Generalized Zero-Shot Learning using Generated Proxy Unseen Samples and Entropy Separation

Omkar Gune, Biplab Banerjee, Subhasis Chaudhuri
guneomkar@ee.iitb.ac.in,bbanerjee@iitb.ac.in,sc@ee.iitb.ac.in
Indian Institute of Technology Bombay, India

Fabio Cuzzolin
fabio.cuzzolin@brookes.ac.uk
Oxford Brookes University, UK

ABSTRACT

The recent generative model-driven Generalized Zero-shot Learning (GZSL) techniques overcome the prevailing issue of the model bias towards the seen classes by synthesizing the visual samples of the unseen classes through leveraging the corresponding semantic prototypes. Although such approaches significantly improve the GZSL performance due to data augmentation, they violate the principal assumption of GZSL regarding the unavailability of semantic information of unseen classes during training. In this work, we propose to use a generative model (GAN) for synthesizing the visual *proxy* samples while strictly adhering to the standard assumptions of the GZSL. The aforementioned proxy samples are generated by exploring the *early* training regime of the GAN. We hypothesize that such proxy samples can effectively be used to characterize the average entropy of the label distribution of the samples from the unseen classes. Further, we train a classifier on the visual samples from the seen classes and proxy samples using entropy separation criterion such that an average entropy of the label distribution is low and high, respectively, for the visual samples from the seen classes and the proxy samples. Such entropy separation criterion generalizes well during testing where the samples from the unseen classes exhibit higher entropy than the entropy of the samples from the seen classes. Subsequently, low and high entropy samples are classified using supervised learning and ZSL rather than GZSL. We show the superiority of the proposed method by experimenting on AWA1, CUB, HMDB51, and UCF101 datasets.

CCS CONCEPTS

• **Computing methodologies** → **Object recognition**.

KEYWORDS

Generalized zero-shot learning; generative models

ACM Reference Format:
Omkar Gune, Biplab Banerjee, Subhasis Chaudhuri and Fabio Cuzzolin. 2020. Generalized Zero-Shot Learning using Generated Proxy Unseen Samples and Entropy Separation. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394171.3413657

MM '20, October 12–16, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3413657

1 INTRODUCTION

The Generalized Zero-shot Learning (GZSL) [34] addresses the problem of identifying the visual samples of previously *unseen* or *seen* classes when the machine learning model is trained on the visual-semantic data only from the seen classes. GZSL models exhibit poor performance due to model bias towards the seen classes as the training data is not available for the unseen classes. To overcome the issue of a model bias towards the seen classes, recently, generative models such as Generative Adversarial Network (GAN) and Variational Auto-encoder (VAE) have been used in GZSL [4, 11, 35]. Specifically, such methods train the GAN or VAE using visual-semantic data from the seen classes. The prototypes of the unseen classes are then utilized to generate visual samples of corresponding unseen classes. A standard softmax based classifier is subsequently trained using the already available visual samples of the seen classes and the generated visual samples of the unseen classes. The complex GZSL problem is consequently transmuted into a relatively simpler problem of supervised classification, thus implicitly reducing the model-bias problem considerably.

However, the approach above suffers from the following issues. Firstly, the generative model-driven approaches clearly violate the holy grail of GZSL in the sense that *no information is available about the unseen classes during training*. Although the generative models are trained using the data from the seen classes, subsequent steps make use of the prototypes of the unseen classes to generate the visual samples of corresponding unseen classes. It is evident that accessing the prototypes of the unseen classes during training benefits such models, resulting in superior performance as compared to other GZSL approaches, which do not violate the assumptions of GZSL. Secondly, although generative models trained on visual-semantic data of the seen classes can generate the visual samples of the unseen classes, the quality of such samples is generally abysmal under the aforesaid model-bias problem [14, 27].

We address the above issues of GZSL in this work, as explained below. Firstly, we emphasize that we do not use data from the unseen classes during training, thus adhering to the standard assumption of GZSL. Secondly, as we do not attempt to generate visual samples of the unseen classes, the issue of poor quality of generated samples of the unseen classes does not arise. Instead, we exploit the poor quality of the samples from the seen classes generated by GAN during its early training regime.

We use conditional Wasserstein GAN [6] (CWGAN) as our backbone generative model trained on the visual-semantic data from the seen classes. As we do not use any information (semantic prototypes) related to the unseen classes during training, generating the visual samples of the unseen classes is clearly not feasible. Hence, we synthesize the visual samples, termed as *proxy samples*, during

the *early* training regime of CWGAN. The exact training iteration is identified based on the *Confidence Score* (CS) of the proxy samples in that training iteration. The CS obtained from the classifier pre-trained on the seen classes is similar to the Inception Score [25]. CS is indicative of the diversity of the samples generated by GAN, where a higher CS corresponds to the fact that generated data distribution matches the real data distribution. A lower CS is indicative of the poor quality of the generated samples where generated data distribution is very different from the real data distribution.

Furthermore, it is a known fact that samples generated during early training regime are far from the real samples (referring to the visual samples from the seen classes in the (G)ZSL setup) having a low CS. However, it does not necessarily mean that generated proxy samples belong to the unseen classes. We hypothesize that proxy samples are representative of the samples belonging to the *open-set* of classes. (Here, we refer to the close-set as the set of seen classes and open-set as the set of all the classes excluding the seen classes. In principle, the open-set is vast and contains the unseen classes.) Therefore, we use the proxy samples to characterize the samples from the open-set classes in terms of the entropy of the label distribution. Towards that end, we train a separate classifier on visual samples of the seen classes and proxy samples. We train the classifier to have low and high average entropy of the label distribution for visual samples of the seen classes and proxy samples, respectively. Such entropy separation criterion generalizes well during testing, where we observe that the average entropy of the samples from the unseen classes is higher than the average entropy of the samples from the seen classes. During testing, low and high entropy test samples are identified as belonging to the seen classes and the unseen classes, respectively. The labels of low and high entropy samples are then separately identified using supervised learning (SL) and ZSL. To minimize the effect of misclassification due to hard division of samples based on low and high entropy, we also keep a margin around the decision boundary. The test samples whose entropies fall into this margin are classified using GZSL.

To the best of our knowledge, we are the first to use the evolution of GAN training to address GZSL. Our four-fold contributions are

- We deal with GZSL while adhering to the standard assumption that no data from the unseen classes are available during training. While doing so, we solve the harder GZSL problem in terms of simpler SL or ZSL problems.
- We propose to use the early training regime of GAN to synthesize the proxy samples, which are representatives of the samples from the open-set containing the unseen classes. The exact training regime is identified using the Confidence Score of the generated proxy samples.
- We increase the separation between the entropies of visual samples from the seen classes and proxy samples. Subsequently, based on the low or high entropy of a test sample, the GZSL problem is solved using SL or ZSL, respectively.
- We demonstrate the efficacy of the proposed approach on diverse tasks such as image object recognition (AWA1 and CUB) and video action recognition (HMDB51 and UCF101), where our approach outperforms the existing methods.

2 RELATED WORK

GZSL has gained much attention in the last few years to address the label scarcity issue in the open-world visual recognition systems [13, 34]. GZSL makes use of the semantic information to transfer knowledge from seen classes to unseen classes. In this respect, semantic space can be derived in terms of semantic attributes [22] or distributed word-vector embeddings [17]. Amongst different GZSL approaches, non-generative models aim to learn deterministic or stochastic functions given the semantic and the visual spaces [9, 10, 15, 21, 23, 24, 29, 31, 37–39]. On the other hand, generative techniques for GZSL focus to combat the class-imbalance issue in GZSL by modeling the underlying data distributions [4, 12, 14, 16, 19, 27, 35, 36]. Notably, [35] uses WGAN [6], which is trained to generate visual samples of the seen classes from the corresponding seen class-prototypes. The generator of WGAN is then bestowed to synthesize the visual features of unseen classes from the respective class-prototypes. Synthesized features of unseen classes, along with the features of the seen classes, are used to train a supervised classifier. However, it is observed that generated samples of unseen classes are poor in quality in the sense that they do not represent unseen class distribution [14, 27]. To overcome this issue and improve the quality of samples of unseen classes, cycle consistency is proposed in [4], where generated visual samples are mapped back to their corresponding semantic prototype. [27] proposes to match the gradients of generated and real samples to improve sample generation further. [16] extend the approach in [35] and train a separate classifier for seen and unseen classes. [14] proposes to alleviate confusion between seen and unseen class samples and introduce feature confusion scores. [36] uses GAN and VAE to generate unseen class samples.

The existing generative-model driven methods using GAN or VAE differ from our approach in two ways. First, these methods train the generator for large enough iterations such that it can match the distribution of the seen classes. On the other hand, we are interested in the early training regime to synthesize the proxy samples, which are far from the samples from the seen classes. Second, the existing methods train the softmax classifier on seen and unseen classes using the visual samples from seen classes and synthesized visual samples from unseen classes by optimizing the cross-entropy loss. However, we make use of the entropy separation criterion to train the classifier on the visual samples of the seen classes and proxy samples.

3 THE GZSL PROBLEM

We first formally define the problem of ZSL and GZSL, following which the proposed approach is discussed.

Consider a dataset $\mathcal{D}$ consisting of S seen classes and U unseen classes such that $\mathcal{D} = \mathcal{D}_S \bigcup \mathcal{D}_U$ where $\mathcal{D}_S = \{x_i, y_i, a_{y_i}\}_{i=1}^{n_s}$ and $\mathcal{D}_U = \{x_i, y_i, a_{y_i}\}_{i=1}^{n_u}$ denote the data belonging to the seen/train classes and unseen/test classes, respectively. Here, $x_i \in \mathbf{R}^d$, $y_i \in \mathbf{Z}$, $a_{y_i} \in \mathbf{R}^k$ represent a visual feature, its class label, and the corresponding class prototype while n_s and n_u represent the numbers of samples from the seen and unseen classes, respectively. For $\mathcal{D}_S$, $y_i \in \mathcal{Y}_S = \{1, 2, \ldots, S\}$ and for $\mathcal{D}_U$, $y_i \in \mathcal{Y}_U = \{S{+}1, S{+}2, \ldots, S{+}U\}$ such that $\mathcal{Y}_S \cap \mathcal{Y}_U = \phi$. Furthermore, $\mathcal{D}_S = \mathcal{D}_{tr} \bigcup \mathcal{D}_{tr}^{'}$ where $\mathcal{D}_{tr}$ represents the data of the seen classes used during training and $\mathcal{D}_{tr}^{'}$

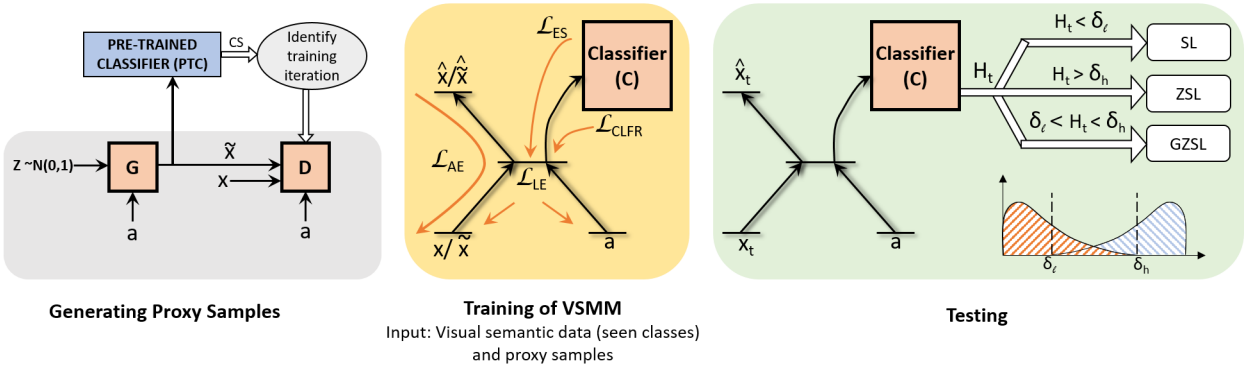

Figure 1: The three stages of the proposed method are shown. Left: A CWGAN is trained, and the generator G is used to synthesize the proxy samples. The exact training iterations are identified based on the Confidence Score (CS) of the generated proxy samples given by the pre-trained classifier PTC (trained on seen classes). Middle: A Visual-Semantic Mapping Module (VSMM) having the classifier C is trained on the visual-semantic data from the seen classes and synthesized proxy samples. The VSMM is trained using different loss criteria, including the entropy separation criterion. Right: During testing, the domain of a test sample (seen/unseen or unknown) is identified based on its entropy H_t of the label distribution given by C. Subsequently, based on entropy thresholds δ_l and δ_h, a sample is classified using SL, ZSL, or GZSL techniques.

represents the held out data of the seen classes which is used during testing. Let $f_v(\cdot)$ and $f_s(\cdot)$ denote the embedding functions projecting the visual and semantic features to a shared latent space with m dimensions. The goal of ZSL training is to use $\mathcal{D}_{tr}$ to maximize the compatibility between visual embedding $f_v(x_i)$ and semantic embedding $f_s(a_{y_i})$. Let $\mathcal{F}(\cdot)$ denote a compatibility function (e.g. inner product) which gives a higher score for pairs of visual and semantic embeddings associated with the same samples in $\mathcal{D}_{tr}$ and gives lower score otherwise. During testing, the class label of a test sample x_t is inferred as

$$\hat{y_t} = \underset{i \in \mathcal{Y}_U}{\operatorname{argmax}} \mathcal{F}(f_v(x_t), f_s(a_i)). \tag{1}$$

In GZSL, a test sample may come either from unseen class data $\mathcal{D}_U$ or seen class data ($\mathcal{D}'_{tr}$). The class label is obtained using the following criterion.

$$\hat{y_t} = \underset{i \in \{\mathcal{Y}_S \bigcup \mathcal{Y}_U\}}{\operatorname{argmax}} \mathcal{F}(f_v(x_t), f_s(a_i)) \tag{2}$$

4 METHODOLOGY

Motivation: Existing GZSL methods focus on improving the visual and semantic embedding functions to boost the model performance. We take an alternate path, where our method acts as a pre-processing step for the existing GZSL methods. In the pre-processing step, the domain (seen or unseen) of a test sample is first identified. If a sample is identified as belonging to the unseen domain, then its class label is inferred using Eq. (1) rather than Eq.(2). Furthermore, if a test sample is identified as belonging to the seen domain, its class label can be inferred using a supervised learning (SL) framework. Hence, the GZSL problem can be solved using relatively simpler ZSL and SL problems.

One way to identify the domain of the samples is to train a binary classifier for the task of identifying domain as seen or unseen. Alternatively, the above problem translates to the close/open-set recognition problem where the close-set is formed by the seen classes and open-set consists of all classes other than the seen classes (including the unseen classes). For such a task, one would need the visual samples of the close-set (seen) and open-set classes. Synthesizing the visual samples of the vast open-set classes is a difficult task in itself. One can resort to synthesizing the visual samples of the unseen classes as the representative samples of the open-set classes. However, synthesizing the visual samples of the unseen classes using corresponding semantic prototypes violates the GZSL setting.

We would like to get the benefits of synthesizing the visual samples of unseen classes. However, the use of prototypes for doing the same is prohibited. While synthesizing the visual samples of unseen classes looks infeasible in such a situation, we aim at generating the *proxy* samples, which are representatives of the samples from the open-set classes. To this end, we carefully assess the evolution of the training regime of GAN and claim that the early training regime of GAN is indeed capable of generating such samples. Next, we train the classifier such that the average entropy of label distribution is low and high, respectively, for the visual samples of the seen classes and synthesized proxy samples. During testing, based on the entropy of the test sample, one can identify its domain.

Our overall method is shown in Figure 1 has three stages. First, we use conditional Wasserstein GAN [35] (CWGAN) to generate proxy samples. The exact training iteration for generating proxy samples is identified using the classifier (PTC) pre-trained on the visual samples of the seen classes. Second, we train the Visual-Semantic Mapping Module (VSMM) containing the classifier (C) (different than the PTC) using the proposed entropy separation criterion. Third, we carry out the testing where the domain of the test sample is identified based on its entropy, followed by classification using SL or ZSL or GZSL.

4.1 Visual-semantic Mapping Module (VSMM)

We use a Neural Network based Visual-Semantic Mapping Module (VSMM) to learn the visual and semantic embedding functions to carry out the GZSL. Our VSMM is similar to the model in [38], which consists of an autoencoder (AE) but additionally uses a classifier which plays a vital role in identifying the domain of a test sample. The proposed VSMM may be simple but provides robust embedding in the latent space due to AE. Further, the classifier in the latent space makes the embedding discriminative. The standard AE is trained to minimize the following loss on the visual samples from the seen classes.

$$\mathcal{L}_{AE} = \sum_{i=1}^{n_{tr}} ||\hat{x}_i - x_i||_2^2, \tag{3}$$

where $\hat{x}_i = g(f_v(x_i))$. Here, encoder function $f_v(\cdot)$ is the visual embedding function and $g(\cdot)$ is the decoder function. We simultaneously learn a separate semantic embedding function $f_s(\cdot)$ such that semantic embedding is matched with the corresponding class visual embedding in the shared latent space of the AE. This is achieved by minimizing the following loss.

$$\mathcal{L}_{LE} = \sum_{i=1}^{n_{tr}} ||f_v(x_i) - f_s(a_{y_i})||_2^2. \tag{4}$$

Next, we use a standard softmax classifier, denoted by C, which is trained on the visual embeddings in the latent space to minimize the cross-entropy loss $\mathcal{L}_{CLFR}$ for the samples belonging to the seen classes. The overall objective function to be minimized by training on the visual-semantic data from the seen classes is given by

$$\mathcal{L}_S = \mathcal{L}_{AE} + \mathcal{L}_{LE} + \mathcal{L}_{CLFR} \tag{5}$$

In the next section, we describe the generation of the proxy samples in the early training regime of the GAN.

4.2 Exploiting Early Training Regime of GAN

In order to generate proxy samples further to be used to construct the seen/unseen domain classifier, we built upon the conditional Wasserstein GAN (CWGAN) [6] based approach of [35]. Our CWGAN consists of a generator G and a discriminator (critic) D being conditioned on the semantic class prototype a. The adversarial game between G and D optimizes the following objective function

$$\min_G \max_D \mathcal{L}_{CWGAN} = \mathbf{E}[D(x,a)] - \mathbf{E}(D(G(z,a)),a)] - \lambda \mathbf{E}[(||\nabla_{\hat{x}} D(\hat{x},a)||_2 - 1)^2], \tag{6}$$

where $\hat{x} = \alpha x + (1-\alpha x)$ with $\alpha \sim U(0,1)$, $\mathbf{E}[\cdot]$ is an expectation operator and z is noise from the standard normal distribution. The third term in Eq.(6) corresponds to gradient penalty to ensure a stable training where λ is a hyper-parameter.

It is well understood that generative models can produce the samples which match the original data distribution when G and D are locked into equilibrium. Furthermore, during the initial phase of training of CWGAN, D is confident in identifying the real and generated samples. From a different perspective, samples generated by G during the early training phase are far from the samples belonging to the seen classes. Such samples belong to the open-set classes but may not necessarily represent the samples from the unseen classes. However, we hypothesize that they can effectively be used to characterize the samples from the unseen classes in terms of entropies of label distribution. However, a natural question may arise as to what would one mean by the early training phase of GAN? Or is there any criterion to decide how early it is early? To decide on the exact iteration index of training, we are interested in; we use the criterion similar to the Inception Score (IS) [26].

The IS is used to measure the quality and the diversity of samples generated by G in a typical deep generative model. We make use of an IS-like criterion to identify the training iteration of CWGAN where proxy samples of interest can be generated. We call such criterion as the Confidence Score (CS). We formally define it first and then mention how it helps decide the training iteration to generate proxy samples. Let $p(y|x)$ denote the conditional label distribution obtained by applying the pre-trained classifier (PTC) (trained on seen classes and different than the classifier C in Section 4.1) to generated samples. Let $p(y)$ denote the distribution obtained by marginalizing over the label distributions obtained by passing the generated samples through the PTC. The CS is given by

$$CS = exp(\mathbf{E}[KL(p(y|x)||p(y))]) \tag{7}$$

where $KL(\cdot)$ represents the KL divergence. The IS uses Inception model [30] to obtain $p(y|x)$ and $p(y)$. We emphasize that we do not use the Inception model to obtain $p(y|x)$ and $p(y)$ but use a PTC on seen classes of the given dataset. This is because the Inception model [30] is trained on images of objects, whereas we also experiment on videos of actions. In such a scenario, the Inception model may not be appropriate to obtain the conditional and marginal probabilities.

For well trained CWGAN, the generator can produce samples from all the seen classes. Marginalizing the label probability distributions of these samples results in $p(y)$ to be a uniform distribution. On the other hand, $p(y|x)$ changes from uniform distribution to a peaky distribution as training of CWGAN progresses. During the initial training phase, samples produced by G are far from the real samples of the seen classes. Hence, when such samples are passed through the PTC, their label probability distribution $p(y|x)$ follows a uniform distribution, resulting in lower CS. As training progresses, G can produce realistic samples belonging to the seen classes. When passed through the PTC, these images generate a peaky distribution, indicating that they belong to one of the seen classes. This results in larger KL-divergence between a marginal and conditional distribution and hence the larger CS.

We identify the training iterations for which $s_l < CS < s_h$ where thresholds s_l and s_h are set using validation. The larger value of CS ($CS > s_h$) corresponds to G producing realistic samples of the seen classes. Hence corresponding iterations are inappropriate for synthesizing the proxy samples. On the other hand, a smaller value of CS ($CS < s_l$) also has limited use. This is because G conditioned on the semantic prototypes and normally distributed noise can produce only a normally distributed samples.

Next, we describe how the synthesized proxy samples and visual samples of the seen classes can be used to train the GZSL model, infer the domain label, and hence the class label.

4.3 Entropy Separation Criterion

Our goal is to identify whether a test sample belongs to the seen (close-set) or the unseen domain (open-set). One way to identify this is by looking at the entropy of the label distribution of the

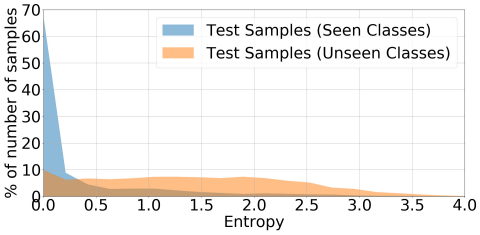

Figure 2: Histogram of the entropies of test samples from the seen and unseen classes for the AWA1 dataset. The entropy is obtained from the output of the classifier C of the VSMM, which is trained using Eq.(5). Uniform distribution of entropies of the samples from the unseen classes makes them challenging to distinguish from the samples of the seen classes based on the threshold.

sample obtained using the classifier pre-trained on the seen classes. One may expect the lower average entropy for the samples from the seen classes and higher average entropy for the samples coming from outside the seen classes. This is shown in Figure 2, where histogram of entropies of the test samples from seen and unseen classes are plotted for the HMDB51 dataset. The classifier (C) from the VSMM is used to obtain the entropies of the test samples when the VSMM is trained using the objective function in Eq.(5) on visual-semantic data from the seen classes. We observe that the samples from the seen classes have low entropies. However, the entropies of the samples from the unseen classes range from smaller to larger values. In such a case, the samples from the unseen classes would be difficult to identify based on the entropy. Therefore, we aim to increase the separation between the entropies of the samples from the seen and unseen classes. As the GZSL setting does not allow us to use any data from the unseen classes, we resort to the proxy samples that are generated, as explained in Section 4.2. Let H_S and H_P denote the average entropy of samples from the seen classes and the proxy samples. Then the entropy separation loss with margin γ is given by

$$\mathcal{L}_{ES} = \max[0, H_S - H_P + \gamma], \tag{8}$$

The overall objective for training the VSMM is given by

$$\mathcal{L} = \mathcal{L}_S + \mathcal{L}_{ES}. \tag{9}$$

4.4 Inference

Once the VSMM is trained by optimizing the loss in Eq.(9), in the first step of the inference process, the domain of a test sample is identified. For this, entropy H_t of the label distribution is calculated using the softmax activations of the classifier C. The following decision rule gives the domain of the test sample

$$\hat{y_t} \in \begin{cases} \mathcal{Y}_S & \text{if } H_t < \delta_l\,, \\ \mathcal{Y}_U & \text{if } H_t > \delta_h, \\ \mathcal{Y}_S \bigcup \mathcal{Y}_U & \text{if } \delta_l < H_t < \delta_h, \end{cases} \tag{10}$$

where the thresholds δ_l and δ_h are set using validation. The test samples for which $\delta_l < H_t < \delta_h$, the classifier C is not confident about the domain of the sample, hence its domain can be seen or unseen classes ($\mathcal{Y}_S \bigcup \mathcal{Y}_U$). In the second step, a separate inference module based on the domain label is used to identify the final class label of the test sample. Specifically, if the domain is identified as the seen class domain, then the class label is inferred using SL (directly from the classifier C output). If the domain is identified as the unseen class domain, then the class label is inferred using ZSL (Eq.(1)). If the domain is seen or unseen, then GZSL (Eq.(2)) is used to infer the class label.

The proposed VSMM is end-to-end in the sense that it can infer the domain labels and final class labels of the test samples. However, it can also be used as a pre-processing step to determine the domain labels of the samples followed by existing GZSL methods to infer the final class labels.

Importance of two thresholds: We use two thresholds instead of a single threshold to avoid the hard division and hence misclassification due to the wrong assignment of the sample to either seen or unseen classes. From a different perspective, we expect that the entropies of proxy samples and test samples from the unseen classes may differ slightly. This is because although the proxy samples are representative of the samples from the extensive open-set, they may not necessarily correspond to the samples from the unseen classes. Hence, the single threshold separating the samples from the seen classes and proxy samples may not be effective in separating the samples from the seen and unseen classes during testing. To overcome this difficulty, we use a two threshold system resulting in a margin of the entropies. Furthermore, we ideally want all the test samples to be classified using either SL or ZSL. However, some of the test samples are classified using GZSL due to two threshold system. We experimentally observe that the number of samples classified using GZSL is small. Algorithm 1 explains the GZSL training and testing process using the proposed method.

Algorithm 1: GZSL using the proposed method.

Training:

1. Train CWGAN using Eq.(6);
2. Generate the proxy samples and calculate the CS (Eq.(7)) using PTC in each training iteration;
3. Identify the training iterations of interest such that $s_l < CS < s_h$;
4. Train the classifier C by optimizing Eq.(9) using the visual-semantic samples from the seen classes and the proxy samples generated in training iterations of Step 3;

Inference:

5. For a test sample, identify the domain using Eq.(10);
6. Based on the domain of the test sample, identify the class label using SL, ZSL or GZSL;

5 EXPERIMENTS

Datasets and Feature: We evaluate the proposed method on four diverse datasets. AWA1 [34] and CUB [32] are coarse grained and fine-grained datasets, respectively, for recognition of animals and birds from images. HMDB51 and UCF101 are two datasets for action recognition from video. The description of the datasets is given in Table 1. We use d = 2048-dimension ResNet101 [8] features for

Table 1: The description of different datasets for GZSL.

Item	AWA1	CUB	HMDB51	UCF101
$\lvert\mathcal{Y}_S\rvert$	40	150	26	51
$\lvert\mathcal{Y}_U\rvert$	10	50	25	50
$\lvert\mathcal{D}\rvert$	30K	11K	6.7K	13K
k	85	312	300	300

Table 2: GZSL performance comparison using average top-1 accuracy on the seen (S) and the unseen classes (U) and their harmonic mean (H). UD corresponds to using (Y) or not using (N) unseen class data during training. GM corresponds to using (Y) or not using (N) generative models.

			AWA1			CUB		
UD	**Method**	**GM**	**S**	**U**	**H**	**S**	**U**	**H**
Y	DSEN [18]	N	-	-	-	71.1	59.1	64.5
	f-CLSWGAN [35]	Y	61.4	57.9	59.6	57.7	43.7	49.7
	SE-GZSL [11]	Y	68.1	58.3	62.8	53.3	41.5	46.7
	f-VAEGAND2 [36]	Y	76.1	57.1	65.2	75.6	63.2	68.9
N	DeViSE [5]	N	68.7	13.4	22.4	74.7	17.1	27.8
	SYNC [3]	N	87.3	8.9	16.2	90.5	10.0	18.0
	SJE [1]	N	74.6	11.3	19.6	73.9	8.0	14.4
	ALE [33]	N	76.1	16.8	27.5	81.8	14.0	23.9
	SAE [10]	N	77.1	1.8	3.5	82.2	1.1	2.2
	DZSL [38]	N	68.4	32.8	47.3	57.9	19.6	29.2
	PSR [37]	N	-	-	-	73.8	20.7	32.3
	Ours	Y	65.0	51.9	**57.8**	42.7	37.6	**40.0**

images in AWA1 and CUB while manually defined attributes of dimension $k = 85$ and $k = 312$ for AWA1 and CUB, respectively. For HMDB51 and UCF101, we use the I3D [2] video feature of $d = 8196$ dimension and $k = 300$-dim word2vec as semantic prototypes provided by [16]. For AWA1 and CUB we use the new seen/unseen split proposed by [34] while for HMDB1 and UCF101 we use the seen/unseen splits provided by [16] where we experiment on 30 splits and report the average performance. We use the evaluation criterion proposed by [34] where we report the average top 1 accuracy on the seen (S) and the unseen (U) classes and the harmonic mean (H) of S and U.

Model Architecture: We use fully connected NN models for the proposed method. The details of the model architecture are given in supplementary material due to space constraints. We fix $\gamma = 1, \delta_l = 0.5$ and $\delta_h = 1.5$ for all the datasets while s_l and s_h are set separately for each dataset based on validation.

Comparison with the state of the art methods: We compare the performance of our model with the existing generative and non-generative GZSL methods. We emphasize that many of the existing generative model-based methods violate the principal assumption of GZSL of not using any data related to the unseen classes during training. *It is imperative that such methods get the added advantage of using the unseen class semantic information during training and hence are not directly comparable with our method.* Nonetheless, we still report the performance of these methods and show that ours perform comparably with these methods.

Table 3: GZSL performance comparison using average top-1 accuracy on the seen (S) and the unseen classes (U) and their harmonic mean (H). UD corresponds to using (Y) or not using (N) the data from the unseen classes during training. GM corresponds to using (Y) or not using (N) generative models. $*$ represents results are reported from [16]

			HMDB51			UCF101		
UD	**Method**	**GM**	**S**	**U**	**H**	**S**	**U**	**H**
Y	GGM* [20]	Y	-	-	20.1	-	-	17.5
	f-CLSWGAN* [35]	Y	52.6	23.7	32.7	74.8	20.7	32.4
	CEWGAN [16]	Y	55.6	26.8	36.1	75.9	24.8	37.3
N	ESZSL [24]	N	71.1	4.3	8.0	96.4	3.2	3.4
	SJE* [1]	N	-	-	10.5	-	-	8.9
	ConSE* [21]	N	-	- 15.4	-	-	12.7	
	SADLE [7]	N	76.2	16.0	26.3	98.1	12.7	22.5
	Ours	N	61.1	25.3	**35.6**	69.2	17.1	**27.4**

Results: Table 2 compares the proposed method with the existing generative and non-generative methods for AWA1 and CUB datasets. All the generative model-driven methods outperform those with non-generative models. This is because synthesizing the visual samples from the unseen classes and using them during training reduces the model bias towards the seen classes. Although generative, our method synthesizes the proxy samples and not the visual samples from the unseen classes during training. In the same experimental setting, our method beats the H of existing state of the art DZSL[38] (in AWA1) by 10.5 and PSR [37] (in CUB) by 7.7. Our method performs comparably with other generative methods such as f-CLSWGAN [35], SE-GZSL [11], which uses CWGAN and VAE, respectively. Our performance is comparable with f-VAEGAND2 [36] on AWA, while our model performs poorly on CUB as compared to f-VAEGAND2 [36].

Table 3 shows the results for HMDB51 and UCF101. These are relatively complex datasets, but we still outperform the existing models. We observe that the generative-model driven methods outperform the non-generative methods. In the standard GZSL setting, we improve by 9.3 and 4.9 in H over the non-generative model SADLE in [28], for HMDB51 and UCF101, respectively. Generative-model driven methods GGM [20], f-CLSWGAN [35] and CEWGAN [16] uses unseen class prototypes during training to generate unseen class visual samples. We still outperform GGM, f-CLSWGAN, and deliver comparable performance with CEWGAN. We note that although we use the approach similar to CEWGAN for converting GZSL into SL and ZSL, CEWGAN trains separate classifiers for the unseen classes based on synthesized features.

The improvement due to the proposed entropy separation loss can also be observed from the Figure 4 which shows the effect of entropy separation during training and testing for AWA1 dataset. The left plot of Figure 4 shows the histogram of entropies of the samples from the seen classes and entropies of the proxy samples at the end of the training of the VSMM. It can be observed that the visual samples from the seen classes have a low entropy, while the proxy samples possess high entropy. The right plot of Figure 4 shows that a large number of test samples from the seen classes and the unseen classes have low and high entropies, respectively. By using decision rule in Eq.(10), the test samples can be labeled using

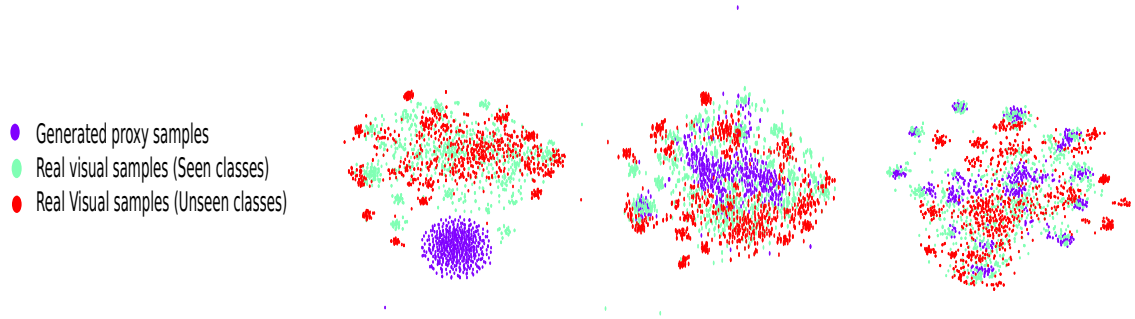

Figure 3: HMDB51: t-SNE plots show the generated samples as training of CWGAN progresses. Left: Initial training phase where discriminator (D) of the CWGAN can easily discriminate between generated fake samples and real samples (train samples) of the seen classes. Centre: Generated samples are still far from the real samples of seen classes and can be used as proxy samples representing the samples from the open-set of classes. Right: Generator (G) of CWGAN can generate fake samples that are indistinguishable from the real samples of the seen classes.

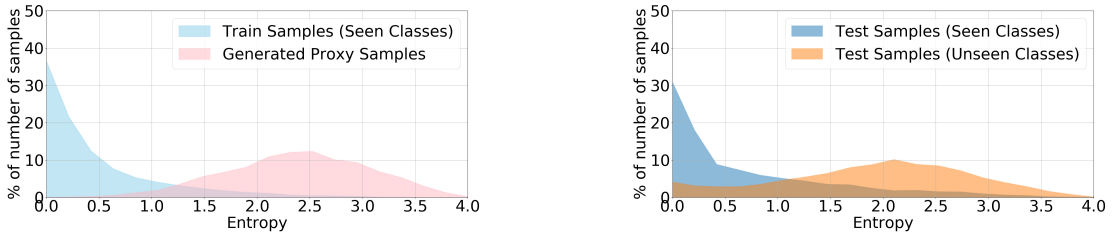

Figure 4: AWA1: Left: Histogram of entropies of the samples from the seen classes and generated proxy samples during training. Right: Histogram of entropies of the samples from the seen and unseen classes during testing. The model trained with entropy separation criterion generalizes well during testing providing better separation of the entropies of the samples from the seen and unseen classes.

Table 4: The effect of entropy separation criterion over the Baseline. For Ours, the average value and standard deviation (in brackets) of H are reported. The average is reported as different models of CWGAN can be selected based on training iterations to generate proxy samples such that $s_l < CS < s_h$.

Model	AWA1	CUB	HMDB51	UCF101
Baseline	33.2	28.0	18.8	17.0
Ours	57.8 (±3.5)	40.0(±2.7)	35.6(±1.6)	27.4(±2.0)

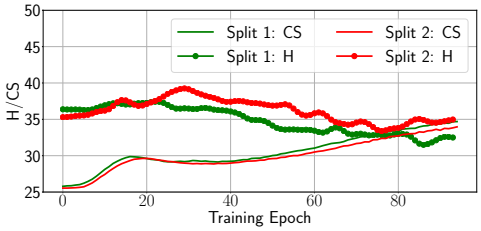

Figure 5: The plot is showing the evolution of training of CWGAN and its effect on CS and H for two different splits of HMDB51. CS increases as the training of CWGAN progress whereas H initially increases and then decreases.

SL, ZSL, or GZSL. The two histograms highlight the importance of the two thresholds. It can be observed that there is a good separation between the entropies of the visual samples from the seen classes and proxy samples during training. As the proxy samples may not necessarily constitute the unseen classes; there exists a small overlap between the entropies of the visual samples from the seen and unseen classes during testing (right plot Figure 4). To avoid misclassification due to hard thresholding, we use two thresholds where samples whose entropies fall in the $[\delta_l, \delta_h]$ are classified using GZSL.

5.1 Ablation Study

Effect of entropy separation criterion: We use Baseline to denote the performance of the VSMM, where we do not include the entropy separation loss during training, and all the test samples are classified using GZSL. Baseline suffers from the model bias as there is no provision to overcome the model bias towards the seen classes. The proxy samples and entropy separation criterion help in reducing the model bias by converting the GZSL problem into SL and ZSL. From Table 4, it can be observed that training VSMM with entropy separation loss improves the H over the Baseline by 24.6 (AWA1), 12.0 (CUB), 16.8 (HMDB51), and 10.4 (UCF101).

Selection of training iteration of CWGAN: It is important to identify the appropriate training iteration for generating proxy

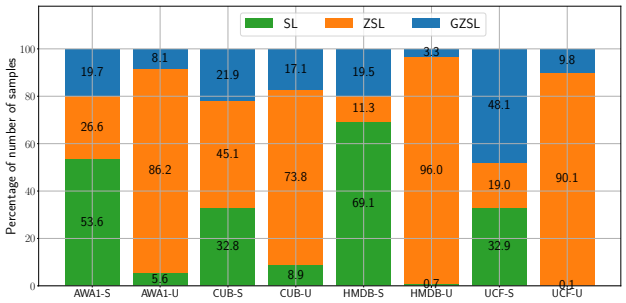

Figure 6: Percentage of the test samples that are classified using SL, ZSL, or GZSL. Dataset names followed by S or U correspond to the results for the test samples from the seen classes or unseen classes, respectively.

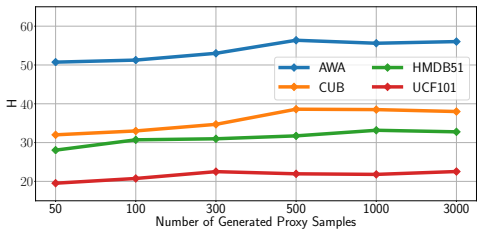

Figure 7: Effect of the number of generated proxy samples on H. Plots for HMDB51 and UCF101 are for a random split.

samples for unseen classes. Once we decide upon the training iteration, we use the G of CWGAN, which is trained until that iteration to generate the proxy samples. Selecting the training iteration at the start of the training is not desirable. As the G is conditioned on the prototypes of seen classes as well as normally distributed noise, generated samples would be normally distributed but very far from the seen classes (Figure 3 (left)). The training iteration, when D of CWGAN can not distinguish between real and generated samples (this can be identified by looking at the Wasserstein's distance), is not desirable. Because G in such a case would generate the visual samples from the seen classes (which is abundantly available) (Figure 3 (right)). Hence, we select the training iteration in the early training phase of CWGAN (Figure 3 (center)) where normally distributed samples start to *disperse*. To exactly arrive at the training iteration of interest, we use CS defined in Eq.(7). In Figure 5, the the CS and H are plotted against the training epochs for a couple of splits for HMDB51. It can be noted that CS increases as the training progresses, indicating that G can produce the data that matches the real data. On the other hand, H initially increases and then decreases as the training progresses. The initial increase in the H is because the VSMM training can separate the entropies of the proxy samples from the entropies of the visual samples from the seen classes. The decrease in H in the later part of CWGAN training is because generated proxy samples get closer to the visual samples of the seen classes, and hence the entropy separation loss during the training of VSMM has no effect. It leads to confusion about separating the entropies of the visual samples from the seen classes and proxy samples during training. This results in poor performance while separating the visual samples from the seen and the unseen classes during testing. In Table 4 we report the average and standard deviation of H for different training iterations for which $s_l < CS < s_h$.

Effect of two thresholds δ_l and δ_h: Ideally, we would like to convert the GZSL problem into SL and ZSL. However, as explained in section 4.4, we employ two threshold system for the inference. Hence, class labels of some of the samples are inferred using GZSL. Figure 6 shows the percentage of test samples from the seen and unseen classes that are classified using SL, ZSL, and GZSL. We note that for all the datasets, the relatively large number of test samples from the unseen classes are classified using ZSL. This can significantly reduce the misclassification due to unseen class test samples. We also note that although a significant fraction of the test samples from the seen classes are classified using GZSL, they are less likely to get misclassified. This is due to the strong model bias towards the seen classes. We note that although the numbers of samples that are classified using SL, ZSL, and GZSL are encouraging, this does not reflect into the overall performance in terms of H. This can be attributed to relatively simpler VSMM. As our method acts as a pre-processing step, one can use any existing GZSL model to infer the class label after the domain of the test sample is identified by our method. To that end, one can expect further improvement in the GZSL performance in terms of H.

Effect of number of generated proxy samples on H: We evaluate the proposed method by generating different numbers of proxy samples. Figure 7 shows that the H increases as the number of proxy sample increases. However, we do not observe the increase in H when the number of samples is more than 3K. As the proxy samples are not the same as the visual samples from the unseen classes, using a large number of proxy samples is recommended to capture the diversity of the visual samples from the open-set.

6 CONCLUSION

In this paper, we have proposed a method that makes use of the early training regime of the generative model and entropy separation criterion to address the problem of GZSL. The proxy samples generated using the early training regime are representative of the classes from the open-set, which contains the unseen classes. To identify the exact training iteration, we used the Confidence Score of the generated proxy samples given by a pre-trained classifier. The proxy samples, along with samples from the seen classes, are used to train a separate classifier with entropy separation loss in an NN-based visual-semantic mapping module (VSMM). Such entropy separation loss keeps the average entropy of the samples from the seen classes low while increasing the average entropy of the proxy samples. This an objective helps in identifying the test samples from the seen and unseen classes by looking at their entropies obtained from the classifier output. The coarse separation of the test samples in the seen or unseen domain then follows the identification of class labels using supervised learning, ZSL or GZSL. During an entire training process, our method does not use any data from the unseen classes; hence it strictly follows the GZSL settings. We demonstrate the excellent performance on two image datasets for animal and bird recognition and two video datasets for action recognition.

REFERENCES

[1] Zeynep Akata, Scott Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. 2015. Evaluation of output embeddings for fine-grained image classification. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE, 2927–2936.

[2] Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6299–6308.

[3] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. 2016. Synthesized classifiers for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5327–5336.

[4] Rafael Felix, Vijay BG Kumar, Ian Reid, and Gustavo Carneiro. 2018. Multi-modal cycle-consistent generalized zero-shot learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 21–37.

[5] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, et al. 2013. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*. 2121–2129.

[6] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of wasserstein gans. In *Advances in neural information processing systems*. 5767–5777.

[7] Omkar Gune, Biplab Banerjee, and Subhasis Chaudhuri. 2018. Structure Aligning Discriminative Latent Embedding for Zero-Shot Learning.. In *BMVC*. 218.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[9] Huajie Jiang, Ruiping Wang, Shiguang Shan, Yi Yang, and Xilin Chen. 2017. Learning Discriminative Latent Attributes for Zero-Shot Classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4223–4232.

[10] Elyor Kodirov, Tao Xiang, and Shaogang Gong. 2017. Semantic autoencoder for zero-shot learning. *arXiv preprint arXiv:1704.08345* (2017).

[11] Vinay Kumar Verma, Gundeep Arora, Ashish Mishra, and Piyush Rai. 2018. Generalized zero-shot learning via synthesized examples. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4281–4289.

[12] Vinay Kumar Verma, Gundeep Arora, Ashish Mishra, and Piyush Rai. 2018. Generalized zero-shot learning via synthesized examples. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4281–4289.

[13] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. 2009. Learning to detect unseen object classes by between-class attribute transfer. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 951–958.

[14] Jingjing Li, Mengmeng Jing, Ke Lu, Lei Zhu, Yang Yang, and Zi Huang. 2019. Alleviating Feature Confusion for Generative Zero-shot Learning. In *Proceedings of the 27th ACM International Conference on Multimedia*. 1587–1595.

[15] Yang Long, Li Liu, Ling Shao, Fumin Shen, Guiguang Ding, and Jungong Han. 2017. From zero-shot learning to conventional supervised classification: Unseen visual data synthesis. (2017).

[16] Devraj Mandal, Sanath Narayan, Sai Kumar Dwivedi, Vikram Gupta, Shuaib Ahmed, Fahad Shahbaz Khan, and Ling Shao. 2019. Out-Of-Distribution Detection for Generalized Zero-Shot Action Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[17] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[18] Shaobo Min, Hantao Yao, Hongtao Xie, Zheng-Jun Zha, and Yongdong Zhang. 2019. Domain-Specific Embedding Network for Zero-Shot Recognition. In *Proceedings of the 27th ACM International Conference on Multimedia*. 2070–2078.

[19] Ashish Mishra, Anubha Pandey, and Hema A Murthy. 2020. Zero-shot learning for action recognition using synthesized features. *Neurocomputing* (2020).

[20] Ashish Mishra, Vinay Kumar Verma, M Shiva Krishna Reddy, S Arulkumar, Piyush Rai, and Anurag Mittal. 2018. A generative approach to zero-shot and few-shot action recognition. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 372–380.

[21] Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S Corrado, and Jeffrey Dean. 2013. Zero-shot learning by convex combination of semantic embeddings. *arXiv preprint arXiv:1312.5650* (2013).

[22] Devi Parikh and Kristen Grauman. 2011. Relative attributes. In *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 503–510.

[23] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. 2016. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 49–58.

[24] Bernardino Romera-Paredes and Philip Torr. 2015. An embarrassingly simple approach to zero-shot learning. In *International Conference on Machine Learning*. 2152–2161.

[25] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. In *Advances in neural information processing systems*. 2234–2242.

[26] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. In *Advances in neural information processing systems*. 2234–2242.

[27] Mert Bulent Sariyildiz and Ramazan Gokberk Cinbis. 2019. Gradient Matching Generative Networks for Zero-Shot Learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[28] Yi-Ren Yeh Shih-Yen Tao, Yao-Hung Hubert Tsai and Yu-Chiang Frank Wang. 2017. Semantics-Preserving Locality Embedding for Zero-Shot Learning. In *BMVC*.

[29] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. 2013. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*. 935–943.

[30] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2818–2826.

[31] Donghui Wang, Yanan Li, Yuetan Lin, and Yueting Zhuang. 2016. Relational Knowledge Transfer for Zero-Shot Learning.. In *AAAI*, Vol. 2. 7.

[32] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. 2010. Caltech-UCSD birds 200. (2010).

[33] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele. 2016. Latent embeddings for zero-shot classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 69–77.

[34] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. 2017. Zero-shot learning-A comprehensive evaluation of the good, the bad and the ugly. *arXiv preprint arXiv:1707.00600* (2017).

[35] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. 2018. Feature generating networks for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5542–5551.

[36] Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. 2019. F-VAEGAN-D2: A Feature Generating Framework for Any-Shot Learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[37] Soma Biswas Yashas Annadani. 2018. Preserving Semantic Relations for Zero-Shot Learning. In *CVPR*.

[38] Li Zhang, Tao Xiang, Shaogang Gong, et al. 2017. Learning a deep embedding model for zero-shot learning. (2017).

[39] Ziming Zhang and Venkatesh Saligrama. 2015. Zero-shot learning via semantic similarity embedding. In *Proceedings of the IEEE International Conference on Computer Vision*. 4166–4174.