Generating Visual Representations for Zero-Shot Classification

Maxime Bucher, Stéphane Herbin, Frédéric Jurie

Generating Visual Representations for Zero-Shot Classification

Maxime Bucher, Stéphane Herbin
ONERA - The French Aerospace Lab
Palaiseau, France
maxime.bucher@onera.fr,
stephane.herbin@onera.fr

Frédéric Jurie
Normandie Univ, UNICAEN, ENSICAEN, CNRS
Caen, France
frederic.jurie@unicaen.fr

Abstract

*This paper addresses the task of learning an image classifier when some categories are defined by semantic descriptions only (*e.g. *visual attributes) while the others are defined by exemplar images as well. This task is often referred to as the Zero-Shot classification task (ZSC). Most of the previous methods rely on learning a common embedding space allowing to compare visual features of unknown categories with semantic descriptions. This paper argues that these approaches are limited as i) efficient discriminative classifiers can't be used ii) classification tasks with seen and unseen categories (Generalized Zero-Shot Classification or GZSC) can't be addressed efficiently. In contrast, this paper suggests to address ZSC and GZSC by i) learning a conditional generator using seen classes ii) generate artificial training examples for the categories without exemplars. ZSC is then turned into a standard supervised learning problem. Experiments with 4 generative models and 5 datasets experimentally validate the approach, giving state-of-the-art results on both ZSC and GZSC.*

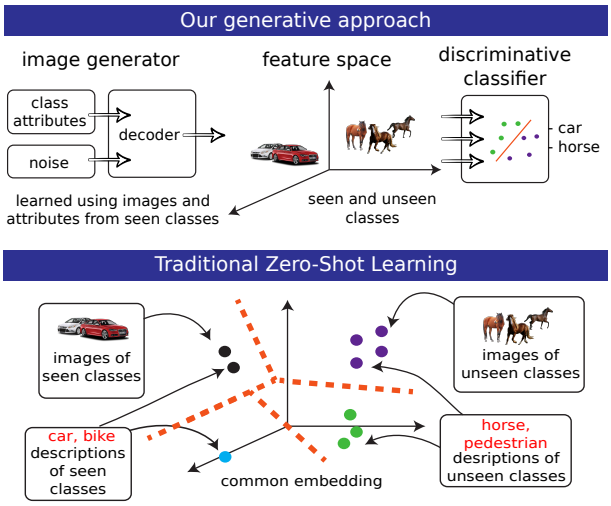

Figure 1: Our method consists in i) learning an image feature generator capable of generating artificial image representations from given attributes ii) learning a discriminative classifier from the artificially generated training data.

1. Introduction and related works

Zero-Shot Classification (ZSC) [22] addresses classification problems where not all the classes are represented in the training examples. ZSC can be made possible by defining a high-level description of the categories, relating the new classes (*the unseen classes*) to classes for which training examples are available (*seen classes*). Learning is usually done by leveraging an intermediate level of representation, the attributes, that provide semantic information about the categories to classify. As pointed out by [32] this paradigm can be compared to how human can identify a new object from a description of it, leveraging similarities between its description and previously learned concepts.

Recent ZSC algorithms (*e.g.* [1, 5]) do the classification by defining a zero-shot prediction function that outputs the class y having the maximum compatibility score with the image x: $f(x) = \arg\max_y S(x, y)$. The compatibility function, for its part, is often defined as $S(x, y; W) = \theta(x)^t W \phi(y)$ where θ and ϕ are two projections and W is a bilinear function relating the two in a common embedding. There are different variants in the recent literature on how the projections or the similarity measure are computed [11, 8, 15, 29, 32, 40, 41, 43], but in all cases the class is chosen as the one maximizing the compatibility score. This embedding and maximal compatibility approach, however, does not exploit, in the learning phase, the information potentially contained in the semantic representation of the unseen categories. The only step where a discriminating capability is exploited is in the final label selection which uses an $\arg\max_y$ decision scheme, but not in the setting of the compatibility score itself.

A parallel can be easily done between the aforementioned approaches and *generative models* such as defined

in the machine learning community. Generative models estimate the joint distribution $p(y, x)$ of images and classes, often by learning the class prior probability $p(y)$ and the class-conditional density $p(x|y)$ separately. However, as it has been observed for a long time [37], discriminative approaches trained for predicting directly the class label have better performance than model-based approaches as long as the learning database reliably samples the target distribution.

Despite one can expect discriminative methods to give better performance [37], they can't be used directly in the case of ZSC for obvious reasons: as no images are available for some categories, discriminative classifiers cannot be learned out-of-the-box.

This paper proposes to overcome this difficulty by generating training features for the unseen classes, in such a way that standard discriminative classifiers can be learned (Fig. 1). Generating data for machine learning tasks has been studied in the literature *e.g.*, [18] or [3] to compensate for imbalanced training sets. Generating novel training examples from the existing ones is also at the heart of the technique called *Data Augmentation*, frequently used for training deep neural networks [23]. When there is no training data at all for some categories, some underlying parametric representation can be used to generate missing training data, assuming a mapping from the underlying representation to the image space. [12] generated images by applying warping and other geometric / photometric transformations to prototypical logo exemplars. A similar idea was also presented in [19] for text spotting in images. [7] capture what they call *The Gist of a Gesture* by recording human gestures, representing them by a model and use this model to generate a large set of realistic gestures.

We build in this direction, in the context of ZSC, the underlying representation being some attribute or text based description of the unseen categories, and the transformation from attributes to image features being learned from the examples of the seen classes. A relevant way to learn this transformation is to use generative models such as *denoising auto encoders* [4] and *generative adversarial nets* (GAN) [16] or their variants [10, 26]. GANs consist in estimating generative models via an adversarial process simultaneously learning two models, a generative model that captures the data distribution, and a discriminative model that estimates the probability that a sample came from the training data rather than the generator. The *Conditional Generative Adversarial Nets* of [28] is a very relevant variant adapted to our problem.

In addition to the advantage of using discriminative classifiers – which is expected to give better performance – our approach, by nature, can address the more realistic task of Generalized Zero-Shot Classification (GZSC). This problem, introduced in [9], assumes that both seen and unseen categories are present at test time, making the traditional approaches suffering from bias decision issues. In contrast, the proposed approach uses (artificial) training examples of both seen and unseen classes during training, avoiding the aforementioned issues.

Another reason to perform classification inference directly in the visual feature space rather than in an abstract attribute or embedding space is that data are usually more easily separated in the former, especially when using discriminant deep features that are now commonly available.

This paper experimentally validates the proposed strategy on 4 standard Zero-Shot classification datasets (Animals with Attributes (AWA) [22], SUN attributes (SUN) [31], Apascal&Ayahoo (aP&Y) [14] and Caltech-UCSD Birds-200-2011 (CUB) [38]), and gives insight on how the approach scales on large datasets such as ImageNet [11]. It shows state-of-the-art performance on all datasets for both ZSC and GZSC.

2. Approach

2.1. Zero shot classification

As motivated in the introduction, we address in this paper the problem of learning a classifier capable of discriminating between a given set of classes where empirical data is only available for a subset of it, the so-called *seen* classes. In the vocabulary of zero-shot classification, the problem is usually qualified as inductive — we do not have access to any data from the unseen classes — as opposed to transductive where the *unseen* data is available but not the associated labels. We do not address in this paper the transductive setting, considering that the availability of target data is a big constraint in practice.

The learning dataset $\mathcal{D}_s$ is defined by a series of triplets $\{x_i^s, a_i^s, y_i^s\}_{i=1}^{N_s}$ where $x_i^s \in \mathcal{X}$ is the raw data (image or features), $y_i^s \in \mathcal{Y}_s$ is the associated class label and a_i^s is a rich semantic representation of the class (attributes, word vector or text) belonging to $\mathcal{A}_s$. This semantic representation is expected to i) contain enough information to discriminate between classes by itself, ii) be predictable from raw data and iii) infer unambiguously the class label $y = l(\mathbf{a})$.

In an inductive ZSC problem, all that is known regarding the new target domain is the set of semantic class representations $\mathcal{A}_u$ of the *unseen* classes. The goal is to use this information and the structure of the semantic representation space to design a classification function f able to predict the class label $\hat{y} = f(\mathbf{x}; \mathcal{A}_u, \mathcal{D}_s)$. The classification function f is usually parametric and settled by the optimization of an empirical learning criterion.

2.2. Discriminative approach for ZSC

In ZSC, the main problem is precisely the fact that no data is available for the unseen classes. The approach taken

in this paper is to artificially generate data for the unseen classes given that seen classes and their semantic representations provide enough information to do so, and then apply a discriminative approach to learn the class predictor.

The availability of data for the unseen classes has two main advantages: it can make the classification of seen *and* unseen classes as a single homogeneous process, allowing to address Generalized Zero Shot Classification as a single supervised classification problem; it potentially allows a larger number of unseen classes, which is for instance required for datasets such ImageNet [11].

Let $\widehat{\mathcal{D}}_u = \{\hat{x}_i^u, a_i^u, y_i^u\}_{i=1}^{N_u}$ be a database generated to account for the unseen semantic class representation $a^u \in \mathcal{A}_u$. The ZSC classification function becomes: $\hat{y} = f_D(\mathbf{x}; \widehat{\mathcal{D}}_u, \mathcal{D}_s)$ and can be used in association with the seen data $\mathcal{D}_s$, to learn a homogeneous supervised problem.

2.3. Generating unseen data

Our generators of unseen data build on the recently proposed approaches for conditional data generation as presented in section 1. The idea is to learn globally a parametric random generative process G using a differentiable criterion able to compare, as a whole, a target data distribution and a generated one.

Given $\mathbf{z}$ a random sample from a fixed multivariate prior distribution, typically uniform or Gaussian, and $\mathbf{w}$ the set of parameters, new sample data consistent with the semantic description $\mathbf{a}$ are generated by applying the function: $\widehat{x} = G(\mathbf{a}, \mathbf{z}; \mathbf{w})$. A simple way to generate conditional $\widehat{x}$ data is to concatenate the semantic representation $\mathbf{a}$ and the random prior $\mathbf{z}$ as the input of a multi-layer network, as shown in Fig. 2.

We now present 4 different strategies to design such a conditional data generator, the functional structure of the generator being common to all the described approaches.

Generative Moment Matching Network A first approach is to adapt the Generative Moment Matching Network (GMMN) proposed in [24] to conditioning. The generative process will be considered as good if for each semantic description $\mathbf{a}$ two random populations $\mathcal{X}(\mathbf{a})$ from $\mathcal{D}_s$ and $\widehat{\mathcal{X}}(\mathbf{a}; \mathbf{w})$ sampled from the generator have low *maximum mean discrepancy* which is a probability divergence measure between two distributions. This divergence can be approximated using a Hilbert kernel based statistics [17] – typically a linear combination of Gaussian functions with various widths — which has the big advantage of being differentiable and may be thus exploited as a machine learning cost. Network parameters $\mathbf{w}$ are then obtained by optimizing the differentiable statistics by stochastic gradient descent, using batches of generated and real data conditioned by the semantic description $\mathbf{a}$.

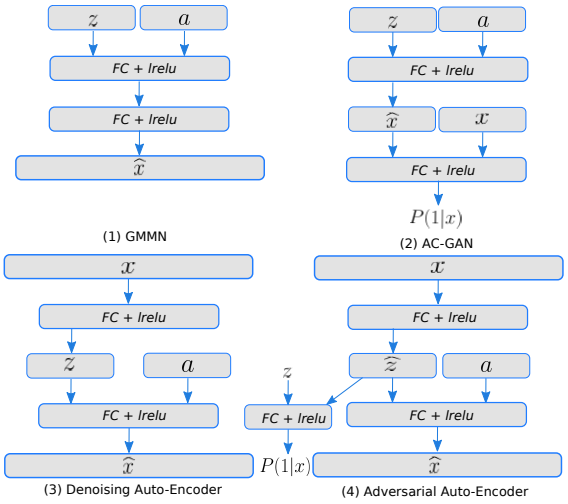

Figure 2: Architecture of the different generative models studied.

Conditional Generative adversarial models Our second model builds on the principles of the generative adversarial networks (GAN), which is to learn a discrepancy measure between a true and a generated distributions — the *Discriminator* — simultaneously with the data generator. One extension allowing to produce conditional distributions is the AC-GAN [30] (Fig. 2) where the generated and the true distributions are compared using a binary classifier, and the quality of the conditional generation is controlled by the performance of this auxiliary task.

This model bears similarities with the GMMN model, the key difference being that in the GMMN distributions of true and generated data are compared using the kernel based empirical statistics while in the AC-GAN case it is measured by a learned discriminative parametric model.

Denoising Auto-Encoder Our third generator relies on the work presented in [4], where an encoder/decoder structure is proposed to design a data generator, the latent code playing the role of the random prior $\mathbf{z}$ used to generate the data. A simple extension able to introduce a conditional data generation control has been developed by concatenating the semantic representation $\mathbf{a}$ to the code that is fed to the decoder (Fig. 2).

In practice, this model is learned as a standard auto-encoder, except that i) some noise is added to the input and ii) the semantic representation $\mathbf{a}$ is concatenated to the code in the hidden layer. For generating novel examples, only the decoder part, *i.e.* the head of the network using $\mathbf{z}$ and $\mathbf{a}$ as input to produce $\widehat{\mathbf{x}}$ is used.

Adversarial Auto-Encoder Our fourth generator is inspired by [26], which is an extension of the denoising auto-

encoder. It introduces an adversarial criterion to control the latent code produced by the encoder part, so that the code distribution matches a fixed prior distribution. This extra constraint is expected to ensure that all parts of the sampling prior space will produce meaningful data.

During training, both the auto-encoder and the discriminator are learned simultaneously. For generating novel examples, as for the denoising auto-encoder, only the decoder part is used.

2.4. Implementing the generators

We implemented our 4 generative models with neural networks, whose architectures are illustrated Fig. 2. Hidden layers are fully connected (FC) with leaky-relu nonlinearity [25] (leakage coefficient of 0.2). For the models using a classifier (the AC-GAN and the Adversarial auto-encoder), the classifier is a linear classifier (fully connected layer + Softmax activation function). The loss used to measure the quality of the reconstruction in the two auto-encoders is the L2 norm.

Regarding how to sample the noise $\mathbf{z}$, we did not observe any difference between sampling it from a Gaussian distribution or from a uniform distribution.

3. Experiments

In this section, after presenting the datasets and the experimental settings, we start by comparing the different generative models described in the previous section. We then show how our approach can be used for the Generalized Zero-shot Classification Task, which is one of the key contributions of the paper, provide some experiments on a large scale zero shot classification task, and finally compare our approach with state-of-the art Zero-Shot approaches on the regular Zero-shot Classification Task.

3.1. Datasets and Settings

A first experimental evaluation is done on 4 standard ZSC datasets: Animals with Attributes (AWA) [22], SUN attributes (SUN) [31], Apascal&Ayahoo (aP&Y) [14] and Caltech-UCSD Birds-200-2011 (CUB) [38] . These benchmarks exhibit a great diversity of concepts; SUN and CUB are for fine-Grained categorization, and include respectively birds and scenes images; AwA contains images of animals from 50 different categories; finally, aP&Y has broader concepts, from cars to animals. For each dataset, attributes descriptions are given, either at the class level or at image level. aP&Y, CUB and SUN have per image binary attributes that we average to produce per class real valued representations. In order to make comparisons with other works, we follow the same training/testing splits for AwA [22], CUB [2] and aP&Y [14]. For SUN we experiment two different settings: one with 10 unseen classes as in [20], a second, more competitive, with ten different folds randomly chosen and averaged, as proposed by [8] (72/71 splits).

Table 1: Zero-Shot classification accuracy (mean) on the validation set, for the 4 generative models.

Model	aP&Y	AwA	CUB	SUN	Avg
Denois. Auto-encod. [4]	62.0	66.4	42.8	82.5	63.4
AC-GAN [30]	55.2	66.0	44.6	83.5	62.3
Adv. Auto-encod. [26]	59.5	**68.4**	49.8	83.7	65.3
GMMN [24]	**65.9**	67.0	**52.4**	**84.0**	**67.3**

Image features are computed using two deep networks, the VGG-VeryDeep-19 [34] and the GoogLeNet [36] networks. For the VGG-19 we use the 4,096-dim top-layer hidden unit activations (fc7) while for the GoogLeNet we use the 1,024-dim top-layer pooling units. We keep the weights learned on ImageNet fixed *i.e.*, we don't apply any fine-tuning.

The classifiers are obtained by adding a standard Fully Connected with Softmax layer to the pre-trained networks. We purposively chose a simple classifier to better observe the behavior of the generators. In all our experiments we generated 500 artificial image features by class, which we consider to be a reasonable trade-off between accuracy and training time; we have not observed any significant improvement when adding more images.

Each architecture has its own set of hyper-parameters (typically the number of units per layer, the number of hidden layers, the learning rate, *etc.*). They are obtained trough a 'Zero-shot' cross-validation procedure. In this procedure, 20% of the seen classes are considered as unseen (hence used as validation set), allowing to choose the hyper-parameters maximizing the accuracy on this so-obtained validation set. In practice, typical values for the number of neurons (resp. the number of hidden layers) are in the range of [500-2000] (resp. 1 or 2).

Model parameters are initialized according to a centered Gaussian distribution ($\sigma = 0.02$). They are optimized with the Adam solver [21] with a cross-validated learning rate (typically of 10^{-4}), using mini-batches of size 128 except for the GMMN where each batch contains all the training images of one class, to make the estimation of the statistics more reliable. In order to avoid over-fitting, we used dropout [35] at every layer (probability of drop of 0.2 for the inputs layers and of 0.5 for the hidden layers). Input data (both image features and w2c vectors) are scaled to [0,1] by applying an affine transformation. With the TensorFlow framework [13] running on a Nvidia Titan X pascal GPU, the learning stage takes around 10 minutes for a given set of hyper-parameters. Our code will be made publicly available.

Table 2: Generalized Zero-Shot classification accuracy on AWA. Image features are obtained with the GoogLeNet. [36] CNN.

	AwA			
Method	$u \rightarrow u$	$s \rightarrow s$	$u \rightarrow a$	$s \rightarrow a$
Lampert *et al*. [22]^{dap}	51.1	78.5	2.4	77.9
Lampert *et al*. [22]^{iap}	56.3	77.3	1.7	76.8
Norouzi *et al*. [29]	63.7	76.9	9.5	75.9
Changpinyo *et al*. [8]$^{o-vs-o}$	70.1	67.3	0.3	67.3
Changpinyo *et al*. [8]^{struct}	73.4	81.0	0.4	81.0
Ours	**81.48**	**82.73**	**32.32**	**81.32**
Ours. (VGG-19)	87.78	85.61	38.21	83.14

Table 3: Generalized Zero-Shot classification accuracy on CUB. Image features are obtained with the GoogLeNet [36] CNN.

	CUB			
Method	$u \rightarrow u$	$s \rightarrow s$	$u \rightarrow a$	$s \rightarrow a$
Lampert *et al*. [22]^{dap}	38.8	56.0	4.0	55.1
Lampert *et al*. [22]^{iap}	36.5	69.6	1.0	69.4
Norouzi *et al*. [29]	35.8	70.5	1.8	69.9
Changpinyo *et al*. [8]$^{o-vs-o}$	53.0	67.2	8.4	66.5
Changpinyo *et al*. [8]^{struct}	54.4	**73.0**	13.2	72.0
Ours.	**61.05**	72.38	**26.87**	**72.00**
Ours. (VGG-19)	59.70	71.21	20.12	69.45

3.2. Comparing the different generative models

Our first round of experiments consists in comparing the performance of the 4 generative models described in Section 2.3, on the regular Zero-shot classification task. Our intention is to select the best one for further experiments. Performance on the validation set is reported Table 1. We can see that the GMMN model outperforms the 3 others on average, with a noticeable 5% improvement on aP&Y. Its optimization is also computationally more stable than the adversarial versions. We consequently chose this generator for the following.

We explain the superiority of the GMMN model by the fact it aligns the distributions by using an explicit model of the divergence of the distributions while the adversarial autoencoder and the AC-GAN have to learn it. For its part, the denoising autoencoder doesn't have any guaranty that the distributions are aligned, explaining its weak performance compared to the 3 other generators.

3.3. Generalized Zero-Shot Classification task

In this section, we follow the Generalized Zero-Shot Learning (GZSC) protocol introduced by Chao *et al*. [9]. In this protocol, test data are from any classes, seen or unseen. This task is more realistic and harder, as the number of class candidates is larger.

We follow the notations of [9], *i.e*.
$u \rightarrow u$: test images from unseen classes, labels of unseen classes (conventional ZSC)
$s \rightarrow s$: test images from seen classes, labels of seen classes (multi-class classication for seen classes)
$u \rightarrow a$: test images from unseen classes, labels of seen and unseen classes (GZSC)
$s \rightarrow a$: test images from seen classes, labels of seen and unseen classes (GZSC)

In the first two cases, only the seen/unseen classes are used in the training phase. In the last two cases, the classifier is learned with training data combining images generated for all classes (seen and not seen).

Most of the recent ZSC works *e.g*., [2, 6, 5, 32] are focused on improving the embedding or the scoring function. However, [9] has shown that this type of approach is unpractical with GZSC. Indeed the scoring function is in this case biased toward seen classes, leading to very low accuracy on the unseen classes. This can be seen on Table 2 and 3 ($u \rightarrow a$ column), where the accuracy drops significantly compared to regular ZSC performance. The data distribution of the ZSC datasets are strongly subject to this bias, as unseen classes are very similar to seen classes both in terms of visual appearance and attribute description. When seen and unseen classes are candidates, it becomes much harder to distinguish between them. For example, the horse (seen) and the zebra classes (unseen) of the AwA dataset cannot be distinguished by standard ZSC methods.

As we can see on Table 2 and 3, our generative approach outperforms any other previous approach. In the hardest case, $u \rightarrow a$, it gives the accuracy of 30% (resp. 10%) higher than state-of-the-art approaches on the AwA (resp. CUB) dataset. It can be easily explained by the fact that it doesn't suffer from the scoring function problem we mentioned, as the Softmax classifier is learned to discriminate both seen and unseen classes, offering a decisive solution to the bias problem.

3.4. Large Scale Zero-Shot Classification

We compared our approach with state-of-the-art methods on a large-scale Zero-Shot classification task. These experiences mirror those presented in [15]: 1000 classes from those of the ImageNet 2012 1K set [33] are chosen for training (seen classes) while 20.345 others are considered to be unseen classes with no image available. Image features are computed with the GoogLeNet network [36].

In contrast with ZSC datasets, no attributes are provided for defining unseen classes. We represent those categories using a skip-gram language model [27]. This model is learned on a dump of the Wikipedia corpus (≈3 billion words). Skip-gram is a language model learned to predict context from words. The neural network has 1 input layer, 1 hidden layer and 1 output layer having the size of the vocabulary (same size as the input layer). The hidden layer has 500 neurons in our implementation. In the literature, the hidden layer has been reported to be an in-

Table 4: Zero-shot and Generalized ZSC on ImageNet.

Scenario	Method	Flat Hit @K 1	2	5	10	20
2-hop	Frome [15]	6.0	10.0	18.1	26.4	36.4
	Norouzi [29]	9.4	15.1	24.7	32.7	41.8
	Changpinyo [8]	10.5	16.7	28.6	40.1	52.0
	Ours.	**13.05**	**21.52**	**33.71**	**43.91**	**57.31**
2-hop (+1K)	Frome [15]	0.8	2.7	7.9	14.2	22.7
	Norouzi [29]	0.3	7.1	17.2	24.9	33.5
	Ours.	**4.93**	**13.02**	**20.81**	**31.48**	**45.31**
3-hop	Frome [15]	1.7	2.9	5.3	8.2	12.5
	Norouzi [29]	2.7	4.4	7.8	11.5	16.1
	Changpinyo [8]	2.9	4.9	9.2	14.2	20.9
	Ours.	**3.58**	**5.97**	**11.03**	**16.51**	**23.88**
3-hop (+1K)	Frome [15]	0.5	1.4	3.4	5.9	9.7
	Norouzi [29]	0.2	2.4	5.9	9.7	14.3
	Ours.	**1.99**	**4.01**	**6.74**	**11.72**	**16.34**
All	Frome [15]	0.8	1.4	2.5	3.9	6.0
	Norouzi [29]	1.4	2.2	3.9	5.8	8.3
	Changpinyo [8]	1.5	2.4	4.5	7.1	10.9
	Ours.	**1.90**	**3.03**	**5.67**	**8.31**	**13.14**
All (+1K)	Frome [15]	0.3	0.8	1.9	3.2	5.3
	Norouzi [29]	0.2	1.2	3.0	5.0	7.5
	Ours.	**1.03**	**1.93**	**4.98**	**6.23**	**10.26**

teresting embedding space for representing word. Consequently, We use this hidden layer to describe each class label by embedding the class name into this 500-dimensional space. Some classes cannot be represented as their name is not contained in the vocabulary established by parsing the Wikipedia corpus. Such classes are ignored, bringing the number of classes from 20,842 to 20,345 classes. For fair comparison, we take the same language model as [8] with the same classes excluded.

As in [8, 15] our model is evaluated on three different scenarios, with an increasing number of unseen classes: i) 2-hop: 1,509 classes ii) 3-hop: 7,678 classes, iii) All: all unseen categories.

For this task we use the Flat-Hit@K metric, the percentage of test images for which the model returns the true labels in the top K prediction scores.

Table 4 summarizes the performance on the 3 hops. As one can see, our model gets state-of the art performance for each configuration. As it can be observed from these experiments, our generative model is very suitable for this large scale GZSC problem *e.g.*, our approach improves by 5% best competitors for the Flat-Hit 1 metric on the 2-hop scenario.

3.5. Classical Zero-Shot Classification task

In this last section, we follow the protocol of the standard ZSC task: during training, only data from seen classes are available while at test time new images (from unseen classes only) have to be assigned to one of the unseen classes.

As explained in the introduction, the recent ZSC literature [2, 6, 5, 32] mostly focuses on developing a good embedding for comparing attributes and images. One of our motivations for generating training images was to make the training of discriminative classifiers possible, assuming it would result in better performance. This section aims at validating this hypothesis on the regular ZSC task.

Table 5: Zero-shot classification accuracy (mean±std) on 5 runs. We report results with VGG-19 and GoogLeNet features. SUN dataset is evaluated on 2 different splits (see 3.1). * [8] features extracted from an MIT Places[45] pre-trained model.

Feat.	Method	aP&Y	AwA	CUB	SUN
GoogLeNet[36]	Lampert *et al*. [22]	-	60.5	39.1	-/44.5
	Akata *et al*. [2]	-	66.7	50.1	-/-
	Changpinyo *et al*. [8]	-	72.9	54.7	**90.0/62.8***
	Xian *et al*. [41]	-	71.9	45.5	-
	Ours.	**55.34**	**77.12**	**60.10**	**85.50/56.41**
VGG-VeryDeep [34]	Lampert *et al*. [22]	38.16	57.23	-	72.00/-
	Romera-Paredes [32]	24.22	75.32	-	82.10/-
	Zhang *et al*. [43]	46.23	76.33	30.41	82.50/-
	Zhang *et al*. [44]	50.35	80.46	42.11	83.83/-
	Wang *et al*. [39]	-	78.3	48.6	-/-
	Bucher *et al*. [5]	53.15	77.32	43.29	84.41/-
	Bucher *et al*. [6]	56.77	86.55	45.87	86.21/-
	Ours.	**57.19**	**87.78**	**59.70**	**88.01/-**

Table 5 summarizes our experiments, reporting the accuracy obtained by state of the art methods on the 4 ZSC datasets, with 2 different deep image features. Each entry is the mean/standard deviation computed on 5 different runs.

With the VGG network, our method give above state-of-the-art performance on each dataset, with a noticeable improvement of more than 15% on CUB. On the SUN dataset, Changpinyo *et al*. [8]'s seems to give better performance but used the MIT Places dataset to learn the features. It has been recently pointed out in sec. 5.1 of Xiang et al. [42] that this database "intersects with both training and test classes of SUN, which could explain their better results compared to ours.

4. Conclusions

This paper introduces a novel way to address Zero-Shot Classification and Generalized Zero-Shot Classification tasks by learning a conditional generator from seen data and generating artificial training examples for the categories without exemplars, turning ZSC into a standard supervised learning problem. This novel formulation addresses the two main limitation of previous ZSC method *i.e.*, their intrinsic bias for Generalized Zero-Shot Classification tasks and their limitations in using discriminative classifiers in the deep image feature space. Our experiments with 4 generative models and 5 datasets experimentally validate the approach and give state-of-the-art performance.

References

[1] Zeynep Akata, Mateusz Malinowski, Mario Fritz, and Bernt Schiele. Multi-cue Zero-Shot Learning with Strong Supervision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR*, pages 59–68. IEEE, June 2016.

[2] Zeynep Akata, Scott Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. Evaluation of Output Embeddings for Fine-Grained Image Classification. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[3] Mohamed Bahy Bader-El-Den, Eleman Teitei, and Mo Adda. Hierarchical classification for dealing with the Class imbalance problem. *IJCNN*, 2016.

[4] Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In *Advances in Neural Information Processing Systems*, pages 899–907, 2013.

[5] M Bucher, S Herbin, and F Jurie. Improving semantic embedding consistency by metric learning for zero-shot classiffication. In *European Conference on Computer Vision*, 2016.

[6] Maxime Bucher, Stéphane Herbin, and Frédéric Jurie. Hard negative mining for metric learning based zero-shot classification. In *Computer Vision–ECCV 2016 Workshops*, pages 524–531. Springer, 2016.

[7] Maria E Cabrera and Juan P Wachs. Embodied gesture learning from one-shot. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN*, pages 1092–1097. IEEE, 2016.

[8] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Synthesized classifiers for zero-shot learning. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 5327–5336. IEEE, 2016.

[9] Wei-Lun Chao, Soravit Changpinyo, Boqing Gong, and Fei Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *European Conference on Computer Vision*, pages 52–68. Springer, 2016.

[10] Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, and Wenjie Li. Mode Regularized Generative Adversarial Networks. *arXiv*, December 2016.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[12] Christian Eggert, Anton Winschel, and Rainer Lienhart. On the Benefit of Synthetic Data for Company Logo Detection. In *ACM Multimedia*, 2015.

[13] Martín Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[14] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[15] Andrea Frome, Gregory S Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. DeViSE: A Deep Visual-Semantic Embedding Model. In *Conference on Neural Information Processing Systems (NIPS)*, 2013.

[16] I Goodfellow, J Pouget-Abadie, and M Mirza. Generative adversarial nets. In *NIPS*, 2014.

[17] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

[18] Hongyu Guo and Herna L Viktor. Learning from imbalanced data sets with boosting and data generation - the DataBoost-IM approach. *SIGKDD Explorations*, 2004.

[19] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading Text in the Wild with Convolutional Neural Networks. *International Journal of Computer Vision*, 116(1):1–20, 2016.

[20] Dinesh Jayaraman and Kristen Grauman. Zero-shot recognition with unreliable attributes. In *Conference on Neural Information Processing Systems (NIPS)*, 2014.

[21] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-Based Classification for Zero-Shot Visual Object Categorization. *IEEE Trans Pattern Anal Mach Intell*, 36(3):453–465, 2014.

[23] Y LeCun, L Bottou, Y Bengio, and P Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

[24] Yujia Li, Kevin Swersky, and Richard S Zemel. Generative moment matching networks. In *ICML*, pages 1718–1727, 2015.

[25] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*. Citeseer, 2013.

[26] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[27] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[28] Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets. *arXiv*, November 2014.

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

[30] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. *arXiv preprint arXiv:1610.09585*, 2016.

[31] Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2751–2758. IEEE, 2012.

[32] Bernardino Romera-Paredes and Philip HS Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, pages 2152–2161, 2015.

[33] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[34] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2014.

[35] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[36] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[37] Ilkay Ulusoy and Christopher M Bishop. Generative versus Discriminative Methods for Object Recognition. In *CVPR*, 2005.

[38] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[39] Qian Wang and Ke Chen. Zero-shot visual recognition via bidirectional latent embedding. *arXiv preprint arXiv:1607.02104*, 2016.

[40] Jason Weston, Samy Bengio, and Nicolas Usunier. Wsabie: scaling up to large vocabulary image annotation. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, pages 2764–2770. AAAI Press, 2011.

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

[42] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning-the good, the bad and the ugly. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[43] Ziming Zhang and Venkatesh Saligrama. Zero-Shot Learning via Semantic Similarity Embedding. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[44] Ziming Zhang and Venkatesh Saligrama. Zero-shot learning via joint latent similarity embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6034–6042, 2016.

[45] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.