# Discriminative Learning of Latent Features for Zero-Shot Recognition

Yan Li [1,2], Junge Zhang [1,2], Jianguo Zhang [3], Kaiqi Huang [1,2,4]
[1] CRIPAC & NLPR, CASIA [2] University of Chinese Academy of Sciences
[3] Computing, School of Science and Engineering, University of Dundee, UK
[4] CAS Center for Excellence in Brain Science and Intelligence Technology
yan.li@cripac.ia.ac.cn, jgzhang@nlpr.ia.ac.cn, j.n.zhang@dundee.ac.uk, kqhuang@nlpr.ia.ac.cn

## Abstract

*Zero-shot learning (ZSL) aims to recognize unseen image categories by learning an embedding space between image and semantic representations. For years, among existing works, it has been the center task to learn the proper mapping matrices aligning the visual and semantic space, whilst the importance to learn discriminative representations for ZSL is ignored. In this work, we retrospect existing methods and demonstrate the necessity to learn discriminative representations for both visual and semantic instances of ZSL. We propose an end-to-end network that is capable of 1) automatically discovering discriminative regions by a zoom network; and 2) learning discriminative semantic representations in an augmented space introduced for both user-defined and latent attributes. Our proposed method is tested extensively on two challenging ZSL datasets, and the experiment results show that the proposed method significantly outperforms state-of-the-art methods.*

## 1. Introduction

In recent years, zero-shot learning (ZSL) has gained its popularity in object recognition task [1, 8, 9, 10, 12, 13, 15, 28]. Unlike traditional object recognition methods that seek to predict the presence of an object instance by assigning its image label as one of the categories *seen* in the training set, zero-shot learning aims to recognize an object instance from a new category *never seen* before. Therefore, in the ZSL task, the seen categories in the training set and the unseen categories in the test set are disjoint. Typically, the descriptors of categories (*e.g.* user-defined attribute annotations [1, 13], the text descriptions of the categories [20], the word vectors of the class names [6, 17], *etc.*) are provided for both seen and unseen classes; some of those descriptors are shared between categories. Those descriptors are often called *side information* or *semantic* representations. In this work, we focus on learning for ZSL with attributes.

As shown in Figure 1, a general assumption under the typical ZSL methods is that there exists a shared embedding space, in which a mapping function, $F(x, y; \mathbf{W}) = \phi(x)^T \mathbf{W} \psi(y)$, is defined to measure the *compatibility* between the image features $\phi(x)$ and the semantic representations $\psi(y)$ for both seen and unseen classes. $\mathbf{W}$ is the visual-semantic mapping matrix to be learned. Existing approaches of ZSL mainly focus on introducing linear or non-linear modelling methods, utilizing various optimization objectives and designing different specific regularization terms to learn the visual-semantic mapping, more specially, to learn $\mathbf{W}$ for ZSL.

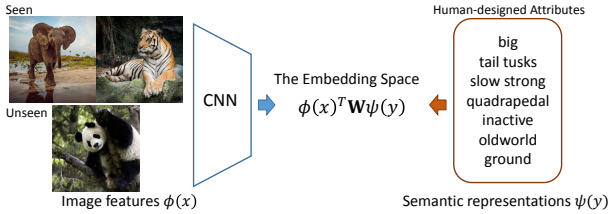


Figure 1. The typical ZSL approaches aim to find an embedding space where the image features $\phi(x)$ and semantic representations $\psi(y)$ are embedded.

To date, the learning of the mapping matrix $\mathbf{W}$, though important to ZSL, is mainly driven by minimizing the alignment loss between the visual and semantic space. However, the final goal of ZSL is to classify unseen categories. Therefore, the visual features $\phi(x)$ and semantic representations $\psi(y)$, should arguably be *discriminative* to recognize different objects. Unfortunately, this issue has been thus far neglected in ZSL and almost all the methods follow the same paradigm: 1) extracting image features by hand-crafting or using pre-trained CNN models; and 2) utilizing the human-designed attributes as the semantic representations. There are some pitfalls existed in this paradigm.

Firstly, the image features $\phi(x)$ either crafted manually or from a pre-trained CNN model may be not representative enough for zero-short recognition task. Though the features from a pre-trained CNN model are learned, yet restricted to a fixed set of images (*e.g.*, ImageNet [22]), which is not optimal for a particular ZSL task.

Secondly, the user-defined attributes $\psi(y)$ are semantically descriptive, but they are not exhaustive, thus limiting its discriminativeness in classification. There may exist discriminative visual clues not reflected by the pre-defined attributes in ZSL datasets, *e.g*., the huge mouths of *hippos*. On the other hand, as shown in Figure 1, the annotated attributes, such as *big*, *strong* and *ground*, are shared in many object categories. This is desired for knowledge transfer between categories, especially from seen to unseen categories. However, if two categories (*e.g*. *cheetah* and *tiger*) share too many (user-defined) attributes, they will be hardly distinguishable in the space of attribute vectors.

Thirdly, low-level feature extraction and embedding space construction in existing ZSL approaches are treated separately, and usually carried out in isolation. Therefore, few existing work ever considers those two components in a unified framework.

To address those pitfalls, we propose an end-to-end model capable of learning latent discriminative features (LDF) for ZSL in both visual and semantic space. Specifically, our contributions are:

1) A cascaded *zooming* mechanism to learn features from object-centric regions. Our model can automatically identify the most discriminative region in an image and then zoom it into a larger scale for learning in a cascaded network structure. In this way, our model can concentrate on learning features from a region with object as a focus.

2) A framework to jointly learn the latent attributes and the user-defined attributes. We formulate the learning of latent attributes as a category-ranking problem to ensure the learned attributes are discriminative. Meanwhile, the discriminative region mining and the latent attributes modelling are jointly learned in our model and assist each other to gain further improvement.

3) An end-to-end network structure for ZSL. The obtained image features can be regulated to be more compatible with the semantic space, which contains both the user-defined attributes and latent discriminative attributes.

## 2. Related Work

Early works of zero-shot learning (ZSL) follow an intuitive way to object recognition that first trains different *attribute classifiers* and then recognizes an image by comparing its predicted attributes with descriptions of unseen classes [5, 13]. Among these works, Direct Attribute Prediction (DAP) model [14] predicts the posterior of each attribute, and then the class posteriors for an image are calculated by maximizing a posterior. Whilst in Indirect Attribute Prediction (IAP) [14] model, the attribute posteriors are computed from the class posterior of seen classes. In these methods, each attribute classifier is trained individually and the relationship between attributes for a class is not considered.

To address this issue, most of recent ZSL works are *embedding-based* methods, which seek to build a common embedding space for images and their semantic features. The DeViSE model [6] and the ALE model [1] are based on a bilinear embedding model, where a linear transformation matrix $\mathbf{W}$ is learned with a hinge ranking loss. The ESZSL model [21] adds a Frobenius norm regularizer into the embedding space construction. The SJE model [2] combines several compatibility functions linearly to form a joint embedding space. The LatEM model [27] improves SJE with more nonlinearity by incorporating latent variables. Recently, the SCoRe model [16] adds a semantically consistent regularization to make the learned transformation matrix perform better on test images. The MFMR model [29] learns the projection matrix by decomposing the visual feature matrix. The majority of ZSL methods thus far extract image features from whole image with fixed pre-trained CNN models. In contrast, image features in our model are learned to be more representative with the mining of latent discriminative regions and the end-to-end training style.

In typical embedding space construction approach, only the space of user-defined attributes is used to embed the seen and unseen classes. Different from this, the JSLA model [18, 19] and the LAD model [11] propose to model latent attributes for ZSL, which are similar to our work. JSLA learns latent discriminative attributes by minimizing the intra class distance between the attributes; while in LAD the discriminativeness of latent attributes is indirectly achieved by training seen class classifiers over the latent attributes. Different from them, our model proposes to directly regulate both inter-class and intra-class distances between latent attributes to achieve the discriminativeness. What's more, JSLA and LAD still utilize the fixed pre-extracted image features, which are less representative than ours.

Another branch of ZSL approaches are based on *hybrid models*, which aim to use the combination of seen classes to classify unseen images. The ConSE model [17] convexly combines the classification probabilities of seen classes to classify unseen objects. The SynC model [3] introduces synthetic classifiers of unseen classes by linearly combining the classifiers of seen classes. In our method, when the learned latent attributes are utilized for ZSL prediction, the latent attribute prototype for an unseen class is obtained by combining the prototypes of seen classes. To this end, our prediction model is among the family of hybrid models; and beyond that our model also learns embeddings for both user-defined attributes and latent attributes in one network.

## 3. Task Definition

In the zero-shot learning task, the training set, *i.e*., the *seen classes*, is defined as $\mathcal{S} \equiv \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$, where $x_i^s \in \mathcal{X}_\mathcal{S}$ is the $i$-th image of the seen class and $y_i^s \in \mathcal{Y}_\mathcal{S}$ is its corresponding class label. The test set, *i.e*., the *unseen classes*, is defined as $\mathcal{U} \equiv \{(x_j^u, y_j^u)\}_{j=1}^{n_u}$, where $x_j^u \in \mathcal{X}_\mathcal{U}$ denotes

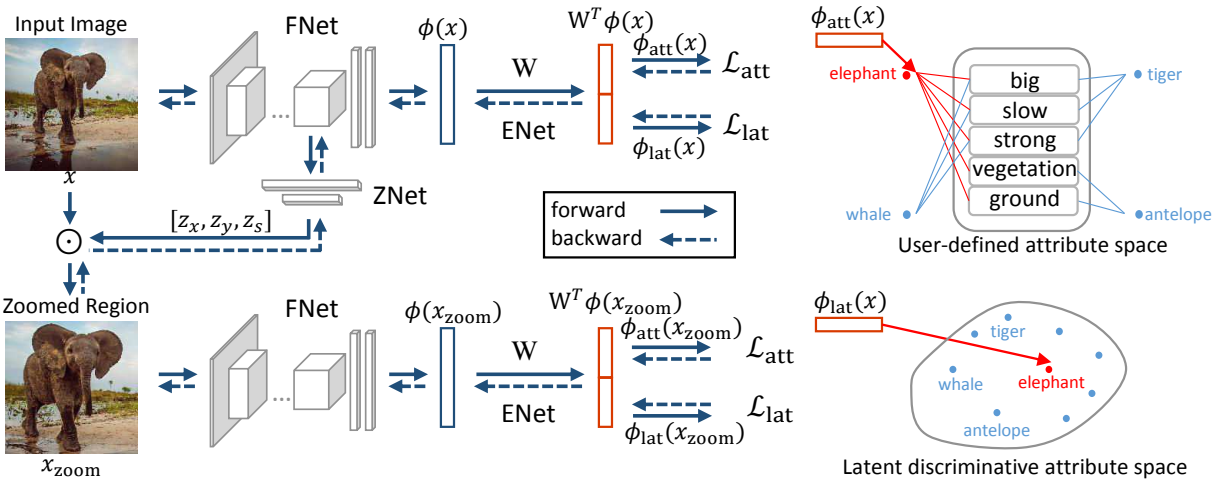


Figure 2. The framework of the proposed Latent Discriminative Features Learning (LDF) model. The coarse-to-fine image representations are projected into both user-defined attributes and latent attributes. The user-defined attributes are usually shared between different categories while the latent attributes are learned to be discriminative by regulating inter/intra class distances.

the $j$-th unseen image and $y_i^s \in \mathcal{Y}_\mathcal{U}$ is the label of it. The seen and unseen classes are disjoint, *i.e.*, $\mathcal{Y}_\mathcal{S} \cap \mathcal{Y}_\mathcal{U} = \emptyset$. Additionally, the user-defined attributes for both seen and unseen classes can be denoted as $\mathcal{A}_\mathcal{S} \equiv \{\mathbf{a}_i^s\}_{i=1}^{c_s}$ and $\mathcal{A}_\mathcal{U} \equiv \{\mathbf{a}_j^u\}_{j=1}^{c_u}$, where $\mathbf{a}_i^s$ and $\mathbf{a}_j^u$ indicate the attribute vectors for the $i$-th seen class and the $j$-th unseen class, respectively. At the test stage, given a test image $x^u$ and the attribute annotations of test classes $\mathcal{A}_\mathcal{U}$, the goal of ZSL is to predict the corresponding category $y^u$ for $x^u$.

## 4. Our Method

The framework of the proposed method is illustrated in Figure 2. Note that the architecture in principle contains multiple scales and for clarity, we illustrate the network with two scales as an example. In each scale, the network consists of three different components, 1) the image feature network (FNet) to extract image representations, 2) the zoom network (ZNet) to locate the most discriminative region and then zoom it to larger scale and 3) the embedding network (ENet) to build the embedding space where the visual and semantic information are associated. For the first scale, the input of the FNet is the image of its original size and the ZNet is responsible for producing the zoomed region. Then for the second scale, the zoomed region is fed into the FNet to obtain more discriminative image features.

### 4.1. The Image Feature Network (FNet)

Different from existing works [4, 16, 29], we would like to learn image features together with embedding for zero-shot learning. Therefore, our framework starts with a compartment of convolutional nets responsible for learning image features, which is termed as FNet. The choice of the architecture of FNet is flexible; and two possible variants are considered in our approach, *i.e.*, the VGG19 and the GoogLeNet. For VGG19, the FNet starts from conv1 to fc7; for GoogLeNet, it starts from conv1 to pool5. Given an image or a zoomed region $x$, the image representation is denoted as:

$$\phi(x) = \mathbf{W}_{\text{IF}} * x \tag{1}$$

where $\mathbf{W}_{\text{IF}}$ indicates the overall parameters of the FNet, and $*$ denotes a set of operations of the FNet. Different from traditional ZSL approaches, the parameters of FNet are jointly trained with other parts in our framework; thus the obtained features are regulated well with the embedding component. We show that this leads to an performance improvement.

### 4.2. The Zoom Network (ZNet)

The final goal of zero-shot learning is to classify different object categories. There exist studies showing that learning from object regions could benefit object categorization at image level [7, 30]. Inspired by these studies, we hypothesize that there may exist some *discriminative* regions in an image which benefit the zero-shot learning. Such a region could contain only object instance or object parts [7]. On the other hand, for ZSL, a candidate region will also need to reflect the user-defined attributes, some of which describe the background, such as *swim*, *tree* and *mountains*. Therefore, a target region is expected to contain some background to enhance the attributes embedding. We name this type of regions as *object-centric* region. To identify them, we introduce the zoom network (ZNet) that adopts an incrementally *zoom-in* approach to let the network automatically search a proper discriminative region from coarse to fine. The *proper* in ZSL task means that the target region is discriminative for classification and meanwhile matched with the annotated attributes.

Specifically, our ZNet takes the output of the last convolutional layer in the FNet (*e.g.*, conv5_4 in VGG19) as the input. For computational efficiency, the candidate region is

assumed as a square and its location can be represented with three parameters:

$$[z_x, z_y, z_s] = \mathbf{W}_{\mathrm{Z}} * \phi(x)_{\text{conv}} \tag{2}$$

where $z_x, z_y$ indicate the x-axis and y-axis coordinates for the center of the searched square, respectively, and $z_s$ represents the length of the square. The $\phi(x)_{\text{conv}}$ denotes the output of the last convolutional layer of the FNet. The ZNet is a two-stacked fully-connected layers (1024-3) followed by the sigmoid activation function and $\mathbf{W}_{\mathrm{Z}}$ denotes the parameters of the ZNet.

After obtaining the location of the square, the searched region can be obtained by directly cropping from the original image. However, it is not convenient to optimize the non-continuous cropping operation in backward-propagation. Inspired by [7], the sigmoid function is utilized to first produce a two-dim continuous mask $\mathbf{M}(\mathrm{x}, \mathrm{y})$. Formally,

$$\begin{aligned} \mathbf{M}_{\mathrm{x}} &= f(\mathrm{x} - z_x + 0.5z_s) - f(\mathrm{x} - z_x - 0.5z_s) \\ \mathbf{M}_{\mathrm{y}} &= f(\mathrm{y} - z_y + 0.5z_s) - f(\mathrm{y} - z_y - 0.5z_s) \end{aligned} \tag{3}$$

where $f(x) = 1/(1 + \exp(-kx))$ and $k$ is set to 10 in all experiments.

Then the cropped region can be obtained by implementing element-wise multiplication $\odot$ between the original image $x$ and the continuous mask $\mathbf{M}$:

$$x^{\text{crop}} = x \odot \mathbf{M} \tag{4}$$

Finally, to obtain better representation for finer localized cropped region, we further use the bilinear interpolation to adaptively zoom the cropped region to the same size with the original image. The zoomed region is then fed into a copy of the FNet in the next scale to extract more discriminative representation.

## 4.3. The Embedding Network (ENet)

### 4.3.1 The Baseline Embedding Model

The embedding network (ENet) aims to learn an embedding space where the visual and semantic information are associated. In this section, we first introduce a baseline embedding model, where the semantic representations, $\psi(y)$, is defined with the user-defined attributes $\mathcal{A}$. In this model, the mapping function to be learned is therefore defined as: $F(x, y; \mathbf{W}) = \phi(x)^T \mathbf{W} \mathbf{a}^y$.

The attribute space $\mathcal{A}$ is adopted as the embedding space and the compatibility score is defined by the inner product:

$$\mathbf{s} = \langle \mathbf{W}^T \phi(x), \mathbf{a}^y \rangle \tag{5}$$

where $\phi(x)$ is the $d$-dim image representation obtained by the FNet and $\mathbf{a}^y$ is the $k$-dim annotated attribute vector of category $y$. $\mathbf{W} \in \mathbb{R}^{d \times k}$ is the weight to learn in a fully connected layer, which can be considered as a linear project matrix that maps $\phi(x)$ to the attribute space $\mathcal{A}$.

The compatibility score measures the similarity between an image and the attribute annotations of classes. It is similar to the classification score in traditional object recognition task. Thus, to learn the matrix $\mathbf{W}$, a standard softmax loss can be used:

$$\mathcal{L} = -\frac{1}{N} \sum_{i}^{n} \log \frac{\exp(\mathbf{s})}{\sum_c \exp(\mathbf{s}^c)}, c \in \mathcal{Y}_{\mathcal{S}} \tag{6}$$

### 4.3.2 The Augmented Embedding Model

The baseline embedding model, adopted by most of existing ZSL methods, has achieved promising performance. However, it is based on user-defined attributes, which is of limited size, and usually not discriminative. To address this issue, we introduce an *augmented* attribute space, where an image is projected into both user-defined attributes (UA) and *latent* discriminative attributes (LA).

Specifically, our embedding network (ENet) learns a matrix $\mathbf{W}_{\text{aug}} \in \mathbb{R}^{d \times 2k}$ mapping the image features to a $2k$-dim augmented space, and the embedded image features $\phi_{\mathrm{e}}(x)$ are computed as follows:

$$\phi_{\mathrm{e}}(x) = \mathbf{W}_{\text{aug}}^T \phi(x), \quad \phi_{\mathrm{e}}(x) \in \mathbb{R}^{2k} \tag{7}$$

The goal is to associate the embedded image features $\phi_{\mathrm{e}}(x)$ with both the UA and the LA. For simplicity, we equally divide $\phi_{\mathrm{e}}(x)$ into two $k$-dim parts:

$$\phi_{\mathrm{e}}(x) = [\phi_{\text{att}}(x); \phi_{\text{lat}}(x)], \quad \phi_{\text{att}}(x), \phi_{\text{lat}}(x) \in \mathbb{R}^k \tag{8}$$

Then we let the first $k$-dim embedded feature $\phi_{\text{att}}(x)$ correspond to the UA and the second $k$-dim component $\phi_{\text{lat}}(x)$ being associated with the LA. Based on this assumption, for $\phi_{\text{att}}(x)$, similar to the baseline model, the softmax loss is utilized to train the ZSL model. Formally,

$$\mathcal{L}_{\text{att}} = -\frac{1}{N} \sum_{i}^{n} \log \frac{\exp(\langle \phi_{\text{att}}(x), \mathbf{a} \rangle)}{\sum_c \exp(\langle \phi_{\text{att}}(x), \mathbf{a}^c \rangle)}, c \in \mathcal{Y}_{\mathcal{S}} \tag{9}$$

For the second embedded feature $\phi_{\text{lat}}(x)$, the goal is to make the learned features be discriminative for object recognition. We propose to utilize the triplet loss [26] to learn the latent discriminative attributes with regulating the inter/intra class distances between latent attributes features:

$$\mathcal{L}_{\text{lat}} = \max(0, m + d(\phi_{\text{lat}}(x_i), \phi_{\text{lat}}(x_k)) - d(\phi_{\text{lat}}(x_i), \phi_{\text{lat}}(x_j))) \tag{10}$$

where $x_i, x_k$ are images from the same class and $x_j$ is from a different class. $d(x, y)$ is the squared Euclidean distance between $x$ and $y$. $m$ is the margin of the triplet loss and is set to 1.0 for all experiments.

From (7) and (8), it can be observed that the UA and LA features are mapped from the same image representation, but with two different matrices:

$$\begin{aligned}\phi_{\text{att}}(x) &= \mathbf{W}_{\text{att}}^T\phi(x),\\ \phi_{\text{lat}}(x) &= \mathbf{W}_{\text{lat}}^T\phi(x), \quad [\mathbf{W}_{\text{att}};\mathbf{W}_{\text{lat}}] = \mathbf{W}_{\text{aug}}\end{aligned} \tag{11}$$

It is noted that $\mathbf{W}_{\text{att}}$ and $\mathbf{W}_{\text{lat}}$ are associated with different loss functions. $\phi_{\text{lat}}$ can be *learned* to be discriminative by specifically exploiting the category information in (10).

For each scale, the network is trained with both the softmax loss and the triplet loss. For a two-scale network (*i.e.*, $s1$ and $s2$), the whole LDF model is trained by the following loss function:

$$\mathcal{L} = \mathcal{L}_{\text{att}}^{s1} + \mathcal{L}_{\text{lat}}^{s1} + \mathcal{L}_{\text{att}}^{s2} + \mathcal{L}_{\text{lat}}^{s2} \tag{12}$$

The final objective function for a multi-scale network could be constructed similarly by aggregating all the loss functions of all of scales.

### 4.4. ZSL Prediction

In the proposed LDF model, the test images can be projected into both user-defined attributes (UA) and latent attributes (LA) as in (7). Thus, ZSL prediction can be performed in both the UA space and the LA space.

**Prediction with UA.** Given a test image $x$, it can be projected to the UA representation $\phi_{\text{att}}(x)$. To predict its class label, the compatibility scores can be used to select the most matched unseen categories:

$$y^* = \arg\max_{c\in\mathcal{Y}_\mathcal{U}}(\mathbf{s}^c) = \arg\max_{c\in\mathcal{Y}_\mathcal{U}}\langle\phi_{\text{att}}(x),\mathbf{a}^c\rangle \tag{13}$$

**Prediction with LA.** The test image $x$ can also be projected to the LA representation, $\phi_{\text{lat}}(x)$. To perform ZSL in the LA space, the LA prototypes for unseen classes are required.

Firstly, the LA prototypes for seen classes are computed. Concretely, all samples $x_i$ from the seen class $s$ are projected to their LA features and the mean of features are utilized as the LA prototype of class $s$, *i.e.*, $\overline{\phi_{\text{lat}}^s} = \frac{1}{N}\sum_i \phi_{\text{lat}}(x_i)$.

Then, for an unseen class $u$, we compute the relationship between class $u$ and all the seen classes $\mathcal{S}$ in the UA space. This relationship can be obtained by solving the following ridge regression problem:

$$\beta_c^u = \arg\min \|\mathbf{a}^u - \sum \beta_c^u\mathbf{a}^c\|_2^2 + \lambda\|\beta_c^u\|_2^2, \quad c\in\mathcal{Y}_\mathcal{S} \tag{14}$$

By applying the same relationship to the LA space, the prototype for unseen class $u$ can be obtained:

$$\overline{\phi_{\text{lat}}^u} = \sum \beta_c^u\overline{\phi_{\text{lat}}^c}, \quad c\in\mathcal{Y}_\mathcal{S} \tag{15}$$

Finally, the classification result of a test image $x$ with LA representation $\phi_{\text{lat}}(x)$ can be achieved as following:

$$y^* = \arg\max_{c\in\mathcal{Y}_\mathcal{U}}\langle\phi_{\text{lat}}(x),\overline{\phi_{\text{lat}}^c}\rangle \tag{16}$$

**Combining multiple spaces.** We can consider both the UA and LA spaces and utilize the concated UA-LA feature $[\phi_{\text{att}}(x);\phi_{\text{lat}}(x)]$ to perform ZSL prediction. Formally,

$$\begin{aligned}y^* &= \arg\max_{c\in\mathcal{Y}_\mathcal{U}}(\langle[\phi_{\text{att}}(x);\phi_{\text{lat}}(x)],[\mathbf{a}^c;\overline{\phi_{\text{lat}}^c}]\rangle)\\ &= \arg\max_{c\in\mathcal{Y}_\mathcal{U}}(\langle\phi_{\text{att}}(x),\mathbf{a}^c\rangle + \langle\phi_{\text{lat}}(x),\overline{\phi_{\text{lat}}^c}\rangle)\end{aligned} \tag{17}$$

**Combining multiple scales.** For a two-scale LDF model (*i.e.*, $s1$ and $s2$). The UA and LA features are obtained in each scale, and the obtained multi-scale features can be combined to gain further improvement.

For multi-scale UA features, *i.e.*, $\phi_{\text{att}}^{s1}, \phi_{\text{att}}^{s2}$, we first concatenate the two features $[\phi_{\text{att}}^{s1};\phi_{\text{att}}^{s2}] \in \mathbb{R}^{2k}$, and then train a new project matrix $\mathbf{W}_{\text{com}} \in \mathbb{R}^{2k\times k}$ to obtain the combined UA feature, *i.e.*, $\phi_{\text{att}}^{\text{com}} = \mathbf{W}_{\text{com}}^T[\phi_{\text{att}}^{s1};\phi_{\text{att}}^{s2}]$. For multi-scale LA features, *i.e.*, $\phi_{\text{lat}}^{s1}, \phi_{\text{lat}}^{s2}$, the combined feature can be obtained by directly concatenating the normalized two features, *i.e.*, $\phi_{\text{lat}}^{\text{com}} = [\widehat{\phi_{\text{lat}}^{s1}};\widehat{\phi_{\text{lat}}^{s2}}]$. Finally, the ZSL prediction can be performed using (17) with the combined UA feature $\phi_{\text{att}}^{\text{com}}$ and the combined LA feature $\phi_{\text{lat}}^{\text{com}}$.

## 5. Experiments

### 5.1. Datasets

The proposed LDF model is evaluated on two representative ZSL benchmarks: Animals with Attributes (AwA) [14] and Caltech-UCSD Birds 200-2011 (CUB) [25]. AwA includes 30,475 images from 50 common animals categories. The 85 class-level attributes (continuous) and the standard 40/10 zero-shot split are adopted in our experiments. The dataset of CUB is a fine-grained bird dataset with 200 different birds and 11,788 images. Following SynC [3], we use a split of 150/50 for zero-shot learning and utilize 312-dim attribute vectors at class level.

### 5.2. Implementation Details

The FNets are initialized using two different CNN models pre-trained on ImageNet, *i.e.*, GoogLeNet [24] and VGG19 [23] respectively, to learn, $\phi(x)$. For AwA, only one zoom operation is performed and the LDF model contains **two** scales, as the objects in AwA images are usually large and centered [1]; for CUB, the LDF model includes **three** scales with two zoom-in operations (*i.e.*, having two ZNets). In each scale, the size of each input image or zoomed region is 224×224, following the same setting as the existing ZSL methods. During training, the LDF model is trained for 5 epoches for AwA and 20 epoches for CUB. The learning rates of GoogLeNet and VGG19 are *fixed* and set to 0.0005 and 0.0001, respectively throughout all of the experiments. At the test stage, $\lambda$ in (14) is set to 1.0 for all datasets.

[1] In supplementary materials, we will show that if we use three scales on AwA, the third scale is actually **useless** for object recognition.

**Training strategy**: We first adopt the strategy used in [7] to initial the ZNet. Then the other components in the LDF model are learned. The detailed process is as follows:

**Step 1**: The FNet in each scale is initialized with the *same* GoogLeNet (or VGG19) pre-trained on ImageNet. Notice that in the subsequent steps of training, the parameters in each scale are *not* shared.

**Step 2**: In each scale, the initialized FNet is utilized to search a discriminative square, which is then used to pre-train the ZNet. The size of the searched square is assumed to be the half size of the original image (*i.e.*, $z_s = 0.5$). Then we slide over the last convolutional layer in the FNet and select the region with the highest activations. Finally, the coordinates of the searched region ($[z_x, z_y, z_s]$) are utilized to train the zoom net with L2 loss.

**Step 3**: We keep the parameters of the ZNet fixed and train both the FNet and the ENet.

**Step 4**: Finally, the parameters of the whole LDF model are fine-tuned in an end-to-end approach.

### 5.3. Baselines

To verify the effectiveness of the different components in our LDF model, four baselines are designed to compare with the proposed LDF model.

- **SS-BE-Fixed** (Single Scale & Baseline Embedding Model & Fixed Image Representations). In this baseline, the ZNet is removed, and only the full-size images are utilized to extract image features. Moreover, the FNet is *fixed* during the training. For semantic representations, only the user-defined attributes are considered (Section 4.3.1).
- **SS-BE-Learned** (Single Scale & Baseline Embedding Model & Learned Image Representations). Compared with the SS-BE-Fixed baseline, the only difference is that the FNet *can be learned* in this baseline.
- **SS-AE-Learned** (Single Scale & Augmented Embedding Model & Learned Image Representations). Compared with the SS-BE-Learned baseline, this baseline aims to build the *augmented* embedding space (Section 4.3.2) with considering both UA and LA.
- **MS-BE-Learned** (Multi Scale & Baseline Embedding Model & Learned Image Representations). Compared with the SS-BE-Learned baseline, the only difference is the ZNet is added into this model (Section 4.2).

### 5.4. Experimental Results

The multi-way classification accuracy (MCA) is used for evaluating the ZSL models. The comparison results using two different CNN models are shown in Table 1.

**Effect of feature learning.** From Table 1, we first notice that, without any specially designed regularization terms, the SS-BE-Learned baseline has already achieved comparable performance with state-of-the-arts and marginally surpass the SS-BE-Fixed baseline. Most of existing ZSL methods use the fixed image feature and only focus on learning visual-semantic mapping with various human-designed regularization terms. We show that feature learning neglected in image feature extraction process is also important to ZSL, which should be paid more attentions. By simply fine-tuning the FNet in an end-to-end framework, SS-BE-Learned can make the image features associate with the semantic information of attributes for different ZSL tasks and obtain better performance.

Table 1. ZSL results (MCA, %) on all the datasets using the deep features of VGG19 and GoogLeNet (numbers in parentheses).

| Method | AwA | CUB |
|---|---|---|
| DAP [13] | 57.2 (60.5) | 44.5 (39.1) |
| ESZSL [21] | 75.3 (59.6) | - (44.0) |
| SJE [2] | - (66.7) | - (50.1) |
| LatEM [27] | - (71.9) | - (45.5) |
| SynC [3] | - (72.9) | - (54.5) |
| JLSE [31] | 80.46 (-) | 42.11 (-) |
| MFMR [29] | 79.8 (76.6) | 47.7 (46.2) |
| Low-Rank [4] | 82.8 (76.6) | 45.2 (56.2) |
| SCoRe [16] | 82.8 (78.3) | 59.5 (58.4) |
| LAD [11] | 82.48 (-) | 56.63 (-) |
| JSLA [19] | 82.9 (-) | 57.1 (-) |
| SS-BE-Fixed (Ours) | 75.20 (73.70) | 50.51 (50.31) |
| SS-BE-Learned (Ours) | 79.35 (75.19) | 59.32 (58.26) |
| SS-AE-Learned (Ours) | 81.36 (77.77) | 65.99 (66.96) |
| MS-BE-Learned (Ours) | 81.80 (78.31) | 64.85 (64.39) |
| LDF (Ours) | **83.40** (79.13) | 67.12 (**70.37**) |

**Effect of ZNet.** The MS-BE-Learned baseline aims to use the ZNet to automatically discover discriminative regions from full-size images and leverage the coarse-to-fine representations to obtain better performance. We can see that the performance of MS-BE-Learned baseline outperforms both the SE-BE-Learned baseline and most of the state-of-the-art methods (Table 1, 81.80% on AwA, 64.85% on CUB).

We further analyze the performance of each scale in MS-BE-Learned model, and show the results in Table 2. It can be seen that, the performance of the first scale, *i.e.*, MS-BE-Learned (Scale 1), is comparable with the single scale baseline, SS-BE-Learned. With more discriminative image features utilized, the performance of the second and the third scale improves continuously.

**Effect of the latent attribute modelling.** The SS-AE-Learned baseline aims to build an augmented embedding space. It is more reasonable to associate image features with both user-defined and latent attributes in our augmented space. It can be observed from Table 1 that the SS-AE-Learned model outperforms SE-BE-Learned baseline for

Table 2. The detailed ZSL results (%) on each scale.

| Method | AwA | CUB |
|---|---|---|
| SS-BE-Learned | 79.35 (75.19) | 59.32 (58.26) |
| MS-BE-Learned (Scale 1) | 79.20 (75.68) | 59.88 (58.87) |
| MS-BE-Learned (Scale 2) | 79.87 (77.02) | 61.04 (61.81) |
| MS-BE-Learned (Scale 3) | - (-) | 62.04 (62.72) |
| MS-BE-Learned (All Scale) | 81.80 (78.31) | 64.85 (64.39) |

MS-BE-Learned (Scale X) denotes the ZSL results using the image features of scale X only.

Table 3. ZSL results (%) with UA features or LA features only.

| Method | AwA | CUB |
|---|---|---|
| SS-BE-Learned | 79.35 (75.19) | 59.32 (58.26) |
| SS-AE-Learned (UA) | 80.97 (77.24) | 62.17 (59.40) |
| SS-AE-Learned (LA) | 78.76 (75.75) | 63.08 (66.11) |
| SS-AE-Learned (UA & LA) | 81.36 (77.77) | 65.99 (66.96) |

SS-AE-Learned (UA/LA) denotes the results predicted with the UA features $\phi_{\text{att}}(x)$ only or the LA features $\phi_{\text{lat}}(x)$ only.

both AwA (81.36%) and CUB (66.96%) datasets.

We believe that, in the augmented attribute space, the learning of LA will help the learning of UA. Further experiments are conducted to verify this. The results are shown in Table 3. For SS-AE-Learned baseline, we only utilize the obtained UA representation $\phi_{\text{att}}(x)$ to perform ZSL prediction as in (13), denoted as SS-AE-Learned (UA). We can see that, when using UA features only, the performance of SS-AE-Learned (UA) is higher than the SS-BE-Learned. (*e.g.*, 80.97% *vs*. 79.35%). It proves that better UA representations are obtained in the augmented attribute space.

**Comparisons with state-of-the-art methods.** Compared with previous methods in Table 1, the LDF model improves the state-of-the-art performance on both datasets. In general, the proposed model based on VGG19 performs better on AwA, while the GoogLeNet-based model shows superiority on CUB. On AwA, our LDF achieves 83.40%, which is slightly higher than JLSA [19] (82.81%). For more challenging CUB dataset that 50 bird species need to be classified, our model obtains more obvious improvement. On CUB, the LDF model reaches 70.37%, with an impressive gain over the state-of-the-art SCoRe (from 58.4% to 70.37%).

Furthermore, the components of the latent discriminative regions mining (the ZNet) and the latent discriminative attribute modelling (the ENet) are jointly learned in the proposed LDF model. We believe the two components could assist each other in the joint learning framework. To verity this assumption, a further analysis of the LDF model is performed, and the results are shown in Table 4. It can be seen that, when using the combined UA features only to perform ZSL prediction, *i.e.*, LDF (UA), the performance of LDF is higher than the MS-BE-Learned baseline. When using the combined LA features only, the performance of the LDF (LA) also exceeds the SS-AE-Learned (LA). It confirms the advantages of the jointly learning approach.

Table 4. The comparisons between the joint training and separated training for ZNet and ENet.

| Method | AwA | CUB |
|---|---|---|
| SS-AE-Learned (LA) | 78.76 (75.75) | 63.08 (66.11) |
| LDF (LA) | 79.35 (76.84) | 66.47 (69.94) |
| MS-BE-Learned (UA) | 81.80 (78.31) | 64.85 (64.39) |
| LDF (UA) | 82.47 (78.77) | 65.94 (65.78) |
| LDF (LA & UA) | 83.40 (79.13) | 67.12 (70.37) |

LDF (LA/UA) denotes the ZSL results predicted with the combined LA features $\phi_{\text{lat}}^{\text{com}}$ only or the combined UA features $\phi_{\text{att}}^{\text{com}}$ only.

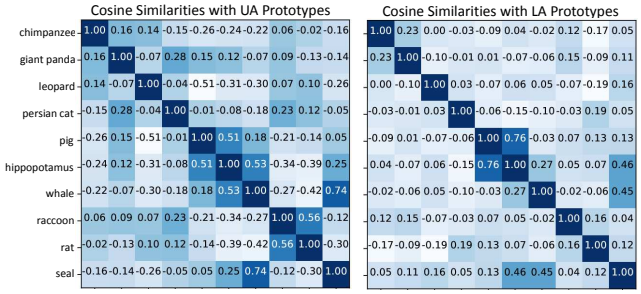


Figure 3. The cosine similarities computed with the UA (left panel) and the LA (right panel) for 10 unseen AwA classes.

**Discriminativeness of LA.** The LA features are learned to be discriminative by exploiting the category information as in (10), and we believe the learned LA space is more discriminative than the UA space. To illustrate this, we show some examples on AwA in Figure 4. The test images are projected to their UA features and LA features with (11). Then for a UA element or a LA element, the images which have largest and smallest activations of the component are shown. It can be observed that, for LA features, the images with large activations belong to one same category and the images with small activations are of the other category. In contrast, the user-defined attributes are usually shared in multiple categories. It confirms the apparent discriminative property of the learned latent attributes.

Additionally, to quantitatively compare the learned LA space with the UA space, we calculate cosine similarities between unseen classes with both the LA and UA prototypes, and the results are shown in Figure 3. The LA prototypes are obtained by directly averaging the LA features, *i.e.*, $\overline{\phi_{\text{lat}}} = \frac{1}{N}\sum_i \phi_{\text{lat}}(x_i)$, for each unseen class, and the UA prototypes are the class-level attribute annotations, *i.e.*, $\mathbf{a}^c$. It can be seen that, compared with the UA prototypes, the cosine similarities between different LA prototypes are obviously smaller for most categories, except for the *pig* and the *hippopotamus*. Compared with attributes annotated by experts, our LA prototypes are learned from the images only. Thus, the categories with similar appearances, *e.g.*, *pig vs*. *hippopotamus*, get closer in the LA space.

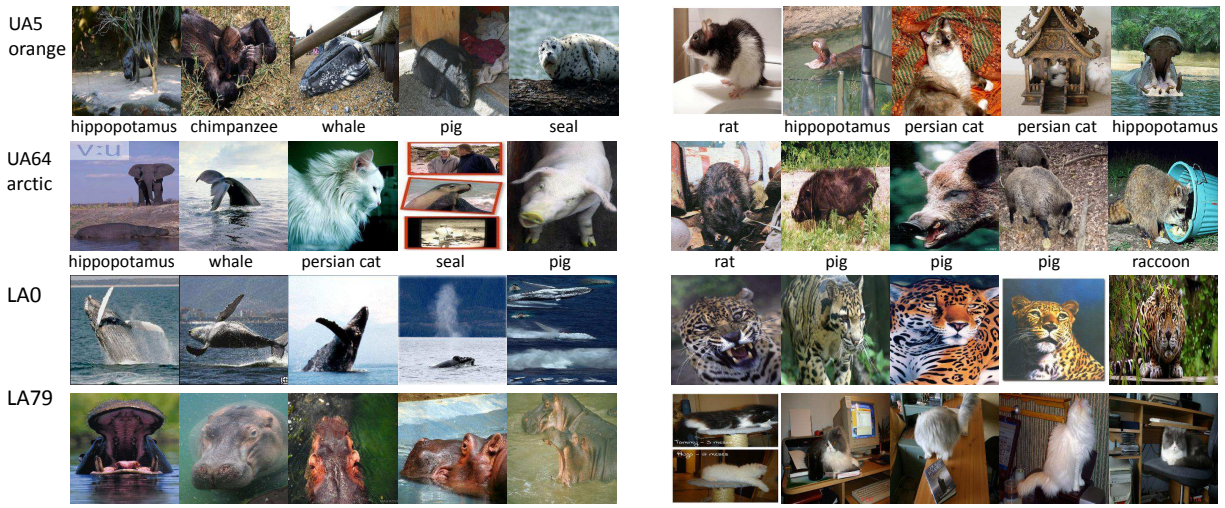


Figure 4. The visual examples on AwA with VGG19 SS-AE-Learned. 'UA/LAX' denotes the X-th element of the attribute features. In each row, the first five images are top-5 images with largest activations and the last five images are selected images with smallest activations.

It is noted that when we perform ZSL prediction with LA features, a LA representation (prototype) of a test *category* is needed, but absent in the dataset. Thus, the LA prototypes for unseen classes have to be computed with (15) leveraging the relationship $\beta_c$. However, $\beta_c$ is computed in the UA space and it cannot exactly reflect the true relationship between LA prototypes. This bias finally degrades the ZSL performance when LA prototypes are utilized for prediction with (16). This bias explains why, in Table 3, the performance of SS-AE-Learned (LA) is lower than SS-AE-Learned (UA) on AwA, although the learned LA space is actually more discriminative than the UA space.

**Visualizations of discriminative regions.** In Figure 5, we show the discovered regions with the LDF model. The left three columns show the examples selected from AwA. We can see that, for images with a single instance, the LDF model progressively searches for finer regions until it finds the main object; for images with multiple instances, the model tends to find a large square including the multiple objects. Another interesting discovery on AwA is that, for some specific categories, *e.g.*, *whale*, the identified regions will include obvious more background elements than others. The reason is that the searched regions of the *humpback whale* are required to be matched with their user-defined attributes, some of which, such as *swims*, *water* and *ocean*, highly relate to the background waters in the images.

The examples in right three columns are sampled from CUB. It is aware that the CUB dataset provides bounding box annotations, however, our model could automatically discover object-centric regions without such annotations, which shows another advantage of our framework. It is noted that, the network in [7] performs fine-grained object recognition, a different task from us; and it could discover some object parts. In contrast, in our ZSL model, the searched regions should be associated to the user-defined attributes, which, for example, correspond to the whole body of the birds from bills to tails. Thus, it is expected that the model will focus on regions containing the whole object rather than its parts; and our analysis confirms this.

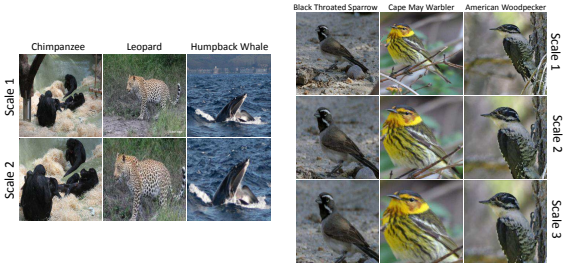


Figure 5. The examples of the learned regions at different scales.

## 6. Conclusion

In this paper, an end-to-end model is proposed to learn the latent discriminative features for ZSL in both visual and semantic space. For visual space, we introduce the zoom net to automatically search for discriminative regions. For semantic space, we propose an augmented attribute space with both the user-defined attributes and the latent attributes. The latent attributes are learned to be discriminative with category information. Finally, the two components could assist each other in the end-to-end joint learning framework.

## 7. Acknowledgement

This work is funded by the National Key Research and Development Program of China (Grant 2016YFB1001004 and Grant 2016YFB1001005), the National Natural Science Foundation of China (Grant 61673375, Grant 61721004 and Grant 61403383) and the Projects of Chinese Academy of Sciences (Grant QYZDB-SSW-JSC006 and Grant 173211KYSB20160008).

## References


[1] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(7):1425–1438, 2016.

[2] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In *CVPR*, pages 2927–2936, 2015.

[3] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha. Synthesized classifiers for zero-shot learning. In *CVPR*, pages 5327–5336, 2016.

[4] Z. Ding, M. Shao, and Y. Fu. Low-rank embedded ensemble semantic dictionary for zero-shot learning. In *CVPR*, pages 2050–2058, 2017.

[5] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, pages 1778–1785, 2009.

[6] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In *NIPS*, pages 2121–2129, 2013.

[7] J. Fu, H. Zheng, and T. Mei. Look closer to see better: recurrent attention convolutional neural network for fine-grained image recognition. In *CVPR*, pages 4438–4446, 2017.

[8] Y. Fu, T. M. Hospedales, T. Xiang, Z. Fu, and S. Gong. Transductive multi-view embedding for zero-shot recognition and annotation. In *ECCV*, pages 584–599, 2014.

[9] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong. Transductive multi-view zero-shot learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37(11):2332–2345, 2015.

[10] C. Huang, C. C. Loy, and X. Tang. Local similarity-aware deep feature embedding. In *NIPS*, pages 1262–1270, 2016.

[11] H. Jiang, R. Wang, S. Shan, Y. Yang, and X. Chen. Learning discriminative latent attributes for zero-shot classification. In *ICCV*, pages 4223–4232, 2017.

[12] N. Karessli, Z. Akata, A. Bulling, and B. Schiele. Gaze embeddings for zero-shot image classification. In *CVPR*, pages 4525–4534.

[13] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, pages 951–958, 2009.

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

[15] Y. Li, D. Wang, H. Hu, Y. Lin, and Y. Zhuang. Zero-shot recognition using dual visual-semantic mapping paths. In *CVPR*, pages 3279–3287, 2017.

[16] P. Morgado and N. Vasconcelos. Semantically consistent regularization for zero-shot recognition. In *CVPR*, pages 6060–6069, 2017.

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

[18] P. Peng, Y. Tian, T. Xiang, Y. Wang, and T. Huang. Joint learning of semantic and latent attributes. In *ECCV*, pages 336–353, 2016.

[19] P. Peng, Y. Tian, T. Xiang, Y. Wang, M. Pontil, and T. Huang. Joint semantic and latent attribute modelling for cross-class transfer learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017.

[20] S. Reed, Z. Akata, H. Lee, and B. Schiele. Learning deep representations of fine-grained visual descriptions. In *CVPR*, pages 49–58, 2016.

[21] B. Romera-Paredes and P. Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, pages 2152–2161, 2015.

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[23] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.

[25] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[26] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research (JMLR)*, 10(Feb):207–244, 2009.

[27] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele. Latent embeddings for zero-shot classification. In *CVPR*, pages 69–77, 2016.

[28] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *arXiv preprint arXiv:1707.00600*, 2017.

[29] X. Xu, F. Shen, Y. Yang, D. Zhang, H. T. Shen, and J. Song. Matrix tri-factorization with manifold regularizations for zero-shot learning. In *CVPR*, pages 3798–3807, 2017.

[30] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International journal of computer vision (IJCV)*, 73(2):213–238, 2007.

[31] Z. Zhang and V. Saligrama. Zero-shot learning via joint latent similarity embedding. In *CVPR*, pages 6034–6042, 2016.