ZSCRGAN: A GAN-based Expectation Maximization Model for Zero-Shot Retrieval of Images from Textual Descriptions

Anurag Roy
Department of CSE, IIT Kharagpur, India

Vinay Kumar Verma
Department of ECE, Duke University, USA

Kripabandhu Ghosh
Department of CSA, IISER Kolkata, India

Saptarshi Ghosh
Department of CSE, IIT Kharagpur, India

ABSTRACT
Most existing algorithms for cross-modal Information Retrieval are based on a supervised train-test setup, where a model learns to align the mode of the query (e.g., text) to the mode of the documents (e.g., images) from a given training set. Such a setup assumes that the training set contains an exhaustive representation of all possible classes of queries. In reality, a retrieval model may need to be deployed on previously unseen classes, which implies a zero-shot IR setup. In this paper, we propose a novel GAN-based model for zero-shot text to image retrieval. When given a textual description as the query, our model can retrieve relevant images in a zero-shot setup. The proposed model is trained using an Expectation-Maximization framework. Experiments on multiple benchmark datasets show that our proposed model comfortably outperforms several state-of-the-art zero-shot text to image retrieval models, as well as zero-shot classification and hashing models suitably used for retrieval.

CCS CONCEPTS
• Computing methodologies → Machine learning; Ranking;
• Information systems → Multimedia and multimodal retrieval; Image search;

KEYWORDS
Zero-shot; Text to image retrieval; GAN; E-M

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org. CIKM ’20, October 19–23, 2020, Virtual Event, Ireland © 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 978-1-4503-6859-9/20/10... $15.00 https://doi.org/10.1145/3340531.3411995

1 INTRODUCTION
Today, information is generated in several modes, e.g., text, image, audio, video, etc. Thus, for a query in one mode (e.g., text), the relevant information may be present in a different mode (e.g., image). Cross-modal Information Retrieval (IR) algorithms are being developed to cater to such search requirements.

Need for Zero-shot Information Retrieval (ZSIR): A train-test setup of an IR task comprises parameter learning for various classes/categories of queries. Standard cross-modal retrieval methods require training data of all classes of queries to train the retrieval models. But such methods can fail to retrieve data for queries of new or unseen classes. Such a situation conforms to the “zero-shot” setup [11, 14, 17] which focuses on recognizing new/unseen classes with limited training classes. Such situations are relevant in any modern-day search system, where new events, hashtags, etc. emerge every day. So, the zero-shot paradigm needs to be incorporated in an IR setting.

Specifically, zero-shot cross-media retrieval intends to achieve retrieval across multiple modes (e.g., images to be retrieved in response to a textual query) where there is no overlap between the query-classes in training and test data. Zero-Shot IR (ZSIR) is especially challenging since models need to handle not only different semantics across seen and unseen query-classes, but also the heterogeneous features of data across different modes.

Present work and differences with prior works: Though lot of research has been reported on general multimodal and cross-modal retrieval [23], to our knowledge, only a few prior works have attempted cross-modal IR in a zero-shot setting [2, 3, 9, 13, 19, 30] (see Section 2 and Section 4.3 for details of these methods). Some of these prior works assume additional information about the class labels (e.g., a measure of semantic similarity between class labels) which may not always be available.

In this paper, we propose a novel model for cross-modal IR in zero-shot setting, based on Conditional Generative Adversarial Networks (GANs) [15], that can retrieve images relevant to a given textual query. Our model – which we name ZSCRGAN (Zero-Shot Cross-modal Retrieval GAN) – relies only on the textual data to perform the retrieval, and does not need additional information about class labels. Though prior ZSIR models [2, 3, 30] also use GANs, the main novel contributions of the proposed model can be summarized as follows: (1) We propose use of wrong classes to enable the generator to generate features that are unique to a specific class, by distinguishing it from other classes. (2) We develop a Common Space Embedding Mapper (CSEM) to map both the image embeddings and the text embeddings to a common space where retrieval can be performed. This is the key step that enables our model to perform retrieval without relying on additional semantic information of class labels. (3) We develop an Expectation-Maximization...
We start by formally defining the zero-shot text to image retrieval problem, and then describe our proposed model ZSCRGAN (implementation available at https://github.com/ranarag/ZSCRGAN). Table 1 gives the notations used in this paper.

### 3 Proposed Approach

We start by formally defining the zero-shot text to image retrieval problem, and then describe our proposed model ZSCRGAN (implementation available at https://github.com/ranarag/ZSCRGAN). Table 1 gives the notations used in this paper.

#### Table 1: Notations used in this paper.

| Description                                      | Description                                      |
|--------------------------------------------------|--------------------------------------------------|
| \( \Gamma \) Generator                        | \( \Omega \) Discriminator                      |
| \( \mathcal{L}_D \) Discriminator Loss Function | \( \mathcal{L}_G \) Generator Loss Function     |
| \( \phi \) Text embedding of unseen class       | \( \mathbf{I}_t \) Image embedding of unseen class |
| \( \hat{\phi} \) Common Space embedding gener-  | \( \mathbf{I}_t \) Common space embedding gener-  |
| ated from \( \phi \)                           | ated from \( \mathbf{I}_t \)                     |
| \( \hat{\mathbf{I}}_r \) Real Image Embedding  | \( \mathbf{I}_w \) Wrong Image Embedding         |
| \( \hat{\mathbf{I}}_r \) Real Text Embedding    | \( \mathbf{I}_w \) Wrong Text Embedding          |
| \( \mathbf{c}_r \) Real Latent Embedding         | \( \mathbf{c}_w \) Wrong Latent Embedding         |
| \( \mathbf{w}_c \) trainable parameters of CSEM | \( \mathbf{w}_c \) trainable parameters of Generator |
| \( \hat{\phi}_c \) Common Space embedding gener- | \( \mathbf{I}_w \) Common space embedding gener-   |
| ated from \( \phi_c \)                          | ated from \( \phi_w \)                           |
| \( \mathbf{w}_c \) trainable parameters of CSEM | \( \mathbf{w}_c \)                              |
| \( \phi \) Latent Embedding Generated from \( \phi \) | \( \mathbf{z}_w \) noise vector sampled from a nor- |

#### 3.1 Problem Definition

In the zero-shot Text to Image retrieval setup, we consider training data \( \mathcal{D} = \{ \mathbf{I}^k_r, \phi^k_r, \mathbf{y}^k \}_{k=1}^K \) with \( K \) samples. \( \mathbf{I}^k_r \) is the real image embedding of the \( k \)-th image. \( \phi^k_r \) is the real text embedding of the text accompanying the \( k \)-th image. \( \mathbf{y}^k \in \mathcal{Y} \) is a class label, and \( \mathcal{Y} = \{1, 2, \ldots, S\} \) is the set of seen classes, where \( S \) is the number of seen classes (only seen classes are available at training time).

Let \( \mathcal{U} = \{1, 2, \ldots, U\} \) be the set of unseen classes, where \( U \) is the number of unseen classes (not available at training time). For each unseen class query \( u \in \mathcal{U} \) a relevant set of images are present that we have to retrieve. At test-time, for each unseen class \( u \in \mathcal{U} \), we use a textual embedding \( \phi_u \) from \( u \) as query. Textual embedding \( \phi_u \) and unseen class image are projected into joint space to perform the retrieval. In the zero-shot setup \( \mathcal{U} \cap \mathcal{Y} = \Phi \), i.e. training and test classes are disjoint.

#### 3.2 Overview of our approach

As part of addressing this challenge, we use a Generative Adversarial Network (GAN) [7] to generate a per class ‘representative embedding’ for all image embeddings of a particular class. This generated embedding is desired to have high similarity with image embeddings from the same class and low similarity with image embeddings from other classes.

We chose a generative model rather than a discriminative one for this purpose, because generative models are most robust to visual imagination, and this helps the model to map the images from unseen classes more robustly. Two popular generative models are Variational Autoencoders (VAE) [10] and Generative Adversarial Networks (GAN) [7]. Samples from VAE models tend to be blurry (i.e., with less information content) as compared to GANs, because of the probability mass diffusion over the data space [21]. Therefore, we found GANs to be the most suitable for this problem.

While there have been prior works using GANs for zero-shot retrieval tasks [2, 3, 19], they rely on class labels for training. Some of the prior models [2, 3] make use of word-embeddings of class labels as class level information. However, in many cases, class label information may not be available. In contrast, our proposed model does not need class labels to perform its task.

Our proposed model ZSCRGAN (shown in Figure 1) works as follows. We take two text embeddings, one belonging to class \( y \in \mathcal{Y} \) and the other belonging to some other class \( \hat{y} \in \mathcal{Y} \). We pass the two text embeddings through a Text Encoder (TE) which generates (i) a latent embedding \( \mathbf{c}_r \) (a.k.a. real text embedding) for class \( y \) and
The example here demonstrates training of the architecture for learning a common space embedding for the class ‘Black
(through which textual embeddings of the wrong class are passed) are identical copies of the corresponding blocks in solid
very poor retrieval performance. The reason being, images from
line (through which textual embeddings of the correct class are passed).

embedding which would be a representative for the images of a
stance,

log

Let

\( ZSCRGAN \)

and GAN.

\[ Q(\hat{\mathbf{I}}, \hat{\phi}, \hat{\psi}_C) \]

\[ \log \sum_{i \in I} Q(i, \phi|\psi_C) \]

\[ \sum_{i \in I} P(i|I, \phi, \psi_C) \cdot \log Q(i, \phi|\psi_C) \cdot \log Q(\mathbf{I}, \phi|\psi_C) \]

\[ \sum_{i \in I} P(i|I, \phi, \psi_C) \cdot \log Q(i, \phi|\psi_C) \cdot \log Q(\mathbf{I}, \phi|\psi_C) \]

\[ \sum_{i \in I} P(i|I, \phi, \psi_C) \cdot \log Q(i, \phi|\psi_C) \bullet \log \frac{Q(\mathbf{I}, \phi|\psi_C)}{P(i|I, \phi, \psi_C)} \]

\[ \log \frac{Q(\mathbf{I}, \phi|\psi_C)}{P(i|I, \phi, \psi_C)} \]

3.3 Details of our proposed model ZSCRGAN

Let \( Q(I, \phi) \) be the joint probability distribution denoting the probability of text embeddings \( \phi \) and relevant image embeddings \( I \). Maximizing this probability is expected to ensure high similarity between \( \phi \) and relevant image embeddings \( I \) therefore leading to better performance of the retrieval model. We plan to do this maximization using a machine learning model having \( \psi_C \) as its parameters. Hence, our aim will be to maximize the probability distribution \( Q(I, \phi|\psi_C) \).

For simplicity, we will maximize the log of this distribution log \( Q(I, \phi|\psi_C) \). Let \( \psi_C \) be the random variable representing the values that can be taken by the trainable parameters of the neural network model. Thus, our log probability function becomes log \( Q(I, \phi|\psi_C) \). However, when we try to maximize log \( Q(I, \phi|\psi_C) \) directly using a machine learning model, we see very poor retrieval performance. The reason being, images from one class are very similar to images from another class. For instance, ‘Crested Auklet’ and ‘Rhinoceros Auklet’ in the CUB dataset are two very similar looking birds, and are almost indistinguishable to the human eye. Due to these overlaps among images from classes, training the neural network model becomes difficult, as it encounters very similar positive and negative examples during training. Thus, we introduce a latent variable \( I' \) – an unobserved embedding which would be a representative for the images of a

\[
\log Q(I, \phi|\psi_C) = \log \sum_{i \in I} Q(i, \phi|\psi_C)
\]

\( \sum_{i \in I} P(i|I, \phi, \psi_C) \cdot \log Q(i, \phi|\psi_C) \cdot \log Q(\mathbf{I}, \phi|\psi_C) \]

\[
\geq \sum_{i \in I} P(i|I, \phi, \psi_C) \cdot \log Q(i, \phi|\psi_C) \cdot \log Q(\mathbf{I}, \phi|\psi_C)
\]

by Jensen’s Inequality

\[
= \sum_{i \in I} P(i|I, \phi, \psi_C) \cdot \log Q(i, \phi|\psi_C) - \sum_{i \in I} P(i|I, \phi, \psi_C) \cdot \log \frac{Q(\mathbf{I}, \phi|\psi_C)}{P(i|I, \phi, \psi_C)}
\]

\[
= \sum_{i \in I} P(i|I, \phi, \psi_C) \cdot [\log Q(i, \phi|\psi_C) + \log Q(\mathbf{I}, \phi|\psi_C)]
\]

\[ - \sum_{i \in I} P(i|I, \phi, \psi_C) \cdot \log Q(\mathbf{I}, \phi|\psi_C) \]

\[ \text{since } I \text{ is conditionally independent of } i \text{ and } \phi \text{ given } \psi_C \]

\[
= \sum_{i \in I} P(i|I, \phi, \psi_C) \cdot \log Q(i, \phi|\psi_C) + \sum_{i \in I} P(i|I, \phi, \psi_C) \cdot \log Q(\mathbf{I}, \phi|\psi_C)
\]

\[ - \sum_{i \in I} P(i|I, \phi, \psi_C) \cdot \log Q(\mathbf{I}, \phi|\psi_C) \]

\[ = \sum_{i \in I} P(i|I, \phi, \psi_C) \cdot \log Q(i, \phi|\psi_C) + 1 \cdot \log Q(I) \]

\[ - \sum_{i \in I} P(i|I, \phi, \psi_C) \cdot \log Q(\mathbf{I}, \phi|\psi_C) \]

\[
(2)
\]
following:
\[
\sum_{i \in T} \log Q(i, I, \phi|\psi_C) - \log Q(I) \\
\geq \sum_{i \in T} P(i|I, \phi, \psi_C) \cdot \log Q(i, I, \phi|\psi_C) - \sum_{i \in T} P(i|I, \phi, \psi_G) \cdot \log P(i|I, \phi, \psi_C) \\
= F(P, \psi_C)
\]

\(Q(i, \phi|\psi_C)\) denotes the joint probability of \(i\) and \(\phi\) being similar. Thus, \(F(P, \psi_C)\) is a lower bound for \(\log Q(I, \phi|\psi_C)\), and maximizing the lower bound will ensure a high minimum value of \(\log Q(I, \phi|\psi_C)\).

1 We train a neural network architecture using the E-M algorithm to maximize \(F(P, \psi_C)\) where the E and M steps at iteration \(it\) are:

\[E\text{-step} : \quad P^{it}(I' = i|I, \phi) = \arg \max_P F(P^{it}, \psi_C)\]

\[M\text{-step} : \quad \psi_C^{it} = \arg \max \psi F(P^{it}, \psi_C)\]

So, the challenge now is to maximize \(F(P, \psi_C)\). To this end, we propose to use neural networks as follows.

### 3.3.2 Using neural networks to approximate \(F(P, \psi_C)\):

Eqn. 2 can be re-written as:

\[
F(P, \psi_C) = \mathbb{E}_{i \in i}(P(i|I, \phi, \psi_C)) \cdot \log Q(i, I, \phi|\psi_C) \\
- \mathbb{E}_{i \in i}(P(i|I, \phi, \psi_C)) \cdot \log P(i|I, \phi, \psi_C)
\]

We take the help of two neural networks to approximate the two parts of the function \(F\) as shown in Eqs. 6 - (1) Common Space Embedding Mapper (CSEM) to represent the first term, and (2) GAN to represent the second term in Eqn. 6.

**Common Space Embedding Mapper (CSEM):** This module is a feed-forward neural network trained to maximize the probability \(Q(i, \phi|\psi_C)\) which denotes the joint probability of \(i\) and \(\phi\) being similar. We define such a \(Q\) as:

\[
Q(i, \phi|\psi_C) = \frac{e^{\varphi_p}}{e^{\varphi_p} + e^{\varphi_m}}
\]

where \(\varphi_p\) and \(\varphi_m\) are scores calculated as:

\[\varphi_p = \text{CosineSim}(\text{CSEM}(i), c_{I\phi})\]

\[\varphi_m = \text{CosineSim}(\text{CSEM}(G(z, c_{I\hat{w}})), c_{I\hat{w}})\]

The CSEM is trained using the cost function Triplet Loss [5] \(L_T\), which can be written as:

\[
L_T = -\mathbb{E}_{i \in i}(P(i|I, \phi, \psi_C), c_{I\phi} - \text{TE}(\phi)) \cdot \log Q(i, \phi|\psi_C) \\
- \mathbb{E}_{i \in i}(P(i|I, \phi, \psi_C), c_{I\phi} - \text{TE}(\phi)) \cdot \log \left[\frac{e^{\varphi_p}}{e^{\varphi_p} + e^{\varphi_m}}\right] \\
= \mathbb{E}_{i \in i}(P(i|I, \phi, \psi_C), c_{I\phi} - \text{TE}(\phi)) \cdot \log (1 + e^{\varphi_m - \varphi_p})
\]

We call this module the **Common Space Embedding Mapper** because it learns to map the image embeddings to a space where the resulting embeddings will have high cosine similarity among themselves if they are from the same class and low cosine similarity with images from different classes. We train it by using the triplet loss \(L_T\) considering \(c_{I\phi}\) as the pivot, \(i\) as the positive example and \(G(z, c_{I\hat{w}})\) as the negative example. Here, \(c_{I\phi}\) and \(c_{I\hat{w}}\) are generated by \(\text{TE}(\phi_{I\phi})\) and \(\text{TE}(\phi_{I\hat{w}})\) respectively, \(\text{TE}(\cdot)\) being the Text Encoder (described in Section 3.3.2).

\[R(I' = i, I) = \frac{1}{1 - p(I' = i, I)} \]

\[p(\cdot, \cdot) = \frac{e^{-d_p}}{e^{-d_p} + e^{-d_m}}\]

\[d_p = |I' - i| \quad \text{and} \quad d_m = |I' - i| - \lambda\]
have also provided a margin λ in order to ensure that i is at least λ separated from Iw, in terms of Manhattan distance. Our formulation of probability p(i, I) also ensures that p ∈ (0, 1), since for p to attain the value 0 or 1, dp or dn will have to be tending to ±∞. However, in our formulation, dp and dn always attain finite values. This ensures that the term 𝐺(𝑖′ = i, I) is not undefined anywhere during our optimization. The log of odds ratio function can be further simplified as:

\[
\log \frac{p(i, I)}{1 - p(i, I)} = \log \frac{e^{-dp}}{e^{-dp} + e^{dp}} - \log \frac{e^{-dn}}{e^{-dn} + e^{dn}} = d_n - d_p \quad (15)
\]

Here the discriminator (𝐷) and generator (𝐺) are trained by alternatively maximizing \(L_D\) and minimizing \(L_G\). Also, \(\alpha\) and \(\beta\) are hyper-parameters to be tuned to achieve optimal results.

**Discriminator:** In this adversarial game between the \(D()\) and the \(G()\), the \(D()\) is trained to separate a real image from a fake one. In our task of generating representative embedding \(i \in I\) from text embeddings of a particular class, an image embedding from a different class \(I_w\) should also be identified by the discriminator as fake given the text embedding \(c_I\). For example as shown in Figure 1, if the text embedding of Black-Footed-Albatross is given and \(G()\) generates an embedding of any other class, say American Goldfinch, then \(D()\) should label it as fake and force \(G()\) to generate for Black-Footed-Albatross from text-embedding of Black-Footed-Albatross. Hence we added another condition to the discriminator loss of classifying the image \(I_w\) as fake and gave it equal weightage as compared to discriminator loss of classifying the generated embedding as fake. Hence, as shown in Eqn. 16, we add another discriminator loss to the CWGAN discriminator loss:

\[
L_D = \mathbb{E}_{z \sim \mathcal{N}(\mu, \Sigma)} \left[ D(G(z, c_I), \mathtt{φ}_{I_w}) \right] - \mathbb{E}_{I \sim \mathcal{P}(\text{data})} \left[ D(I, \mathtt{φ}_{I_w}) \right] + \mathbb{E}_{I \sim \mathcal{P}(\text{data})} \left[ D(I, \mathtt{φ}_{I}) \right] \times 0.5 \\
+ \mathbb{E}_{I \sim \mathcal{P}(\text{data})} \left[ D(I, \mathtt{φ}_{I_w}) \right] - \mathbb{E}_{I \sim \mathcal{P}(\text{data})} \left[ D(I, \mathtt{φ}_{I}) \right] \times 0.5
\]

**Generator:** As shown in Figure 1, we design a \(G()\) with the loss function stated in Eqn. 12 which takes the latent embedding vector as input and tries to generate image embedding of the same class. In order to achieve this goal, we add two regularizers to \(G()\) - (1) Negative Log of odds regularizer (Eqn. 13),(2) Jensen-Shannon Divergence (Eqn. 17). The intention for jointly learning the text encoder model instead of training the text encoder separately is to ensure that the text encoder model learns to keep the important features required for the generator to generate image embedding.

**Learning text encodings:** As shown in Figure 1, the text embedding \(\phi_l\) is first fed into an encoder which encodes \(\phi_l\) into a Gaussian distribution \(\mathcal{N}(\mu(\phi_l), \Sigma(\phi_l))\), where mean \(\mu\) and standard deviation \(\Sigma\) are functions of \(\phi_l\). This helps to increase the samples of the text encoding and hence reduces the chance of overfitting. This kind of encoding provides small perturbations to \(\phi_l\) thus yielding more train pairs given a small number of image-text pairs. We train this model by optimizing the following condition as a regularization term while training the generator.

\[
D_{JS}(\mathcal{N}(\mu(\phi_l), \Sigma(\phi_l))) \sim \mathcal{N}(0, I)) \quad (17)
\]

where \(D_{JS}\) is the Jensen-Shannon divergence (JS divergence) between the conditioning distribution and the standard Gaussian distribution. Unlike the previous conditional GANs [15] our model does not append \(z\) with the conditioned \(c\) directly. Instead it learns the distribution over each embedding and thereby learns a new embedding space. The samples of the learned space are passed on to generator with \(z\). This approach is more discriminative in the latent space, and helps to separate two classes in the original generated space.

**3.3.3 Training setup and implementation details.** The values of the hyper-parameters in the model are set to \(\alpha = 0.5\), \(\beta = 2\), and \(\lambda = 2\) using grid-search. The generator and the discriminator are trained in an iterative manner with the given objective functions (Eqns. 11 and 12). Both the generator and the discriminator are optimized with the root mean squared propagation (RmsProp) optimizer. The generator is a fully connected neural network with two hidden layers. The first hidden layer has 2,048 units and the second hidden layer has 4,096 units. Leaky ReLU activation function has been used in the hidden layer, and the output of the generator is passed through a ReLU function to ensure that there are no negative values. The discriminator is also a fully connected neural network with 1,024 hidden layer units and 1 output unit. Leaky ReLU activation function is used in the hidden layer and no activation function is used for the output layer. The Text Encoder is a fully connected neural network with 2048 output units and 1024 dimension for the mean \(\mu\) and 1024 dimension for the standard deviation \(\Sigma\). The CSEM is implemented as a single layer feed forward network with ReLU activation. The weights and biases of both the generator and the discriminator are initialized with a random normal initializer having mean 0.0 and standard deviation 0.02.

**3.4 Applying model for retrieval**

Once the model ZSCRGAN is trained, the retrieval of images for a novel/unseen class proceeds as shown in Algorithm 2. The query for retrieval is the text embedding \(\phi_l\) of a novel/unseen class. Given \(\phi_l, c_{I_t}\) is generated by the text encoder. Then \(G()\) produces the image embedding \(I_t\) which is passed through CSEM() to produce \(\theta_t\). Now, for each image \(I_{\text{us}}\) from an unseen class in the test set, we obtain
the corresponding image embedding $I_t$ which is also passed through CSEM to get $\theta_t$. Let $\text{sim}_t = \cosineSim(CSEM(I_t), CSEM(G(z, \hat{c}_t)))$ be the cosine similarity between $\theta_t$ and $\theta_t$, where $z$ is a noise vector sampled from a random normal distribution. Thereafter, a 2-tuple $(\text{sim}_t, I_{us})$ is formed and appended to a list called $\text{simList}$. The list is then sorted in descending order of the $\text{sim}_t$ values. The top $k$ images are extracted from the sorted $\text{simList}$ and stored in $imageList$ which is returned as the ranked list of retrieved images.

4 EXPERIMENTS AND ANALYSIS
This section details our experiments through which we compare the performance of the proposed model with that of several state-of-the-art baselines, over several standard datasets.

4.1 Datasets
We use five datasets that are summarized in Table 2. For each dataset, the models work on image embeddings and text/attribute embeddings (e.g., of the image captions). Table 2 also states the sources for the various embeddings for each dataset. For fair evaluation, every model, including the proposed and the baseline models, use the same text and image embeddings.

(1) Oxford Flowers-102 (Flowers) dataset contains 8,189 images of flowers from 102 classes [16]. Each class consists of between 40 and 258 images. Each image is accompanied by 10 sentences, each describing the image [19]. The data is split into 82 training and 20 test classes, with no overlap among the training and test classes [16].

As text embeddings, we use charCNN-RNN embeddings of the image captions provided by [19]. We use the image embeddings provided by [26], which are generated by passing each image through ResNet-101 architecture [8] pre-trained on the Imagenet [4], and taking the pre-trained ResNet’s final layer embeddings.

(2) Caltech-UCSD Birds (CUB) dataset, which contains 11,788 images of birds, divided into 200 species (classes), with each class containing approx. 60 images [24]. Each image has 10 associated sentences describing the image [19]. Following [26], the images in CUB are split into 150 training classes and 50 test classes, such that there is no overlap among the classes in training and test sets. Similar to the Flowers dataset, we use the text embeddings provided by [19] and image embeddings provided by [26].

(3) North American Birds (NAB) dataset is a larger version of the CUB dataset, with 48,562 images of birds categorized into 1,011 classes. The dataset was modified by Elhoseiny et. al. [6] by adding a Wikipedia article for each class and merging some of the classes to finally have 404 classes. Two types of split have been proposed in [6] in terms of how the seen (S) and unseen (U) classes are related – (1) Super-Category-Shared (SCS), and (2) Super-Category-Exclusives (SCE). In SCS, unseen classes are chosen such that for each unseen class, there exists at least one seen class that have the same parent category. However, in SCE split, there is no overlap in the parent category among the seen and unseen classes.

We use the text and image embeddings provided by [30]. The 13, 217-dimensional text embeddings are actually TF-IDF vectors obtained from the Wikipedia articles (suitably preprocessed) corresponding to the classes. The image embeddings are obtained by feeding the images into VPDE-net [28], and extracting the activations of the part-based FC layer of the VPDE-net. The NAB dataset has six semantic parts ('head', 'back', 'belly', 'breast', 'wing', and 'tail'). A 512-dimension feature vector is extracted for each part and concatenated in order. The resulting 3072-dimensional vectors are considered as image embeddings.

(4) Animals with Attributes (AWA) dataset consists of 30, 475 images of animals from 50 different classes. 85 attributes are used to characterize each of the 50 categories, giving a class-level attribute embedding of 85 dimensions. The 2048-dimension image embeddings are taken from the final layer ResNet-101 model trained on ImageNet [4]. There exists two kinds of splits of the data according to [26] – Standard-split and proposed split. The standard-split does not take into consideration the ImageNet classes, while the proposed-split consists test classes which have no overlap with the ImageNet classes. We used the proposed-split [26] of the dataset for the ZSIR models (including the proposed model, DADN, ZSL-GAN).

(5) Wikipedia (Wiki) dataset consists of 2, 866 image-text pairs taken from Wikipedia documents [18]. The image-text pairs have been categorized into 10 semantic labels (classes). Images are represented by 128-dimensional SIFT feature vectors while the textual data are represented as probability distributions over the 10 topics, which are derived from a Latent Dirichlet Allocation (LDA) model. Following [13], the classes are split randomly in an 80:20 train: test ratio, and average of results over 10 such random splits is reported.

Ensuring zero-shot setup for some of the datasets: The Flowers, CUB, and AWA datasets use image embeddings from ResNet-101 that is pretrained over ImageNet. Hence, for these datasets, it is important to ensure that the test/unseen classes should not have any overlap with the ImageNet classes, otherwise such overlap will violate the ZSL setting [26]. To enforce the ZSL setting, we use the train-test split as proposed by [26], which ensures that there is no overlap among the test classes and the ImageNet classes on which ResNet-101 is pretrained.

4.2 Evaluation Metrics
For each class in the test set, we consider as the query its per-class text embedding $\varphi_i$ (or the per-class attribute embedding in case...
of the AWA dataset). The physical interpretation of the query for each dataset is explained in Table 2 (last column). For each class, we retrieve 50 top-ranked images (according to each model). Let the number of queries (test classes) be \( Q \). We report the following evaluation measures:

1. **Precision@50**, averaged over all queries: For a certain query \( q \), Precision@50(\( q \)) = \( \frac{k}{Q} \), where \( k \) is the number of relevant images among the top-50 images retrieved for \( q \). We report Precision@50 averaged over all \( Q \) queries.

2. **Mean Average Precision (mAP@50):** mAP@50 is the mean of Average Precision at rank 50, where the mean is taken over all \( Q \) queries. mAP@50 = \( \frac{1}{Q} \sum_{q=1}^{Q} \text{AveP}_{50}(q) \) where AveP_{50}(\( q \)) for query \( q \) is \( \frac{\sum_{r=1}^{R} \text{Precision}(r|q)}{50} \), where \( R \) is the set of ranks (in \( [1, 50] \)) at which a relevant image has been found for \( q \).

3. **Top-1 Accuracy (Top-1 Acc):** This metric measures the fraction of queries (unseen classes) for which the top-ranked retrieval result is relevant [19, 30]. In our experiments we report the average Top-1 Accuracy of the models over the set of all unseen classes.

### 4.3 Baselines

We compare the proposed ZSRCGAN model with different kinds of baseline models, as described below.

**Zero-Shot Information Retrieval (ZSIR) models:** We consider three state-of-the-art models for ZSIR:

1. **Reed et al.** developed the **DS-SJE** [19] model that jointly learns the image and text embeddings using a joint embedding loss function. We use the codes and pre-trained models provided by the authors at https://github.com/reedsct/cvpr2016.

2. **Choi et al.** proposed two very similar models for zero-shot IR [2, 3]. The **DADN model** [3] outperforms the one in [2]. Hence, we consider DADN as our baseline. DADN uses dual GANs to extract the category label embeddings to project the image embeddings and text embeddings to have a common representation in a semantic space [3]. We use the codes provided by the authors (at https://github.com/PKU-ICST-MIPL/DADN_TCSVT2019).

3. **Image generation models** are optimized to generate high-fidelity images from given textual descriptions. These models can be used for text-to-image retrieval as follows – given the textual query, we can use such a model to generate an image, and then retrieve images that are ‘similar’ to the generated image as answers to the query. We used the **ZSL-GAN** model developed by Zhu et al. [30].

We use the codes provided by the authors at https://github.com/EthanZhu90/ZSL_GAN.

**Zero-Shot Classification (ZSC) models:** We consider two state-of-the-art ZSC models [22, 25] (both of which use generative models), and adopt them for the retrieval task as described below.

1. **f-CLSWGAN** [25] used the GAN to synthesize the unseen class images (image embeddings) using the class attribute. Then, using the synthesized samples of the unseen class, they trained the softmax classifier. We used the codes provided by the authors (at http://datasets.d2.mpi-inf.mpg.de/xian/ImageNet2011_res101_feature.zip). To use this model for retrieval, we used the image embeddings generated by the generator for the unseen classes, and ranked the image embeddings of the unseen classes using their cosine similarity with the generated image embedding.

2. **SE-ZSL** [22] used the Conditional-VAE (CVAE) with feedback connection for Zero-Shot Learning (implementation obtained on request from the authors). We adopt the same architecture for the text to image retrieval as follows. Using the CVAE, we first trained the model over the training class samples. At test time, we generated the image embedding of the unseen classes conditioned on the unseen text query. In the image embedding space, we perform the nearest neighbour search between the generated image embedding and the original image embedding.

**Zero-Shot Hashing models:** We compare the proposed model with some Zero-Shot Hashing models, namely DCMH [27], SePH [12], and the more recent AgNet [9] and CZHash [13]. The implementations of most of these hashing models are not available publicly; hence we adopt the following approach. We apply our method to the same datasets (e.g., AWA, Wiki) for which results have been reported in the corresponding papers [9, 13], and use the same experimental setting as reported in those papers.

Note that different prior works reported different performance metrics. For those baselines whose implementations are available to us, we changed the codes minimally, to report Precision@50, mAP@50, and Top-1 Accuracy for all models. For comparing with the hashing models (whose implementations are not available), we report only mAP which is the only metric reported in [9, 13].

### 4.4 Comparing performances of models

Table 3 and Table 4 respectively compare the performance of the various models on the CUB and Flower datasets. Similarly, Table 5
Table 3: Zero-Shot Retrieval on CUB dataset. The proposed model outperforms all baselines (bold-font indicates the best results in all tables). The superscripts S, F, J, G, and D indicate that the proposed method is statistically significantly better at 95% confidence level (p < 0.05) than SE-ZSL, fCLSWGAN, DS-SJE, ZSL-GAN and DADN respectively. The superscripts have the same meaning in later tables.

| Retrieval Model | Prec@50(%) | mAP@50(%) | Top-1 Acc(%) |
|-----------------|------------|-----------|--------------|
| Zero-shot retrieval models adopted for retrieval |
| SE-ZSL [22]     | 79.3%      | 84.6%     | 59.6%        |
| fCLSWGAN [25]   | 36.1%      | 52.3%     | 64%          |
| ZSL-GAN [30]    | 42.2%      | 59.2%     | 66%          |
| DADN [3]        | 48.9%      | 62.7%     | 68%          |
| ZSCRGAN (proposed) | 52.3%<sup>SFGD</sup> | 65.4%<sup>SFGD</sup> | 74.6%<sup>SFGD</sup> |

Table 4: Zero-Shot Retrieval on Flower dataset. The proposed model outperforms all the baselines.

| Retrieval Model | Prec@50(%) | mAP@50(%) | Top-1 Acc(%) |
|-----------------|------------|-----------|--------------|
| Zero-shot retrieval models adopted for retrieval |
| SE-ZSL [22]     | 41.1%      | 63.1%     | 56.4%        |
| fCLSWGAN [25]   | 44.1%      | 67.2%     | 71.2%        |
| ZSL-GAN [30]    | 38.7%      | 46.6%     | 45%          |
| DADN [3]        | 20.8%      | 28.6%     | 25%          |
| ZSCRGAN (proposed) | 59.5%<sup>SFJGD</sup> | 69.4%<sup>SFJGD</sup> | 80%<sup>SFJGD</sup> |

Table 5: Zero-Shot Retrieval on NAB dataset (SCE split). The proposed model outperforms all the baselines.

| Retrieval Model | Prec@50(%) | mAP@50(%) | Top-1 Acc(%) |
|-----------------|------------|-----------|--------------|
| Zero-shot retrieval models adopted for retrieval |
| SE-ZSL [22]     | 7.5%       | 3.6%      | 7.2%         |
| ZSL-GAN [30]    | 6%         | 9.3%      | 6.2%         |
| DADN [3]        | 4.7%       | 7.3%      | 2.5%         |
| ZSCRGAN (proposed) | 8.4%<sup>FGD</sup> | 11.8%<sup>FGD</sup> | 7.4%<sup>FGD</sup> |

Table 6: Zero-Shot Retrieval on NAB dataset on (SCS split). The proposed model outperforms all the baselines.

| Retrieval Model | Prec@50(%) | mAP@50(%) | Top-1 Acc(%) |
|-----------------|------------|-----------|--------------|
| Zero-shot retrieval models adopted for retrieval |
| SE-ZSL [22]     | 31.5%      | 34.7%     | 11.4%        |
| ZSL-GAN [30]    | 32.6%      | 39.4%     | 34.6%        |
| DADN [3]        | 26.5%      | 28.6%     | 17.3%        |
| ZSCRGAN (proposed) | 36.4%<sup>FGD</sup> | 43%<sup>FGD</sup> | 49.4%<sup>FGD</sup> |

Table 7: Zero-shot retrieval on (i) AwA dataset, and (ii) Wiki dataset. Results of hashing models reproduced from [9] and [13]. Other metrics could not be reported due to unavailability of the implementations of the hashing models.

| Retrieval Model | mAP(%) on AwA | mAP(%) on Wiki |
|-----------------|---------------|---------------|
| Zero-shot Hashing models |
| DCMH [27]       | 16.3% (with 64-bit hash) | 24.83% (with 128-bit hash) |
| AgNet [9]       | 58.4% (with 64-bit hash) | 55.1% (with 128-bit hash) |
| SePH [12]       | 50.44% (with 128-bit hash) | 50.44% (with 128-bit hash) |
| CZHash [13]     | 25.87% (with 128-bit hash) | 25.87% (with 128-bit hash) |
| Zero-shot Retrieval models |
| ZSL-GAN [30]    | 12.5%         | -             |
| DADN [3]        | 27.9%         | 58.94%        |
| ZSCRGAN (proposed) | 62.2%<sup>FGD</sup> | 56.9%         |

And Table 6 compare performances over the NAB dataset (SCE and SCS splits respectively). Table 7 compares performances over the AwA and Wiki datasets. The main purpose of Table 7 is to compare performance with the Zero-Shot Hashing models whose implementations are not available to us; hence we report only mAP which is the only metric reported in [9, 13]. The superscripts S, F, J, G, and D in the tables indicate that the proposed method is statistically significantly better at 95% confidence level (p < 0.05) than SE-ZSL, fCLSWGAN, DS-SJE, ZSL-GAN and DADN respectively, according to the Wilcoxon signed-rank statistical significance test. We find that the results of the proposed model are significantly better than most of the baselines.

We now perform a detailed analysis of why our model performs better than the baselines.

**ZSCRGAN vs. DS-SJE**: DS-SJE [19] uses discriminative models to create text and image embeddings. This discriminative approach of DS-SJE lacks the visual imaginative capability required to capture the different postures of birds in the CUB dataset (which is not so much necessary for flowers) which can be captured using a generative approach like in ZSCRGAN.

---

3 We could not run DS-SJE over some of the datasets, as we were unable to modify the Lua implementation to suit these datasets. Also, the Prec@50 of DS-SJE on the Flower dataset (in Table 4) is what we obtained by running the pre-trained model provided by the authors, and is slightly different from what is reported in the original paper.

4 We could not run ZSL-GAN on the Wiki dataset, since this model assumes a single text description $T_c$ for each class $c$, whereas each class has more than one textual description in the Wiki dataset.

---

5 https://code.google.com/archive/p/word2vec/
4.5 Analysing two design choices in ZSCRGAN

In this section, we analyse two design choices in our proposed model — (1) why we adopted an Expectation-Maximization approach, instead of jointly training the GAN and the CSEM, and (2) why we chose to select wrong class embeddings randomly.

4.5.1 E-M vs Joint Optimization: In the proposed model, the CSEM and the GAN are trained alternately using an E-M setup (see Sec. 3.3). However, it is also possible to train both these networks jointly; i.e., when the Generator is trained, the CSEM loss is also optimized. We performed a comparison between these two approaches, and observed that the performance of the model drops by a significant amount when jointly trained. Figure 2 compares the performance of the two approaches in terms of Precision@50 over the CUB dataset, and shows that the E-M setup results in better retrieval. Similar observations were made for other performance metrics, across all datasets (details omitted for brevity).

The reason why the jointly training approach, where the CSEM loss is optimized along with the generator loss, does not work well is as follows. Using the triplet Loss $L_T$ (defined in Eqn. 9), the CSEM learns maximize similarity with the relevant embeddings and minimize similarity with the irrelevant embeddings. Thus, during backpropagation, the relevant representative embeddings have a different gradient than the irrelevant representative embeddings $G(z, c^*_w)$. When the CSEM is jointly trained with the generator, the weights of the CSEM get updated after every iteration, causing different gradients (or weights) for each of the wrong class embeddings of the generator, thus causing a distorted space of wrong class embeddings, and thus causing hindrance to the learning of the Generator. This problem, however, has been removed in the EM setup where the parameters of the generator are frozen while training the CSEM.

4.5.2 Choice of wrong class embeddings: In the proposed model, for given $L_T$ and $\phi_{tr}$ for a certain target class $c_{\text{target}}$, we learn the representative embedding for the images of that class. To this end, we use wrong class embeddings (WCE) selected randomly from among all other classes. One might think that, instead of randomly selecting wrong classes, we should employ some intelligent strategy of selecting wrong classes. For instance, one can think of selecting WCE such that they are most similar to $\phi_{tr}$, which may have the apparent benefit that the model will learn to distinguish between confusing (similar) classes well. However, we argue otherwise.

Restricting the choice of the wrong classes distorts the space of the wrong embeddings, and hence runs the risk of the model identifying embeddings from classes outside the space of distorted embeddings as relevant. In other words, though the model can learn to distinguish the selected wrong classes well, it fails to identify other classes (that are not selected as the wrong classes) as non-relevant.

To support our claim, we perform two experiments — (1) The wrong class is selected as the class whose text embedding has the highest cosine similarity to $\phi_{tr}$, and (2) The images are clustered using K-Means clustering algorithm, and the wrong class is selected as that class whose images co-occur with the maximum frequency in the same clusters as the images from $c_{\text{target}}$. Figure 3 compares the performance of the proposed model (where WCE are selected randomly) with these two modified models. Specifically, Figure 3(a) shows how Prec@50 varies with number of iterations for which the model is trained, over the CUB dataset. Random selection of WCE performs the best.

4.6 Error Analysis of ZSCRGAN

We analyse the failure cases where ZSCRGAN retrieves an image from a different class, compared to the query-class (whose text embedding has been issued as the query). Figure 4 shows some such examples, where the images enclosed in thick red boxes are not from the query-class. In general, we find that the wrongly retrieved images are in fact very similar to some (correctly retrieved) images in the query-class. For instance, in the CUB dataset, for the query-class Cerulean Warbler, the textual description of an image from this class (this bird is blue with white and has a very short beak) matches exactly with that of an image from a different class (which was retrieved by the model). Other cases can be observed where an image of some different class that has been retrieved, matches almost exactly with the description of the query-class. For instance, in the Flower dataset, for the query class Giant White Arum Lily, the wrongly retrieved class also has flowers with white petals and a yellow stigma, which matches exactly with many of the flowers in the Giant White Arum Lily class.
Figure 4: [color online] Images from the top three classes retrieved by ZSCRGAN, for the query-classes shown on the left. Top panel for Flower dataset, bottom panel for CUB dataset. The images with thick red boundaries are from some class that is not the query-class (hence not considered relevant), but they are very similar to some images in the query-class.

Table 8: Results of ablation analysis on the proposed model. Precision@50 reported on CUB and Flowers datasets.

4.7 Ablation Analysis of ZSCRGAN

Table 8 reports an ablation analysis, meant to analyze the importance of different components of our proposed architecture. For brevity, we report only Prec@50 for the two datasets CUB and Flowers (observations on other datasets are qualitatively similar).

The largest drop in performance occurs when the wrong class embedding is not used. As stated earlier, this use of wrong class embeddings is one of our major contributions. Another crucial factor is the generation of representative embeddings \( i \in I' \) for each class using a GAN. Removing this step also causes significant drop in performance. The Triplet loss and the Negative Log of odds ratio regularizer (R) are also crucial – removal of either leads to significant degradation in performance. Especially, if R is removed, the performance drop is very high for the Flower dataset. R is more important for the Flower dataset, since it is common to find different flowers having similar shape but different colors, and R helps to distinguish flowers based on their colors.

5 CONCLUSION

We propose a novel model for zero-shot text to image (\( T \to I \)) retrieval, which outperforms many state-of-the-art models for ZSIR as well as several ZS classification and hashing models on several standard datasets. The main points of novelty of the proposed model ZSCRGAN are (i) use of an E-M setup in training, and (ii) use of wrong class embeddings to learn the representation of classes. The implementation of ZSCRGAN is publicly available at https://github.com/ranarag/ZSCRGAN.

In future, we look to apply the proposed model to other types of cross-modal retrieval (\( I \to T \)), as well as to the zero-shot multi-view setup (\( T_I \to I, T_I \to T, I \to IT, \) etc.) where multiple modes can be queried or retrieved together.

Acknowledgements: The authors acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

REFERENCES

[1] Martin Arjovsky, Soumith Chintal, and Leon Bottou. 2017. Wasserstein Generative Adversarial Networks. In Proc. ICLR, 214–223.
[2] Jingfei Chi and Yuxin Peng. 2018. Dual Adversarial Networks for Zero-shot Cross-media Retrieval. In Proc. IJCAI 661–669.
[3] J. Chi and Y. Peng. 2019. Zero-shot Cross-media Embedding Learning with Dual Adversarial Distribution Network. IEEE Transactions on Circuits and Systems for Video Technology 30, 4 (2019), 1173–1187.
[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In Proc. IEEE CVPR.
[5] Xingqing Dong and Jianbing Shen. 2018. Triplet Loss in Siamese Network for Object Tracking. In The European Conference on Computer Vision (ECCV).
[6] M. Elhoseiny, Y. Zhu, H. Zhang, and A. Elgammal. 2017. Link the Head to the “Beak”. Zero Shot Learning from Noisy Text Description at Part Precision. In Proc. IEEE CVPR. 6288–6297.
[7] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. 2014. Generative adversarial nets. In Proc. NIPS.
[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In Proc. IEEE CVPR.
[9] Peng Kong, Yuxin Sun, Yunlong Yu, Yuwei Pang, and Jangong Han. 2020. Attribute-Guided Network for Cross-modal Zero-Shot Hashing. IEEE Transactions on Neural Networks and Learning Systems 31, 321–330.
[10] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. In Proc. ICLR.
[11] Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. 2008. Zero-data Learning of New Tasks. In Proc. AAAI - Volume 2.
[12] Z. Lin, G. Ding, J. Han, and J. Wang. 2017. Cross-View Retrieval via Probability-Based Semantics-Preserving Hashing. IEEE Transactions on Cybernetics 47, 12 (2017), 4342–4355.
[13] X. Liu, Z. Li, J. Wang, G. Yu, C. Domeniconi, and X. Zhang. 2019. Cross-Modal Zero-Shot Hashing. In Proc. IEEE ICDM.
[14] Tomas Mikolov, Andrea Frome, Samy Bengio, Jonathon Shlens, Yerom Singer, Greg S Corrado, Jeffrey Dean, and Mohammad Norouzi. 2013. Zero-shot Learning by Convex Combination of Semantic Embeddings. In Proc. ICLR.
[15] M. Mirza and S. Osindero. 2014. Conditional generative adversarial nets. In arXiv:1411.1784.
[16] M-E. Nilsback and A. Zisserman. 2008. Automated Flower Classification over a Large Number of Classes. In Proc. ICCV/VIP.
[17] Mark Palatucci, Dean Pomerleau, Geoffrey E Hinton, and Tom M Mitchell. 2009. Zero-shot Learning with Semantic Output Codes. In Proc. NIPS. 1419–1418.
[18] N. Rasiwasia, J. Costa Pereira, E. Covelli, G. Doyle, G.R.G Lanckriet, R. Levy, and N. Vasconcelos. 2010. A New Approach to Cross-modal Multimodal Retrieval. In ACM International Conference on Multimedia. 251–260.
[19] A. Reed, Z. Akata, B. Schiele, and H. Lee. 2016. Learning deep representations of fine-grained visual descriptions. In Proc. IEEE CVPR.
[20] Richard Socher, Miland Janjoo, Christopher D Manning, and Andrew Ng. 2013. Zero-shot learning through cross-modal transfer. In Proc. NIPS.
[21] L. Theis, A. van den Oord, and M. Bethge. 2016. A note on the evaluation of large-scale generative image models. In Proc. ICLR.
[22] Vinay Kumar Verma, Gundeep Arora, Ashish Mishra, and Piyush Rai. 2018. Generalized zero-shot learning via synthesized examples. In Proc. IEEE CVPR.
[23] Kaye Wang, Quyue Yin, Wei Wang, Shu Wu, and Liang Wang. 2016. A Compre- hensive Survey on Cross-modal Retrieval. CoRR abs/1607.06215 (2016).
[24] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. 2010. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001. California Institute of Technology.
[25] Yongjin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. 2018. Feature generating networks for zero-shot learning. In Proc. IEEE CVPR.
[26] Yongjin Xian, Bernt Schiele, and Zeynep Akata. 2017. Zero-Shot Learning – The Good, the Bad and the Ugly. In Proc. IEEE CVPR Learning deep representations of fine-grained visual descriptions. In Proc. ICLR.
[27] Erkun Yang, Cheng Deng, Wei Liu, Xianglong Liu, Dacheng Tao, and Xiaoke Gao. 2017. Fairwise Relationship Guided Deep Hashing for Cross-modal Retrieval. In Proc. AAAI. 1618–1625.
[28] Han Zhang, Tao Xu, Mohamed Elhoseiny, Xiaolong Huang, Shaoting Zhang, Ahmed Elgammal, and Dimitris Metaxas. 2016. SPDA-CNN: Unifying semantic part detection and abstraction for fine-grained recognition. In Proc. IEEE CVPR.
[29] Li Zhang, Tao Xiang, and Shaogang Gong. 2017. Learning a deep embedding model for zero-shot learning. In Proc. IEEE CVPR.
[30] Yizhe Zhu, Mohamed Elhoseiny, Bingchen Liu, Xi Peng, and Ahmed Elgammal. 2018. A Generative Adversarial Approach for Zero-Shot Learning from Noisy Texts. In Proc. IEEE CVPR.