The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Adversarial Zero-Shot Learning with Semantic Augmentation

Bin Tong,[*] Martin Klinkigt,[*] Junwen Chen,[†] Xiankun Cui,[†]
Quan Kong, Tomokazu Murakami, Yoshiyuki Kobayashi
R&D Group, Hitachi, Japan
{bin.tong.hh, martin.klinkigt.ut, quan.kong.xz}@hitachi.com
{tomokazu.murakami.xr, yoshiyuki.kobayashi.gp}@hitachi.com

Abstract

In situations in which labels are expensive or difficult to obtain, deep neural networks for object recognition often suffer to achieve fair performance. Zero-shot learning is dedicated to this problem. It aims to recognize objects of *unseen* classes by transferring knowledge from *seen* classes via a shared intermediate representation. Using the manifold structure of *seen* training samples is widely regarded as important to learn a robust mapping between samples and the intermediate representation, which is crucial for transferring the knowledge. However, their irregular structures, such as the lack in variation of samples for certain classes and highly overlapping clusters of different classes, may result in an inappropriate mapping. Additionally, in a high dimensional mapping space, the hubness problem may arise, in which one of the *unseen* classes has a high possibility to be assigned to samples of different classes. To mitigate such problems, we use a generative adversarial network to synthesize samples with specified semantics to cover a higher diversity of given classes and interpolated semantics of pairs of classes. We propose a simple yet effective method for applying the augmented semantics to the hinge loss functions to learn a robust mapping. The proposed method was extensively evaluated on small- and large-scale datasets, showing a significant improvement over state-of-the-art methods.

Introduction

The significant performance improvement of deep neural networks in recent years is in part due to the wide availability of large labeled datasets. However, objects in the wild follow a long-tailed distribution. For some uncommon objects, only a limited number of samples can be provided, and new categories of objects may even emerge dynamically. In such cases, many state-of-the-art methods fail to deliver the same high performance as when trained with sufficiently large labelled datasets. This challenge motivates a different learning paradigm in which the number of labelled training samples is limited or the number of new classes to be recognized increases. One such candidate is zero-shot learning (Lampert, Nickisch, and Harmeling 2009).

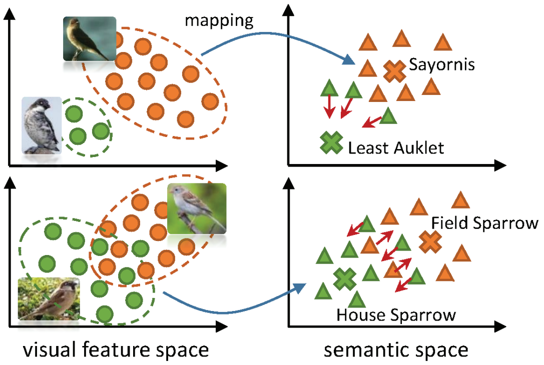


Figure 1: Visualization of mappings from visual feature space to semantic space. On top row, for ‘least auklet’, only a limited number of images are available compared with the other class ‘sayornis’, making embedding of this class difficult. Its embeddings tend to move away from its semantic representation. The bottom row shows two classes that have overlapping clusters of their visual representation with a similar scale. This overlap may remain in semantic space, which degrades recognition performance. Note that red arrows show the expected direction for embeddings.

Zero-shot learning aims to recognize novel classes for which no training samples were provided. In other words, the test and training class sets are disjoint. Such a recognition task can be addressed by using an intermediate semantic representation that is shared between *seen* and *unseen* classes, and knowledge can be transferred from *seen* classes to *unseen* classes. In (Lampert, Nickisch, and Harmeling 2009), so-called attributes were used as the intermediate representation. For *seen* and *unseen* classes, a binary vector expressing their attributes is assigned. Given an example of such attributes, we can look at animals like cows and horses. Both share common attributes, such as ‘long legs’ and ‘long tails’, but also have different attributes, such as ‘horns” for cows (which horses do not have). However, such attributes are difficult to obtain and require expert knowledge about the target domain to design. In (Frome et al. 2013), a so-called word embedding (Bengio et al. 2006) was introduced as a drop-in replacement for the intermediate representation, which can currently be efficiently trained on a large text cor-

[*] contributed equally to this paper.
[†] contributed equally during their internship at Hitachi.

pus (Mikolov et al. 2013).

One of the most frequently used approaches to transfer the knowledge is to embed features of *seen* classes and their corresponding intermediate representations into the same semantic space under the condition that the two are located close to each other. In an ideal case, the *unseen* classes with their intermediate representations and samples are embedded in the same semantic space close to each other. Therefore, the closest embedding of a class to a test sample is returned as the result of a nearest-neighbor search. Learning such an ideal embedding is crucial for zero-shot learning.

Using the inherent structure of *seen* samples has been widely regarded as helpful for boosting recognition performance (Changpinyo et al. 2016), (Changpinyo, Chao, and Sha 2017). However, the existing methods implicitly ignore more irregular structures for the distribution of *seen* classes explained in Figure 1. In these situations, learning the mapping based on either regression (Shigeto et al. 2015) or hinge loss function (Frome et al. 2013) will degrade recognition performance. In addition, it has been pointed out in (Lazaridou, Dinu, and Baroni 2015) (Li, Tao, and Shaogang 2017) (Shigeto et al. 2015) that, in a high dimensional semantic space, a few unseen classes will always become the nearest neighbor of many feature embeddings of test samples, which is called the hubness problem (Radovanović, Nanopoulos, and Ivanović 2010).

To mitigate the above two problems, we integrate a generative adversarial network (GAN) (Goodfellow et al. 2014) into the framework of zero-shot learning. By using GAN, we aim to generate two types of samples. First, we generate samples of given classes, which are called *semantically same* samples, to cover more variations, increasing their diversity. Second, we synthesize *semantically compound* samples from more than one class by interpolating the intermediate representations of known classes. It is argued in (Lazaridou, Dinu, and Baroni 2015) that a mapping function with a max-margin ranking loss can significantly mitigate the hubness problem. For this reason, we integrate the relations among semantics into hinge loss functions to learn a robust mapping. Our contributions are two folds:

1. To the best of our knowledge, this is the first study that uses GAN in the field of zero-shot learning. This integration is simple, effective, and easy to implement. With GAN, variations of samples belonging to the same class and *semantically compound* samples belonging to more than one class are synthesized. The synthesized samples are used with hinge loss functions for learning a robust mapping to the semantic space.
2. With extensive evaluation on small- and large-scale datasets, we confirm a moderate improvement over other state-of-the-art methods in the task of object recognition by 2% to 4%, but a significant improvement for the task of image retrieval by 10% to over 30%.

Related Work

Zero-shot learning has become a popular research topic in computer vision and machine learning. Many studies in this field inherently have a two-step process. In the first step, an embedding function to map the intermediate representations and visual features into the same semantic space is learned. There are three different kinds of methods of learning the embedding function. The first one maps visual features onto the space of intermediate representations (Frome et al. 2013); the second one maps the intermediate representations onto the space of visual features (Shigeto et al. 2015) (Li, Tao, and Shaogang 2017); the last one maps both the visual features and intermediate representations into a common semantic space (Yang and Hospedales 2015). In the second step, a nearest neighbor search in the mapped space is carried out to predict the class label. Some studies have only a one-step process, in which the embedding function and prediction are jointly learned in a unified framework (Changpinyo et al. 2016), (Akata et al. 2013), (Romera-Paredes and Torr 2015). The proposed method belongs to the one that has the two-step process.

As intermediate representation, attribute vectors and word embeddings of class labels serve as two popular sources of side information in zero-shot learning. Recently, textual description of an image category (Elhoseiny, Saleh, and Elgammal 2013)(Ba et al. 2015)(Li, Tao, and Shaogang 2017) and gaze information (Karessli et al. 2017) have been used as side information. The proposed method only uses attribute vectors and word embeddings.

According to how the test data is used, zero-shot learning is categorized into two types of methods, which are inductive and transductive zero-shot learning. Inductive methods process *unseen* samples sequentially. In contrast, transductive methods (Fu et al. 2015a) (Zhang and Saligrama 2016) often use the manifold structure of all unseen samples. In fact, it is known that the distribution difference in visual appearance for a given attribute between seen and unseen samples exists, which may lead to the domain shift problem. Using the manifold structure of unseen samples can reduce the distribution difference. However, such methods have not received much attention, as the distribution of unseen samples is required. New unseen classes cannot be added dynamically, limiting their use in practice. The proposed method belongs to inductive zero-shot learning.

GAN is a neural network model trained in an unsupervised manner, aiming to generate new data with the same distribution as the data of interest. It is widely applied in computer vision and natural language processing tasks, such as generating samples of images (Denton et al. 2015) and generating sequential words (Li et al. 2017). Recently, in (Akshay Mehrotra 2017), a GAN model was introduced into residual pairwise networks for one shot learning. This framework cannot be directly applied to zero-shot learning, as at least one sample of a given target class is required to train the network. The idea of adversarial training was used in (Wang Xiaolong 2017) to make object detection more robust by learning occlusion and deformation.

Preliminary on GAN

In this section, the basics of GAN are introduced. A GAN model consists of a generator G and discriminator D that compete in a turn-wise min-max game. The discriminator attempts to distinguish real training data from synthetic data, and the generator attempts to fool the discriminator by generating synthetic data that looks like real data. The D and G play the following game on $V(D,G)$

$$\min_G \max_D V(D,G) = \mathbb{E}_{x \in p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \in p_z(z)}[\log(1 - D(G(z)))], \tag{1}$$

where x represents a sample. p_{data} and p_z represent the distribution of real samples and synthetic samples, and z represents a noise vector.

In the original GAN model, only z is used to generate samples. In a variation called conditional GAN (CGAN), a condition y, which is often a class label, is included in addition to z to control the sample generation. The objective function becomes

$$\min_G \max_D V(D,G) = \mathbb{E}_{x \in p_{data}(x)}[\log D(x|y)] + \mathbb{E}_{z \in p_z(z)}[\log(1 - D(G(z|y)))], \tag{2}$$

where y could be a one-hot representation of the class label. During training of the CGAN model, y is used to instruct the generator G to synthesize samples for this given class.

Proposed Method

We denote the training data as $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\mathbf{x}_i \in \mathcal{R}^d$ and y_i represent a sample and its class label, respectively. N is the number of samples. In the context of zero-shot learning, y_i is often used for the intermediate representation, which is typically a per-class attribute vector or word embedding. Both are dense vectors with continuous values. The proposed method is built up on two components: discriminative embedding and training with augmented semantics. Given two samples with different class labels, the discriminative embedding is trained to differentiate the feature representations of those samples. During the training with a GAN model, new samples that do not exist in the training data are synthesized, making the embedding more robust to recognizing unseen classes.

Discriminative Embedding

It has been stated in (Lazaridou, Dinu, and Baroni 2015) that the max-margin loss function can mitigate the hubness problem that may occur for training embeddings with regressive loss functions. Inspired by (Frome et al. 2013), we use the hinge-loss based objective function for the discriminative embedding. Its purpose is to make the similarity within samples and their class labels larger than that between those samples and the other class labels. The objective function can be written as

$$L = \sum_{i \neq j} \max[0, m - s(\mathbf{x}_i, y_i) + s(\mathbf{x}_i, y_j)], \tag{3}$$

where $s(\cdot,\cdot)$ measures the similarity between samples and class labels based on embedding functions, and m is a margin. To reduce the computational cost, we only randomly select K different class labels from y_i. In the later section, we introduce how the similarity is calculated.

Training with a CGAN model

This subsection focuses on the integration of a CGAN model into the training of the discriminative embedding. We generate two types of samples to make a robust mapping: *semantically same* samples of one class and *semantically compound* samples of different classes. There are different motivations behind the generation of those two types of samples. First, some classes may not have sufficient samples to cover all their visual variations. Generating *semantically same* but visually different samples provides a higher degree of diversity to those classes. Second, samples belonging to two different class labels may overlap in the feature space. As indicated in Figure 1, with insufficient mapping, this overlap will also occur in the embedding space. Generating *semantically compound* samples is helpful for discriminating such overlapping classes.

To control the semantics of synthesized samples, we use the CGAN model, whose generator is denoted as $G(z|y_i)$, where y_i is the class label of sample $\mathbf{x}_i$. Mathematically, a synthesized sample that has the same semantic as a given class label y_i is denoted as $\widehat{\mathbf{x}}_i$. It is expected to come from the same distribution as $\mathbf{x}_i$. The *semantically compound* sample is denoted as $\widehat{\mathbf{x}}_{ij}$. It has a semantic derived from the interpolation of two different class labels, such as $y_{ij} = \alpha y_i + (1-\alpha) y_j$ where $\alpha \in (0,1)$. The interpolated class label y_{ij} is used as the input of the generator, such as $G(z|y_{ij})$.

We have the following two loss functions for $\widehat{\mathbf{x}}_i$.

$$L_{g_1} = \sum_{i \neq j} \max[0, m_g - s(\widehat{\mathbf{x}}_i, y_i) + s(\widehat{\mathbf{x}}_i, y_j)] \tag{4}$$

$$L_{g_2} = ||s(\mathbf{x}_i, y_i) - s(\widehat{\mathbf{x}}_i, y_i)||^2, \tag{5}$$

where m_g is a margin. Equation 4 requires that the mapping of $\widehat{\mathbf{x}}_i$ is closer to that of y_i than y_j, as $\widehat{\mathbf{x}}_i$ is semantically the same as y_i. Equation 5 constrains $\widehat{\mathbf{x}}_i$ to be semantically similar to $\mathbf{x}_i$. We also have the following two loss functions for $\widehat{\mathbf{x}}_{ij}$:

$$L_{m_1} = \sum_{i \neq j} \max[0, m_c - s(\mathbf{x}_i, y_i) + s(\widehat{\mathbf{x}}_{ij}, y_i)] \tag{6}$$

$$L_{m_2} = \sum_{i \neq j} \max[0, m_c - s(\widehat{\mathbf{x}}_{ij}, y_i) + s(\mathbf{x}_i, y_j)], \tag{7}$$

where m_c is a margin. Equation 6 makes the similarity between $\widehat{\mathbf{x}}_{ij}$ and y_i smaller than that between $\mathbf{x}_i$ and y_i, as $\widehat{\mathbf{x}}_{ij}$ is the sample derived from the interpolated semantics of y_i and y_j. Equation 7 makes the similarity between $\widehat{\mathbf{x}}_{ij}$ and y_i larger than that between $\mathbf{x}_i$ and y_j. Since $\widehat{\mathbf{x}}_{ij}$ is generated from the interpolated semantics of y_i and y_j, it is natural to have it semantically being between $\mathbf{x}_i$ and $\mathbf{x}_j$. The *between* relation serves as a criterion to separate the overlapped samples from two different class labels.

Similarity on Embedding Space

There are various approaches for calculating the similarity between a sample and its class label. In any cases, mapping heterogeneous objects into the same space is mandatory. Inspired by (Li, Tao, and Shaogang 2017), we map the visual features of samples and their intermediate representations into a common space. In the experiment section, we argue that embedding into the common space is more likely to result in better recognition performance than other ways of embedding, such as mapping into the space of intermediate representation directly.

Two individual neural networks are trained to map the visual features of samples and their class labels into a common space. The similarity is calculated as the dot product of their embeddings in the common space. The similarity between $\mathbf{x}_i$ and y_i can be obtained by:

$$s(\mathbf{x}_i, y_i) = f_{\Theta_i}(\mathbf{x}_i)^T \cdot f_{\Theta_c}(y_i), \tag{8}$$

where $f_{\Theta_i}(\cdot)$ represents the mapping from the visual features into the common space, and $f_{\Theta_c}(\cdot)$ represents the mapping from class labels into the common space. We use $\Theta = \{\Theta_i, \Theta_c\}$ as short notation for the mapping parameters to learn, where Θ_i and Θ_c represent the parameters for mapping the visual features and class labels, respectively.

Learning Algorithm

In the CGAN model, both generator and discriminator are designed as fully connected networks. The generator receives the intermediate representation of a given class as well as a randomized noise vector and outputs a synthesized sample. The discriminator receives both true and synthesized samples of the given class, and outputs probabilities of being determined as true samples. The generator and discriminator are trained with the intermediate representations of class labels. When generating *semantically same* samples, the generator is required to feed the intermediate representation of a class label. When generating *semantically compound* samples, the generator is required to feed an interpolated semantic representation of two different class labels.

Generating good-quality samples is crucial for the optimization of loss functions, such as Equations 4-7. Thus, we pre-train the CGAN model before optimizing the hinge loss functions to mitigate this cold start problem. In the simplest way, we can train all loss functions together by combining Equations 3-7 with some balancing parameters. However, we observed through experiments that training such a combined objective function often results in inappropriate recognition performance. Fortunately, the optimization of our method can be decomposed into three independent threads: learning the discriminative embedding using Equation 3, learning with *semantically same* samples, and learning with *semantically compound* samples. To decouple the multiple loss functions, the CGAN model is optimized independently from the hinge loss functions due to the fact that their parameters are disjoint. We denote Ψ as the parameters of the CGAN model. The notations Ψ_g and Ψ_d represent the parameters of the generator and discriminator, respectively. We name our method GANZrl, and its learning process is shown in Algorithm 1.

Algorithm 1 GANZrl

```
Input: training data $\mathbf{x}_i$ and its label $y_i$ where $i \in [1, \dots, N]$
Output: $\Theta$
Initialize $\alpha$, $\Theta$, $\Psi_d$ and $\Psi_g$.
Pretrain the CGAN model.
repeat
    $\Psi_d \leftarrow \bigtriangledown_{\Psi_d}\{\mathbb{E}_{\mathbf{x}_i \in p_{data}(\mathbf{x}_i)}[\log D(\mathbf{x}_i|y_i)]$
        $+\mathbb{E}_{z \in p_z(z)}[\log(1 - D(G(z|y_i)))]\}$
    $\Psi_g \leftarrow \bigtriangledown_{\Psi_g}\mathbb{E}_{z \in p_z(z)}[\log(1 - D(G(z|y_i)))]$
    $\widehat{\mathbf{x}}_i \leftarrow G(z|y_i)$
    $\widehat{\mathbf{x}}_{ij} \leftarrow G(z|(\alpha y_i + (1-\alpha) y_j))$
    $\Theta \leftarrow \bigtriangledown_\Theta L$
    $\Theta \leftarrow \bigtriangledown_\Theta (L_{g_1} + L_{g_2})$
    $\Theta \leftarrow \bigtriangledown_\Theta (L_{m_1} + L_{m_2})$
until
```

Experiments

We conducted experiments on various datasets to verify the effectiveness of GANZrl. First, we introduce the datasets used for evaluation. We then discuss two tasks of the experiments, object recognition and image retrieval. Finally, we give in-depth analysis to gain more insight into GANZrl.

In the experiments, we used three small-scale and two large-scale benchmark datasets. The small-scale datasets were Animals with Attribute (AwA), CUB-200-2011 (CUB) and SUN with Attribute (SUN). The large-scale datasets were ILSRC2010 (ImageNet-1) and ILSVRC2012/ILSVRC2010 (ImageNet-2). In ImageNet-2, the 1000 classes of ILSVRC2012 were used as seen classes, and 360 classes of ILSVRC2010, which are not included in ILSVRC2012, as unseen classes. The details of the datasets are given in Table 1.

Table 1: Statistics on benchmark datasets. IR refers to type of intermediate representation, IR-D represents dimension of this semantic space, and A and W represent attribute vector and word embedding, respectively.

dataset	#instances	IR	IR-D	#seen	#unseen
AwA	30475	A	85	40	10
CUB	11788	A	312	150	50
SUN	14340	A	102	707	10
ImageNet-1	1.28×10^6	W	500	800	200
ImageNet-2	1.69×10^6	W	500	1000	360

In the experiments, the intermediate representations of the attribute vector and word embedding are treated as continuous and dense vectors. The per-class attribute vector is calculated as the average of the binary per-image attribute vectors of a given class. The word embedding was trained on a corpus of 4.6M Wikipedia documents by using the skip-gram word2vec model. For AwA, we used VGG-19 features (Simonyan and Zisserman 2014) provided by the official site. For the other datasets, we used a variant of GoogLeNet features, called Inception-ResNet (Szegedy et al. 2017). For the mapping from visual features and intermediate representations to the common space, we empirically found that one

Table 2: The comparison between GANZrl and the state-of-the-art methods. IR refers to the type of intermediate representation utilized. References: DeViSE (Frome et al. 2013), ConSE (Mohammad et al. 2014), SSE (Zhang and Ziming 2015), JLSE (Zhang and Ziming 2016), SECML (Maxime, Stéphane, and Frédéric 2016), VIL (Fu and Sigal 2016), DEM (Li, Tao, and Shaogang 2017), UVDS (Long et al. 2017), SAE (Kodirov, Xiang, and Gong 2017), RRZSL (Shigeto et al. 2015), ESZSL (Romera-Paredes and Torr 2015), AMP (Fu et al. 2015b), SS-VOC (Fu et al. 2015a), PDDM (Huang, Loy, and Tang 2016). Note that results of DeViSE and ConSE are reimplemented by DEM and SS-VOC.

small-scale dataset					large-scale dataset			
Method	IR	AwA	CUB	SUN	Method	IR	ImageNet-1	ImageNet-2
DeViSE	A	50.4	33.5	-	DeViSE	W	31.8	12.8
SAE	A	84.7	61.4	91.5	SAE	W	46.1	27.2
DEM	A	78.7	59.0	-	DEM	W	60.7	25.7
SSE	A	76.3	30.4	82.5	ConSE	W	28.5	15.5
RRZSL	A	80.4	52.4	84.5	VIL	W	16.8	-
JLSE	A	80.5	42.1	83.9	AMP	W	41.0	-
ESZSL	A	75.3	48.7	82.1	SS-VOC	W	-	16.8
SECML	A	77.3	43.3	84.4	PDDM	W	48.2	-
UVDS	A	82.1	45.7	86.5				
GANZrl-IR	A	82.97 ± 1.05	55.32 ± 0.26	90.31 ± 0.80	GANZrl-IR	W	52.40 ± 0.97	29.36 ± 0.44
GANZrl-SS	A	84.38 ± 0.53	58.89 ± 1.05	91.56 ± 0.58	GANZrl-SS	W	56.13 ± 0.94	30.02 ± 0.37
GANZrl-SC	A	82.76 ± 0.98	61.04 ± 0.49	91.41 ± 1.21	GANZrl-SC	W	**61.10** ± 0.68	**30.80** ± 0.49
GANZrl	A	**86.23** ± 0.44	**62.56** ± 0.30	**93.59** ± 0.58	GANZrl	W	54.95± 1.11	29.58 ± 0.31

layer of fully connected neurons with an activation function, followed by a batch normalization (Ioffe and Szegedy 2015) tends to result in higher classification performance. The design of the CGAN model is described as follows. In the generator, the noise prior and intermediate representation are independently fed into a fully connected layer with an activation function. These two layers' individual outputs are concatenated and mapped back to the visual feature space by a different fully connected layer with an activation function. In the discriminator, the visual feature and intermediate representation are first concatenated and fed into two layers of fully connected neurons followed by activation functions.

The setting of parameters shared by all datasets is as follows. RMSprop was used as the gradient descent algorithm for training the CGAN model with an initial learning rate of 10^{-4}. Adam was used for learning the mapping, and its initial learning rate was set to 10^{-4} and 5×10^{-5}. The activation functions were chosen from sigmoid, tanh, and leakyrelu. The batch size was set to 64. The margins m and m_g of Equation 3 and Equation 4 were set to 0.1 or 0.2. The margin m_c of Equations 6 and 7 was set to 1 or 2 times the margin of m. The dimension of the noise vector was set to 30, and the dimension of the common space was set to 1024 or 2048. We empirically found that above parameters tend to result in the highest classification performance.

In the object recognition task, flat hit@k classification accuracy is reported. For hit@k, a test image is successfully classified if the correct label is among the top k labels returned. In the small-scale datasets, k was set to 1; in the large-scale datasets, k was set to 5. In the task of image retrieval, the mean average precision (MAP) is used. The larger the MAP is, the better the ranking performance. We only compare our results to other inductive zero-shot methods, since GANZrl belongs to this category.

Object Recognition

Table 2 shows the classification performance of GANZrl compared to those of the state-of-the-art methods on the small- and large-scale datasets. The average accuracies with standard deviations are reported from 5 independent runs. Attributes (A) and word embeddings (W) were used for the small- and large-scale datasets, respectively. We also provide results for some variants of GANZrl. For the variant called GANZrl-IR, visual features are mapped to the space of *intermediate representation* (IR), where the similarity is measured. The variant called GANZrl-SS optimizes the loss functions of Equation 3 and the combination of Equations 4 and 5, meaning that only *semantically same* visual features generated by the CGAN model are used. The variant called GANZrl-SC optimizes the loss functions of Equation 3 and the combination of Equations 6 and 7, meaning that only *semantically compound* visual features generated by the CGAN model are used.

We observed that, for all the small- and large-scale datasets, GANZrl and its variants were able to achieve the best performance. Particularly, GANZrl-SC even outperformed SAE by 3.6% on the ImageNet-2 dataset. We can also see that, on the small-scale datasets, GANZrl-SS and GANZrl-SC were inferior to GANZrl. However, on the large-scale datasets, GANZrl-SC outperformed GANZrl-SS and GANZrl. The reason might be that the number of samples for each given class in the small-scale datasets was much smaller than that for the large-scale datasets. On large-scale datasets, the synthesized samples of a class did not necessarily cover more visual variation as covered by the large number of samples given for that class. In such a situation, *semantically compound* samples become more important for discriminating two classes.

We also noticed that GANZrl-IR was inferior to GANZrl on all datasets. It is in part due to the fact that the space of intermediate representation is inferior in differentiating visual information compared to a well-trained common space. We empirically verified that mapping intermediate representations and visual features into a common space would be the preferable choice. We also empirically found that using word embeddings or the combination of attribute vectors and word embeddings does not improve classification performance in the small-scale datasets. It seems like human-designed attributes are semantically more discriminative than word embeddings trained in an unsupervised manner.

Image Retrieval

For the image retrieval task, images having a certain class label that we are searching for are returned. Given a class label together with its attribute vector, the nearest five images close to this attribute vector in the common space are returned. In Table 3, we can see that GANZrl significantly outperformed the state-of-the-art methods. With 31.62%, the improvement on CUB was the most significant. This is in part due to the fact that, together with the generation of visual examples, a very robust mapping of the visual features and class labels into a common space is achieved. As a result, GANZrl can catch the small changes in the details of the birds better than JLSE.

Table 3: Comparison on MAP. SSE (Zhang and Ziming 2015), JLSE (Zhang and Ziming 2016), SECML (Maxime, Stéphane, and Frédéric 2016)

Method	AwA	CUB	SUN
SSE-ReLU	42.60	3.70	44.55
SSE-INT	46.25	4.69	58.94
JLSE	67.66	29.15	80.01
SECML	68.10	25.33	52.68
GANZrl	**75.63**	**60.77**	**90.12**

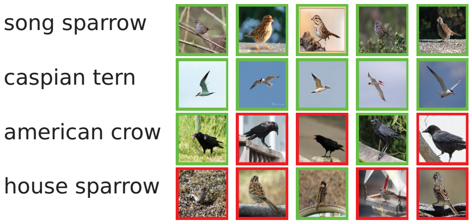


Figure 2: Image retrieval on CUB dataset

Besides measuring MAP, we also visualized the ranking results on CUB and SUN, as shown in Figures 2 and 3, respectively. We cannot visualize the ranking results for the AwA dataset, as no images are provided. For each dataset, two good cases and two bad cases are shown. The green boader indicates correct retrievals belonging to the target class, while the red boader indicates incorrect retrievals not belonging to the target class. We can see that, on both datasets, GANZrl was able to achieve reasonable ranking results. More specifically, in SUN, for 80% of the test classes, all images in the top 5 are correct. Even for the worst 10% of the classes, at least three correct images were within the top 5 ranking. In CUB, for 52% of the test classes, all images in the top 5 were correct, while even for the worst 10% of the classes, at least one correct image was within the top 5 ranking.

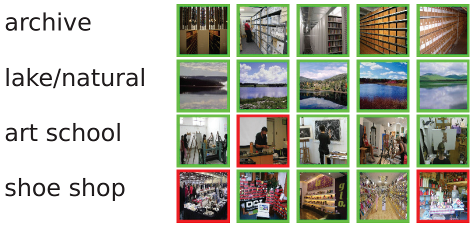
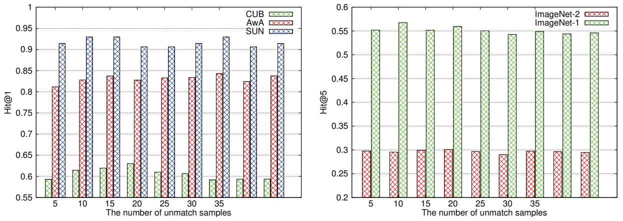


Figure 3: Image retrieval on SUN dataset

Analysis and Discussion

To gain an insight into GANZrl, we examined how it performs in different parameter settings. First, we examined how the number of unmatched samples affects classification performance. The number of unmatched samples represents K different class labels, which is used in Equations 3, 5, 6, and 7. The K was tuned from the set $\{5, 10, 15, 20, 25, 30\}$. Figure 4 shows the performance changes for small- and large-scale datasets. We can see that the number of unmatched samples did not significantly affect performance. Even for small numbers of unmatched samples, reasonable performance was achieved. The performance of GANZrl in the large-scale datasets was more stable than in the small-scale datasets. As shown in Figure 4(a), GANZrl tended to perform best in the SUN data set when the number of unmatched samples was set to 15.

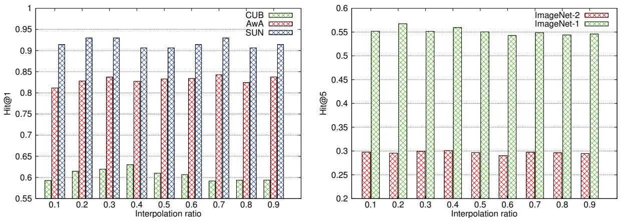
Second, we examined how α values affect classification performance. The α determines the ratio of two different semantics to be combined into one semantic. The α was tuned from the set $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. Figures 5(a) and 5(b) show the change in performance for small- and large-scale datasets when the ratio of interpolation for two different class labels changes. Similar to the number of



(a) small-scale

(b) large-scale

Figure 4: Classification performance when number of unmatched samples changes



(a) small-scale (b) large-scale

Figure 5: Classification performance when ratio of interpolation changes

unmatched samples, the ratio did not significantly affect performance. Again, the performance of the large-scale datasets was more stable than that of the small-scale datasets. As shown in Figure 5(a), GANZrl was likely to perform best for AwA and SUN datasets when the interpolation ratios were set to 0.3 and 0.7. For CUB, setting the interpolation ratio to 0.4 tended to provide the best performance. With the above observations, GANZrl shows its robustness, which makes it easy to apply in different datasets and parameter settings.

To verify the effectiveness of the synthesized samples generated with the CGAN model, we visualized the embeddings of the visual features and class labels in the common semantic space. To ease visualization, we chose AwA and CUB, which have less than 200 class labels in the training data. Note that class labels in both AwA and CUB datasets are represented by their attribute vectors. Figure 6 shows the embeddings of class labels and synthesized visual features in a 2-dimensional space using t-SNE (van der Maaten and Hinton 2008). Take Figure 6(a) as an example. The class labels 'grizzly+bear' and 'horse' are denoted as '2' and '7', respectively. We can see that the blue triangles, which are synthesized samples of class label '7', were located near to the embedding of class label '7'. The blue squares, which are synthesized samples of class label '2', were located near the embedding of the class label '2'. The *semantically compound* visual features generated by the interpolated semantics from class labels '2' and '7' are denoted with green circles. As expected, those compound samples were located between the embeddings of class labels '2' and '7'.

For AwA and SUN, we visualized the confusion matrix for the test data with 10 class labels, as shown in Figure 7. Due to limited space, we do not show the confusion matrix for CUB as it has more than 10 test classes. The numbers associated with the confusion matrix of AwA represent these classes: 'chimpanzee', 'giant+panda', 'leopard', 'persian+cat', 'pig', 'hippopotamus', 'humpback+whale', 'raccoon', 'rat', and 'seal'. The numbers associated with the confusion matrix of SUN represent these classes: 'inn_indoor', 'flea_market_indoor', 'lab_classroom', 'outhouse_outdoor', 'chemical_plant', 'mineshaft', 'lake_natural', 'shoe_shop', 'art_school', and 'archive'. As shown in Figure 7(a), the classification accuracy of the class 'humpback+whale' was relatively low. Its samples were often missclassified as 'chimpanzee' or 'seal'. Worth noting is that (Lampert, Nickisch, and Harmeling 2009) with their attribute prediction method performed best on this class due to the unique attribute combination of the *unseen* class 'humpback+whale' to the two *seen* classes 'blue+whale' and 'killer+whale'. In the attribute vector space, all those whales form a unique and dense cluster easily separable from the other animals. However, in our case, a synthesized sample derived from a whale and any other animal disturbs this separation. This makes the sample move towards the attribute vector of 'chimpanzee', which is the nearest neighbour to the mean of all attribute vectors of *seen* classes. This disturbance of synthetic compound samples can also be identified as one of the reasons for the low classification accuracy of the class 'outhouse_outdoor' in SUN, as shown in Figure 7(b).

Conclusion

We proposed a method for zero-shot learning that mitigates the problem of inappropriately handling irregular manifold structures of *seen* classes and the hubness problem. We synthesized samples with specified semantics through a GAN model. These synthetic samples increase the visual diversity of a given class as well as the compound semantics from two different classes. These samples are used to learn a robust semantic mapping by applying them to hinge loss functions. To the best of our knowledge, this is the first work that integrates the CGAN model into zero-shot learning. The extensive experiments on small- and large-scale datasets show significant improvements over the state-of-the-art methods in tasks of object recognition and image retrieval.

Acknowledgments.

We would like to acknowledge Yanwei Fu, Takuma Shibahara and Katsuyuki Nakamura for helpful discussions.

References

Akata, Z.; Perronnin, F.; Harchaoui, Z.; and Schmid, C. 2013. Label-embedding for attribute-based classification. In *CVPR*, 819–826.

Akshay Mehrotra, A. D. 2017. Generative adversarial residual pairwise networks for one shot learning. In *arXiv:1703.08033*.

Ba, J. L.; Swersky, K.; Fidler, S.; and Salakhutdinov, R. 2015. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *ICCV*, 4247–4255.

Bengio, Y.; Schwenk, H.; Senécal, J.-S.; Morin, F.; and Gauvain, J.-L. 2006. *Neural Probabilistic Language Models*. Berlin, Heidelberg: Springer Berlin Heidelberg. 137–186.

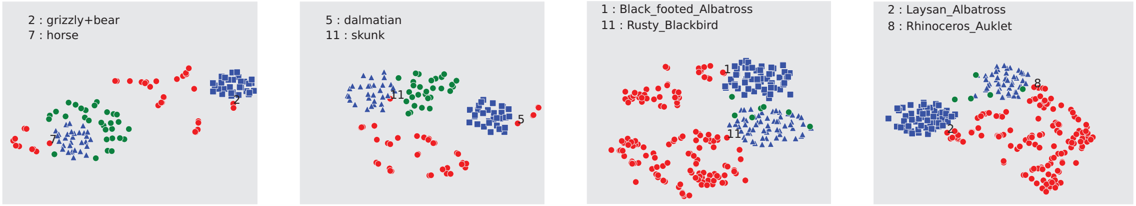
Changpinyo, S.; Chao, W.; Gong, B.; and Sha, F. 2016. Synthesized classifiers for zero-shot learning. In *CVPR*, 5327–5336.

Changpinyo, S.; Chao, W.; and Sha, F. 2017. Predicting visual exemplars of unseen classes for zero-shot learning. In *ICCV*.

Denton, E.; Chintala, S.; Szlam, A.; and Fergus, R. 2015. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*, 1486–1494.

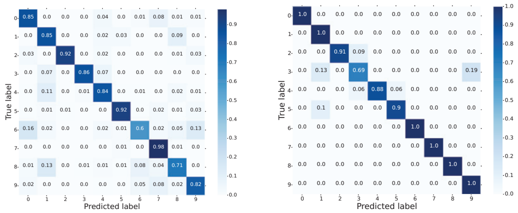
Elhoseiny, M.; Saleh, B.; and Elgammal, A. M. 2013. Write a classifier: Zero-shot learning using purely textual descriptions. In *ICCV*, 2584–2591.

Frome, A.; Corrado, G. S.; Shlens, J.; Bengio, S.; Dean, J.; Ranzato, M. A.; and Mikolov, T. 2013. Devise: A deep visual-semantic embedding model. In *NIPS*, 2121–2129.



(a) AwA-1 (b) AwA-2 (c) CUB-1 (d) CUB-2

Figure 6: Visualization of embeddings of class labels and synthesized image features.



(a) AwA (b) SUN

Figure 7: Confusion matrix of AwA and SUN

Fu, Y., and Sigal, L. 2016. Semi-supervised vocabulary-informed learning. In *CVPR*, 5337–5346.

Fu, Y.; Hospedales, T. M.; Xiang, T.; and Gong, S. 2015a. Transductive multi-view zero-shot learning. *IEEE Trans. PAMI* 37(11):2332–2345.

Fu, Z.; Xiang, T.; Kodirov, E.; and Gong, S. 2015b. Zero-shot object recognition by semantic manifold distance. In *CVPR*, 2635–2644.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NIPS*, 2672–2680.

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

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 448–456.

Karessli, N.; Akata, Z.; Schiele, B.; and Bulling, A. 2017. Gaze embeddings for zero-shot image classification. In *CVPR*, 4525–4534.

Kodirov, E.; Xiang, T.; and Gong, S. 2017. Semantic autoencoder for zero-shot learning. In *CVPR*, 3174–3183.

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

Lazaridou, A.; Dinu, G.; and Baroni, M. 2015. Hubness and pollution: Delving into cross-space mapping for zero-shot learning. In *ACL*, 270–280.

Li, J.; Monroe, W.; Shi, T.; Ritter, A.; and Jurafsky, D. 2017. An embarrassingly simple approach to zero-shot learning. In *EMNLP*.

Li, Z.; Tao, X.; and Shaogang, G. 2017. Learning a deep embedding model for zero-shot learning. In *CVPR*, 2021–2030.

Long, Y.; Liu, L.; Shao, L.; Shen, F.; Ding, G.; and Han, J. 2017. From zero-shot learning to conventional supervised classification: Unseen visual data synthesis. In *CVPR*, 1627–1636.

Maxime, B.; Stéphane, H.; and Frédéric, J. 2016. Improving semantic embedding consistency by metric learning for zero-shot classiffication. In *ECCV*, 730–746.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781.

Mohammad, N.; Tomas, M.; Samy, B.; Yoram, S.; Jonathon, S.; Andrea, F.; Greg, C.; and Jeffrey, D. 2014. Zero-shot learning by convex combination of semantic embeddings. In *ICLR*.

Radovanović, M.; Nanopoulos, A.; and Ivanović, M. 2010. Hubs in space: Popular nearest neighbors in high-dimensional data. *JMLR* 11:2487–2531.

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

Shigeto, Y.; Suzuki, I.; Hara, K.; Shimbo, M.; and Matsumoto, Y. 2015. Ridge regression, hubness, and zero-shot learning. In *ECML-PKDD*, 135–151.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *CoRR* abs/1409.1556.

Szegedy, C.; Ioffe, S.; Vanhoucke, V.; and Alemi, A. A. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 4278–4284.

van der Maaten, L., and Hinton, G. 2008. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research* 9:2579–2605.

Wang Xiaolong, Shrivastava Abhinav, G. A. 2017. A-fast-rcnn: Hard positive generation via adversary for object detection. In *CVPR*, 2606–2615.

Yang, Y., and Hospedales, T. M. 2015. A unified perspective on multi-domain and multi-task learning. In *ICLR*.

Zhang, Z., and Saligrama, V. 2016. Zero-shot recognition via structured prediction. In *ECCV*, 533–548.

Zhang, and Ziming, Saligrama, V. 2015. Zero-shot learning via semantic similarity embedding. In *ICCV*, 4166–4174.

Zhang, and Ziming, Saligrama, V. 2016. Zero-shot learning via joint latent similarity embedding. In *CVPR*, 6034–6042.