

Multi-modal Cycle-consistent Generalized Zero-Shot Learning

Rafael Felix, Vijay Kumar B G, Ian Reid, Gustavo Carneiro

The University of Adelaide, SA 5005, Australia
{rafael.felix,vijay.kumar,ian.reid,gustavo.carneiro}@adelaide.edu.au

Abstract. In generalized zero shot learning (GZSL), the set of classes are split into seen and unseen classes, where training relies on the semantic features of the seen and unseen classes and the visual representations of only the seen classes, while testing uses the visual representations of the seen and unseen classes. Current methods address GZSL by learning a transformation from the visual to the semantic space, exploring the assumption that the distribution of classes in the semantic and visual spaces is relatively similar. Such methods tend to transform unseen testing visual representations into one of the seen classes' semantic features instead of the semantic features of the correct unseen class, resulting in low accuracy GZSL classification. Recently, generative adversarial networks (GAN) have been explored to synthesize visual representations of the unseen classes from their semantic features - the synthesized representations of the seen and unseen classes are then used to train the GZSL classifier. This approach has been shown to boost GZSL classification accuracy, but there is one important missing constraint: there is no guarantee that synthetic visual representations can generate back their semantic feature in a multi-modal cycle-consistent manner. This missing constraint can result in synthetic visual representations that do not represent well their semantic features, which means that the use of this constraint can improve GAN-based approaches. In this paper, we propose the use of such constraint based on a new regularization for the GAN training that forces the generated visual features to reconstruct their original semantic features. Once our model is trained with this multi-modal cycle-consistent semantic compatibility, we can then synthesize more representative visual representations for the seen and, more importantly, for the unseen classes. Our proposed approach shows the best GZSL classification results in the field in several publicly available datasets.

Keywords: generalized zero-shot learning; generative adversarial networks; cycle consistency loss

1 Introduction

Generalized Zero-shot Learning (GZSL) separates the classes of interest into a sub-set of seen classes and another sub-set of unseen classes. The training process uses the semantic features of both sub-sets and the visual representations of only the seen classes; while the testing process aims to classify the visual representations of both sub-sets [2, 3]. The semantic features available for both the

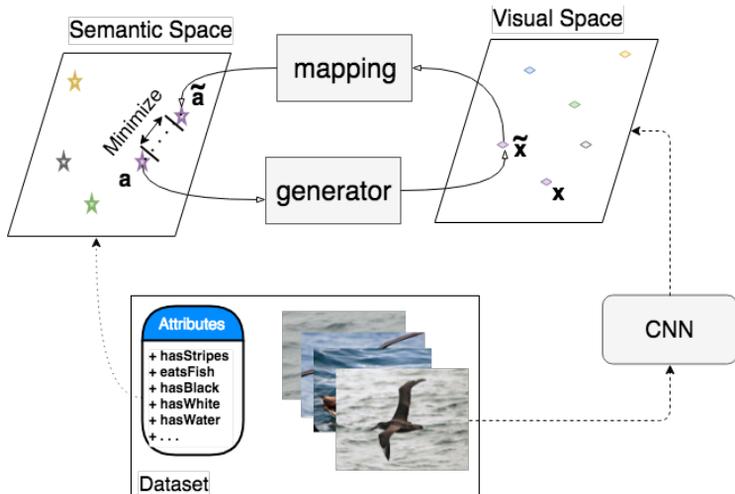


Fig. 1. Overview of the proposed multi-modal cycle-consistent GZSL approach. Our approach extends the idea of synthesizing visual representations of seen and unseen classes in order to train a classifier for the GZSL problem [1]. The main contribution of the paper is the use of a new multi-modal cycle consistency loss in the training of the visual feature generator that minimizes the reconstruction error between the semantic feature \mathbf{a} , which was used to synthesize the visual feature $\tilde{\mathbf{x}}$, and the reconstructed semantic feature $\tilde{\mathbf{a}}$ mapped from $\tilde{\mathbf{x}}$. This loss is shown to constrain the optimization problem more effectively in order to produce useful synthesized visual features for training the GZSL classifier.

training and testing classes are typically acquired from other domains, such as visual features [4], text [5, 6, 3], or learned classifiers [7]. The traditional approach to address this challenge [2] involves the learning of a transformation from the visual to the semantic space of the seen classes. Testing is then performed by transforming the visual representation of the seen and unseen classes into this semantic space, where classification is typically achieved with a nearest neighbor classifier that selects the closest class in the semantic space. In contrast to Zero-shot Learning (ZSL), which uses only the unseen domain for testing, GZSL approaches tend to be biased towards the seen classes, producing poor classification results, particularly for the unseen testing classes [1].

These traditional approaches rely on the assumption that the distributions observed in the semantic and visual spaces are relatively similar. Recently, this assumption has been relaxed to allow the semantic space to be optimized together with the transformation from the visual to the semantic space [8] - this alleviates the classification bias mentioned above to a certain degree. More recent approaches consist of building a generative adversarial network (GAN) that synthesizes visual representations of the seen and unseen classes directly from their semantic representation [9, 8]. These synthesized features are then used to train a multi-class classifier of seen and unseen classes. This approach has been shown to improve the GZSL classification accuracy, but an obvious weakness is that

the unconstrained nature of the generation process may let the approach generate unrepresentative synthetic visual representations, particularly of the unseen classes (i.e., representations that are far from possible visual representations of the test classes).

The **main contribution** of this paper is a **new regularization of the generation of synthetic visual representations in the training of GAN-based methods that address the GZSL classification problem**. This regularization is **based on a multi-modal cycle consistency loss term that enforces good reconstruction from the synthetic visual representations back to their original semantic features** (see Fig. 1). This regularization is motivated by the cycle consistency loss applied in training GANs [10] that forces the generative training approach to produce more constrained visual representations. We argue that this constraint preserves the semantic compatibility between visual features and semantic features. Once our model is trained with this multi-modal cycle consistency loss term, we can then synthesize visual representations for unseen classes in order to train a GZSL classifier [11, 1].

Using the experimental setup described by Xian et al. [1], we show that our proposed regularization provides significant improvements not only in terms of GZSL classification accuracy, but also ZSL on the following datasets: Caltech-UCSD-Birds 200-2011 (CUB) [12, 2], Oxford-Flowers (FLO) [13], Scene Categorization Benchmark (SUN) [14, 2], Animals with features (AWA) [4, 2], and *ImageNet* [15]. In fact, the experiments show that our proposed approach holds the current best ZSL and GZSL classification results in the field for these datasets.

2 Literature Review

The starting point for our literature review is the work by Xian et al. [2, 1], who proposed new benchmarks using commonly accepted evaluation protocols on publicly available datasets. These benchmarks allow a fair comparison among recently proposed ZSL and GZSL approaches, and for this reason we explore those benchmarks to compare our results with the ones obtained from the current state of the art in the field. We provide a general summary of the methods presented in [2], and encourage the reader to study that paper in order to obtain more details on previous works. The majority of the ZSL and GZSL methods tend to compensate the lack of visual representation of the unseen classes with the learning of a mapping between visual and semantic spaces. For instance, a fairly successful approach is based on a bi-linear compatibility function that associates visual representation and semantic features. Examples of such approaches are ALE [16], DEVISE [17], SJE [18], ESZSL [19], and SAE [20]. Despite their simplicity, these methods tend to produce the current state-of-the-art results on benchmark datasets [2]. A straightforward extension of the methods above is the exploration of a non-linear compatibility function between visual and semantic spaces. These approaches, exemplified by LATEM [21] and CMT [6], tend not to be as competitive as their bi-linear counterpart, probably because the more complex models need larger training sets to generalize more effectively. Seminal ZSL and GZSL methods were based on models relying on learning intermediate feature classifiers, which are combined to predict image classes (e.g., DAP and

IAP) [4] – these models tend to present relatively poor classification results. Finally, hybrid models, such as SSE [3], CONSE [22], SYNC [23], rely on a mixture model of seen classes to represent images and semantic embeddings. These methods tend to be competitive for classifying the seen classes, but not for the unseen classes.

The main disadvantage of the methods above is that the lack of visual training data for the unseen classes biases the mapping between visual and semantic spaces towards the semantic features of seen classes, particularly for unseen test images. This is an issue for GZSL because it has a negative effect in the classification accuracy of the unseen classes. Recent research address this issue using GAN models that are trained to synthesize visual representations for the seen and unseen classes, which can then be used to train a classifier for both the seen and unseen classes [8, 9]. However, the unconstrained generation of synthetic visual representations for the unseen classes allows the production of synthetic samples that may be too far from the actual distribution of visual representations, particularly for the unseen classes. In GAN literature, this problem is known as unpaired training [10], where not all source samples (e.g., semantic features) have corresponding target samples (e.g., visual features) for training. This creates a highly unconstrained optimization problem that has been solved by Zhu et al. [10] with a cycle consistency loss to push the representation from the target domain back to the source domain, which helped constraining the optimization problem. In this paper, we explore this idea for GZSL, which is a novelty compared to previous GAN-based methods proposed in GZSL and ZSL.

3 Multi-modal Cycle-consistent Generalized Zero Shot Learning

In GZSL and ZSL [2], the dataset is denoted by $\mathcal{D} = \{(\mathbf{x}, \mathbf{a}, y)_i\}_{i=1}^{|\mathcal{D}|}$ with $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^K$ representing visual representation (e.g., image features from deep residual nets [24]), $\mathbf{a} \in \mathcal{A} \subseteq \mathbb{R}^L$ denoting L -dimensional semantic feature (e.g., set of binary attributes [4] or a dense *word2vec* representation [25]), $y \in \mathcal{Y} = \{1, \dots, C\}$ denoting the image class, and $|\cdot|$ representing set cardinality. The set \mathcal{Y} is split into seen and unseen subsets, where the seen subset is denoted by \mathcal{Y}_S and the unseen subset by \mathcal{Y}_U , with $\mathcal{Y} = \mathcal{Y}_S \cup \mathcal{Y}_U$ and $\mathcal{Y}_S \cap \mathcal{Y}_U = \emptyset$. The dataset \mathcal{D} is also divided into mutually exclusive training and testing subsets: \mathcal{D}^{Tr} and \mathcal{D}^{Te} , respectively. Furthermore, the training and testing sets can also be divided in terms of the seen and unseen classes, so this means that \mathcal{D}_S^{Tr} denotes the training samples of the seen classes, while \mathcal{D}_U^{Tr} represents the training samples of the unseen classes (similarly for \mathcal{D}_S^{Te} and \mathcal{D}_U^{Te} for the testing set). During training, samples in \mathcal{D}_S^{Tr} contain the visual representation \mathbf{x}_i , semantic feature \mathbf{a}_i and class label y_i ; while the samples in \mathcal{D}_U^{Tr} comprise only the semantic feature and class label. During ZSL testing, only the samples from \mathcal{D}_U^{Te} are used; while in GZSL testing, all samples from \mathcal{D}^{Te} are used. Note that for ZSL and GZSL problems, only the visual representation of the testing samples is used to predict the class label.

Below, we first explain the f-CLSWGAN model [1], which is the baseline for the implementation of the main contribution of this paper: the multi-modal cycle

consistency loss used in the training for the feature generator in GZSL models based on GANs. The loss, feature generator, learning and testing procedures are explained subsequently.

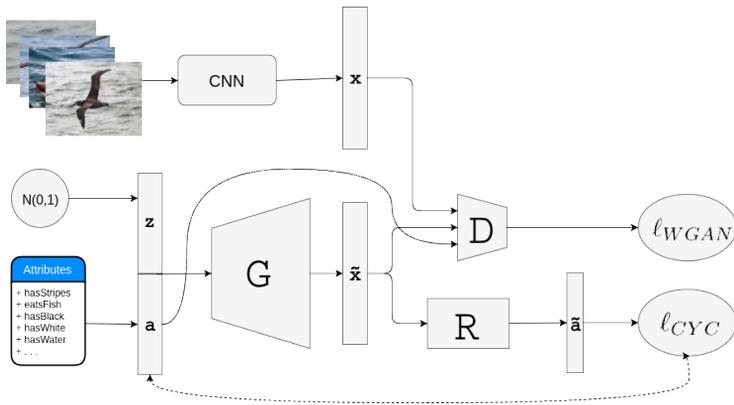


Fig. 2. Overview of the multi-modal cycle-consistent GZSL model. The visual features, represented by \mathbf{x} , are extracted from a state-of-art CNN model, and the semantic features, represented by \mathbf{a} , are available from the training set. The generator $G(\cdot)$ synthesizes new visual features $\tilde{\mathbf{x}}$ using the semantic feature and a randomly sampled noise vector $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and the discriminator $D(\cdot)$ tries to distinguish between real and synthesized visual features. Our main contribution is focused on the integration of a multi-modal cycle consistency loss (at the bottom) that minimizes the error between the original semantic feature \mathbf{a} and its reconstruction $\tilde{\mathbf{a}}$, produced by the regressor $R(\cdot)$.

3.1 f-CLSWGAN

Our approach is an extension of the feature generation method proposed by Xian et al. [1], which consists of a classification regularized generative adversarial network (f-CLSWGAN). This network is composed of a generative model $G: \mathcal{A} \times \mathcal{Z} \rightarrow \mathcal{X}$ (parameterized by θ_G) that produces a visual representation $\tilde{\mathbf{x}}$ given its semantic feature \mathbf{a} and a noise vector $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ sampled from a multi-dimensional centered Gaussian, and a discriminative model $D: \mathcal{X} \times \mathcal{A} \rightarrow [0, 1]$ (parameterized by θ_D) that tries to distinguish whether the input \mathbf{x} and its semantic representation \mathbf{a} represent a true or generated visual representation and respective semantic feature. Note that while the method developed by Yan et al. [26] concerns the generation of realistic images, our proposed approach, similarly to [1, 8, 9], aims to generate visual representations, such as the features from a deep residual network [24] - the strategy based on visual representation has shown to produce more accurate GZSL classification results compared to the use of realistic images. The training algorithm for estimating θ_G and θ_D follows a minimax game, where $G(\cdot)$ generates synthetic visual representations that are supposed to fool the discriminator, which in turn tries to distinguish the

real from the synthetic visual representations. We rely on one of the most stable training methods for GANs, called Wasserstein GAN, which uses the following loss function [27]:

$$\theta_G^*, \theta_D^* = \arg \min_{\theta_G} \max_{\theta_D} \ell_{WGAN}(\theta_G, \theta_D), \quad (1)$$

with

$$\begin{aligned} \ell_{WGAN}(\theta_G, \theta_D) = & \mathbb{E}_{(\mathbf{x}, \mathbf{a}) \sim \mathbb{P}^{x, a}} [D(\mathbf{x}, \mathbf{a}; \theta_D)] - \mathbb{E}_{(\tilde{\mathbf{x}}, \mathbf{a}) \sim \mathbb{P}_G^{x, a}} [D(\tilde{\mathbf{x}}, \mathbf{a}; \theta_D)] \\ & - \lambda \mathbb{E}_{(\hat{\mathbf{x}}, \mathbf{a}) \sim \mathbb{P}_{\alpha}^{x, a}} [(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}}, \mathbf{a}; \theta_D)\|_2 - 1)^2], \end{aligned} \quad (2)$$

where $\mathbb{E}[\cdot]$ represents the expected value operator, $\mathbb{P}_S^{x, a}$ is the joint distribution of visual and semantic features from the seen classes (in practice, samples from that distribution are the ones in \mathcal{D}_S^{Tr}), $\mathbb{P}_G^{x, a}$ represents the joint distribution of semantic features and the visual features produced by the generative model $G(\cdot)$, λ denotes the penalty coefficient, and $\mathbb{P}_{\alpha}^{x, a}$ is the joint distribution of the semantic features and the visual features produced by $\hat{\mathbf{x}} \sim \alpha \mathbf{x} + (1 - \alpha)\tilde{\mathbf{x}}$ with $\alpha \sim \mathcal{U}(0, 1)$ (i.e., uniform distribution).

Finally, the f-CLSWGAN is trained with the following objective function:

$$\theta_G^*, \theta_C^*, \theta_D^* = \arg \min_{\theta_G, \theta_C} \max_{\theta_D} \ell_{WGAN}(\theta_G, \theta_D) + \beta \ell_{CLS}(\theta_C, \theta_G), \quad (3)$$

where $\ell_{CLS}(\theta_C, \theta_G) = -\mathbb{E}_{(\tilde{\mathbf{x}}, y) \sim \mathbb{P}_G^{x, y}} [\log P(y|\tilde{\mathbf{x}}, \theta_C)]$, with

$$P(y|\tilde{\mathbf{x}}, \theta_C) = \frac{\exp((\theta_C(y))^T \tilde{\mathbf{x}})}{\sum_{c \in \mathcal{Y}} \exp((\theta_C(c))^T \tilde{\mathbf{x}})} \quad (4)$$

representing the probability that the sample $\tilde{\mathbf{x}}$ has been predicted with its true label y , and β is a hyper-parameter that weights the contribution of the loss function. This regularization with the classification loss was found by Xian et al. [1] to enforce $G(\cdot)$ to generate discriminative visual representations. The model obtained from the optimization in (3) is referred to as **baseline** in the experiments.

3.2 Multi-modal Cycle Consistency Loss

The main issue present in previously proposed GZSL approaches based on generative models [1, 8, 9] is that the unconstrained nature of the generation process (from semantic to visual features) may produce image representations that are too far from the real distribution present in the training set, resulting in an ineffective multi-class classifier training, particularly for the unseen classes. The approach we propose to alleviate this problem consists of constraining the synthetic visual representations to generate back their original semantic features - this regularization has been inspired by the cycle consistency loss [10]. Figure 2 shows an overview of our proposal. This approach, representing the main contribution of this paper, is represented by the following loss:

$$\begin{aligned} \ell_{CYC}(\theta_R, \theta_G) = & \mathbb{E}_{\mathbf{a} \sim \mathbb{P}_S^a, \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} [\|\mathbf{a} - R(G(\mathbf{a}, \mathbf{z}; \theta_G); \theta_R)\|_2^2] \\ & + \mathbb{E}_{\mathbf{a} \sim \mathbb{P}_U^a, \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} [\|\mathbf{a} - R(G(\mathbf{a}, \mathbf{z}; \theta_G); \theta_R)\|_2^2], \end{aligned} \quad (5)$$

where \mathbb{P}_S^a and \mathbb{P}_U^a denote the distributions of semantic features of the seen and unseen classes, respectively, and $R: \mathcal{X} \rightarrow \mathcal{A}$ represents a regressor that estimates the original semantic features from the visual representation generated by $G(\cdot)$.

3.3 Feature Generation

Using the losses proposed in Sections 3.1 and 3.2, we can propose several feature generators. First, we pre-train the regressor $R(\cdot)$ defined below in (6), by minimizing a loss function computed only from the seen classes, as follows:

$$\ell_{REG}(\theta_R) = \mathbb{E}_{(\mathbf{a}, \mathbf{x}) \sim \mathbb{P}_S^{a,x}} [\|\mathbf{a} - R(\mathbf{x}; \theta_R)\|_2^2], \quad (6)$$

where $\mathbb{P}_S^{a,x}$ represents the real joint distribution of image and semantic features present in the seen classes. In practice, this regressor is defined by a multi-layer perceptron, whose output activation function depends on the format of the semantic vector.

Our first strategy to build a feature generator consists of pre-training a regressor (using samples from seen classes) optimized by minimizing ℓ_{REG} in (6), which produces θ_R^* and training the generator and discriminator of the WGAN using the following optimization function:

$$\theta_G^*, \theta_D^* = \arg \min_{\theta_G} \max_{\theta_D} \ell_{WGAN}(\theta_G, \theta_D) + \lambda_1 \ell_{CYC}(\theta_R^*, \theta_G), \quad (7)$$

where ℓ_{WGAN} is defined in (2), ℓ_{CYC} is defined in (5), and λ_1 weights the importance of the second optimization term. The optimization in (7) can use both the seen and unseen classes, or it can rely only the seen classes, in which case the loss ℓ_{CYC} in (5) has to be modified so that its second term (that depends on unseen classes) is left out of the optimization. The feature generator model in (7) trained with seen and unseen classes is referred to as **cycle-(U)WGAN**, while the feature generator trained with only seen classes is labeled **cycle-WGAN**.

The second strategy explored in this paper to build a feature generator involves pre-training the regressor in (6) using samples from seen classes to produce θ_R^* , and pre-training a softmax classifier for the seen classes using ℓ_{CLS} , defined in (3), which results in θ_C^* . Then we train the combined loss function:

$$\theta_G^*, \theta_D^* = \arg \min_{\theta_G} \max_{\theta_D} \ell_{WGAN}(\theta_G, \theta_D) + \lambda_1 \ell_{CYC}(\theta_R^*, \theta_G) + \lambda_2 \ell_{CLS}(\theta_C^*, \theta_G). \quad (8)$$

The feature generator model in (8) trained with seen classes is referred to as **cycle-CLSWGAN**.

3.4 Learning and Testing

As shown in [1] the training of a classifier using a potentially unlimited number of samples from the seen and unseen classes generated with $\mathbf{x} \sim G(\mathbf{a}, \mathbf{z}; \theta_G^*)$ produces more accurate classification results compared with multi-modal embedding models [16–19]. Therefore, we train a final softmax classifier $P(y|\mathbf{x}, \theta_C)$, defined in (4), using the generated visual features by minimizing the negative log likelihood loss $\ell_{CLS}(\theta_C, \theta_G^*)$, as defined in (3), where θ_G^* has been learned from one of the feature learning strategies discussed in Sec. 3.3 - the training of the classifier produces θ_C^* . The samples used for training the classifier are generated based on the task to be solved. For instance, for ZSL, we only use generated visual representations from the set of unseen classes; while for GZSL, we use the generated samples from seen and unseen classes.

Finally, the testing is based on the prediction of a class for an input test visual representation \mathbf{x} , as follows:

$$y^* = \arg \max_{y \in \tilde{\mathcal{Y}}} P(y|\mathbf{x}, \theta_C^*), \quad (9)$$

where $\tilde{\mathcal{Y}} = \mathcal{Y}$ for GZSL or $\tilde{\mathcal{Y}} = \mathcal{Y}_U$ for ZSL.

4 Experiments

In this section, we first introduce the datasets and evaluation criteria used in the experiments, then we discuss the experimental set-up and finally show the results of our approach, comparing with the state-of-the-art results.

4.1 Datasets

We evaluate the proposed method on the following ZSL/GZSL benchmark datasets, using the experimental setup of [2], namely: CUB-200-2011 [12, 1], FLO [13], SUN [2], and AWA [28, 2] – where CUB, FLO and SUN are fine-grained datasets, and AWA coarse. Table 4.1 shows some basic information about these datasets in terms of number of seen and unseen classes and number of training and testing images. For CUB-200-2011 [12, 1] and Oxford-Flowers [13], the semantic feature has 1024 dimensions produced by the character-based CNN-RNN [29] that encodes the textual description of an image containing fine-grained visual descriptions (10 sentences per image). The sentences from the unseen classes are not used for training the CNN-RNN and the per-class sentence is obtained by averaging the CNN-RNN semantic features that belong to the same class. For the FLO dataset [13], we used the same type of semantic feature with 1024 dimensions [29] as was used for CUB (please see description above). For the SUN dataset [2], the semantic features have 102 dimensions. Following the protocol from Xian et al.[2], visual features are represented by the activations of the 2048-dim top-layer pooling units of ResNet-101 [24], obtained from the entire image. For AWA [28, 2], we use a semantic feature containing 85 dimensions denoting per-class attributes. In addition, we also test our approach on *ImageNet* [15], for a split containing 100 classes for testing [30].

Table 1. Information about the datasets CUB[12], FLO[13], SUN [31], AWA[2], ImageNet [15]. Column (1) shows the number of seen classes, denoted by $|\mathcal{Y}_S|$, split into the number of training and validation classes (train+val), (2) presents the number of unseen classes $|\mathcal{Y}_U|$, (3) displays the number of samples available for training $|\mathcal{D}^{Tr}|$ and (4) shows number of testing samples that belong to the unseen classes $|\mathcal{D}_U^{Te}|$ and number of testing samples that belong to the seen classes $|\mathcal{D}_S^{Te}|$.

Name	$ \mathcal{Y}_S $ (train+val)	$ \mathcal{Y}_U $	$ \mathcal{D}^{Tr} $	$ \mathcal{D}_U^{Te} + \mathcal{D}_S^{Te} $
CUB	150 (100+50)	50	7057	1764+2967
FLO	82 (62+20)	20	1640	1155+5394
SUN	745 (580+65)	72	14340	2580+1440
AWA	40 (27+13)	10	19832	4958+5685
ImageNet	1000 (1000 + 0)	100	1.2×10^6	0+5200

The input images do not suffer any pre-processing (cropping, background subtraction, etc.) and we do not use any type of data augmentation. This ResNet-101 is pre-trained on ImageNet with 1K classes [15] and is not fine tuned. For the synthetic visual representations, we generate 2048-dim CNN features using one of the feature generation models, presented in Sec. 3.3.

For CUB, FLO, SUN, and AWA we use the zero-shot splits proposed by Xian et al. [2], making sure that none of the training classes are present on ImageNet [15]. Differently from these datasets (i.e., CUB, FLO, SUN, AWA), we observed that there is a lack of standardized experimental setup for GZSL on *Imagenet*. Recently, published papers have used *ImageNet* for GZSL using several splits (e.g., 2-hop, 3-hop), but we noticed that some of the supposedly unseen classes can actually be seen during training (e.g., in split **2-hop**, we note that the class *American mink* is assumed to be unseen, while class *Mink* is seen, but these two classes are arguably the same). Nevertheless, in order to demonstrate the competitiveness of our proposed **cycle-WGAN**, we compare it to the **baseline** using carefully selected 100 unseen classes (i.e., no overlap with 1k training seen classes) from *ImageNet* [30].

4.2 Evaluation Protocol

We follow the evaluation protocol proposed by Xian et al. [2], where results are based on average per-class top-1 accuracy. For the ZSL evaluation, top-1 accuracy results are computed with respect to the set of unseen classes \mathcal{Y}_U , where the average accuracy is independently computed for each class, which is then averaged over all unseen classes. For the GZSL evaluation, we compute the average per-class top-1 accuracy on seen classes \mathcal{Y}_S , denoted by s , the average per-class top-1 accuracy on unseen classes \mathcal{Y}_U , denoted by u , and their harmonic mean, i.e. $H = 2 \times (s \times u) / (s + u)$.

4.3 Implementation Details

In this section, we explain the implementation details of the generator $G(\cdot)$, the discriminator $D(\cdot)$, the regressor $R(\cdot)$, and the weights used for the hyper-parameters in the loss functions in (2),(3),(7) and (8) - all these terms have

Table 2. Summary of cross-validated hyper-parameters in our experiments.

	$R(\cdot)$			GAN: $G(\cdot)$ and $D(\cdot)$				Classifier		
	$lr_{R(\cdot)}$	batch	#ep	$lr_{G(\cdot)}$	$lr_{D(\cdot)}$	batch	#ep	lr	batch	#ep
CUB	$1e^{-4}$	64	100	$1e^{-4}$	$1e^{-3}$	64	926	$1e^{-4}$	4096	80
FLO	$1e^{-4}$	64	100	$1e^{-4}$	$1e^{-3}$	64	926	$1e^{-4}$	2048	100
SUN	$1e^{-4}$	64	100	$1e^{-2}$	$1e^{-2}$	64	926	$1e^{-4}$	4096	298
AWA	$1e^{-3}$	64	50	$1e^{-4}$	$1e^{-3}$	64	350	$1e^{-4}$	2048	37
<i>ImageNet</i>	$1e^{-4}$	2048	5	$1e^{-4}$	$1e^{-3}$	256	300	$1e^{-3}$	2048	300

Table 3. Comparison between the reported results of **f-CLSWGAN** [1] and our implementation of it, labeled **baseline**, where we show the top-1 accuracy on the unseen test \mathcal{Y}_U (GZSL), the top-1 accuracy for seen test \mathcal{Y}_S (GZSL), the harmonic mean H (GZSL), and the top-1 accuracy for ZSL ($T1_Z$).

Classifier	CUB				FLO				SUN				AWA			
	\mathcal{Y}_U	\mathcal{Y}_S	H	$T1_Z$												
f-CLSWGAN [1]	43.7	57.7	49.7	57.3	59.0	73.8	65.6	67.2	42.6	36.6	39.4	60.8	57.9	61.4	59.6	68.2
baseline	43.8	60.6	50.8	57.7	58.8	70.0	63.9	66.8	47.9	32.4	38.7	58.5	56.0	62.8	59.2	64.1

been formally defined in Sec. 3 and depicted in Fig. 2. The generator consists of a multi-layer perceptron (MLP) with a single hidden layer containing 4096 nodes, where this hidden layer is activated by LeakyReLU [32], and the output layer, with 2048 nodes, has a ReLU activation [33]. The weights of $G(\cdot)$ are initialized with a truncated normal initialization with mean 0 and standard deviation 0.01 and the biases are initialized with 0. The discriminator $D(\cdot)$ is also an MLP consisting of a single hidden layer with 4096 nodes, which is activated by LeakyReLU, and the output layer has no activation. The initialization of $D(\cdot)$ is the same as for $G(\cdot)$. The regressor $R(\cdot)$ is a linear transform from the visual space \mathcal{X} to the semantic space \mathcal{A} . Following [1], we set $\lambda = 10$ in (2), $\beta = 0.01$ in (3) and $\lambda_1 = \lambda_2 = 0.01$ in (7) and (8). We ran an empirical evaluation with the training set and noticed that when λ_1 and λ_2 share the same value, the training becomes stable, but a more systematic evaluation to assess the relative importance of these two hyper-parameters is still needed. Table 2 shows the learning rates for each model (denoted by $lr_{\{R(\cdot), G(\cdot), D(\cdot)\}}$), batch sizes (**batch**) and number of epochs (**#ep**) used for each dataset and model – the values for $G(\cdot)$ and $D(\cdot)$ have been estimated to reproduce the published results of our implementation of f-CLSWGAN (explained below), and the values for $R(\cdot)$ have been estimated by cross validation using the training and validation sets.

Regarding the number of visual representations generated to train the classifier, we performed a few experiments and reached similar conclusions, compared to [1]. For all experiments in the paper, we generated 300 visual representations per class [1]. We reached this number after a study that shows that for a small number of representations (below 100), the classification results were not competitive; for values superior to 200 or more, results became competitive, but unstable; and above 300, results were competitive and stable.

Since our approach is based on the f-CLSWGAN [1], we re-implemented this methodology. In the experiments, the results from our implementation of f-

CLSWGAN using a softmax classifier is labeled as **baseline**. The results that we obtained from our baseline are very similar to the reported results in [1], as shown in Table 3. For ImageNet, note that we use a split [30] that is different from previous ones used in the literature, as explained above in Sec. 4.1, so it is not possible to have a direct comparison between f-CLSWGAN [1] and our **baseline**. Nevertheless, we show in Table 6 that the results we obtain for the split [30] are in fact similar to the reported results for f-CLSWGAN [1] for similar ImageNet splits. We developed our code ¹ and perform all experiments using Tensorflow [34].

5 Results

In this section we show the GZSL and ZSL results using our proposed models **cycle-WGAN**, **cycle-(U)WGAN** and **cycle-CLSWGAN**, the baseline model f-CLSWGAN, denoted by **baseline**, and several other baseline methods previously used in the field for benchmarking [2]. Table 4 shows the **GZSL results** and Table 5 shows the **ZSL results** obtained from our proposed methods, and several baseline approaches on CUB, FLO, SUN and AWA datasets. The results in Table 6 shows that the top-1 accuracy on ImageNet for **cycle-WGAN** and **baseline** [1].

Table 4. GZSL results using per-class average top-1 accuracy on the test sets of unseen classes \mathcal{Y}_U , seen classes \mathcal{Y}_S , and the harmonic mean result H – all results shown in percentage. Results from previously proposed methods in the field extracted from [2].

Classifier	CUB			FLO			SUN			AWA		
	\mathcal{Y}_U	\mathcal{Y}_S	H									
DAP [28]	4.2	25.1	7.2	–	–	–	1.7	67.9	3.3	0.0	88.7	0.0
IAP [28]	1.0	37.8	1.8	–	–	–	0.2	72.8	0.4	2.1	78.2	4.1
DEVISE [17]	23.8	53.0	32.8	9.9	44.2	16.2	16.9	27.4	20.9	13.4	68.7	22.4
SJE [18]	23.5	59.2	33.6	13.9	47.6	21.5	14.7	30.5	19.8	11.3	74.6	19.6
LATEM [21]	15.2	57.3	24.0	6.6	47.6	11.5	14.7	28.8	19.5	7.3	71.7	13.3
ESZSL [19]	12.6	63.8	21.0	11.4	56.8	19.0	11.0	27.9	15.8	6.6	75.6	12.1
ALE [16]	23.7	62.8	34.4	13.3	61.6	21.9	21.8	33.1	26.3	16.8	76.1	27.5
SAE [20]	8.8	18.0	11.8	–	–	–	7.8	54.0	13.6	1.8	77.1	3.5
baseline [1]	43.8	60.6	50.8	58.8	70.0	63.9	47.9	32.4	38.7	56.0	62.8	59.2
cycle-WGAN	46.0	60.3	52.2	59.1	71.1	64.5	48.3	33.1	39.2	56.4	63.5	59.7
cycle-CLSWGAN	45.7	61.0	52.3	59.2	72.5	65.1	49.4	33.6	40.0	56.9	64.0	60.2
cycle-(U)WGAN	47.9	59.3	53.0	61.6	69.2	65.2	47.2	33.8	39.4	59.6	63.4	59.8

6 Discussion

Regarding the GZSL results in Table 4, we notice that there is a clear trend of all of our proposed feature generation methods (**cycle-WGAN**, **cycle-(U)WGAN**),

¹ <https://github.com/rfeligmg/frwgan-eccv18>

Table 5. ZSL results using per-class average top-1 accuracy on the test set of unseen classes \mathcal{Y}_U – all results shown in percentage. Results from previously proposed methods in the field extracted from [2].

Classifier	ZSL			
	CUB	FLO	SUN	AWA
DEWISE [17]	52.0	45.9	56.5	54.2
SJE [18]	53.9	53.4	53.7	65.6
LATEM [21]	49.3	40.4	55.3	55.1
ESZSL [19]	53.9	51.0	54.5	58.2
ALE [16]	54.9	48.5	58.1	59.9
baseline [1]	57.7	66.8	58.5	64.1
cycle-WGAN	57.8	68.6	59.7	65.6
cycle-CLSWGAN	58.4	70.1	60.0	66.3
cycle-(U)WGAN	58.6	70.3	59.9	66.8

Table 6. ZSL and GZSL using per-class average top-1 accuracy on the test sets of unseen classes \mathcal{Y}_U – all results shown in percentage.

Classifier	ZSL	GZSL
baseline [1]	7.5	0.7
cycle-WGAN	8.7	1.5

and **cycle-CLSWGAN**) to perform better than **baseline** on the unseen test set. In particular, it seems advantageous to use the synthetic samples from unseen classes to train the **cycle-(U)WGAN** model since it achieves the best top-1 accuracy results in 3 out of the 4 datasets, with improvements from 0.7% to more than 4%. In general, the top-1 accuracy improvement achieved by our approaches in the seen test set is less remarkable, which is expected given that we prioritize to improve the results for the unseen classes. Nevertheless, our approaches achieved improvements from 0.4% to more than 2.5% for the seen classes. Finally, the harmonic mean results also show that our approaches improve over the **baseline** in a range of between 1% and 2.2%. Notice that this results are remarkable considering the outstanding improvements achieved by f-CLSWGAN [1], represented here by **baseline**. In fact, our proposed methods produce the current state of the art GZSL results for these four datasets.

Analyzing the ZSL results in Table 5, we again notice that, similarly to the GZSL case, there is a clear advantage in using the synthetic samples from unseen classes to train the **cycle-(U)WGAN** model. For instance, top-1 accuracy results show that we can improve over the **baseline** from 0.9% to 3.5%. The results in this table show that our proposed approaches currently hold the best ZSL results for these datasets.

It is interesting to see that, compared to GZSL, the ZSL results from previous method in the literature are far more competitive, achieving results that are relatively close to ours and the **baseline**. This performance gap between ZSL and GZSL, shown by previous methods, enforces the argument in favor of using generative models to synthesize images from seen and unseen classes to train GZSL models [1, 9, 8]. As argued throughout this paper, the performance

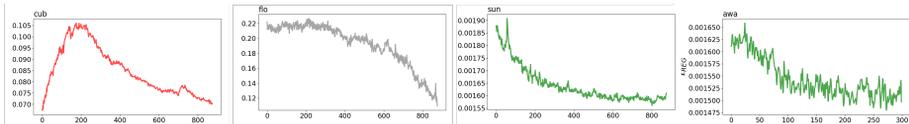


Fig. 3. Evolution of ℓ_{REG} in terms of the number of epochs for the 4 datasets.

produced by generative models can be improved further with methods that help the training of GANs, such as the cycle consistency loss [10].

In fact, the experiments clearly demonstrate the advantage of using our proposed multi-modal cycle consistency loss in training GANs for GZSL and ZSL. In particular, it is interesting to see that the use of synthetic examples of unseen classes generated by **cycle-(U)WGAN** to train the GZSL classifier provides remarkable improvements over the **baseline**, represented by f-CLSWGAN [1]. The only exception is with the SUN dataset, where the best result is achieved by **cycle-CLSWGAN**. We believe that **cycle-(U)WGAN** is not the top performer on SUN due to the number of classes and the proportion of seen/unseen classes in this dataset. For CUB, FLO and AWA we notice that there is roughly a (80%, 20%) ratio between seen and unseen classes. In contrast, SUN has a (91%, 9%) ratio between seen and unseen classes. We also notice a sharp increase in the number of classes from 50 to 817 – GAN models tend not to work well with such a large number of classes. Given the wide variety of GZSL datasets available in the field, with different number of classes and seen/unseen proportions, we believe that there is still lots of room for improvement for GZSL models.

Regarding the large-scale study on ImageNet, the results in Table 6 show that the top-1 accuracy classification results for **Baseline** and **cycle-WGAN** are quite low (similarly to the results observed in [1] for several ImageNet splits), but our proposed approach still shows more accurate ZSL and GZSL classification.

An important question about our approach is whether the regularisation succeeds in mapping the generated visual representations back to the semantic space. In order to answer this question, we show in Fig. 3 the evolution of the reconstruction loss ℓ_{REG} in (6) as a function of the number of epochs. In general, the reconstruction loss decreases steadily over training, showing that our model succeeds at such mapping. Another relevant question is if our proposed methods take more or less epochs to converge, compared to the **Baseline** – Fig. 4 shows the classification accuracy of the generated training samples from the seen classes for the proposed models **cycle-WGAN** and **cycle-CLSWGAN**, and also for the **baseline** (note that **cycle-(U)WGAN** is a fine-tuned model from the **cycle-WGAN**, so their loss functions are in fact identical for the seen classes shown in the graph). For three out of four datasets, our proposed **cycle-WGAN** converges faster. However, when the ℓ_{CLS} is included in (7) to form the loss in (8) (transforming **cycle-WGAN** into **cycle-CLSWGAN**), then the convergence of **cycle-CLSWGAN** is comparable to that of the **baseline**. Hence, **cycle-WGAN** tends to converge faster than the **baseline** and **cycle-CLSWGAN**.

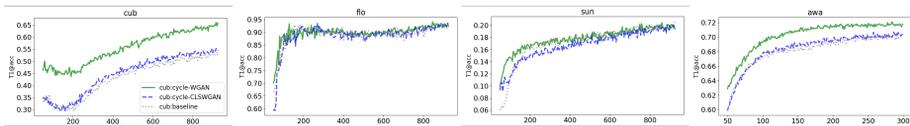


Fig. 4. Top-1 accuracy for the generated training samples from the seen classes.

7 Conclusions and Future Work

In this paper, we propose a new method to regularize the training of GANs in GZSL models. The main argument explored in the paper is that the use of GANs to generate seen and unseen synthetic examples for training GZSL models has shown clear advantages over previous approaches. However, the unconstrained nature of the generation of samples from unseen classes can produce models that may not work robustly for some unseen classes. Therefore, by constraining the generation of samples from unseen classes, we target to improve the GZSL classification accuracy. Our proposed constraint is motivated by the cycle consistency loss [10], where we enforce that the generated visual representations maps back to their original semantic feature – this represents the multi-modal cycle consistency loss. Experiments show that the use of such loss is clearly advantageous, providing improvements over the current state of the art f-CLS2WGAN [1] both in terms of GZSL and ZSL.

As noticed in Sec. 6, GAN-based GZSL approaches offer indisputable advantage over previously proposed methods. However, the reliance on GANs to generate samples from unseen classes is challenging because GANs are notoriously difficult to train, particularly in unconstrained and large scale problems. Therefore, future work in this field should be focused on targeting these problems. In this paper, we provide a solution that addresses the unconstrained problem, but it is clear that other regularization approaches could also be used. In addition, the use of GANs in large scale problems (regarding the number of classes) should also be more intensively studied, particularly when dealing with real-life datasets and scenarios. Therefore, we will focus our future research activities in solving these two issues in GZSL.

References

1. Xian, Y., Lorenz, T., Schiele, B., Akata, Z.: Feature generating networks for zero-shot learning. In: 31st IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018), Salt Lake City, UT, USA (2018)
2. Xian, Y., Schiele, B., Akata, Z.: Zero-shot learning – the Good, the Bad and the Ugly. In: 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, HI, USA, IEEE Computer Society (2017) 3077–3086
3. Zhang, Z., Saligrama, V.: Zero-shot learning via semantic similarity embedding. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 4166–4174
4. Lampert, C.H., Nickisch, H., Harmeling, S.: Attribute-based classification for zero-shot visual object categorization. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(3) (2014) 453–465

5. Qiao, R., Liu, L., Shen, C., van den Hengel, A.: Less is more: zero-shot learning from online textual documents with noise suppression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 2249–2257
6. Socher, R., Ganjoo, M., Manning, C.D., Ng, A.: Zero-shot learning through cross-modal transfer. In: Advances in neural information processing systems. (2013) 935–943
7. Yu, F.X., Cao, L., Feris, R.S., Smith, J.R., Chang, S.F.: Designing category-level attributes for discriminative visual recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2013) 771–778
8. Long, Y., Liu, L., Shen, F., Shao, L., Li, X.: Zero-shot learning using synthesised unseen visual data with diffusion regularisation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017)
9. Bucher, M., Herbin, S., Jurie, F.: Generating visual representations for zero-shot classification. In: International Conference on Computer Vision (ICCV) Workshops: TASK-CV: Transferring and Adapting Source Knowledge in Computer Vision. (2017)
10. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Computer Vision (ICCV), 2017 IEEE International Conference on. (2017)
11. Tran, T., Pham, T., Carneiro, G., Palmer, L., Reid, I.: A bayesian data augmentation approach for learning deep models. In: Advances in Neural Information Processing Systems. (2017) 2794–2803
12. Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., Perona, P.: Caltech-ucsd birds 200. (2010)
13. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on, IEEE (2008) 722–729
14. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE (2009) 1778–1785
15. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09. (2009)
16. Akata, Z., Perronnin, F., Harchaoui, Z., Schmid, C.: Label-embedding for image classification. *IEEE transactions on pattern analysis and machine intelligence* **38**(7) (2016) 1425–1438
17. Frome, A., Corrado, G.S., Shlens, J., Bengio, S., Dean, J., Mikolov, T., et al.: Devise: A deep visual-semantic embedding model. In: Advances in neural information processing systems. (2013) 2121–2129
18. Akata, Z., Reed, S., Walter, D., Lee, H., Schiele, B.: Evaluation of output embeddings for fine-grained image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 2927–2936
19. Romera-Paredes, B., Torr, P.: An embarrassingly simple approach to zero-shot learning. In: International Conference on Machine Learning. (2015) 2152–2161
20. Elyor Kodirov, T.X., Gong, S.: Semantic Autoencoder for Zero-shot Learning. *IEEE CVPR 2017* (2017)
21. Xian, Y., Akata, Z., Sharma, G., Nguyen, Q., Hein, M., Schiele, B.: Latent embeddings for zero-shot classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 69–77
22. Norouzi, M., Mikolov, T., Bengio, S., Singer, Y., Shlens, J., Frome, A., Corrado, G., Dean, J.: Zero-shot learning by convex combination of semantic embeddings. (2014)
23. Changpinyo, S., Chao, W.L., Gong, B., Sha, F.: Synthesized classifiers for zero-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 5327–5336

24. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
25. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. (2013) 3111–3119
26. Yan, X., Yang, J., Sohn, K., Lee, H.: Attribute2image: Conditional image generation from visual attributes. In: European Conference on Computer Vision, Springer (2016) 776–791
27. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan. arXiv preprint arXiv:1701.07875 (2017)
28. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. (June 2009) 951–958
29. Reed, S., Akata, Z., Lee, H., Schiele, B.: Learning deep representations of fine-grained visual descriptions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 49–58
30. Wang, P., Liu, L., Shen, C., Huang, Z., van den Hengel, A., Shen, H.T.: Multi-attention network for one shot learning. In: 2017 IEEE conference on computer vision and pattern recognition, CVPR. (2017) 22–25
31. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: Computer vision and pattern recognition (CVPR), 2010 IEEE conference on, IEEE (2010) 3485–3492
32. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: Proc. icml. Volume 30. (2013) 3
33. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10). (2010) 807–814
34. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: A system for large-scale machine learning. In: OSDI. Volume 16. (2016) 265–283