given the trained parameters $\boldsymbol{W}$, the function in Eqn. (7) is used to predict an unseen image x^u. To associate the visual and semantic information, the score function $F(\cdot)$, parameterized by $\boldsymbol{W}$, is typically formulated as the bilinear compatibility function [1, 13, 37, 3, 22, 42, 27, 46]:

$$F(x, y; \boldsymbol{W}) = \theta(x)\boldsymbol{W}\phi(y), \tag{8}$$

where $\theta(x)$ and $\phi(y)$ are the visual embedding of image x and the semantic embedding of label y, respectively.

In the context of the proposed ARE and ACSE, $\boldsymbol{v}_{\text{ARE}}(x)$ and $\boldsymbol{v}_{\text{ACSE}}(x)$ serve as the visual embeddings of input image x, i.e., Eqn. (8) can be reformulated as follows:

$$\begin{aligned} F_{\text{ARE}}(x, y; \boldsymbol{\Theta}_{\text{ARE}}, \boldsymbol{W}_{\text{ARE}}) &= \boldsymbol{v}_{\text{ARE}}(x)^{\mathsf{T}} \boldsymbol{W}_{\text{ARE}} a^{y*}, \\ F_{\text{ACSE}}(x, y; \boldsymbol{\Theta}_{\text{ACSE}}, \boldsymbol{W}_{\text{ACSE}}) &= \boldsymbol{v}_{\text{ACSE}}(x)^{\mathsf{T}} \boldsymbol{W}_{\text{ACSE}} a^{y*}, \end{aligned} \tag{9}$$

where $\boldsymbol{\Theta}_{\text{ARE}}$ and $\boldsymbol{\Theta}_{\text{ACSE}}$ are the whole learnable parameters w.r.t. $\boldsymbol{v}_{\text{ARE}}(x)$ and $\boldsymbol{v}_{\text{ACSE}}(x)$ respectively, $\boldsymbol{W}_{\text{ARE}}$ and $\boldsymbol{W}_{\text{ACSE}}$ are the embedding parameters for mapping $\boldsymbol{v}_{\text{ARE}}(x)$ and $\boldsymbol{v}_{\text{ACSE}}(x)$ to the semantic embedding a^{y*}, which is the L-2 normalized semantic vector w.r.t. class y.

We further denote the normalized attribute matrix w.r.t. all these $\mathcal{C}^s$ seen classes as $\mathcal{A} \in \mathbb{R}^{Q \times \mathcal{C}^s}$, and the class outputs of image x on the final layer of ARE and ACSE are

$$\begin{aligned} O_{\text{ARE}}(x; \boldsymbol{\Theta}_{\text{ARE}}, \boldsymbol{W}_{\text{ARE}}) &= \mathcal{A}^{\mathsf{T}} \boldsymbol{W}_{\text{ARE}}^{\mathsf{T}} \boldsymbol{v}_{\text{ARE}}(x), \\ O_{\text{ACSE}}(x; \boldsymbol{\Theta}_{\text{ACSE}}, \boldsymbol{W}_{\text{ACSE}}) &= \mathcal{A}^{\mathsf{T}} \boldsymbol{W}_{\text{ACSE}}^{\mathsf{T}} \boldsymbol{v}_{\text{ACSE}}(x), \end{aligned} \tag{10}$$

To learn all these parameters ($\boldsymbol{\Theta}_{\text{ARE}}, \boldsymbol{W}_{\text{ARE}}, \boldsymbol{\Theta}_{\text{ACSE}}, \boldsymbol{W}_{\text{ACSE}}$) in Eqn. (10) in an end-to-end manner, i.e., to train the proposed AREN (Fig. 2), the loss function is

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{ARE}} + \lambda_2 \mathcal{L}_{\text{ACSE}}, \tag{11}$$

where λ_1 and λ_2 are trade-off parameters. $\mathcal{L}_{\text{ARE}}$ and $\mathcal{L}_{\text{ACSE}}$ are specified as

$$\begin{aligned} \mathcal{L}_{\text{ARE}} &= \frac{1}{N_s} \sum_{i=1}^{N_s} L(O_{\text{ARE}}(x_i^s; \boldsymbol{\Theta}_{\text{ARE}}, \boldsymbol{W}_{\text{ARE}}), y_i^s), \\ \mathcal{L}_{\text{ACSE}} &= \frac{1}{N_s} \sum_{i=1}^{N_s} L(O_{\text{ACSE}}(x_i^s; \boldsymbol{\Theta}_{\text{ACSE}}, \boldsymbol{W}_{\text{ACSE}}), y_i^s), \end{aligned} \tag{12}$$

where L is some classification loss. In this paper, cross-entropy (CE) loss is used. Compared with traditional CE loss, the difference lies in that the weights of the CE loss layer are frozen as $\mathcal{A}$ and are fixed without updating during the training phase. In this way, the attribute matrix $\mathcal{A}$ can guide the attentive region discovery, and progressively project the input image to the direction of its semantic representation. To this end, we term the designed two stream loss function $\mathcal{L}$ as compatibility loss.

3.3. Prediction

In the AREN framework, the tested unseen image can be projected into the semantic space by ARE/ACSE, enabling it to perform a separate prediction.

Prediction by ARE: A test image x^u can be projected into the semantic space, thus resulting in the ARE representation $\phi_{\text{ARE}}(x^u)$ $(= \boldsymbol{W}_{\text{ARE}}^{\mathsf{T}} \boldsymbol{v}_{\text{ARE}}(x^u))$. To predict the class label, the location of the maximum compatibility score can be chosen as the predicted label:

$$y^{u*} = \arg\max_{c \in \mathcal{Y}^U} \phi_{\text{ARE}}(x^u)^{\mathsf{T}} a_c^u. \tag{13}$$

Prediction by ACSE: Similarly, suppose the ACSE representation of x^u is $\phi_{\text{ACSE}}(x^u)$ $(= \boldsymbol{W}_{\text{ACSE}}^{\mathsf{T}} \boldsymbol{v}_{\text{ACSE}}(x^u))$. The predicted class label is:

$$y^{u*} = \arg\max_{c \in \mathcal{Y}^U} \phi_{\text{ACSE}}(x^u)^{\mathsf{T}} a_c^u. \tag{14}$$

Combining ARE and ACSE: After obtaining the ARE and ACSE representations of x^u, i.e., $\phi_{\text{ARE}}(x^u)$ and $\phi_{\text{ACSE}}(x^u)$, we first calculate their combined vector, and then predict the label in the same way as Eqn. (13) / Eqn. (14):

$$y^{u*} = \arg\max_{c \in \mathcal{Y}^U} (\gamma_1 \phi_{\text{ARE}}(x^u)^{\mathsf{T}} + \gamma_2 \phi_{\text{ACSE}}(x^u)^{\mathsf{T}}) a_c^u. \tag{15}$$

4. Experiments

4.1. Datasets and Settings

Four widely used ZSL datasets, i.e., CUB [43], AWA2 [46], SUN [30], and APY [11], are employed to validate the proposed AREN. Specifically, CUB contains a total of 11,788 bird images from 200 classes, each of which has a 312D continuous semantic vector. We use the standard split (SS) and the proposed split (PS) of 150/50 (seen/unseen) for evaluation, as done in [46]. AWA2 is an extension of AWA, whose images cannot be accessed. As such, we adopt AWA2, which includes 37,322 images of animals from 50 classes, among which 40/10 (seen/unseen) splits under SS/PS settings are evaluated, an 85D semantic vector is associated with each class. SUN is a scene image dataset, consisting of 14,340 images from 717 categories. SS/PS splits of 645/72 for seen/unseen classes are leveraged, and a 102D continuous semantic vector is constructed for each class. APY, with a total of 15,339 images, contains 32 categories with 64D attribute, and the seen/unseen splits are 20/12, evaluated under SS/PS settings. As in [46], after obtaining the AREN model, we conduct both ZSL and GZSL evaluations under the PS setting, only ZSL evaluation under the SS setting, for all four datasets.

4.2. Training Details and Parameters

For fair comparison with the published approaches, [46] reproduced nearly all leading methods using the 2,048D ResNet101 features. As such, the backbone net in Fig. 2 is taken as the ResNet101 net [16].

As with the initial pre-training on the ImageNet dataset [38], the input image size for these four datasets is 224×224. Therefore, the size of the last convolutional feature map $\boldsymbol{Z}$ for ResNet101 is $2048 \times 7 \times 7$. For each dataset, the AREN is trained for 100 epochs with an initial learning rate selected from $[0.0001, 0.003]$ (which is robust). The parameter (λ_1, λ_2) is fixed as $(0.5, 0.5)$ during the training of AREN, while, when the ARE and ACSE are trained separately, it is taken as $(1, 0)$ and $(0, 1)$, respectively, to ensure that only their own loss function contributes to the gradient updating. In the ARE, the number K of the part regions is experientially selected from $\{k \in \mathbb{N}_+ | 4 \leqslant k \leqslant 12\}$, and the AT parameter α is selected from $\{0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. In the ACSE, the compressed channel number N_{cps} is set to 20.

When testing unseen images, as for the separate testing of the ARE and ACSE, the combination coefficient (γ_1, γ_2) (in Eqn. (15)) is set to $(1, 0)$ and $(0, 1)$, respectively. Meanwhile, for the jointly trained AREN, (γ_1, γ_2) is set to $(0.5, 0.5)$ to achieve the fused matching results.

4.3. Evaluations in ZSL setting

We compare our proposed methods against current state-of-the-art models, on all four aforementioned ZSL datasets, under SS/PS settings [46]. Average Class Accuracy (ACA) is adopted as the evaluation metric. Table 1 presents the experimental results, from which, it can be concluded that **i)** ARE, ACSE and AREN consistently outperform the compared counterparts by a large margin, under both SS/PS settings. For example, ARE achieves 72.5% on CUB under the PS setting, which has improved the ACA up to 17%, compared with the recently proposed RN method whose ACA is only 55.6%. **ii)** For some datasets, the jointly trained AREN model performs slightly worse than the separately trained ARE and ACSE. The reasons lie in that 1) the coefficient (λ_1, λ_2) of the loss function in Eqn. (11) and the prediction coefficient (γ_1, γ_2) in Eqn. (15) are only roughly set, and, thus, may not lead to the best optimized model; and 2) the separate models (ARE and ACSE) are powerful enough, and the joint training disturbs their discrimination. **iii)** Most importantly, under the inductive setting, we are on par with and have even surpassed some of the leading transductive methods (such as QFSL).

Table 1. ZSL results (ACA, in %) on evaluated four datasets. Our methods and most of the compared methods use ResNet101 as the backbone net for fair comparisons. SS = Standard Split, PS = Proposed Split. The best result is marked in red, the second best in blue, and the third best in **bold**.

	Method	CUB		SUN		AWA2		APY	
		SS	**PS**	**SS**	**PS**	**SS**	**PS**	**SS**	**PS**
†	DAP [1]	37.5	40.0	38.9	39.9	58.7	46.1	35.2	33.8
	IAP [1]	27.1	24.0	17.4	19.4	46.9	35.9	22.4	36.6
	CONSE [28]	36.7	34.3	44.2	38.8	67.9	44.5	25.9	26.9
	CMT [39]	37.3	34.6	41.9	39.9	66.3	37.9	26.9	28.0
	SSE [53]	43.7	43.9	54.5	51.5	67.5	61.0	31.1	34.0
	LATEM [44]	49.4	49.3	56.9	55.3	68.7	55.8	34.5	35.2
	ALE [2]	53.2	54.9	59.1	58.1	80.3	62.5	30.9	**39.7**
	DEVISE [13]	53.2	52.0	57.5	56.5	68.6	59.7	35.4	**39.8**
	SJE [3]	55.3	53.9	57.1	53.7	69.5	61.9	32.0	32.9
	ESZSL [37]	55.1	53.9	57.3	54.5	75.6	58.6	34.4	38.3
	SYNC [5]	54.1	55.6	59.1	56.3	71.2	46.6	39.7	23.9
	SAE [20]	33.4	33.3	42.4	40.3	80.7	54.1	8.3	8.3
	PSR [4]	–	56.0	–	**61.4**	–	63.8	–	38.4
‡	SCoRe[27]*	59.5	–	–	–	–	–	–	–
	QFSL$^-$ [20]	58.5	58.8	58.9	56.2	72.6	63.5	–	–
	DEM [52]*	–	51.7	–	40.3	–	**67.1**	–	35.0
	LDF [22]*	67.1	–	–	–	83.4	–	–	–
	SP-AEN [8]	–	55.4	–	59.2	–	–	–	24.1
	RN [50]	–	55.6	–	–	–	64.2	–	–
♮	UDA [19]	39.5	–	–	–	–	–	–	–
	TMV [14]	51.2	–	**61.4**	–	–	–	–	–
	SMS [15]	59.2	–	60.5	–	–	–		
	QFSL [42]	**69.7**	**72.1**	**61.7**	58.3	84.8	**79.7**	–	–
♭	**ARE**	**70.2**	**72.5**	60.8	59.0	**86.3**	66.9	**44.0**	35.5
	ACSE	69.0	71.5	**61.5**	**59.7**	**86.5**	65.2	**43.5**	38.7
	AREN	**70.7**	**71.8**	**61.7**	**60.6**	**86.7**	**67.9**	**44.1**	**39.2**

† : Inductive & ResNet101 feature based methods.
‡ : Inductive & End-to-end trainable CNN based methods.
♮ : Transductive.
♭ : Proposed & Inductive & ResNet101 as backbone, end-to-end trainable.
* : Indicates that ResNet101 is not used as backbone net.

4.4. Evaluations in GZSL Setting

To evaluate the GZSL, the searched label space for a given test image is enlarged to include both unseen ($\mathcal{Y}^U$) and the seen classes ($\mathcal{Y}^S$). Under the PS setting [46], the test images come from both seen and unseen classes. To begin with, we present the evaluation protocol for GZSL. Suppose that the ACA for the testing samples from the unseen classes is $\text{ACA}_{\mathcal{Y}^U}$, and meanwhile, $\text{ACA}_{\mathcal{Y}^S}$ for testing samples

Table 2. GZSL results (in %) in PS setting; our methods and most of the compared methods are taking ResNet101 as the backbone net for fair comparisons. **ts** = ACA on $\mathcal{Y}^U$, **tr**=ACA on $\mathcal{Y}^S$, and **H** = harmonic mean. The best number is marked in **bold**.

	Method	**CUB**			**SUN**			**AWA2**			**APY**		
		ts	**tr**	**H**	**ts**	**tr**	**H**	**ts**	**tr**	**H**	**ts**	**tr**	**H**
†	DAP [1]	1.7	67.9	3.3	4.2	25.1	7.2	0.0	84.7	0.0	4.8	78.3	9.0
	IAP [1]	0.2	72.8	0.4	1.0	37.8	1.8	0.9	87.6	1.8	5.7	65.6	10.4
	CONSE [28]	1.6	72.2	3.1	6.8	39.9	11.6	0.5	90.6	1.0	0.0	**91.2**	0.0
	CMT [39]	7.2	49.8	12.6	8.1	21.8	11.8	0.5	90.0	1.0	1.4	85.2	2.8
	SSE [53]	8.5	46.9	14.4	2.1	36.4	4.0	8.1	82.5	14.8	0.2	78.9	0.4
	LATEM [44]	15.2	57.3	24.0	14.7	28.8	19.5	11.5	77.3	20.0	0.1	73.0	0.2
	ALE [2]	23.7	62.8	34.4	21.8	33.1	26.3	14.0	81.8	23.9	4.6	73.7	8.7
	DEVISE [13]	23.8	53.0	32.8	16.9	27.4	20.9	17.1	74.7	27.8	4.9	76.9	9.2
	SJE [3]	23.5	59.2	33.6	14.7	30.5	19.8	8.0	73.9	14.4	3.7	55.7	6.9
	ESZSL [37]	12.6	63.8	21.0	11.0	27.9	15.8	5.9	77.8	11.0	2.4	70.1	4.6
	SYNC [5]	11.5	70.9	19.8	7.9	**43.3**	13.4	10.0	90.5	18.0	7.4	66.3	13.3
	SAE [20]	7.8	54.0	13.6	8.8	18.0	11.8	1.1	82.2	2.2	0.4	80.9	0.9
	PSR [4]	24.6	54.3	33.9	20.8	37.2	26.7	20.7	73.8	32.3	13.5	51.4	21.4
‡	DEM [52]*	19.6	57.9	29.2	20.5	34.3	25.6	30.5	86.4	45.1	11.1	75.1	19.4
	QFSL [42]	33.3	48.1	39.4	30.9	18.5	23.1	52.1	72.8	60.7	–	–	–
	RN [50]	38.1	61.1	47.0	–	–	–	30.0	93.4	45.3	–	–	–
♭	**ARE**	38.4	76.4	51.2	19.0	29.3	23.1	17.5	**93.2**	29.5	11.6	75.3	20.1
	ACSE	34.6	**80.1**	48.4	15.2	28.8	19.9	18.2	92.9	30.4	9.6	76.5	17.1
	AREN	38.9	78.7	52.1	19.0	38.8	25.5	15.6	92.9	26.7	9.2	76.9	16.4
♭	**ARE+CS**$^\diamondsuit$	61.3	66.6	63.8	**41.7**	35.2	**38.2**	**55.6**	79.8	**65.5**	28.0	53.7	36.8
	ACSE+CS$^\diamondsuit$	61.3	68.4	64.7	36.8	34.9	35.8	53.5	79.2	63.9	**30.8**	50.8	**38.3**
	AREN+CS$^\diamondsuit$	**63.2**	69.0	**66.0**	40.3	32.3	35.9	54.7	79.1	64.7	30.0	47.9	36.9

†: Inductive & ResNet101 feature based methods. ‡: End-to-end trainable CNN based methods. ♭: Proposed & Inductive & ResNet101 as backbone, end-to-end trainable. $*$: Indicates that ResNet101 is not used as backbone. $\diamondsuit$: CS, i.e., Calibrated Stacking [6], means reducing the prediction scores for the seen classes.

from the seen classes. Their Harmonic mean **H** can then be calculated as $\mathbf{H} = \frac{2\times \text{ACA}_{\mathcal{Y}^U} \times \text{ACA}_{\mathcal{Y}^S}}{\text{ACA}_{\mathcal{Y}^U} + \text{ACA}_{\mathcal{Y}^U}}$. To this end, the harmonic mean **H** is taken as the main evaluation criterion for our models under the GZSL setting.

$\text{ACA}_{\mathcal{Y}^U}$ ($\mathbf{ts}$), $\text{ACA}_{\mathcal{Y}^S}$ ($\mathbf{tr}$), and their harmonic mean **H** for the evaluated datasets are listed in Table 2. From Table 2, we can draw the following conclusions: **i)** On CUB, AWA2, and APY datasets, the proposed methods without calibrated stacking (CS) in **H** are comparable to/better than current state-of-the-art methods. **ii)** Our initial results typically achieve a high $\mathbf{tr}$, but a low $\mathbf{ts}$, which indicates that calibrated stacking [6] is needed. As shown in the last three rows, after the CS operation, the **H** mean, $\mathbf{tr}$ and $\mathbf{ts}$ become the best in most cases. **iii)** the overall AREN model, w/o a CS operation, shows a lower performance than the separate ARE and ACSE models, for some datasets. This is likely for the same reasons as in the ZSL model.

4.5. Ablation Study

In the following, the CUB and AWA2 datasets are taken as examples for ablation analysis.

Coefficients in loss function. For the AREN model, there exist two parameters, i.e., λ_1 and λ_2, in Eqn. (11). By varying their values from $\{0.1, 0.5, 1.0, 1.5, 2.0\}$ and fixing other parameters as defaults, we run different models for 10 epochs and produce the ACA maps w.r.t. λ_1 and λ_2 under SS/PS settings for ZSL. The changing tendency of ACA w.r.t. (λ_1, λ_2), overall, is stable and consistent (Fig. 3).

AT coefficient in ARE. To observe the influence of the AT coefficient α for K fixed attentive maps, we conduct experiments varying α from $\{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$, where $\alpha = 0$ indicates ARE without AT, $\alpha = 1.0$ means the strongest AT is added (only the attentive map with largest activate value is preserved, while all other maps are set to zero-elements), and if $\alpha > 1.0$, ARE becomes un-trainable. ZSL results under SS/PS settings are illustrated in Fig. 4. The curves show that improvements in ACA are achieved, which

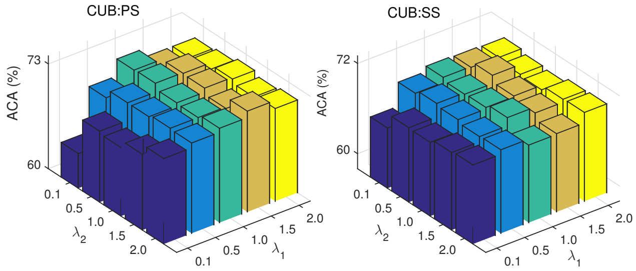


Figure 3. The ACA-(λ_1, λ_2) maps on CUB under SS/PS settings.

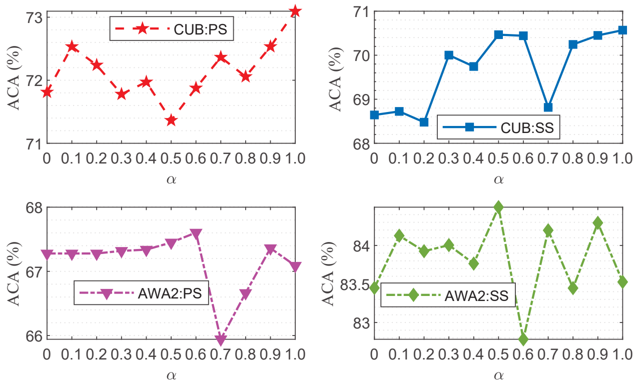


Figure 4. The ACA-α curves under ZSL setting.

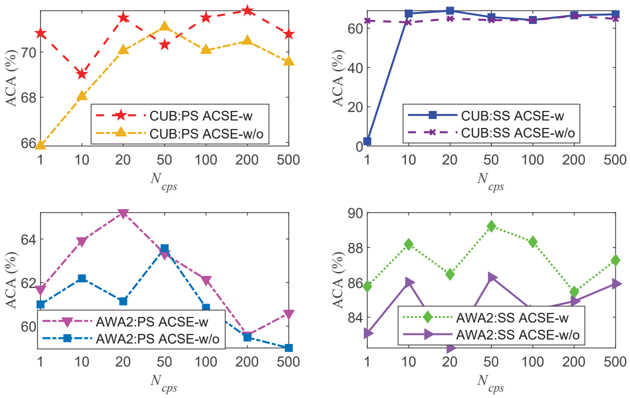


Figure 5. The ACA-N_{cps} curves of ZSL setting w/w/o attention.

well confirms the effectiveness of the AT mechanism.

Channel compression in ACSE. We present the trend of ACA w.r.t. varying values of N_{cps} over a discrete value range of $\{1, 10, 20, 50, 100, 200, 500\}$. ZSL results are shown in Fig. 5. We can see that the ACSE with small values of N_{cps} achieves better ACA results in nearly all cases. In particular, we achieve an accuracy of 89.2% for AWA2:SS, under $N_{cps} = 50$. Only see ACSE-w (ACSE with attention) curves for comparison. **Attention in ACSE.** In Fig. 2, the $\boldsymbol{Z}_{cps}$ from the ACSE and K attentive feature maps from ARE are used collaboratively to generate a second-order vector. We further observe two cases, i.e., 1) the current ACSE with attention (ACSE-w), and 2) the ACSE without attention (ACSE-w/o), i.e., second-order vector is obtained by $\boldsymbol{Z}_{cps} \circledcirc \boldsymbol{Z}_{cps}$. For fair comparisons, we make the dimensions of the final vectors for ACSE-w and ACSE-w/o (almost) the same. To reuse the results in Fig. 5 from ACSE-w, we vary the values of N_{cps} from $\Upsilon = \{46, 144, 203, 320, 453, 640, 1012\}$ for ACSE-w/o attention, thus making the dimensions of the generated vectors from the two cases approximately the same. From Fig. 5, it can be concluded that ACSE-w is consistently better than ACSE-w/o.

Global versus part features: The global model is trained by taking the original ResNet101 with its fully connected layer as the backbone, followed by the projection to semantic space and the same compatibility loss as ours. The ACAs of the global baseline (GB), ARE, and ACSE are listed in Table 3, which shows significant improvements have been made by our models.

4.6. Visualization

The ARE models with PS split are used to visualize what the learned regions look like, and unseen images from AWA2 and CUB are considered (Fig. 7). Based on the above ten m_v values for each image, AT_{max} (Eqn. (5)) is obtained, and T_B is thus acquired by multiplying α with AT_{max}, e.g., for "horse", let $\alpha = 0.8$, only six attention maps (in red rectangle boxes) are reserved, the discarded four masks are backgrounds w.r.t. the sky. To this end, the AT mechanism is automatically suppressing the backgroud noises. Moreover, global objects and semantic parts are addressed by these learned masks, e.g., the 4-th mask of "mallard" corresponds to "whole body", and the 1-st mask of "kingbird" fucuses on "head". ARE/ACSE models of PS splits are further used to visualize the distribution of the unseen test images on AWA2 by t-SNE visualization [55] (Fig. 6).

Table 3. ZSL results (ACA, in %) of global/part features.

Method	CUB		AWA2	
	SS	PS	SS	PS
GB	60.2	62.7	81.7	60.3
ARE	70.2	72.5	86.3	66.9
ACSE	69.0	71.5	86.5	65.2

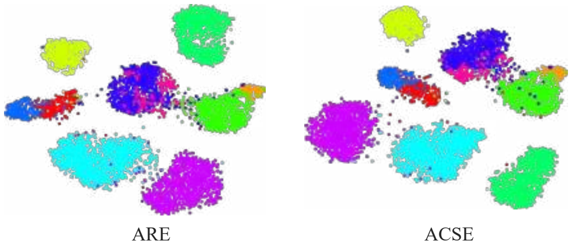
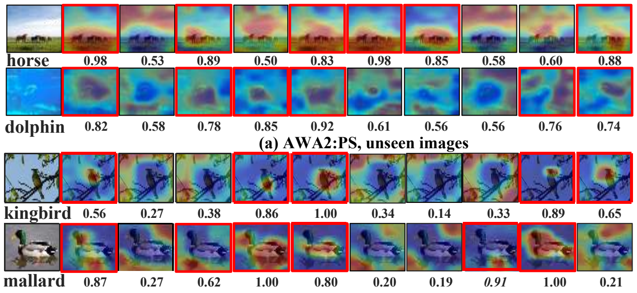


Figure 6. t-SNE visualization of unseen class images on AWA2.



(a) AWA2:PS, unseen images

(b) CUB:PS, unseen images

Figure 7. In both (a) and (b), (attention) masks in red rectangle are selected by AT mechanism. For each row, the first one is the input image, the left ones are its ten attentive feature masks, the number below is the maximum value m_v within the mask.

5. Conclusions

An attentive region embedding network (AREN) is proposed for solving the challenging ZSL/GZSL task, which consists of two branches, i.e., the upper stream attentive region embedding (ARE) and the bottom stream (attention guided) compressed second-order embedding (ACSE). Both ARE and ACSE are embedded into the semantic space, where ZSL/GZSL is conducted through nearest neighbor matching. An adaptive thresholding (AT) is also incorporated into the ARE. Actually, the AT can also be applied to many other general tasks which requires an attention mechanism, such as visual question answering. Integrating ARE and ACSE together leads to the AREN model, which has achieved some new state-of-the-art results for both ZSL and GZSL, on the standard benchmarks.

References

[1] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for attribute-based classification. In *CVPR*, 2013. 1, 2, 4, 5, 6, 7

[2] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for image classification. In *TPAMI*, 2016. 6, 7

[3] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In *CVPR*, 2015. 2, 4, 5, 6, 7

[4] Y. Annadani and S. Biswas. Preserving semantic relations for zero-shot learning. In *CVPR*, 2018. 2, 6, 7

[5] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha. Synthesized classifiers for zero-shot learning. In *CVPR*, 2016. 6, 7

[6] W.-L. Chao, S. Changpinyo, B. Gong, and F. Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *ECCV*, 2016. 7

[7] Y. Long, L. Liu, L. Shao, F. Shen, G. Ding, and J. Han. From zero-shot learning to conventional supervised classification: Unseen visual data synthesis. In *CVPR*, 2017. 2

[8] L. Chen, H. Zhang, J. Xiao, W. Liu, and S.-F. Chang. Zero-shot visual recognition using semantics-preserving adversarial embedding network. In *CVPR*, 2018. 2, 6

[9] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. Attention to scale: Scale-aware semantic image segmentation. In *CVPR*, 2016. 3

[10] M. Elhoseiny, Y. Zhu, H. Zhang, and A. M. Elgammal. Link the head to the" beak": Zero shot learning from noisy text description at part precision. In *CVPR*, 2017. 2

[11] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 2, 5

[12] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. In *TPAMI*, 2010. 2

[13] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In *NeurIPS*, 2013. 2, 5, 6, 7

[14] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong. Transductive multi-view zero-shot learning. In *TPAMI*, 2015. 3, 6

[15] Y. Guo, G. Ding, X. Jin, and J. Wang. Transductive zero-shot recognition via shared model space learning. In *AAAI*, 2016. 6

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 6

[17] D. Jayaraman and K. Grauman. Zero-shot recognition with unreliable attributes. In *NeurIPS*, 2014. 2

[18] N. Karessli, Z. Akata, B. Schiele, A. Bulling, et al. Gaze embeddings for zero-shot image classification. In *CVPR*, 2017. 2

[19] E. Kodirov, T. Xiang, Z. Fu, and S. Gong. Unsupervised domain adaptation for zero-shot learning. In *ICCV*, 2015. 2, 6

[20] E. Kodirov, T. Xiang, and S. Gong. Semantic autoencoder for zero-shot learning. In *CVPR*, 2017. 2, 6, 7

[21] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009. 1, 2

[22] Y. Li, J. Zhang, J. Zhang, and K. Huang. Discriminative learning of latent features for zero-shot recognition. In *CVPR*, 2018. 2, 3, 4, 5, 6

[23] G. Xie, X. Zhang, S. Yan, and C. Liu. SDE: A novel selective, discriminative and equalizing feature representation for visual recognition. In *IJCV*, 2017. 1

[24] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear cnn models for fine-grained visual recognition. In *ICCV*, 2015. 4, 5

[25] G.-S. Xie, X.-Y. Zhang, W. Yang, M. Xu, S. Yan, and C.-L. Liu. LG-CNN: From local parts to global discrimination for fine-grained recognition. In *PR*, 2017. 2

[26] Y. Long, L. Liu, F. Shen, L. Shao, and X. Li. Zero-shot learning using synthesised unseen visual data with diffusion regularisation. In *TPAMI*, 2017. 2

[27] P. Morgado and N. Vasconcelos. Semantically consistent regularization for zero-shot recognition. In *CVPR*, 2017. 2, 3, 4, 5, 6

[28] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. In *arXiv:1312.5650*, 2013. 6, 7

[29] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell. Zero-shot learning with semantic output codes. In *NeurIPS*, 2009. 1

[30] G. Patterson and J. Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, 2012. 5

[31] Y. Yao, F. Shen, J. Zhang, L. Liu, Z. Tang, and L. Shao. Discovering and distinguishing multiple visual senses for web learning. In *TMM*, 2018. 1

[32] Z. Zhang, Y. Xu, L. Shao, and J. Yang. Discriminative block-diagonal representation learning for image recognition. In *TNNLS*, 2018. 1

[33] Z. Zhang, L. Shao, Y. Xu, L. Liu, and J. Yang. Marginal representation learning with graph structure self-adaptation. In *TNNLS*, 2017. 1

[34] J. Qin, L. Liu, L. Shao, F. Shen, B. Ni, J. Chen, and Y. Wang. Zero-shot action recognition with error-correcting output codes. scene attributes. In *CVPR*, 2017. 2

[35] R. Qiao, L. Liu, C. Shen, and A. van den Hengel. Less is more: zero-shot learning from online textual documents with noise suppression. In *CVPR*, 2016. 2

[36] S. Reed, Z. Akata, H. Lee, and B. Schiele. Learning deep representations of fine-grained visual descriptions. In *CVPR*, 2016. 2

[37] B. Romera-Paredes and P. Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, 2015. 2, 5, 6, 7

[38] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. In *IJCV*, 2015. 6

[39] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In *NeurIPS*, 2013. 2, 6, 7

[40] J. Qin, Y. Wang, L. Liu, J. Chen, and L. Shao. Beyond semantic attributes: Discrete latent attributes learning for zero-shot recognition. In *PRL*, 2016. 2

[41] H. Jiang, R. Wang, S. Shan, and X. Chen. Learning class prototypes via structure alignment for zero-shot recognition. In *ECCV*, 2018. 2

[42] J. Song, C. Shen, Y. Yang, Y. Liu, and M. Song. Transductive unbiased embedding for zero-shot learning. In *CVPR*, 2018. 2, 3, 4, 5, 6, 7

[43] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. In *Technical report*, 2011. 5

[44] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele. Latent embeddings for zero-shot classification. In *CVPR*, 2016. 2, 6, 7

[45] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata. Feature generating networks for zero-shot learning. In *CVPR*, 2018. 2

[46] Y. Xian, B. Schiele, and Z. Akata. Zero-shot learning-the good, the bad and the ugly. In *CVPR*, 2017. 1, 2, 3, 5, 6

[47] G.-S. Xie, X.-Y. Zhang, X. Shu, S. Yan, and C.-L. Liu. Task-driven feature pooling for image classification. In *ICCV*, 2015. 1

[48] H. Xu and K. Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *ECCV*, 2016. 3

[49] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 2, 4

[50] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018. 1, 2, 6, 7

[51] M. Ye and Y. Guo. Zero-shot classification with discriminative semantic representation learning. In *CVPR*, 2017. 2

[52] L. Zhang, T. Xiang, S. Gong, et al. Learning a deep embedding model for zero-shot learning. In *CVPR*, 2017. 1, 2, 3, 6, 7

[53] Z. Zhang and V. Saligrama. Zero-shot learning via semantic similarity embedding. In *ICCV*, 2015. 2, 6, 7

[54] Z. Zhang and V. Saligrama. Zero-shot learning via joint latent similarity embedding. In *CVPR*, 2016. 2

[55] L. V. D. Maaten, and G. Hinton. Visualizing data using t-SNE. In *JMLR*, 2008.

8