Co-Representation Network for Generalized Zero-Shot Learning

Fei Zhang [1] Guangming Shi [1]

Abstract

Generalized zero-shot learning is a significant topic but faced with bias problem, which leads to unseen classes being easily misclassified into seen classes. Hence we propose an embedding model called co-representation network to learn a more uniform visual embedding space that effectively alleviates the bias problem and helps with classification. We mathematically analyze our model and find it learns a projection with high local linearity, which is proved to cause less bias problem. The network consists of a cooperation module for representation and a relation module for classification. It is simple in structure and can be easily trained in an end-to-end manner. Experiments show that our method outperforms existing generalized zero-shot learning methods on several benchmark datasets.

1. Introduction

Zero-shot learning (ZSL) is an important field in computer vision. It aims at classifying images of unseen classes through establishing mathematical relationship between the semantic space and the visual space. Usually both seen classes and unseen classes are described through a set of semantic vectors in the same semantic space, including manually defined attribute vectors (Lampert et al., 2014) and word embedding vectors (Mikolov et al., 2013). Existing ZSL researches can be categorized into conventional ZSL (CZSL) and generalized ZSL (GZSL) according to the classification range: A CZSL task classifies only unseen classes while a GZSL task classifies both seen and unseen classes. Our algorithm is dedicated to the more challenging and practical GZSL task.

Since the the concept of ZSL was first proposed by Lampert et al. (2009), algorithms for CZSL have been emerging one after another. During this period, every new algorithm broke the records on several benchmark datasets and has promoted the progress of ZSL. However, CZSL technology has not got practical applications because of its defects. In practical scenarios, it is usually necessary to classify both seen and unseen samples instead of just classifying the unseen ones. But almost all classic CZSL algorithms have a strong classification bias towards seen classes when applying it to a GZSL task (Xian et al., 2017), which is called bias problem. Therefor, it is significant to study the GZSL algorithm. At present, effective GZSL algorithms mostly use generative models (Verma et al., 2018; Xian et al., 2018). They generate synthetic features of unseen classes to control the weights of seen classes and unseen classes so that the bias problem is mitigated.

Generative model is effective in GZSL but usually complex in structure and not easy to train. Instead of generative model, in this paper we propose a novel embedding model called co-representation network (CRnet) for GZSL, which learns more discriminative visual features and a more uniform embedding space that helps to classify. Our model achieves a high accuracy on both seen classes and unseen classes and outperforms existing generative models on most GZSL benchmark datasets. The model has a simple, intuitive and easy to implement structure with high expandability. It consists of two modules: A cooperation module for projecting the semantic space to a visual embedding space; A relation module (Sung et al., 2018) for classification. The cooperation module consists of several parallel single-layer perceptrons, each of which learns a linear projection in a particular direction and is therefore called an expert module. We mathematically analyze our model and find it learns a piecewise linear projection with high local linearity, which is proved to be effective in alleviating the bias problem. We evaluate our model on five GZSL benchmark datasets including AwA1 (Lampert et al., 2014), AwA2 (Xian et al., 2017), CUB (Welinder et al., 2010), aPY (Farhadi et al., 2009) and SUN (Patterson & Hays, 2012), and achieve the state-of-the-art harmonic mean accuracy on the first four datasets and a comparable result on SUN. Our main contributions are summarized as follows:

1. We explore the cause of the bias problem and propose a reasonable metric, Local Relative Distance, to measure it.

2. We propose a novel CRnet with simple structure to solve

[1]School of Artificial Intelligence, Xidian University, China. Correspondence to: Guangming Shi <gmshi@xidian.edu.cn>.

Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

the bias problem in essence and therefor achieve good results on GZSL.

3. We mathematically analyze our model and prove that its high local linearity is effective in alleviating the bias problem, which is an enlightening conclusion for other transfer learning problems.

2. Related Work

We summarize several recent algorithms for CZSL and GZSL, and explain their relationship with ours.

2.1. Conventional Zero-Shot Learning

Early CZSL algorithms focus on building an embedding model with a better mathematical projection between the semantic space and the visual space. Generally speaking, the form of projection can be classified into three main categories as follows.

1) The visual space are set as the source space and classification is based on the semantic space. Famous methods including DeViSE (Frome et al., 2013), ALE (Akata et al., 2016), ESZSL (Romera-Paredes & Torr, 2015), SAE (Kodirov et al., 2017), etc. learn linear or non-linear projections from the visual space to the semantic space through various constraints or loss functions. Besides, classical algorithms like DAP (Lampert et al., 2014) consider CZSL as a probability problem and perform classification based on the predicted posterior of each attribute, so we also classify them as this category.

2) The visual space and the semantic space are simultaneously set as the source space and projected to a latent space in search of better representation that helps to classify, such as SSE (Zhang & Saligrama, 2015). SSE learns an embedding function for semantic vectors and visual vectors to get mixture patterns so that the similarity can be readily measured.

3) The semantic space is set as the source space and classification is performed on the visual space. For example, DEM (Zhang et al., 2017) learns a deep embedding model which projects the semantic space to the visual embedding space to alleviate the hubness problem (Radovanović et al., 2010). This type of projection proves to significantly alleviate the hubness problem compared with the other two types of projection (Shigeto et al., 2015; Zhang et al., 2017). We adopt this conclusion, thus our model learns a projection from the semantic space to the visual space.

2.2. Generalized Zero-Shot Learning

Although the concept of GZSL has been proposed in 2013 (Scheirer et al., 2013), research on it did not appear until nearly two years. Chao et al. (2016) propose an effective calibration method to adapt CZSL algorithms to perform well on GZSL and develop a metric for GZSL evaluation. Xian et al. (2017) not only evaluate the quality of most previous CZSL algorithms in the same benchmark but also reproduce their performance on GZSL tasks, and propose a reasonable protocol for GZSL evaluation, which is widely adopted in later GZSL algorithms, including ours.

Most existing algorithms use generative models for GZSL. Because once generating some synthetic samples with high quality of unseen classes, the problem becomes a traditional classification task and bias problem as well as hubness problem can get solved. Bucher et al. (2017) present 4 different strategies to design conditional data generators and compare their performance. CVAE-ZSL (Mishra et al., 2018) uses a conditional variational autoencoder to implement the generation. SE-GZSL (Verma et al., 2018) also designs a generative model baesd on a variational autoencoder but in a feedback-driven way to generate novel exemplars from seen/unseen classes. f-CLSWGAN (Xian et al., 2018) trains a Wasserstein GAN with a classification loss and is able to generate sufficiently discriminative CNN features to train softmax classifiers.

Compared with embedding models widely used in CZSL, generative models show their powerful performance on GZSL. In this paper, we mathematically look into the essential cause of CZSL algorithms' bad performance on GZSL, the bias problem, and propose a traditional and simple embedding model called CRnet to overcome it rather than pursuing the trend. Especially, Sung et al. (2018) propose the relation network with a learnable relation module for metric learning to solve the few-shot learning problem as well as CZSL/GZSL problem. We adopt this module in CRnet and set the relation network as a contrast in our experiments.

3. Method

3.1. Problems

CZSL & GZSL: Our task is to learn a projection from the semantic space to the visual space. Given a seen set containing M seen classes $\mathcal{S} = (\mathbf{S}^s, \mathbf{Y}^s) = \{(\mathbf{s}_m^s, y_m^s)\}_{m=1}^M$ and an unseen set containg N unseen classes $\mathcal{U} = (\mathbf{S}^u, \mathbf{Y}^u) = \{(\mathbf{s}_n^u, y_n^u)\}_{n=1}^N$, where $\mathbf{s}_m^s, \mathbf{s}_n^u \in \mathbb{R}^{1\times P}$ denote the P-d (dimensional) semantic vectors of the m-th seen class and the n-th unseen class respectively, y_m^s, y_n^u denote their class labels respectively. And they satisfy $\mathbf{Y}^s \cap \mathbf{Y}^u = \varnothing$. Then training samples can be denoted as $\mathbf{V}^s = \{\mathbf{v}_i^s\}_{i=1}^C$, where C is the number of images, $\mathbf{v}_i^s \in \mathbb{R}^{1\times Q}$ denote the Q-d feature vector extracted from the i-th training image through a traditional deep CNN (e.g. GoogleNet, VGG19, ResNet) and its corresponding semantic vector and label are $(\mathbf{s}_i, y_i) \in \mathcal{S}$. Similarly, let $\mathbf{V}^t = \{\mathbf{v}_j^t\}_{j=1}^D$ denote the D test samples but its corresponding label and semantic vector $(\mathbf{s}_j^t, y_j^t)$ remain unknown, which is exactly what ZSL has to solve. So the main task is to learn a projection $f : \mathbf{S}^s \to \mathbf{V}^s$ through training samples so as to infer the labels of test set $\mathbf{V}^t$.

After obtaining a well-learned projection $f : \mathbf{S}^s \to \mathbf{V}^s$, for semantic vector $\mathbf{s}_m^s$ and $\mathbf{s}_n^u$, their actual projection results are $\widehat{\mathbf{v}}_m^s = f(\mathbf{s}_m^s)$ and $\widehat{\mathbf{v}}_n^u = f(\mathbf{s}_n^u)$ respectively, which we call feature anchors. And the space where the feature anchors are located in is called visual embedding space. Then a similarity function $g(*)$ is used to evaluate the similarity r_{jd} between the j-th test image's feature $\mathbf{v}_j^t$ and any predicted feature anchor $\widehat{\mathbf{v}}_d$:

$$r_{jd} = g(\mathbf{v}_j^t, \widehat{\mathbf{v}}_d) \tag{1}$$

For CZSL, the test set only contains images of unseen classes, that is, $y_j^t \in \mathbf{Y}^t = \mathbf{Y}^u$ and the search space is limited to unseen classes, $\widehat{\mathbf{v}}_d \in \{\widehat{\mathbf{v}}_n^u\}_{n=1}^N, d \in \{1, \dots, N\}$. For GZSL, the test set does not just contains images of unseen classes, then $y_j^t \in \mathbf{Y}^t = \mathbf{Y}^u \cup \mathbf{Y}^s$ and search space contains both seen classes and unseen classes, $\widehat{\mathbf{v}}_d \in \{\widehat{\mathbf{v}}_n^u\}_{n=1}^N \cup \{\widehat{\mathbf{v}}_m^s\}_{m=1}^M, d \in \{1, \dots, M+N\}$. The higher the similarity, the more likely it is to belong to this class. So the classification of j-th test image can be achieved by

$$ind(j) = \arg\max_d \ g(\mathbf{v}_j^t, \widehat{\mathbf{v}}_d) \tag{2}$$

where $ind(j)$ denotes the index, so the predicted label $y_j^t = \mathbf{Y}_{ind(j)}^t$. In this paper, we focus on GZSL.

Bias Problem: The key to GZSL is to solve the bias problem. It refers to the problem that in the embedding space the projected feature anchors of unseen classes $\widehat{\mathbf{v}}_m^s$ are distributed too near to that of seen classes $\widehat{\mathbf{v}}_n^u$, hence they are hard to separate and unseen images are easily classified into close seen classes. Consequently, most previous algorithms that perform well on CZSL tasks are invalid on GZSL ones.

The cause of bias problem is similar to that of over-fitting problem in classical machine learning tasks. On one hand, semantic space and visual embedding space are two completely different space and their manifolds are inconsistent, which increases the complexity of projection $f(*)$ between them and requires it to be highly nonlinear. On the other hand, the sample size of the semantic space is limited the same as the seen class number M. These cause $f(*)$ to over-fit the semantic vectors of seen classes and make the feature anchors of seen classes and unseen classes extremely close in the embedding space. Hence, the visual features that is classified based on the similarity function $g(*)$ become less separable. In general, bias problem is highly related with the complexity of $f(*)$ and the uniformity of the distribution of points in the embedding space.

Figure 1 shows a simplest example of projection from 1-d semantic space to the q-th dimension of the embedding space. s_1^s, s_2^s and s_3^s are three points in the semantic space

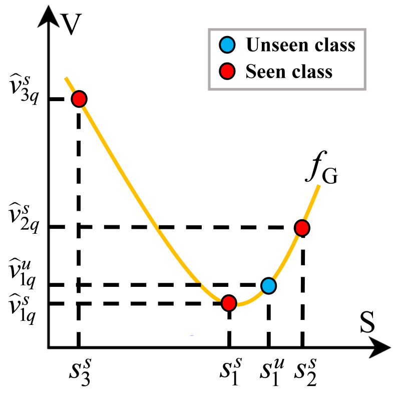

Figure 1. An example of a general fitting curve f_G from 1-d to 1-d. S: Semantic space, V: Embedding space.

and their corresponding feature anchors are $\widehat{\mathbf{v}}_1^s$, $\widehat{\mathbf{v}}_2^s$ and $\widehat{\mathbf{v}}_3^s$. $\widehat{v}_{1q}^s$, $\widehat{v}_{2q}^s$ and $\widehat{v}_{3q}^s$ are the values of feature anchors in the q-th dimension. And the curve f_G is learned through these seen points. Suppose there is a point s_1^u for unseen class between s_1^s and s_2^s and its predicted feature anchor is $\widehat{\mathbf{v}}_1^u$, then $\widehat{v}_{1q}^u = f_G(s_1^u)$. When the distance between $\widehat{\mathbf{v}}_3^u$ and $\widehat{\mathbf{v}}_1^s$ or $\widehat{\mathbf{v}}_2^s$ is too close, the bias problem arises. For quantitative analysis, we use Standardized Euclidean distance $e(*)$ as the distance between any points in the embedding space:

$$e(\widehat{\mathbf{v}}_1, \widehat{\mathbf{v}}_2) = \left(\sum_{q=1}^{Q} \frac{(\widehat{v}_{1q} - \widehat{v}_{2q})^2}{\sigma_q^2} \right)^{\frac{1}{2}} \tag{3}$$

where σ_q^2 is the variance of all points in the q-th dimension and is used for normalization. And we adopt the Local Relative Distance (LRD) to measure the distribution of unseen class anchors:

$$\mathrm{LRD}(\widehat{\mathbf{v}}_j^u) = \frac{e(\widehat{\mathbf{v}}_j^u, \widehat{\mathbf{v}}_{c(j)}^s)}{e(\widehat{\mathbf{v}}_{c(j)}^s, \widehat{\mathbf{v}}_{c(j)}^{s'})} \tag{4}$$

where $\widehat{\mathbf{v}}_{c(j)}^s$ is the seen class anchor closest to the unseen class anchor $\widehat{\mathbf{v}}_{c(j)}^s$, and $\widehat{\mathbf{v}}_{c(j)}^{s'}$ is the seen class anchor closest to $\widehat{\mathbf{v}}_{c(j)}^s$. We propose the metric LRD on the assumption that the local discrimination of similarity function $g(*)$ is certain and is related to the shortest distance between the two local seen class anchors. Hence, unseen class anchors with large LRD can be more easily distinguished from seen class anchors by $g(*)$ and lead to less bias problem.

3.2. Algorithm

The core of CRnet is to decompose the complex projection $f(*)$ into several simple projections to construct a more uniform embedding space with large LRD. The concept of decomposition is common in machine learning and is effective for specific problems, e.g., the Deep Set model adopts a sum-decomposition structure to achieve permutation invariance (Zaheer et al., 2017). While in this paper, CRnet adopts a decomposition structure to alleviate the bias problem. It consists of a cooperation module with several expert modules for $f(*)$ and a relation module for $g(*)$. The steps and concepts of the algorithm are as follows.

Initialization Algorithm: First, divide the semantic vectors $\mathbf{S}^s = \{\mathbf{s}_m^s\}_{m=1}^M$ of training set into $K \in \{1, \dots, M\}$ subsets. We call each subset a semantic field and use the unsupervised K-means clustering algorithm to implement this process. Let $\bar{\mathbf{s}}_k$ denote the clustering center of k-th semantic field, where $k \in \{1, \dots, K\}$, and we refer to it as field center. Then for the k-th field center we specialize in learning a projection $f_k(\mathbf{s}_m^s; \bar{\mathbf{s}}_k)$ through an expert module so that the whole projection $f(*)$ can be expressed as:

$$f(\mathbf{s}_m^s) = \sum_{k=1}^{K} f_k(\mathbf{s}_m^s; \bar{\mathbf{s}}_k) \tag{5}$$

Expert Module: We design expert modules with the same structure for each field center. For the k-th field center, the input of its corresponding expert module is the offset of the m-th semantic vector relative to $\bar{\mathbf{s}}_k$, denoted as $\mathbf{o}_{km}$ and $\mathbf{o}_{km} = \mathbf{s}_m^s - \bar{\mathbf{s}}_k$. Then the set of the input offsets with M seen classes for the k-th expert module is $\mathbf{O}_k = \{\mathbf{o}_{km}\}_{m=1}^M$.

Each expert module contains a single layer perceptron and we implement it with a FC layer (full connected layer) with ReLU (Rectified Linear Unit). Then the projection $f_k(*)$ can be experssed as:

$$f_k(\mathbf{s}_m^s; \bar{\mathbf{s}}_k) = relu\left(\mathbf{W}_k(\mathbf{s}_m^s - \bar{\mathbf{s}}_k) + \mathbf{b}_k\right) \tag{6}$$

where $\mathbf{W}_k$ denotes the weights of FC layer and $\mathbf{b}_k$ denotes the bias.

Figure 2(a) shows an example of three semantic fields ($K = 3$). It's obvious that for a specific expert module, it has small input offsets for semantic vectors in its corresponding field, while for semantic vectors in other fields it has large ones ($\mathbf{o}_{22} > \mathbf{o}_{21}$). In this way, the expert network gets different learning strategies for the semantic vectors belonging to different fields, thus it can have different sensitivity to different directions.

Cooperation Module: The cooperation module is a combination of K expert modules. We perform this combination by summing the output vectors of the K expert networks (Eq 5). Then the entire projection $f(*)$ can be expressed as:

$$f(\mathbf{s}_m^s) = \sum_{k=1}^{K} relu\left(\mathbf{W}_k(\mathbf{s}_m^s - \bar{\mathbf{s}}_k) + \mathbf{b}_k\right) \tag{7}$$

Figure 2(b) shows an ideal state achieved by each expert module after training with Figure 2(a) as the initial state. ReLU of each expert module actually divides the semantic space into two parts. Semantic vectors located in one part

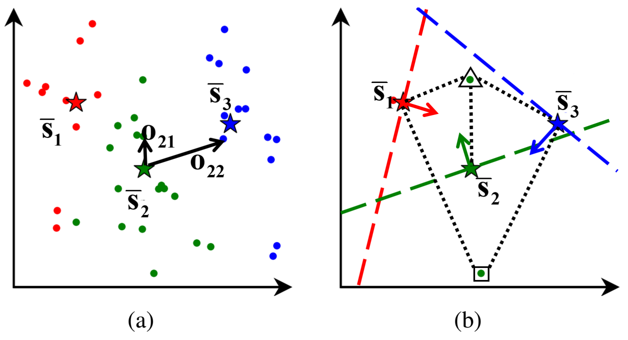

Figure 2. (a) Semantic fields of AwA2 (Section 4.1), stars represent the field centers. (b) The division of sementic space by different expert modules. The dotted colored lines represent the dividing lines, colored arrows indicate the gradient directions. All three modules respond to the point boxed in the triangle while only f_1 and f_3 respond to the point boxed in the square.

are projected to 0 while those located in another part produce responses. And along with $\mathbf{W}_k$, which is the gradient direction of $f_k(*)$, the response produced by the k-th expert module increases the most. Each expert module will eventually be sensitive to data in a specific direction and this is why we name it expert. Hence, cooperation module finally devides the semantic space into several parts, and semantic vectors located in different parts are projected by several different expert modules. In this way, the form of each expert module is very simple, a piecewise linear function $f_k(*)$, but the whole model still retains certain nonlinear ability.

Relation Module: We adopt a relation module (Sung et al., 2018) as the similarity function $g(*)$ and follow most of the original settings. Relation module works in a data driven way so that the similarity metric is self-adaptive to data. It consits of a two-layer MLP and its structure can be written as: FC1-ReLU-FC2-Sigmoid. The input of FC1 is the concatenation of a predicted anchor $\widehat{\mathbf{v}}_m^s = f(\mathbf{s}_m^s)$ and a sample feature vector $\mathbf{v}_i^s$ in depth; FC2 works as a hidden layer to increase nonlinearity and its output dimension is 1; The sigmoid function maps the output to $[0, 1]$ in order to represent the similarity. We randomly sample the entire training set to generate training pairs of $\widehat{\mathbf{v}}_m^s$ and $\mathbf{v}_i^s$, and control the ratio of matched pairs ($\widehat{\mathbf{v}}_m^s$ and $\mathbf{v}_i^s$ correspond to the same class y_m^s) to mismatched pairs ($\widehat{\mathbf{v}}_m^s$ and $\mathbf{v}_i^s$ correspond to different classes) at about $1:30$. The similarity of matched pairs is set to 1 and the similarity of mismatched pairs is set to 0. Then the similarity ground-truth $l(\mathbf{v}_i^s, \widehat{\mathbf{v}}_m^s)$ of training pairs can be expressed as:

$$l(\mathbf{v}_i^s, \widehat{\mathbf{v}}_m^s) = \begin{cases} 1, & y_m^s = y_i \\ 0, & y_m^s \neq y_i \end{cases} \tag{8}$$

The relation network is trained through MSE loss:

$$loss1 = \sum_m \sum_i (g(\mathbf{v}_i^s, \widehat{\mathbf{v}}_m^s) - l(\mathbf{v}_i^s, \widehat{\mathbf{v}}_m^s))^2 \tag{9}$$

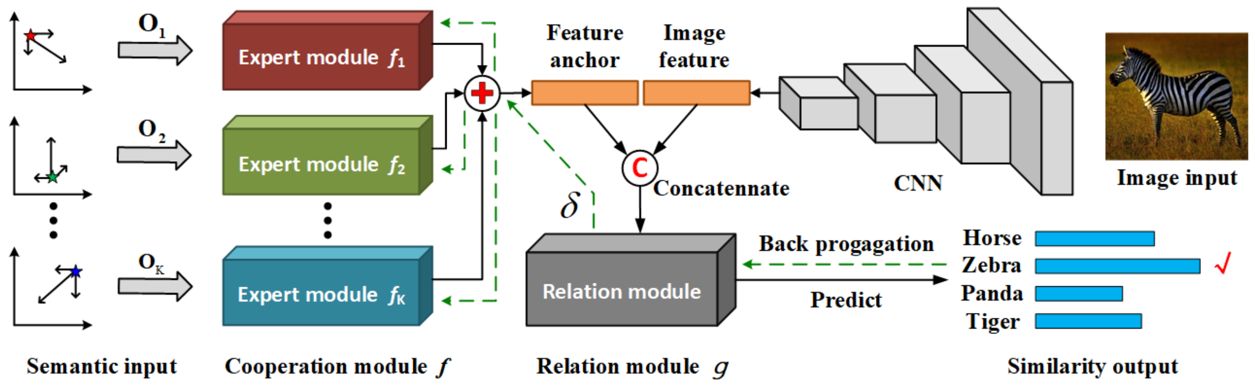

Figure 3. The structure of CRnet

where $\sum_m, \sum_i$ respectively mean the possible values of i and m in a randomly generated training batch (Sung et al., 2018). In this way, the network will have a smaller output for mismatched pairs and a larger output for matched pairs.

We connect the cooperation module and the relation module together to construct the entire CRnet. It is simple in structure and only consists of several perceptrons: several parallel single-layer perceptrons make up the cooperation module and a two-layer perceptron makes up the relation module. Figure 3 shows the entire structure of CRnet for GZSL. It should be noted that the pretrained CNN module is only used for feature extraction and will not be trained.

Trianing: The entire network can be trained in an end-to-end manner with $loss1$. And we also adopt two regularization losses to improve the performance. The total loss function can be expressed as:

$$loss = loss1 + \alpha||\mathbf{w}_f||_2^2 + \beta||\mathbf{w}_g||_2^2 \tag{10}$$

where $\mathbf{w}_f$, $\mathbf{w}_g$ represent trainable parameters of $f(*)$ and $g(*)$ respectively and $||\mathbf{w}_f||_2^2$, $||\mathbf{w}_g||_2^2$ are the regularization loss for them. α, β are two hyperparameters. We argue that it is necessary to use two coefficients to control the regularization terms of the two modules separately. Because there are two completely different high-dimensional spaces and the roles played by the two modules are completely different: cooperation module for representation learning while relation module for classification. Sung et al. (2018) points out that $||\mathbf{w}_f||_2^2$ can alleviate the hubness problem of embedding space, while $||\mathbf{w}_g||_2^2$ can avoid the relation module over-fitting on the training set as well as accelerating convergence. And both of these points help solve the bias problem. So the objective function can be written as:

$$\begin{aligned}\arg\min_{\mathbf{w}_f,\mathbf{w}_g} \sum_m \sum_i (g(\mathbf{v}_i^s, \hat{\mathbf{v}}_m^s) - l(\mathbf{v}_i^s, \hat{\mathbf{v}}_m^s))^2 \\ + \alpha||\mathbf{w}_f||_2^2 + \beta||\mathbf{w}_g||_2^2\end{aligned} \tag{11}$$

where $\hat{\mathbf{v}}_m^s$ can be obtained from Eq (7).

The green dotted line in Figure 3 indicates the back propagation path of loss. There are K branches after loss goes through the relation module and each branch share the same error, denoted as δ. According to Eq (6), the partial derivative of $loss$ to $\mathbf{W}_k$ of the k-th expert module is

$$\frac{\partial loss}{\partial \mathbf{W}_k} = \delta(\mathbf{s}_m^s - \bar{\mathbf{s}}_k) \tag{12}$$

In particular, the common regularization term $||\mathbf{w}_f||_2^2$ is omitted here for simplicity purpose. Then the update processes of $\mathbf{W}_k$ can be expressed as

$$\mathbf{W}_k' = \mathbf{W}_k + \eta\delta(\mathbf{s}_m^s - \bar{\mathbf{s}}_k) \tag{13}$$

where η is the learning rate. Obviously, when the input semantic vector $\mathbf{s}_m^s$ is close to the k-th field center $\bar{\mathbf{s}}_k$, the update rate of $\mathbf{W}_k$ is small. In the meanwhile, expert module whose field center is far away from $\mathbf{s}_m^s$ is faced with just a opposite situation — the update rate of weights is large. Hence, each expert module can converge in different directions and finally reach the state of Figure 2(b).

In fact, the initialization center corresponding to each expert module can be arbitrarily selected, and even all modules can share the same center, such as the mean of all semantic vectors. But in this case, the convergence direction of different modules largely depends on the initial value of weights, which easily cause different expert modules to converge to the same direction and leads to a local optimal solution, especially when K is large. So we use the initialization algorithm to ensure that the expert modules converge in different directions and divide the space as evenly as possible.

Test: Apply the learned projection $f(*)$ to both seen class space $\mathbf{S}^s$ and unseen class space $\mathbf{S}^u$ for test. During test, the input offests $\mathbf{O}_k = \{\mathbf{o}_{kd}\}_{d=1}^{M+N} = \{\mathbf{s}_d - \bar{\mathbf{s}}_\mathbf{k}\}_{d=1}^{M+N}$, where $\mathbf{s}_d \in \mathbf{S}^u \cup \mathbf{S}^s$. Then classification for test image's feature $\mathbf{v}_j^t$ can be easily performed through Eq (1) and Eq (2).

Table 1. Classification accuracies on various datasets following the GZSL setting. As/Au: Average per-class top-1 accuracy in % on seen/unseen classes. H: Harmonic mean accuracy. Cloumn 3-12 are the results of classic CZSL methods on GZSL repruduced by Xian et al. (2017), which show strong bias on As. Cloumn 13-18 are the official results of recent GZSL methods.

	AwA1			**AwA2**			**CUB**			**SUN**			**aPY**		
Method	**As**	**Au**	**H**	**As**	**Au**	**H**	**As**	**Au**	**H**	**As**	**Au**	**H**	**As**	**Au**	**H**
CONSE (Norouzi et al., 2013)	88.6	0.4	0.8	90.6	0.5	1.0	72.2	1.6	3.1	39.9	6.8	11.6	91.2	0.0	0.0
CMT* (Socher et al., 2013)	86.9	8.4	15.3	89.0	8.7	15.9	60.1	4.7	8.7	28.0	8.7	13.3	74.2	10.9	19.0
SSE (Zhang & Saligrama, 2015)	80.5	7.0	12.9	82.5	8.1	14.8	46.9	8.5	14.4	36.4	2.1	4.0	78.9	0.2	0.4
SJE (Akata et al., 2015)	74.6	11.3	19.6	73.9	8.0	14.4	59.2	23.5	33.6	30.5	14.7	19.8	55.7	3.7	6.9
ESZSL (Romera-Paredes & Torr, 2015)	75.6	6.6	12.1	77.8	5.9	11.0	63.8	12.6	21.0	27.9	11.0	15.8	70.1	2.4	4.6
SYNC (Changpinyo et al., 2016)	87.3	8.9	16.2	90.5	10.0	18.0	70.9	11.5	19.8	43.3	7.9	13.4	66.3	7.4	13.3
SAE (Kodirov et al., 2017)	77.1	1.8	3.5	82.2	1.1	2.2	54.0	7.8	13.6	18.0	8.8	11.8	80.9	0.4	0.9
LATEM (Xian et al., 2016)	71.7	7.3	13.3	77.3	11.5	20.0	57.3	15.2	24.0	28.8	14.7	19.5	73.0	0.1	0.2
ALE (Akata et al., 2016)	16.8	76.1	27.5	81.8	14.0	23.9	62.8	23.7	34.4	33.1	21.8	26.3	73.7	4.6	8.7
DEVISE (Frome et al., 2013)	68.7	13.4	22.4	74.7	17.1	27.8	53.0	23.8	32.8	27.4	16.9	20.9	76.9	4.9	9.2
DEM (Zhang et al., 2017)	84.7	32.8	47.3	86.4	30.5	45.1	57.9	19.6	29.2	34.3	20.5	25.6	11.1	75.1	19.4
RN (Sung et al., 2018)	91.3	31.4	46.7	93.4	30.0	45.3	61.1	38.1	47.0	-	-	-	-	-	-
DCN (Liu et al., 2018)	84.2	25.5	39.1	-	-	-	60.7	28.4	38.7	37.0	25.5	30.2	75.0	14.2	23.9
CVAE-ZSL (Mishra et al., 2018)	-	-	47.2	-	-	51.2	-	-	34.5	-	-	26.7	-	-	-
SE-GZSL (Verma et al., 2018)	67.8	56.3	61.5	68.1	58.3	62.8	53.3	41.5	46.7	30.5	40.9	34.9	-	-	-
f-CLSWGAN (Xian et al., 2018)	61.4	57.9	59.6	-	-	-	57.7	43.7	49.7	36.6	42.6	39.4	-	-	-
CRnet (Ours)	74.7	**58.1**	**65.4**	78.8	52.6	**63.1**	56.8	**45.5**	**50.5**	36.5	34.1	35.3	68.4	**32.4**	**44.0**

4. Experiment

We evaluate our approach on five benchmark datasets including AwA1 (Animals with Attributes 1), AwA2 (Animals with Attributes 2), CUB (Caltech UCSD Birds 200), SUN (SUN Scene Recognition) and aPY(Attribute Pascal and Yahoo), following the GZSL settings (Xian et al., 2017) for seen/unseen splits and compare it with other methods including classic CZSL methods and several recent GZSL methods.

4.1. Datasets and Hyperparameters

Among these five datasets, AwA1, AwA2, and aPY are three coarse-grained classification datasets while CUB and SUN are two fine-grained ones. For fair comparison, we use the 2048-d features extracted form resnet-101 pre-trained on Imagenet without any finetune operation for all datasets, consistent with the experiments reproduced by Xian et al. (2017). The specific details and training hyper-parameters of each datasets are summarized in Table 2.

Table 2. Statistics and hyper-parameters of each dataset. Att: The dimensions of semantic vectors; S/U: Seen class size/Unseen class size; Img: Total number of images.

	Statistic			**Parameter**		
Dataset	Att	S/U	Img	K	α	β
AwA1	85	40/10	30475	3	1e-5	1e-4
AwA2	85	40/10	37322	3	1e-5	1e-4
aPY	64	20/12	15339	3	1e-5	1e-4
CUB	312	150/50	11788	4	1e-5	0
SUN	102	645/72	14340	12	1e-6	0

For AWA1, CUB and SUN, the hyper-parameters are determined through a train-validation split of seen classes and are used to train the model on complete data. For AwA2 and aPY, we use the same hyper-parameters as AwA1 because of the similarity of the three datasets. The adjustment of hyper-parameters in our method is not complicated and they follow some rules: The best value of K is roughly positively correlated with the number of seen classes M. And our experiments show that a slight increase in K will bring some redundant parameters to the network, but has little impact on the results. α, β are the coefficients of regularization terms $||\mathbf{w}_f||_2^2$, $||\mathbf{w}_g||_2^2$ respectively. For fine-grained classification datasets CUB and SUN, large β causes under-fitting of $g(*)$, hence it is set to 0. Similarly, SUN with a large class size requires a smaller α to avoid under-fitting of $f(*)$.

Besides, the number of hidden units of relation module for each dataset is set to 2048, which is considered large enough. All models are trained at a learning rate of 10^{-5} with Adam optimizer until the loss converge.

4.2. Comparison of Results

We evaluate the average per-class top-1 accuracy on both seen classes (denoted as As) and unseen classes (denoted as Au) of each dataset, and calculate the harmonic mean of the two accuracy as Xian et al. (2017) suggest to evaluate the overall performance of the model. Let H denote harmonic mean and it can be expressed as $\mathrm{H} = (2 \times \mathrm{Au} \times \mathrm{As})/(\mathrm{Au} + \mathrm{As})$. Table 1 reports the results of different methods on the five datasets following the GZSL settings.

The result shows that our method outperforms recent GZSL methods on the overall evaluation metric H. Especially, RN (relation network) can be seen as a contrast to our network because they share the same relation module and differ only in the implementation of projection $f(*)$: RN uses a two-layer MLP while CRnet uses a cooperation module. Obviously, Au of CRnet has a significant improvement compared with RN, which proves that cooperation module is quite effective in solving the bias problem. Besides, SE-GZSL and f-CLSWGAN use generative models to solve the bias problem so that its Au increases a lot. But the generative model also leads to an obvious decrease on As. While our model learns a projection through seen features directly just like most CZSL methods, leading to a small decrease on As. Therefore, compared with SE-GZSL, our approach obtains a comparable Au (52.6%) with a much more higher As (78.8%) on AwA2. On AwA1, CUB and aPY, our approach even achieves the highest Au (58.1%, 45.5%, 32.4% respectively). The good performance on both Au and As finally results in a high H, which indicates that the model maintains a good balance between seen classes and unseen classes and our method truly takes effects.

But embedding models like CRnet also have a drawback. CRnet is highly dependent on the correct classification of seen class samples. Hence, it is limited by the seen samples and Au is usually no higher than As. While generative models that generate synthesized unseen samples can adjust the weights of unseen and seen classes and is less constrained. The result on SUN is a good embodiment: the classification accuracy on seen classes is quite low (36.5%), resulting in CRnet's even lower accuracy on unseen classes (34.1%). While generative model like f-CLSWGAN achieves a higher Au (42.6%) than As (36.6%) and outperforms CRnet on H.

5. Analysis

It seems that CRnet has achieved a very good effect on GZSL tasks only by replacing the two-layer MLP in RN with several parallel single-layer percptrons. But why is this effective and how does CRnet solve the bias problem? We explain its mechanism and analyze the reasons why it works from the following aspects.

5.1. More Discriminative Features

We visualize the feature anchors of CRnet and RN respectively and find that the former is much more sparse than the latter, as shown in Figure 4. In the subsequent similarity calculation and classification process, only those dimensions with response will take effect. In other words, stronger sparsity means more discriminative features. The single-layer cooperate module in CRnet has learned the most discriminative parts while the two-layer MLP in RN has learned many redundant features which only exist in the training set and cause disturbance to the test set. This is similar to the over-confident phenomenon of multi-layer networks (Guo et al., 2017). We believe single-layer networks have some advantages over multi-layer networks in tasks like GZSL. Compared with RN, more sparse and discriminative features is the direct reason why CRnets accuracy on seen classes decreases while the accuracy on unseen classes increases.

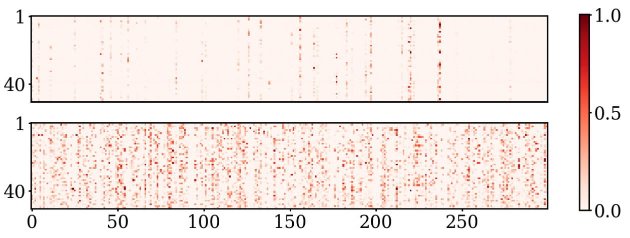

Figure 4. Visualization of the feature anchors of CRnet (above) and RN (below) well-trained on AwA2. The first 300 dimensions' normalization results of the feature anchors of 40 unseen classes and 10 seen classes are presented.

5.2. More Uniform Embedding Space

Such a parallel structure of CRnet allows different expert modules to learn differently and are sensitive to different inputs. In order to have a better understanding of the cooperation process, we visualize the response to different semantic vectors of each expert module (Figure 5). Obviously each expert module has a weak response and even no response to semantic vectors around its corresponding field center $\bar{s}_k$ (Figure 5.(a)(b)(c)). While for semantic vectors located in the center of the space, each expert module has response with parts of units actived (Figure 5.(d)).

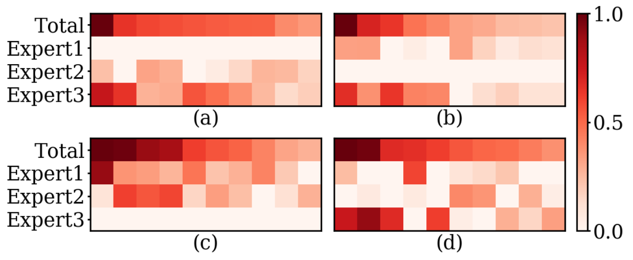

Figure 5. Responses of each expert module of CRnet trained on AwA2 to different inputs, (a) : $\bar{s}_1$, (b) : $\bar{s}_2$, (c) : $\bar{s}_3$, (d) : $\frac{1}{3}(\bar{s}_1 + \bar{s}_2 + \bar{s}_3)$. Total: Totoal response, i.e. preditced feature anchor. The results are normalized and sorted in descending order of each dimensions of total response, we show the ten dimensions with the strongest responses.

The result is consistent with the state we expect to achieve in Figure 2(b). Hence the projection of any semantic vector to visual embedding space is done as well as constrained by multiple parties. In this process, ReLU plays an important role. Intuitively, it resembles a set of switches which determines whether the expert module responds. Mathematically, for each dimension of the embedding space, cooperation module constructs a piecewise linear function with $K+1$ pieces through ReLU. In this way, the local linearity of $f(*)$ increases without a heavy decrease of global nonlinearity. High local linearity ensures that $\widehat{\mathbf{v}}^u_n$ is not too close to $\widehat{\mathbf{v}}^s_m$ and leads to a larger LRD. We have a simple mathematical proof for this as follows.

Figure 6 shows two different fitting curves f_{CR} and f_{G} for the three ponints in Figure 1. On $[s^s_1, s^s_2]$, f_{CR} is monotone increasing and linear whereas f_{G} is convex. Suppose $v_* = 1/2 \cdot (\widehat{v}^s_{1q} + \widehat{v}^s_{2q})$, $s_* = 1/2 \cdot (s^s_1 + s^s_2)$, then $f_{\mathrm{CR}}(s_*) = v_*$. Suppose $f_{\mathrm{G}}(s_*) = v'_*$, obvioulу, $v'_* < v_*$. Then, $\forall s^u_1 \in [s^s_1, s_*]$, $\widehat{v}^u_{1q} = f_{\mathrm{CR}}(s^u_1) > f_{\mathrm{G}}(s^u_1) = \widehat{v}^{u'}_{1q}$ is always satisfied. According to Eq (4), in this 1-d to 1-d case, LRD can be easily caculated through:

$$\mathrm{LRD}(\widehat{v}^u_{1q}) = \frac{\min(\widehat{v}^u_{1q} - \widehat{v}^s_{1q}, \widehat{v}^s_{2q} - \widehat{v}^u_{1q})}{\widehat{v}^s_{2q} - \widehat{v}^s_{1q}} \qquad (14)$$

So, $\forall s^u_1 \in [s^s_1, s_*]$, $\mathrm{LRD}(\widehat{v}^u_{1q}) > \mathrm{LRD}(\widehat{v}^{u'}_{1q})$. In general, we consider $\mathbf{s}^u_n$ to be uniformly distributed in the semantic space. Hence, $\forall s^u_1 \in [s^s_1, s^s_2]$,

$$p\left(\mathrm{LRD}(\widehat{v}^u_{1q}) > \mathrm{LRD}(\widehat{v}^{u'}_{1q})\right) > \frac{s_* - s^s_1}{s^s_2 - s^s_1} = 0.5 \qquad (15)$$

where $p(*)$ is the probability. Similarly, there is the same conclusion when f_{G} is concave or the two functions are monotone decreasing on $[s^s_1, s^s_2]$. Eq (15) indicates that f_{CR} is more likely to get feature anchors with large LRD. In other words, high local linearity between two points leads to less bias problem. The conclusion can be extended to the case of P-d to Q-d according to Eq (3).

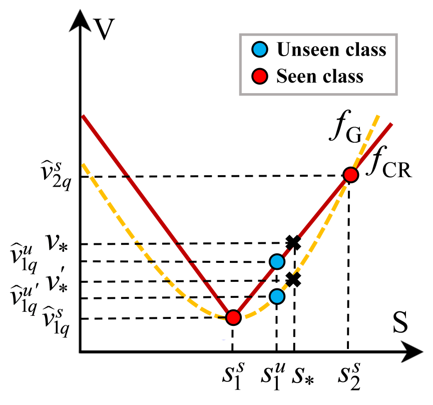

Figure 6. Examples of two different fitting curves for the three points in Figure 1. S: Semantic space, V: Embedding space.

The representation ability of single-layer model is worse than that of the multi-layer model. But CRnet adopts a parallel stucture to avoid this problem and learns a piecewise linear projection similar to f_{CR} in Figure 6 with large LRD. Larger LRD means a more uniform embedding space, so that bias problem can be alleviated to a certain extent. This is the root reason of CRnet's good performance on GZSL.

We also calculate the average LRD of all unseen classes for CRnet and RN on three datasets and report the result in Table 3. The contrast proves that CRnet's embedding space is more uniform than RN's. CRnet outperforms RN on the average LRD on all three datasets, consistent with Au. What's more, when the relative difference between Au of the two networks is smaller, the difference in the average LRD is more likely to be smaller too, which indicates that the proposed LRD is a reasonable metric for measuring the bias problem.

Table 3. Average LRD of all unseen class anchors for RN and CRnet on various datasets.

	AwA1	**AwA2**	**CUB**
RN	0.711	0.756	0.831
CRnet	0.835	0.956	0.843

5.3. Learnable Relation Module

Using a Relation module to implement the similarity function $g(*)$ is also an essential part of our algorithm. The simplest form of $g(*)$ for the final classification is the cosine distance, which is widely used in ZSL algorithm. It is easy to implement and does not introduce extra parameters to the network. But such a fixed pre-specified similarity function is highly dependent on the linear separability of the visual embedding space and leads to sharp drop in classification performance. In contrast, relation module learns a non-linear similarity function in a data driven day and is capable of self-adapting to the embedding space so that it performs a better classification (Sung et al., 2018).

In general, CRnet learns a better visual representation with more discriminative features and a more uniform embedding space with less bias problem by a cooperative approach, so that it achieves better performance on GZSL. Thus we name it "Co-Representation Network".

6. Conclution

We propose the Local Relative Distance metric to analyze the bias problem in GZSL and design a novel embedding model called co-representation network to solve it. It adopts a single-layer cooperation module with parallel structure to learn a more uniform embedding space with better representation. Its high local linearity proves to alleviate the bias problem effectively. The model outperforms existing GZSL methods and we believe it can provide some inspiration for other transfer learning tasks.

Acknowledgements

This work is supported by Natural Science Foundation (NSF) of China (Nos. 61632019, 61836008, 61572387, 61871304, 61672404), the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (No. 61621005), and the Fundamental Research Funds for the Central Universities (No. JB181702).

References

Akata, Z., Reed, S., Walter, D., Lee, H., and Schiele, B. Evaluation of output embeddings for fine-grained image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2927–2936, 2015.

Akata, Z., Perronnin, F., Harchaoui, Z., and Schmid, C. Label-embedding for image classification. *IEEE transactions on pattern analysis and machine intelligence*, 38(7): 1425–1438, 2016.

Bucher, M., Herbin, S., and Jurie, F. Generating visual representations for zero-shot classification. In *International Conference on Computer Vision (ICCV) Workshops: TASK-CV: Transferring and Adapting Source Knowledge in Computer Vision*, 2017.

Changpinyo, S., Chao, W.-L., Gong, B., and Sha, F. Synthesized classifiers for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5327–5336, 2016.

Chao, W.-L., Changpinyo, S., Gong, B., and Sha, F. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *European Conference on Computer Vision*, pp. 52–68. Springer, 2016.

Farhadi, A., Endres, I., Hoiem, D., and Forsyth, D. Describing objects by their attributes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 1778–1785. IEEE, 2009.

Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Mikolov, T., et al. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pp. 2121–2129, 2013.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330, 2017.

Kodirov, E., Xiang, T., and Gong, S. Semantic autoencoder for zero-shot learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4447–4456. IEEE, 2017.

Lampert, C. H., Nickisch, H., and Harmeling, S. Learning to detect unseen object classes by between-class attribute transfer. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 951–958. IEEE, 2009.

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

Liu, S., Long, M., Wang, J., and Jordan, M. I. Generalized zero-shot learning with deep calibration network. In *Advances in Neural Information Processing Systems*, pp. 2009–2019, 2018.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119, 2013.

Mishra, A., Reddy, S. K., Mittal, A., and Murthy, H. A. A generative model for zero shot learning using conditional variational autoencoders. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2269–22698. IEEE, 2018.

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

Patterson, G. and Hays, J. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 2751–2758. IEEE, 2012.

Radovanović, M., Nanopoulos, A., and Ivanović, M. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11(Sep): 2487–2531, 2010.

Romera-Paredes, B. and Torr, P. An embarrassingly simple approach to zero-shot learning. In *International Conference on Machine Learning*, pp. 2152–2161, 2015.

Scheirer, W. J., Rocha, A. D. R., Sapkota, A., and Boult, T. E. Toward open set recognition. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 35(7):1757–1772, 2013.

Shigeto, Y., Suzuki, I., Hara, K., Shimbo, M., and Matsumoto, Y. Ridge regression, hubness, and zero-shot learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 135–151. Springer, 2015.

Socher, R., Ganjoo, M., Manning, C. D., and Ng, A. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, pp. 935–943, 2013.

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., and Hospedales, T. M. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, 2018.

Verma, V. K., Arora, G., Mishra, A., and Rai, P. Generalized zero-shot learning via synthesized examples. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pp. 4, 2018.

Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., and Perona, P. Caltech-ucsd birds 200. 2010.

Xian, Y., Akata, Z., Sharma, G., Nguyen, Q., Hein, M., and Schiele, B. Latent embeddings for zero-shot classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 69–77, 2016.

Xian, Y., Lampert, C. H., Schiele, B., and Akata, Z. Zero-shot learning - a comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, PP(99):1–1, 2017.

Xian, Y., Lorenz, T., Schiele, B., and Akata, Z. Feature generating networks for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. Deep sets. In *Advances in neural information processing systems*, pp. 3391–3401, 2017.

Zhang, L., Xiang, T., and Gong, S. Learning a deep embedding model for zero-shot learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3010–3019. IEEE, 2017.

Zhang, Z. and Saligrama, V. Zero-shot learning via semantic similarity embedding. In *Proceedings of the IEEE international conference on computer vision*, pp. 4166–4174, 2015.