Scientific Research Publishing

Open Journal of Applied Sciences, 2022, 12, 317-335
https://www.scirp.org/journal/ojapps
ISSN Online: 2165-3925
ISSN Print: 2165-3917

Explanatory Multi-Scale Adversarial Semantic Embedding Space Learning for Zero-Shot Recognition

Huiting Li

College of Information Science and Technology, Jinan University, Guangzhou, China
Email: lihuting7@gmail.com

How to cite this paper: Li, H.T. (2022) Explanatory Multi-Scale Adversarial Semantic Embedding Space Learning for Zero-Shot Recognition. *Open Journal of Applied Sciences*, **12**, 317-335.
https://doi.org/10.4236/ojapps.2022.123023

Received: January 15, 2022
Accepted: March 19, 2022
Published: March 22, 2022

Abstract

The goal of zero-shot recognition is to classify classes it has never seen before, which needs to build a bridge between seen and unseen classes through semantic embedding space. Therefore, semantic embedding space learning plays an important role in zero-shot recognition. Among existing works, semantic embedding space is mainly taken by user-defined attribute vectors. However, the discriminative information included in the user-defined attribute vector is limited. In this paper, we propose to learn an extra latent attribute space automatically to produce a more generalized and discriminative semantic embedded space. To prevent the bias problem, both user-defined attribute vector and latent attribute space are optimized by adversarial learning with auto-encoders. We also propose to reconstruct semantic patterns produced by explanatory graphs, which can make semantic embedding space more sensitive to usefully semantic information and less sensitive to useless information. The proposed method is evaluated on the AwA2 and CUB dataset. These results show that our proposed method achieves superior performance.

Keywords

Zero-Shot Recognition, Semantic Embedding Space, Adversarial Learning, Explanatory Graph

1. Introduction

Zero-shot learning is one of the research focuses in the field of transfer learning. Unlike the traditional image classification, it needs to classify classes that have unseen before. Therefore, in the task of Zero-shot learning, the classes in the training set and the classes in the evaluation set are disjoint. Its superiority is that it can solve the problem of insufficient training data for every possible class.

For example, if some scene images of 'kitchen' are collected, some object images of 'bowl' are meanwhile obtained. Once the classifier for 'kitchen' is trained, this classifier can also be used to recognize 'bowl'. To describe the logic relationships between seen and unseen classes, a semantic embedding space should be defined which relies on several visual concepts [1] [2], such as user-defined attributes and Word2vec. Map images in seen and unseen classes into this semantic embedding space. The mapping from semantic embedding space to class labels is pre-defined. In this way, unseen classes can be classified without training data.

To learn the mappings between seen and unseen classes, existing methods can be classified into two categories: recognition using independent semantics (RIS) [3] [4] [5] and recognition using semantic embeddings (RULE) [6] [7] [8]. RIS learns an independent classifier for each semantic concept (one attribute like 'has a tail' or 'green wings') in semantic embedding space. Because of its simplicity, RIS is widely used in attribute recognition. RULE learns a bilinear compatibility function between semantic concepts and class labels. By learning all the semantics at the same time, RULE leverages the advantages of the dependencies between semantics concepts. To exploit the complementarity between RIS and RULE, Morgado and Vasconcelos in [9] reduced them to several constraints in CNN architectures, where RIS learns each independent CNN for each semantic concept, and RULE learns a single CNN for all semantic concepts. The balance between RIS and RULE is guaranteed by hyperparameters in this work.

In [9], RIS and RULE are mapped into regularization constraints. They attempted to optimize a single semantic embedding space by RIS and RULE, which limits the dimensions of the solution space. In this paper, we design two semantic embedding spaces, one optimized by RIS and the other optimized by RULE. This design extends the solution space to a higher dimension. It can optimize user-defined attribute space by RIS, and optimize the discriminative semantic embedding space by RULE. The advantages of RIS and RULE can be better combined by this design.

The goal of both RIS and RULE is to learn the mappings from visual feature space to semantic embedding space. Both two methods face the strong bias problem in which unseen classes tend to be classified as one of the seen classes [10]. To improve the generalization capability of semantic embedding space, existing methods can be divided into three groups:

Knowledge graph. This kind of methods models mappings between different categories by knowledge graphs [11] [12], by which the learned manifold of semantic embedding space is aligned to the knowledge graph.

Embedding reconstruction. This kind of method first reconstructs input images by training an encoder/decoder network, which can encode more generalized features. The output of the encoder is taken as regularization of semantic embedding space in the classification network [7] [13].

Latent attributes. This kind of method learns latent attributes and user-defined attributes jointly [9] [14], which improves the discriminative ability

of semantic embedding space. In [14], optimizing the extra latent attributes space by class labels is based on triplet loss, since this space has no ground-truth labels.

The latent attributes in group 3 are more descriptive and flexible than the user-defined attributes, while the user-defined attributes are more transferable than the latent attributes. To take the advantage of both, we propose to learn the user-defined and the latent attributes jointly, and then concatenate them into a unified semantic embedding space. However, both user-defined and latent attributes suffer from the bias problem, which tends to learn attributes over-fitted to training set. To reduce the bias problem and improve the generalization capacity, we propose to take the user-defined and the latent attributes regularized by the output of the auto-encoder. To achieve this goal, we input both the semantic embedding space and the output of the encoder into a discriminator, which tries to classify two kinds of inputs while a generator tries to confuse the discriminator. This process is always formulated as an adversarial learning objective. The method in [13] has proved the superiority of adversarial learning. In this paper, we also use adversarial learning loss to constrain the learning processes of both user-defined attributes and latent attributes. As presented in [9], RIS and RULE are two kinds of optimization loss functions. In deep learning framework, RIS aims to learn several independent CNNs, one CNN for per semantic attribute, which makes the output of the classification network to be completely the same with ground-truth semantic embedding vectors. RULE aims to learn a single CNN for all semantic attributes automatically. The user-defined attributes can be optimized by RIS and the latent attributes can be optimized by RULE. The user-defined attributes are not exhaustive, especially when two or more categories share too many semantic attributes. The latent attributes of the training set are not discriminative enough on evaluation set, especially when the training set is quite misaligned with the evaluation set. To take the advantage of two kinds of attributes, [9] proposed to constrain optimization objectives of the user-defined attributes and the latent attributes as a single loss function, and optimization of the user-defined attributes is considered as a regularization constraint. [14] proposed to learn extra latent attributes optimized by a triplet loss. The learned latent attributes are connected with user-defined attributes to form a semantic embedding space. The method in [14] achieved much better performance than [9] because [14] took advantage of much higher attribute dimensions. The entire framework is jointly trained. The framework of the method we propose is shown in **Figure 1**.

In group 2, the semantic embedding space is constrained to reconstruct entire original images. The constraint can easily introduce a lot of noise and cause serious bias problems in the training set. In group 3, the learned attributes are only discriminative on training set, and some attributes on zero-shot classes are ignored. If taking the advantages of groups 2 and 3 are combined, learned semantic embedding spaces from training set will be transferred to zero-shot sets much

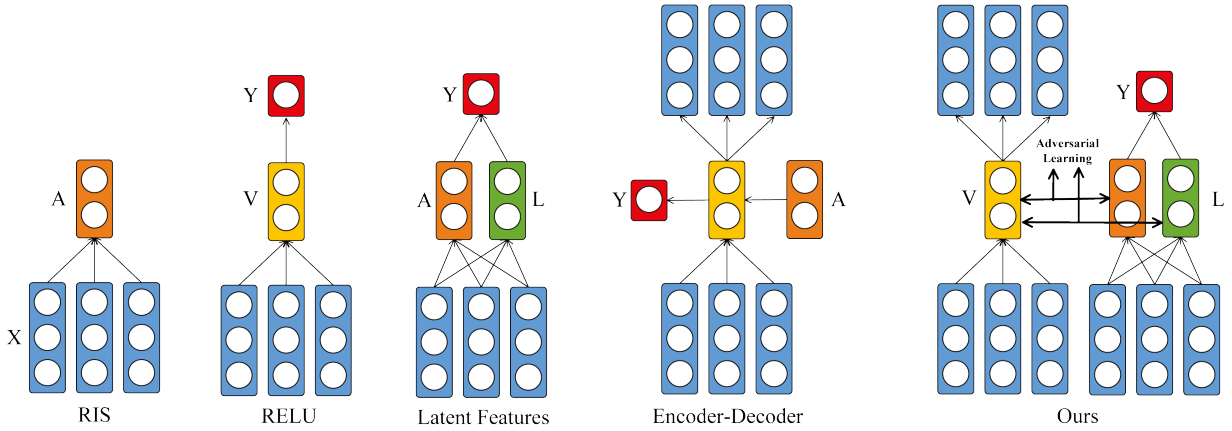

Figure 1. Summarization of existing zero-shot learning frameworks, where 'X' represents input features, 'A' represents user-defined attributes, 'L' represents latent attributes, 'Y' represents class labels and 'V' represents autoencoders' output.

better. Motivated by this, we propose to reconstruct semantic patterns of original images in an organized way to constraint semantic embedding. The semantic patterns are discovered and organized by explanatory graphs proposed in [15]. Zhang *et al.* in [15] revealed the knowledge structures hidden in CNNs. The core idea of this method is to disentangle part patterns of high-level layers to form an explanatory graph. Each graph node corresponds to each part pattern, and each edge represents the cooperation and spatial relationships between part patterns. This explanatory graph can organize chaotic information hidden in trained CNNs into structural features.

Traditional reconstruction methods are more sensitive to variances like rotation, flipping, and so on. When the images have large variances, their linear feature vectors will have large variances, but cooperation and spatial relationships can be still stable. These stable relationships can be learned automatically by explanatory graphs and can be used to bridge a semantic embedding space between seen classes and unseen classes. In this way, the generalized capacity of semantic embedding space is improved to a higher level.

We summarize our contributions as follows:

- Our proposed method optimizes user-defined attributes and latent attributes jointly, which can produce more descriptive and discriminative semantic embedding space.
- Our proposed method improves the generalization capacity of semantic embedding space by adversarial learning with autoencoders.
- Our proposed method extracts interpretable and explanatory semantic patterns based on explanatory graphs in an organized way, which can reduce the influence of rotation, zoom, background noise, and so on.
- Our proposed method integrates the multi-scale explanatory reconstruction network into the classification network and optimizes the overall framework in an alternative way, which combines advantages of RIS, RULE and multi-scale reconstruction at the same time.

2. Related Works

Existing zero-shot learning (ZSL) tasks can be mainly divided into three types: conventional ZSL, Generalised ZSL (GZSL), and Transductive ZSL (TZSL), which are summarized in Figure 2. In the task of conventional ZSL, the training set consists of seen classes and the test set is assumed only from unseen classes. In the task of GZSL, the training set and the test set are the same as conventional ZSL, but GZSL requires recognizing both seen and unseen classes at the same time. In the task of TZSL, the unlabeled images in the test set are available in the training process, which reduces the challenge of zero-shot recognition significantly. Obviously, GZSL is the most difficult task among the three types.

Existing work uses two strategies to obtain semantic embedded spaces, one is recognition using independent semantics (RIS) and the other is recognition using latent embeddings (RULE):

RIS. This kind of method aims to learn an independent classifier for per semantic attribute [3] [4] [5] [16]. RIS provides supervision for per semantic attributes but cannot model the dependencies between different semantic attributes, which is hard to guarantee reliable mappings from visual feature vector to semantic embedding space.

RULE. Such methods directly map images to their classes label [6] [7] [8] [17] [18] [19] [20] [21] by compatibility functions, which learn all semantic features at the same time. Palatucci *et al.* in [17] learned a linear mapping relationship between fMRI-based image space and the semantic space. In [18], a skip-gram model trained on Wikipedia articles is used to produce label features [22]. [19] weighted a group of training label embeddings by using the probabilities of the classifier. [20] learned to justify that if an input image belongs to seen classes in the semantic word spaces by using an outlier detector. SJE [7] [21] and ALE [6] combined multiclass and weighted estimation ranking loss to learn a bilinear compatibility function. Xian *et al.* in [8] proposed to construct a nonlinear compatibility learning model by learning several linear models.

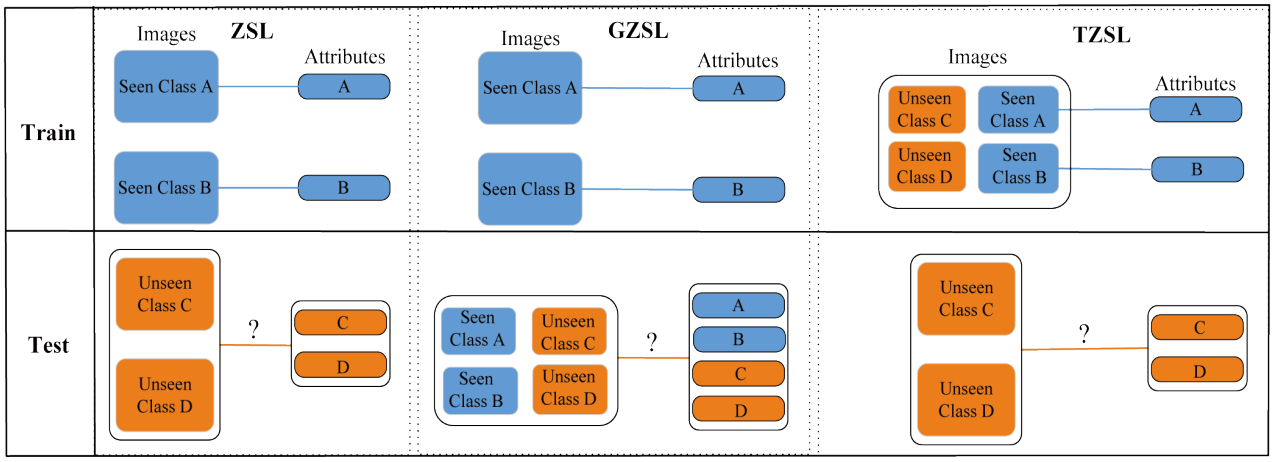

Figure 2. Comparison of existing three types of zero-shot learning tasks.

Although RULE methods model dependencies between different semantic attributes, they leave several semantic combinations unconstrained. To solve this problem, Chen *et al.* in [23] used a conditional random field to model dependencies between semantic attributes to improve independent classifiers. Wang and Ji in [5] unified attribute extraction and object classification in a single probabilistic model. Morgado and Vasconcelos in [9] leveraged the advantages of both RIS and RULE to design a CNN framework and proposed using semantics as constraints for recognition. Chen *et al.* in [13] proposed to reconstruction network to regularize the semantic embedding space of classification network, which makes embedding space more informative. Li *et al.* in [14] learned latent discriminative attributes to supplement unconstrained semantic space covered by user-defined attributes. Zhang and Koniusz in [24] proposed to use nonlinear kernels to learn relationships between visual features and attributes. Xian *et al.* in [25] proposed to generate features based on attributes of unseen categories and the generated features were added to the training set. We summarize existing ZSL frameworks of different tasks in **Figure 1**.

3. The Proposed Approach

In this section, we first perform adversarial learning between user-defined attributes and autoencoders and then perform adversarial learning between latent attributes and autoencoders. We extract the semantic patterns based on the explanatory graphs after training the overall architecture based on the original images. These semantic patterns are used to train the same architecture. The semantic embedded spaces of different semantic patterns are connected to form a final semantic embedded space for zero-shot recognition. The overall framework of the proposed method is illustrated in **Figure 3**.

3.1. Adversarial Semantic Embedding Space Learning

We construct a unified framework including the classification network and the reconstruction network. The classification network aims to optimize user-defined attributes and latent attributes jointly in adversarial manners. The reconstruction network takes autoencoder as the backbone.

1) Classification network: The classification network aims to map visual features space to semantic embedding space. In our method, we propose a novel classification optimization objective, which integrates the advantages of user-defined attributes and latent attributes in augmented space. The augmented space contains two vectors, one is to utilize RIS to learn user-defined attributes, and the other is to utilize RULE to learn latent attributes.

Given a training dataset $D=\left(x^{(i)}, s^{(i)}, y^{(i)}\right)_{i=1}^{N}$, where $s^{(i)}=\left(s_1^{(i)}, \cdots, s_Q^{(i)}\right)$ represent Q attributes, and visual feature extractor $\theta(x;\Theta)$ of parameters Θ, and t_k is independent classifier of attribute k, then user-defined attributes (UA) are optimized by minimizing:

$$L_{ua}=\sum_i \sum_k L_b\left(\sigma\left(t_k^{\mathrm{T}}\theta(x;\Theta)\right), s_k^{(i)}\right) \tag{1}$$

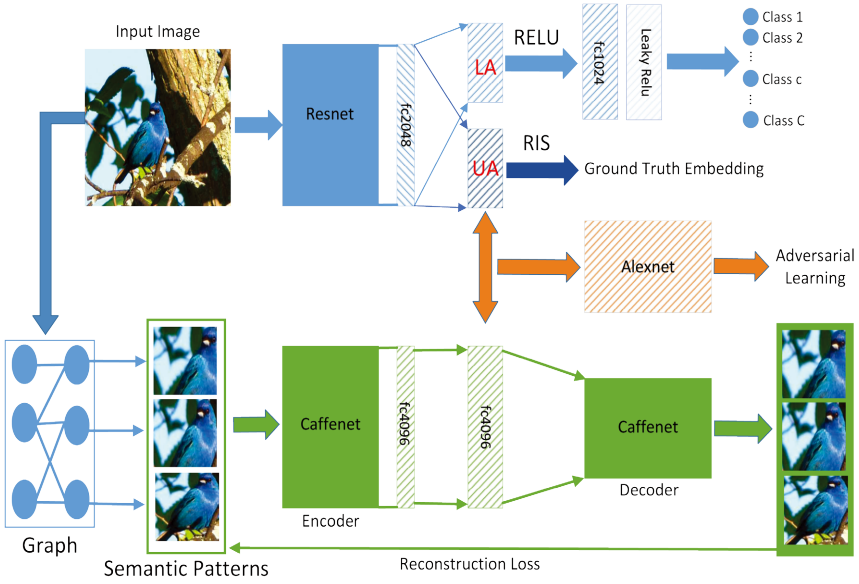

Figure 3. The framework of the proposed method. Color 'blue' represents the classification network, color 'green' presents the reconstruction network, and color 'orange' represents adversarial learning component. Components filled with oblique lines represent parameters that need to be optimized.

where $\sigma(\cdot)$ is the sigmoid function and L_b is the cross-entropy loss function which is $L_b(v,y) = -y\log(v) - (1-y)\log(1-v)$.

Unlike user-defined attributes, latent attributes are learned automatically. Supposing the parameter matrix T maps visual features $\theta(x;\Theta)$ to latent attributes, the parameter matrix Φ maps latent attributes to class labels, then latent attributes (LA) are optimized by minimizing:

$$L_{la} = \sum_i L\left(\Phi^{\mathrm{T}} T^{\mathrm{T}} \theta(x;\Theta), y^{(i)}\right) \tag{2}$$

where L represents softmax loss function, and $\Phi = \left[\phi(1),\cdots,\phi(C)\right] \in \mathbb{R}^{Q\times C}$ is calculated as:

$$\phi_k(y) = \begin{cases} 1 & \text{if class } c \text{ contains attribute } k, \\ -1 & \text{if class } c \text{ does not contain attribute } k. \end{cases} \tag{3}$$

In our method, the loss weight of latent attributes is equal to user-defined attributes.

2) Reconstruction network: The reconstruction network includes the encoder network and the decoder network, and it is constructed based on autoencoders. Supposing the encoder is $E(\cdot)$ and the decoder is $G(\cdot)$, then the reconstruction objective function [13] can be expressed as:

$$L_{rec} = L_{feat} + L_{pixel} \tag{4}$$

where $L_{feat} = \left\| F\left(G\left(E(x)\right)\right) - F(x) \right\|_2^2$ is high-level feature loss and $L_{pixel} = \left\| G\left(E(x)\right) - x \right\|_2^2$ is pixel-wise loss. And the high-level feature extractor is $F(\cdot)$. We take the output of conv5 layer of AlexNet [26] as the feature extractor,

which is suggested by [13].

3) Adversarial learning: The reconstruction network tries to regularize the semantic embedding space generated by the classification network to improve the generalization ability. To achieve this, the semantic embedding space should search for a solution on the manifold of encoded vectors produced by the reconstructed network. This goal can be expressed by the adversarial learning loss function. Similar to the loss function of GAN [27], the adversarial learning loss function can be expressed as:

$$L_{adv} = \mathbb{E}_x\left(\log D\left(E\left(x\right)\right)\right) + \mathbb{E}_{x'}\left(\log\left[1 - D\left(C\left(x'\right)\right)\right]\right) \tag{5}$$

where $E(\cdot)$ denotes the encoder in the reconstruction network, $C(\cdot)$ denotes the semantic embedding space output by the classification network, x and x' denote input features of the reconstruction network and the classification network. The classification network aims to minimize L_{adv} while the discriminator D aims to maximize L_{adv}. In the process of adversarial learning, the manifold of the semantic embedding space becomes closer to the manifold of the reconstructed network output.

4) Zero-shot prediction: We train the overall framework with the full objective function $L = L_{ua} + L_{la} + L_{rec} + L_{adv}$. After training, during the prediction process, for a test image x, assuming that the user-defined attribute vector is $\varphi_{ua}(x)$, the predicted labels can be inferred as follows:

$$\hat{y} = \arg\max_{c \in \mathcal{Y}_U} \left\langle \varphi_{ua}\left(x\right), \mathbf{s}_{ua}^{c} \right\rangle \tag{6}$$

where $\mathbf{s}_{ua}^{c}$ denotes the ground-truth user-defined attributes of unseen class c, $\mathcal{Y}_U$ denotes the label set of unseen classes and $\hat{y}$ denotes the predicted label.

In the latent attributes space, the ground-truth prototypes of unseen classes are unknown. Therefore, we need to estimate the prototypes of unseen classes. To achieve this, we first calculate prototypes of seen classes as $\overline{\varphi_{la}^{c}} = \frac{1}{N}\sum_{i}\varphi_{la}\left(x\right)$, where x is the sample of class c, $\varphi_{la}(\cdot)$ denotes latent attributes extractor for sample x, and N denotes the number of samples of class c. Then, we construct the relationships between unseen class u and seen class in the user-defined attributes space. These relationships can be modeled based on regression problem:

$$\beta_c^u = \arg\min \left\| \mathbf{s}^u - \sum \beta_c^u \mathbf{s}^c \right\|_2^2 + \lambda \left\| \beta_c^u \right\|_2^2, c \in \mathcal{Y}_S \tag{7}$$

The relationship between unseen class u and seen classes can be constructed by solving this regression problem. The prototype of class u in the latent attributes space can be calculated based on the constructed relationship as:

$$\overline{\varphi_{la}^{u}} = \sum \beta_c^u \overline{\varphi_{la}^{c}}, c \in \mathcal{Y}_S \tag{8}$$

After calculating the prototypes of unseen classes, given UA prototype $\mathbf{s}_{ua}^{u}$ of unseen class u, the corresponding LA prototype $\overline{\mathbf{s}_{la}^{u}}$ can be obtained. Then the predicted label in the latent attributes space can be inferred as follows:.

$$\hat{y} = \arg\max_{u \in \mathcal{Y}_{ll}} \left\langle \varphi_{la}(x), \overline{\mathbf{s}_{la}^{u}} \right\rangle \tag{9}$$

To combine user-defined attributes and latent attributes, the predicted label in user-defined attributes space and latent attributes space can be inferred as follows:

$$\hat{y} = \arg\max_{c \in \mathcal{Y}_{ll}} \left\langle \left[\varphi_{la}(x); \varphi_{ua}(x)\right], \left[\mathbf{s}_{la}^{c}; \mathbf{s}_{ua}^{c}\right] \right\rangle \tag{10}$$

To combine different semantic scales, the concatenated multi-scale user-defined attributes space is $\varphi_{mulua}(x) = \left[\varphi_{ua}^{s_1}(x); \varphi_{ua}^{s_2}(x)\right]$, and multi-scale latent attributes space is $\mathbf{s}_{mulla} = \left[\mathbf{s}_{la}^{s_1}; \mathbf{s}_{la}^{s_2}\right]$, then the predicted label can be inferred as follows:

$$\hat{y} = \arg\max_{c \in \mathcal{Y}_{ll}} \left\langle \varphi_{mulua}(x), \mathbf{s}_{mulla}^{c} \right\rangle \tag{11}$$

where s_1 and s_2 denote different input scales.

3.2. Explanatory Multi-Scale Semantic Patterns

Given a series of images and a pre-trained CNN model, the explanatory graph of these images can be constructed. As shown in **Figure 4**, the explanatory graph can grasp the comprehensive semantic information of deep networks. In deep CNNs, higher layers usually represent high-level semantic patterns, such as object patterns, while lower layers describe simple shapes or edges. Therefore, the patterns of higher layers can grasp the main semantic information and filter out noises, and the patterns of lower layers can be considered as components of higher layers.

We summarize the learning process of explanatory graphs in [15]. The explanatory graph should be constructed layer by layer from top to bottom. Let G define the explanatory graph to be constructed. Assuming that the Lth layer convolutional feature map of input image I is C_L^I, the graph node set V in position R_L^I will be inferred. The graph nodes should have two constrains: one is that graph nodes should be well consistent with feature maps C_L^I; the other one is that graph nodes should keep consistent spatial relationship with upper layers R_{L+1}^I. These two constraints can be expressed by parameter θ_L. Then the objective function for learning Lth layer graph node set V is:

$$\arg\min_{\theta_L} \Pi_{I \in \mathbf{I}} P\left(X_L^I \mid R_{L+1}^I, \theta_L\right) \tag{12}$$

The parameter θ_L is used to infer graph nodes that satisfy these two constraints. The graph node set can be interpreted as a hybrid model:

$$\begin{aligned} P\left(X_L \mid R_{L+1}, \theta_L\right) &= \Pi_{x \in X_L} P\left(p_x \mid R_{L+1}, \theta_L\right)^{F(x)} \\ &= \Pi_{x \in X_L} \left\{ \sum_V P(V) P\left(p_x \mid V, R_{L+1}, \theta_L\right) \right\}^{F(x)} \end{aligned} \tag{13}$$

where $P(V)$ denotes a constant prior probability, and $F(x)$ denotes the neural response of each unit $x \in C_L^I$:

$$F(x) = \beta \cdot \max\left\{f_x, 0\right\} \tag{14}$$

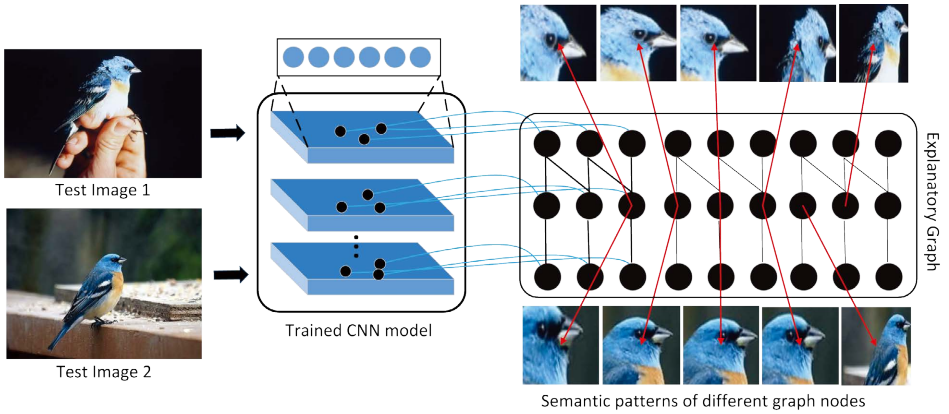

Figure 4. Illustration of semantic patterns extracted from explanatory graphs. We can see that each graph node represents one meaningful semantic. Semantic patterns are much more robust to variations like zooming or rotating than original images.

where f_x denotes the normalized response of unit x and β denotes a constant. Since the relative displacements between V and its connected nodes $\bar{V} \in E_V$ in the upper layer are expected to change little among different images, the compatibility between V and P_x is computed as:

$$P\left(p_x \mid V, R_{L+1}, \theta_L\right) = \alpha \Pi_{\bar{V} \in E_V} P\left(p_x \mid p_{\bar{V}}, \theta_L\right)^{\lambda} \tag{15}$$

where $p_{\bar{V}}$ denotes the inferred positions of $\bar{V}$, λ denotes a normalization constant, and α denotes a constant to guarantee uniform distribution. Supposing the spatial relationship between V and $\bar{V}$ follows a Gaussian distribution, then

$$P\left(p_x \mid p_{\bar{V}}, \theta_L\right) = \mathbf{N}\left(p_x \mid \mu_{\bar{V}\to V}, \sigma_{\bar{V}}^2\right) \tag{16}$$

where $\mu_{\bar{V}\to V} = \mu_V - \mu_{\bar{V}} + P_{\bar{V}}$ denotes the prior position of V based on $\bar{V}$, and $\sigma_{\bar{V}}^2$ denotes the variation. For each graph node V, the parameters μ_V and E_v will be iteratively learned by the Expectation-Maximization (EM) algorithm. The graph is learned layer by layer in a top-down manner [15].

Once the explanatory graph is constructed, corresponding semantic patterns of different graph nodes can be inferred. Each graph node corresponds to the same semantic patterns of different images. During graph building process, positions on convolutional feature maps of different semantic patterns were recorded. Given positions of convolutional feature maps, corresponding positions on original images can be obtained by inverse convolutional operations. Assuming that the position obtained on the input image is (x, y), the expected scale is λ, and the size of original image is o, then the size of the corresponding semantic pattern is calculated as follows:

$$\delta = \min\left(\left[\min(x, y) - 1, \min\left(o - (x, y)\right), (\lambda - 1)/2\right]\right) \tag{17}$$

Given position (x, y) and crop size δ, the semantic pattern I_s is cropped as $I_s = I\left(x - delta : x + delta, y - delta : y + delta\right)$. The obtained semantic patterns can be used to train encoder-decoder to make learned semantic embedding

space generalized and discriminative. Some examples of semantic patterns are shown in Figure 5.

Multiple semantic patterns can be merged to form a larger-scale semantic pattern. If a semantic pattern p is merged from s smaller-scale semantic patterns, we define the scale of semantic pattern p as s. Unlike the scale in [14], which is based on resolution, the scale in our method is based on the number of semantic patterns.

3.3. Implementation Details

The proposed framework should be optimized by four kind of optimizers. The optimization procedure can be concluded as:

1) Building explanatory graphs. We construct an explanatory graph for each training class. We use VGG19 to extract explanatory graphs. The ninth, tenth, twelfth and thirteenth convolutional layers are selected to build a four-layer graph.

2) Extracting semantic patterns. We set the number of graph nodes on each convolutional feature channel as 40. On each graph layer, top-20 activated graph nodes are selected to generate corresponding semantic patterns. Multiple graph nodes can be combined to form larger scale semantic pattern images.

3) Training reconstruction network. We combine multi-scale semantic part images and original images to form new training dataset for the reconstruction network. We take caffenet as backbone.

4) Training classification network. Based on the trained reconstruction network, we train the classification network using the original training data set. For each batch, we first optimize user-defined attributes by loss L_{ua}, then optimize latent attributes by loss L_{la}, and finally optimize the discriminator by loss L_{adv}. We take Resnet101 [28] as backbone of classification network and Alexnet [26] as backbone of discriminator.

To train our framework, data augmentation methods, including random cropping and mirroring are used to reduce overfitting in training. The crop size is $224\times224\times3$. We used the center crop in test process. We used the center crop in the test process. All used architectures are pre-trained on Imagenet dataset. The overall network is trained by finetuning pre-trained CNNs. The adopted optimization method is stochastic gradient descent (SGD). We set the momentum as 0.9 and the weight decay as 0.0005. The learning rate is initialized as 0.0001 and is multiplied by 0.1 when the error is plateauing. Grid search is taken to select hyperparameters.

4. Experiments

In this section, we present the experimental results and compare them to existing state-of-the-art results. Our methods are evaluated on widely used benchmarks for zero-shot learning: Animals with Attributes 2 [31] (AwA2) and Caltech-UCSD Birds-200-2011 [30] (CUB-200-2011).

Figure 5. Some semantic pattern samples discovered by explanatory graphs on CUB dataset. From left to right represents semantic patterns of top 5 activated graph nodes.

4.1. Baseline Methods

To demonstrate the influence of different components, we design the following baseline methods for comparison:

SS-UA (Single Scale and user-defined attributes optimization). This baseline only optimizes user-defined attributes in the classification network.

SS-LA (Single Scale and latent attributes optimization). This baseline only optimizes latent attributes in the classification network.

SS-UA&LA (Single Scale and user-defined attributes + latent attributes optimization). This baseline optimizes user-defined and latent attributes at the same time in the classification network.

SS-UA&LA-AL (Single Scale and user-defined attributes + latent attributes optimization & adversarial learning network). This method uses original images to optimize both classification and reconstruction network in adversarial manner [13].

MS-UA&LA-AL (Multi-Scale and user-defined attributes + latent attributes optimization & adversarial learning network). This method uses multi-scale semantic patterns to optimize both classification and reconstruction network in adversarial manner.

4.2. Effect of Attributes Space

In this section, the effect of different attributes space types is evaluated. In the classification network, user-defined attributes and latent attributes are contained in the final semantic embedding space. User-defined attributes are optimized by

RIS and latent attributes are optimized by RULE. These two kinds of attributes are connected to form the final augmented semantic embedding space.

The experimental results are shown in **Table 1**. Since the experimental results of some settings like 'SS-AE-Fixed' and 'MS-AE-Fixed' are not reported in [14], we report results implemented by ourselves for fair comparison. It can be seen that the augmented semantic embedding space has higher accuracy than single user-defined attribute space or latent attribute space, because it improves the generalized capacity of user-defined attributes and discriminative capacity of latent attributes. Compared with the triple loss in [14], the RULE optimizer for latent attributes obtains higher accuracy, because latent attributes can learn features more compatible to image classification by jointly optimizing semantic embedding space classifiers with semantic embedding space extractors.

4.3. Effect of Adversarial Learning

In this section, the influence of adversarial learning between the classification and reconstruction network is evaluated. The experimental results are presented in **Table 2**. It can be seen that adversarial learning can obviously improve the performance of classification networks. Furthermore, when we use multi-scale semantic patterns to train the reconstruction network, the accuracy is further improved because more semantic patterns can guide the reconstruction network to learn more meaningful semantic information.

4.4. Effect of Semantic Scales

The effect of semantic scales is shown in **Table 3**. It can be seen that a larger semantic scale can obtain higher recognition accuracy. Since the semantic patterns in our method are used in an organized way, combining several semantic scales can achieve better performance than the method in [14].

4.5. Effect of Graph Nodes

From section 3.2, we can know that in the explanatory graph, each graph node corresponds to a semantic pattern. More graph nodes will generate more semantic patterns, which can provide more multi-scale training images for reconstructing the network. Graph node number is determined by graph nodes number on each layer and graph layer number. Supposing a graph contains l layers, each layer contains c channels and each channel contains n graph nodes, then the total number of graph node is $l \times c \times n$.

We first test the effect of graph nodes number n on each channel. All graph nodes on each channel are ranked by activations, and top-n nodes are selected. The experimental results are presented in **Figure 6**. It can be seen that the greater the value of graph nodes number n on each channel, the higher the recognition accuracy, because more graph nodes can generate more semantic patterns. When $n > 40$, the recognition accuracy rises slowly due to the limited number of meaningful semantic patterns. For example, sometimes the three semantic

patterns of the head, torso, and legs are sufficient to represent the most important semantic information.

We then test the effect of graph layer number l with AlexNet as backbone. The number of AlexNet's convolutional layer is increased from 1 to 8, which will generate an 8-layer graph. The experimental results are presented in Figure 7. As you can see, more graph layers correspond to higher accuracy, because more layers contain more useful information and complex structures. However, more layers also require higher computational costs. Therefore, we should consider the balance between recognition accuracy and computational efficiency.

Table 1. Influence of augmented semantic embedding space on zero-shot recognition accuracy (%) using Resnet101 and VGG19 (in brackets).

Method	AwA2	CUB
SS-BE-Fixed [14]	75.20 (73.70)	50.51 (50.31)
SS-AE-Fixed (UA) [14]	76.97 (75.24)	54.17 (51.40)
SS-AE-Fixed (LA) [14]	74.76 (73.75)	55.08 (58.11)
SS-AE-Fixed (UA&LA) [14]	77.36 (75.77)	57.99 (58.96)
SS-UA (Ours)	76.23 (74.56)	56.57 (53.43)
SS-LA (Ours)	79.46 (75.87)	57.73 (59.64)
SS-UA&LA (Ours)	**80.92** (78.52)	**60.51** (59.33)

Table 2. Influence of adversarial learning on zero-shot recognition accuracy (%).

Method	AwA2	CUB
SS-UA&LA	80.92 (78.52)	60.51 (59.33)
SS-UA-AL	82.20 (78.70)	62.46 (61.53)
SS-LA-AL	83.41 (81.82)	63.69 (62.45)
SS-UA&LA-AL	83.92 (82.54)	66.81 (66.79)
MS-UA&LA-AL	84.62 (83.46)	67.32 (67.25)

Table 3. Influence of different adversarial scales comparison with [14] on zero-shot recognition accuracy (%).

Method	AwA2	CUB
MS-BE-Fixed (Scale 1)	75.20 (72.68)	51.88 (50.87)
MS-BE-Fixed (Scale 2)	75.87 (74.02)	53.04 (53.81)
MS-BE-Fixed (Scale 3)	- (-)	54.04 (54.72)
MS-BE-Fixed (All Scale)	77.80 (75.31)	56.85 (56.39)
MS-UA-AL (Scale 1)	77.34 (74.34)	53.65 (52.43)
MS-UA-AL (Scale 2)	77.61 (76.89)	55.32 (55.16)
MS-UA-AL (Scale 3)	78.78 (77.24)	56.12 (56.46)
MS-UA-AL (All Scale)	**79.53** (78.76)	**58.67** (58.21)

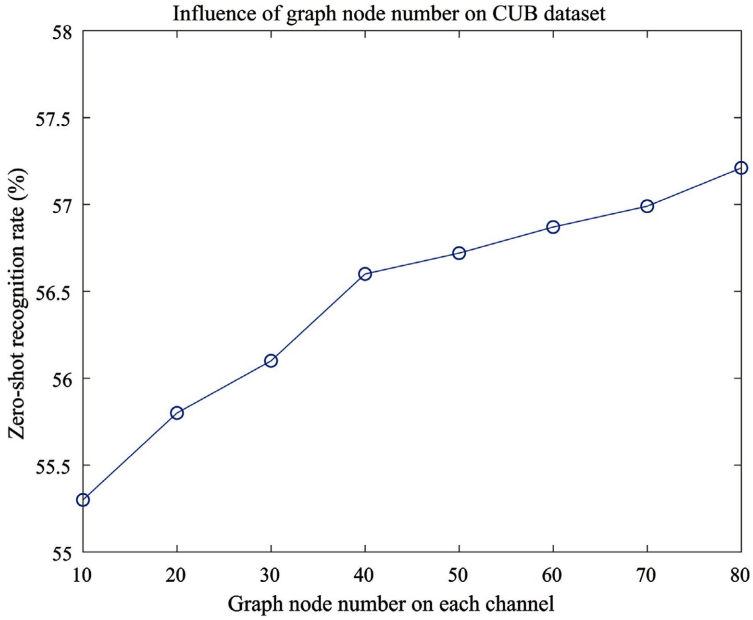

Figure 6. Influence of graph node number on CUB dataset. The used classification network is 'SS-UA'.

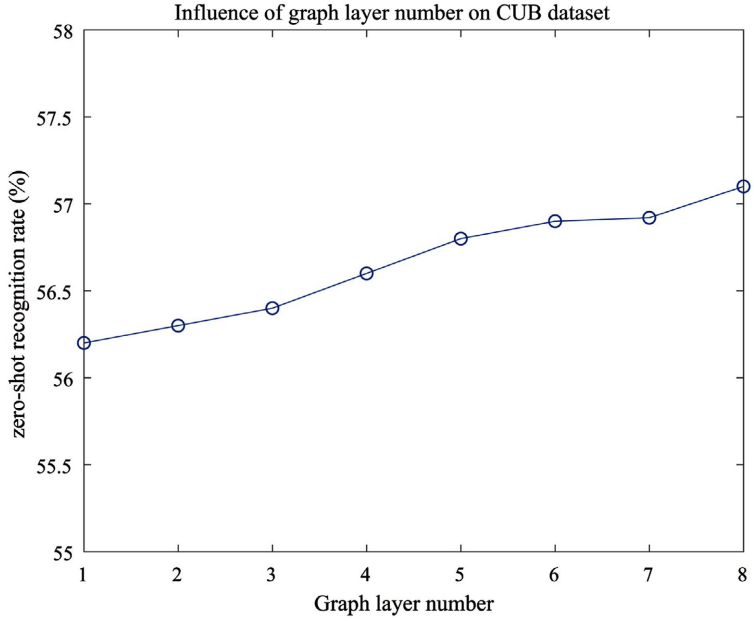

Figure 7. Influence of graph layer number on CUB dataset. The used classification network is 'SS-UA'.

We present some semantic patterns corresponding to Top-k activated explanatory graph nodes on CUB dataset in Figure 8. It can be seen that graph nodes with higher activation values correspond to more detailed features. The semantic patterns in 'head' and 'neck' are much more than those in 'leg' and 'foot'.

4.6. Comparison with State-of-the-Art Methods

As shown in Table 4, we compare our proposed method to the state-of-the-art

results on the AwA2 and CUB dataset. It can be seen that, in zero-shot recognition tasks, our method outperforms other methods. We compare our method with 'MS-AE-Fixed' setting in [14] instead of 'MS-AE-Learned' setting, because in our framework, backbone architecture parameters of classification network, encoder network and decoder network are fixed, and only parameters of fully-connected layers need to be optimized. Our proposed method outperforms f-CLSWGAN [25] above 10% on CUB dataset because our method directly reconstruct original images and combines advantages of user-defined attributes and latent attributes while [25] generates visual feature vectors based on only user-defined attributes.

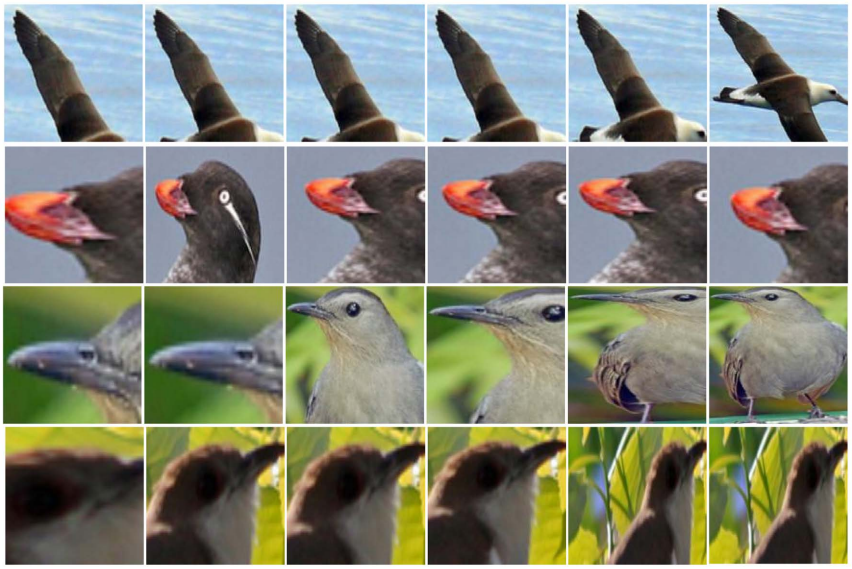

Figure 8. Some semantic patterns correspond to top-*k* explanatory graph nodes on CUB dataset. From left to right represents semantic patterns of top 1-top 10, top 11-top 20, top 21-top 30, top 31-top 40 top 41-top 50 and top 51-top 60 activated graph nodes.

Table 4. Accuracy comparison with other methods on unseen classes by zero-shot recognition methods. 'SS' represents standard splits and 'PS' represents splits proposed by [29].

Method	AwA2		CUB	
	SS	PS	SS	PS
DAP [31]	59.5	-	41.7	-
ES-ZSL [7]	75.6	58.6	55.1	53.9
Deep-SCoRe [9]	82.8	-	59.5	-
SP-AEN [13]	-	58.5	-	58.5
CONSE [19]	67.9	44.5	36.7	34.3
SJE [21]	69.5	61.9	55.3	53.9
MS-AE-Fixed [14]	81.4	-	63.37	-
Kernel [24]	-	64.3	-	57.1
f-CLSWGAN [25]	-	68.2	-	57.3
SE-ZSL [32]	80.8	69.2	60.3	59.6
MS-UA&LA-AL	**84.62**	**69.73**	**68.55**	**67.32**

5. Conclusion

In this paper, we propose a novel zero-shot recognition framework, which improves semantic embedding learning by reconstructing multi-scale explanatory semantic patterns. Semantic patterns are extracted from explanatory graphs. The classification network is trained jointly by two optimizers: RIS for semantic embedding supervision and RULE for class-level supervision. Multi-scale semantic patterns are taken to optimize both classification and reconstruction network by adversarial learning. Extensive experiments demonstrate satisfactory performance on zero-shot recognition tasks, which suggests that our method has good generalization capacity in visual recognition field.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Tom, M., Sutskever, I., Chen, K., Corrado, G.S. and Dean, J. (2013) Distributed Representations of Words and Phrases and Their Compositionality.

[2] Chen, X., Weng, J., Lu, W. and Xu, J. (2017) Multi-Gait Recognition Based on Attribute Discovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **40**, 1697-1710. https://doi.org/10.1109/TPAMI.2017.2726061

[3] Lampert, C.H., Nickisch, H. and Harmeling, S. (2009) Learning to Detect Unseen Object Classes by Between-Class Attribute Transfer. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (*CVPR*), Miami, 20-25 June 2009, 951-958. https://doi.org/10.1109/CVPR.2009.5206594

[4] Rohrbach, M., Stark, M. and Schiele, B. (2011) Evaluating Knowledge Transfer and Zero-Shot Learning in a Large-Scale Setting. 2011 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Colorado Springs, 20-25 June 2011, 1641-1648. https://doi.org/10.1109/CVPR.2011.5995627

[5] Wang, X. and Ji, Q. (2013) A Unified Probabilistic Approach Modeling Relationships between Attributes and Objects. *IEEE International Conference on Computer Vision*, Sydney, 1-8 December 2013, 2120-2127. https://doi.org/10.1109/ICCV.2013.264

[6] Akata, Z., Perronnin, F., Harchaoui, Z. and Schmid, C. (2016) Label-Embedding for Image Classification. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **38**, 1425-1438. https://doi.org/10.1109/TPAMI.2015.2487986

[7] Romera-Paredes, B. and Torr, P.H.S. (2015) An Embarrassingly Simple Approach to Zero-Shot Learning. *Proceedings of the* 32*nd International Conference on Machine Learning* (*ICML*'15), Volume 37.

[8] Xian, Y., Akata, Z., Sharma, G., Nguyen, Q., Hein, M. and Schiele, B. (2016) Latent Embeddings for Zero-Shot Classification. 2016 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Las Vegas, 27-30 June 2016, 69-77. https://doi.org/10.1109/CVPR.2016.15

[9] Morgado, P. and Vasconcelos, N. (2017) Semantically Consistent Regularization for Zero-Shot Recognition. 2017 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Honolulu, 21-26 July 2017, 6060-6069. https://doi.org/10.1109/CVPR.2017.220

[10] Song, J., Shen, C., Yang, Y., Liu, Y. and Song, M. (2018) Transductive Unbiased Embedding for Zero-Shot Learning. 2018 *IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-22 June 2018, 1024-1033. https://doi.org/10.1109/CVPR.2018.00113

[11] Changpinyo, S., Chao, W.L., Gong, B. and Fei, S. (2016) Synthesized Classifiers for Zero-Shot Learning. 2016 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Las Vegas, 27-30 June 2016, 5327-5336. https://doi.org/10.1109/CVPR.2016.575

[12] Wang, X., Ye, Y. and Gupta, A. (2018) Zero-Shot Recognition via Semantic Embeddings and Knowledge Graphs. 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 6857-6866. https://doi.org/10.1109/CVPR.2018.00717

[13] Chen, L., Zhang, H., Xiao, J., Liu, W. and Chang, S.F. (2017) Zero-Shot Visual Recognition Using Semantics-Preserving Adversarial Embedding Networks. 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 1043-1052. https://doi.org/10.1109/CVPR.2018.00115

[14] Li, Y., Zhang, J., Zhang, J. and Huang, K. (2018) Discriminative Learning of Latent Features for Zero-Shot Recognition. 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 7463-7471. https://doi.org/10.1109/CVPR.2018.00779

[15] Zhang, Q., Cao, R., Shi, F., Wu, Y.N. and Zhu, S.C. (2017) Interpreting CNN Knowledge via an Explanatory Graph.

[16] Farhadi, A., Endres, I., Hoiem, D. and Forsyth, D. (2009) Describing Objects by Their Attributes. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Miami, 20-25 June 2009, 1778-1785. https://doi.org/10.1109/CVPR.2009.5206772

[17] Palatucci, M., Pomerleau, D., Hinton, G.E. and Mitchell, T.M. (2009) Zero-Shot Learning with Semantic Output Codes. *International Conference on Neural Information Processing Systems*, Vancouver, 7-10 December 2009, 1410-1418.

[18] Frome, A., Corrado, G.S., Shlens, J., Bengio, S. and Mikolov, T. (2013) DeViSE: A Deep Visual-Semantic Embedding Model. Curran Associates Inc., Red Hook.

[19] Norouzi, M., Mikolov, T., Bengio, S., Singer, Y., Shlens, J., Frome, A., *et al.* (2013) Zero-Shot Learning by Convex Combination of Semantic Embeddings.

[20] Socher, R., Ganjoo, M., Sridhar, H., Bastani, O., Manning, C.D. and Ng, A.Y. (2013) Zero-Shot Learning through Cross-Modal Transfer. Curran Associates Inc., Red Hook.

[21] Akata, Z., Reed, S., Walter, D., Lee, H. and Schiele, B. (2015) Evaluation of Output Embeddings for Fine-Grained Image Classification. *IEEE Computer Vision and Pattern Recognition*, Boston, 7-12 June 2015, 2927-2936. https://doi.org/10.1109/CVPR.2015.7298911

[22] Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013) Efficient Estimation of Word Representations in Vector Space.

[23] Chen, H., Gallagher, A. and Girod, B. (2012) Describing Clothing by Semantic Attributes. In: *European Conference on Computer Vision*, Springer, Berlin, 609-623. https://doi.org/10.1007/978-3-642-33712-3_44

[24] Zhang, H. and Koniusz, P. (2018) Zero-Shot Kernel Learning. 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 7670-7679. https://doi.org/10.1109/CVPR.2018.00800

[25] Xian, Y., Lorenz, T., Schiele, B. and Akata, Z. (2018) Feature Generating Networks

for Zero-Shot Learning. 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Salt Lake City, 18-23 June 2018, 5542-5551. https://doi.org/10.1109/CVPR.2018.00581

[26] Krizhevsky, A., Sutskever, I. and Hinton, G. (2012) ImageNet Classification with Deep Convolutional Neural Networks. *Communications of the ACM*, **60**, 84-90.

[27] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., *et al.* (2014) Generative Adversarial Nets. MIT Press, Cambridge.

[28] He, K., Zhang, X., Ren, S. and Sun, J. (2016) Deep Residual Learning for Image Recognition. 2016 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Las Vegas, 27-30 June 2016, 770-778. https://doi.org/10.1109/CVPR.2016.90

[29] Xian, Y., Schiele, B. and Akata, Z. (2017) Zero-Shot Learning—The Good, the Bad and the Ugly. 2017 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Honolulu, 21-26 July 2017, 3077-3086. https://doi.org/10.1109/CVPR.2017.328

[30] Wah, C., Branson, S., Welinder, P., Perona, P. and Belongie, S. (2011) The Caltech-UCSD Birds-200-2011 Dataset. California Institute of Technology, Pasadena.

[31] Lampert, C.H., Nickisch, H. and Harmeling, S. (2014) Attribute-Based Classification for Zero-Shot Visual Object Categorization. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **36**, 453-465. https://doi.org/10.1109/TPAMI.2013.140

[32] Verma, V.K., Arora, G., Mishra, A. and Rai, P. (2018) Generalized Zero-Shot Learning via Synthesized Examples. 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 4281-4289. https://doi.org/10.1109/CVPR.2018.00450