Zero-shot Learning for Image Generation based on Wikipedia Text Descriptions

Shayan Kousha [1]

Abstract

The purpose of generative Zero-shot learning (ZSL) is to learning from seen classes, transfer the learned knowledge, and create samples of unseen classes from the description of these unseen categories. To achieve better ZSL accuracy the models need to get better at understanding the description of unseen classes. We introduce a regularization model that forces generator models to pay close attention to the description of each category to create meaningful visual features while allowing deviation from seen classes. Our empirical results demonstrate that our model outperforms multiple state-of-the-art models on the task of generalized zero-shot recognition and classification when trained on the CUB and NABirds datasets. Code is available at *https://github.com/shayankousha/TGRZSL*

1. Introduction

Computer vision classification models have advanced in the past few years. They mainly use a huge amount of data per class and learn to classify these classes. However, the data gathering process is usually time consuming and expensive. As a result, we do not have enough training data for many categories and classes. This issue has inspired researchers to find ways to eliminate the need of having a large amount of data at training time. One of these approaches that has gained popularity in recent years is Zero-Shot Learning (ZSL). The idea behind ZSL is to learn from classes with many training samples, transfer the learned knowledge, and use descriptions of classes without any training samples to classify samples of these classes at test time. The classes without training samples are referred to as "unseen" classes and classes with training samples are called "seen" classes.

[1]Department of Electrical Engineering and Computer Science, York University, Toronto, Canada. Correspondence to: Shayan Kousha <shayanko@yorku.ca>.

Project Report for EECS6327 (Fall 2020), Department of Electrical Engineering and Computer Science, York University.

We can go one step further and instead of only classifying these unseen classes by learning from seen classes, we can learn to generate samples of unseen classes. This is a challenging task because not only the model needs to be able to classify samples of unseen classes at test time, it needs to also learn to generate new samples from the unseen classes. Since no sample of the unseen classes is used at training time, the model needs to learn to "imaging" the unseen classes when generating samples. This notion of imaginations has been studied in recent works (Guo et al., 2017; Yizhe Zhu & Elgammal, 2018; Long & Shao, 2017) adapting many deep generative models. Once the "imagined" images are generated, They can be used in other computer vision tasks. For instance, the generated visual examples of unseen classes can be considered as labeled data to a regular classifier.

Despite the advances in the field of generative ZSL, there are still some challenges that are remained unsolved. These generative ZSL methods do not guarantee that the generated visual examples of unseen classes deviate from seen classes. Since generative ZSL models learn from seen classes and later transfer the learned knowledge to create samples of unseen classes, there is a risk that the generated images are too similar to samples from seen classes. The second problem arises when the model is forced to generate samples of unseen classes that deviate from seen classes without careful considerations. If the model and learning process is not setup properly, there is a risk to generate images that do not follow the description of unseen classes and their only property is that they deviate enough from the seen classes. Our work mainly focuses on forcing the generative ZSL model to generate meaningful samples of unseen classes while encouraging the model to learn to deviate from seen classes addressing both mentioned issues and challenges of generative ZSL.

Our work is based on the assumption that the description of each unseen class has enough information to accompany the transferred knowledge to create meaningful visual examples of unseen classes. For example, if we know what a bird is (knowledge transferred from seen bird images) and know that there exist small birds with a black body, we can easily "imagine" a crow without having to see any images of crows

previously. By paying close attention to the description we can "imagine" a meaningful image as well as understanding how the described bird differs from the birds we have seen so far. Therefore, we believe paying attention to the details of the description is the key to solve both mentioned issues. Inspired by these observations, we introduce a regularizer that allows us to force the generator model to pay close attention to the details by creating text descriptions from visual features created by the generator. Therefore, we call our proposed method Text Generator Regularizer ZSL (TGRZSL). Our contributions to the field of generative ZSL are as follows:

1. We propose a regularizer that encourages the generative process to pay close attention to the description each class is accompanied with. The regularizer generates new textual information from the imagined image and compares it to the original textual data used in the generation process. A loss function penalizes the generator and regularizer if the generated textual description is not similar to the inputted description.

2. Our regularizer is unsupervised and does not depend on the generative ZSL approach. Therefore, it can be added to any ZSL approach that uses text data as the description of each class without requiring any modifications to the structure of the ZSL model.

3. Our method is tested on multiple datasets using multiple evaluation metrics, outperforming state-of-the-art methods in most cases.

The remainder of the paper is as follows: In section 2 we review related work and mention some of the contributions made to the field of zero-shot learning and recognition over the past few years. This is followed by a formal definition of our setting and data points in section 3 and a detailed description of our method and models in section 4. Later, in section 5, we describe the datasets that we use in our experiments as well as the pre-processing steps done on them by (Elhoseiny et al., 2017) and (Yizhe Zhu & Elgammal, 2018). Finally, in section 6 we present our results and compare our method to state-of-the-art models.

2. Related Work

There have been some attempts to address some of the limitations and drawbacks of ZSL. Here we mention some of the contributions to the field.

A noticeable early ZSL work is from (Lampert et al., 2009) where they proposed a Direct Attribute Prediction (DAP) model. They had the limiting assumption of independence between the attributes that accompany classes to describe these categories (e.g. text descriptions). Later (Akata et al., 2016) proposed an Attribute Label Embedding(ALE) approach which eliminated the need for the independence assumption.

Both of the proposed methods are unsupervised techniques. However, new approaches mainly focus on generating images of unseen classes using the accompanied attributes or description, converting zero-shot recognition to a conventional classification task with labeled data (Guo et al., 2017; Yizhe Zhu & Elgammal, 2018; Long & Shao, 2017). Given that it is a fairly new approach, relying on learned knowledge from seen classes and unseen class descriptions to generate samples of an unseen class has some issues like not deviating enough from seen classes or generating unrealistic images of unseen classes.

(Elhoseiny & Elfeki, 2019) proposed a new method inspired by the psychology of human creativity. Therefore, they named their approach Creativity Inspired Zero-shot Learning (CIZSL) which is one of the baselines used in section 6. They proposed a learning signal inspired by the psychology of human creativity to explicitly force the model to carefully deviate from seen classes when generating samples for unseen classes. They use a parameterized entropy measure to make the learning process easier. They also introduced a training procedure that uses hallucinated text descriptions to explore the unseen class space. We will also use this technique to train our ZSL model and regularizer.

3. Setting

In our zero-shot learning setting each data point consists of an image (or visual features), a class label, and a semantic representation of the class, which describes the class. Therefore, all samples of the same class are accompanied with the same semantic representation. We use textual features extracted from corresponding Wikipedia articles. In this section, we introduce notations to represent training and test data.

$t_i^s \in \tau$ and $t_i^u \in \tau$ represent semantic representations of seen and unseen classes where τ is the semantic space. N_s is the number of seen (training) image examples, $x_i^s \in X$ is the visual features of the i^{th} image in the visual space X, and y_i^s is the corresponding category label. Therefore, the seen data (training data) is denoted as $D^s = (x_i^s, y_i^s, t_i^s)_{i=1}^{N^s}$ where we have K^s unique seen class labels. Additionally, we denote the set of seen and unseen class labels as S and U where S and U do not have any labels in common. Therefore, we can formulate the zero-shot learning task as predicting the label $y_u \in U$ of an unseen class sample $x^u \in X$. We can also formulate Generalized ZSL (GZSL) , which is more complicated that ZSL, as predicting the label of $y \in U \cup S$.

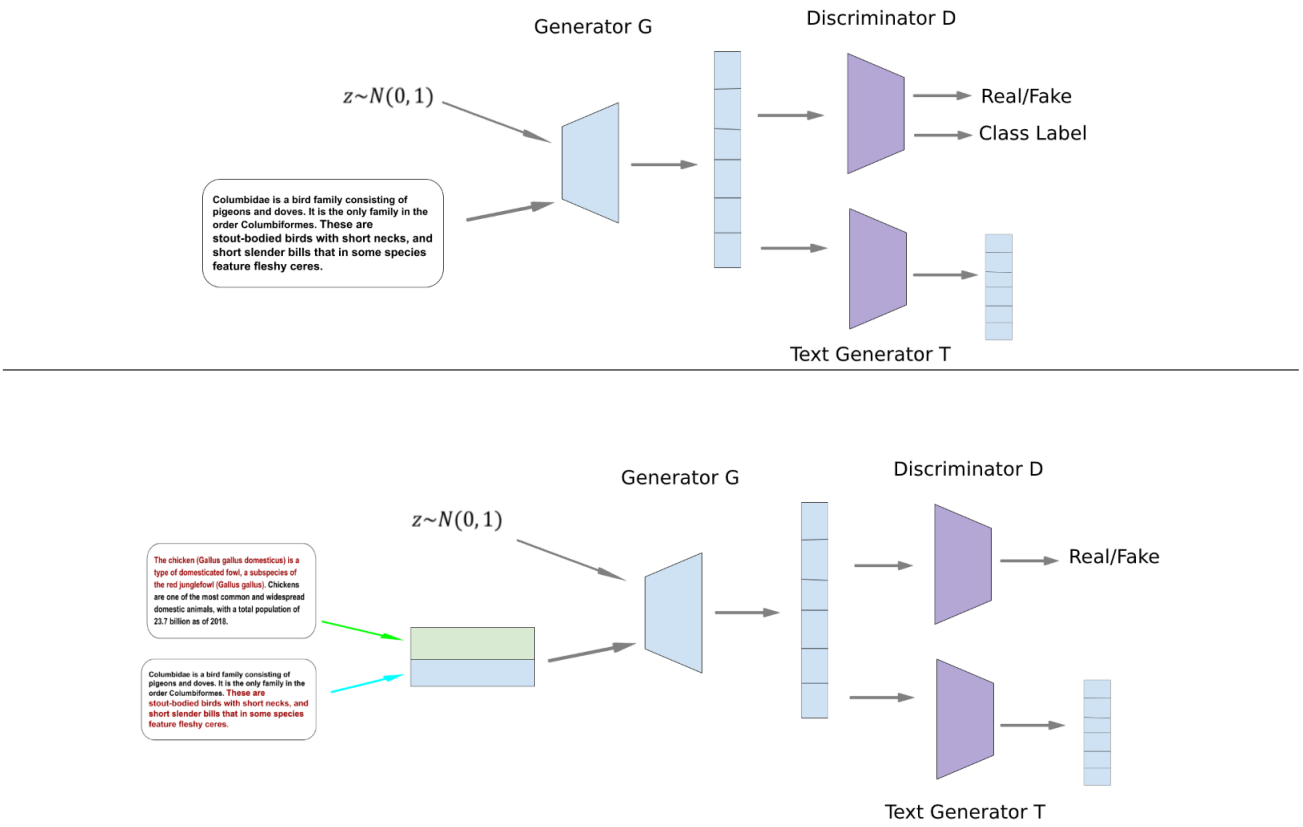

Figure 1. Top image represents the model and its input when train by following the conventional training process of GAN models. Bottom image represents the model and the input data when the model is trained by the hallucinated text introduced in (Elhoseiny & Elfeki, 2019). The second head of the discriminator model which is responsible for classification is not used in this training process. Therefore, it is removed from the graph.

4. Method

Figure 1 shows an overview of our model. Our model is GAN (Goodfellow et al., 2014) based and built on top of the simple GAN model introduced in (Yizhe Zhu & Elgammal, 2018). We define the generator G as $\mathbb{R}^Z \times \mathbb{R}^T \xrightarrow{\theta_G} \mathbb{R}^X$, the discriminator D as $\mathbb{R}^X \xrightarrow{\theta_D} \{0,1\} \times \mathbb{L}_{cls}$, and the text generator T as $\mathbb{R}^X \xrightarrow{\theta_T} \mathbb{R}^T$ where θ_G, θ_D and θ_T are parameter of the three models and $\mathbb{L}_{cls}$ is the list of class labels.

The generator model consists of three fully connected layers with ReLU and Tanh activations. The first layer is responsible for denoising the text representation before concatenating it with z which is sampled from mean zero standard deviation 1 Gaussian distribution. Then it uses the concatenated data to generate sample visual features from the noise and the textual information. Later the discriminator model takes in the generated visual features to not only detect whether the features are real but also to predict the class label. The discriminator has one shared fully connected layer to process the input data. It has two other fully connected layers that each of which is responsible for one of the two discriminator tasks. These two models are from (Yizhe Zhu & Elgammal, 2018) and used without any adjustments and modifications.

Adding a third model called text generator to this already existing model is the main contribution of our research. The text generator is the regularization added to this model to encourages the generator G to pay close attention to the text descriptions. This model has three fully connected layers accompanied with ReLU activations to generate text descriptions from inputted visual features. A loss function is added to penalize this model and the generator G if the generated text description from model T is not similar to the text description inputted to the model G.

There are a few options to measure similarities between text descriptions. Cosine similarity and Mean Squared Error loss functions are the more popular ones. We experimented with both but got better results with cosine similarity because of the structure of our text data which we will talk about more

in section 5.

This loss makes sure the model G generates images that closely represent the inputted textual information. After training these three models we can use the generator G model to sample data from both seen and unseen classes. In the next section, we explain the training process and how we get the model G to learn to generate visual features for unseen classes.

4.1. training

Training this method is challenging because:

1. This method has three models that their behavior and errors can have effects on each other. Therefore, we use the usual GAN training procedure which is training each model separately in each iteration.

2. The training data is from the seen classes. Since the model G does not see any data from unseen classes, it might be too hard for it to "imagine" or generate data for these classes. Therefore, the training process should be modified in a way to allow the model G to explore the unseen area of the class label space implicitly without passing it any samples from unseen classes.

   We employ the training process introduced in (Elhoseiny & Elfeki, 2019). The training process helps the generator G explore its unseen space with new text data referred to as hallucinated text t^h. To construct this hallucinated text we pick two text features from seen classes at random ($t_a^s, t_b^s \in \tau$). Then we use the the weighted sum of these two text features to create the hallucinated text:

   $$t^h = \alpha t_a^s + (1 - \alpha) t_b^s \tag{1}$$

   Where alpha is uniformly sampled from interval [0.2, 0.8] which is the suggested interval by (Elhoseiny & Elfeki, 2019).

   Please note this training process is not meant to replace the regular training process used to train GAN models. One other important note is that the second head of discriminator which is used to classify the visual features is not used in this step of the training process. See 1 for reference.

4.2. Zero-Shot Recognition

Once the models are trained we can use the generator model to create visual features of unseen classes by passing the model a sampled point from Gaussian distribution and the text description of each unseen class. Since there is no limit on the number of generated samples of unseen classes, we can generate enough data to train a classifier to predict labels of visual features from unseen classes. We used the nearest neighbour model but any classifier can be used. Given this setting and the ability to generate arbitrary samples of unseen classes, zero-shot recognition becomes a regular classification task. We later report the accuracy of the nearest neighbour model as the top-1 accuracy for different benchmarks in table 1.

5. Datasets

We use two datasets to train and test our models. The datasets are: *Caltech UCSD Birds-2011* (CUB) (Wah et al., 2011) and *North America Birds* (NAB) (Van Horn et al., 2015). CUB and NAB are both datasets of birds containing 200 and 1011 bird species and 11,788 and 48,562 images, respectively. (Elhoseiny et al., 2017) split both datasets into two groups called *Super-Category-Shared* (SCS) and *Super-Category-Exclusive* (SCE) splits which are commonly referred to as easy and hard splits. *Super-Category-Shared* split is considered to be "easy" because for each unseen class there exist at least one seen class that has the same super-category. On the other hand, *Super-Category-Exclusive* split contains unseen classes that do not share their parent category with any of the seen classes. Therefore, *Super-Category-Shared* group is easier because there are more similarities between unseen and seen classes compared to the seen and unseen classes of *Super-Category-Exclusive* split. From this point forward, we refer to these splits as easy and hard splits.

(Elhoseiny et al., 2017) made another contribution by extending CUB and NAB by adding the Wikipedia article of each class. later (Yizhe Zhu & Elgammal, 2018) processed the articles by tokenizing the articles, removing the stop words, reducing the words to their word stem, and extracting Term Frequency-Inverse Document Frequency(TF-IDF) feature vectors. The dimensionalities of the resulting textual features of CUB and NAB are 7551 and 13217, respectively. We use these feature vectors as text description input to our model.

Additionally, (Yizhe Zhu & Elgammal, 2018) prepossessed the images of both datasets by extracting bird body parts from images by resizing images to 224 × 224 and feeding them into Visual Part Detector/Encoder network (VPDE-net) (Zhang et al., 2016) and extracting activations of the part-based FC layer of VPDE-net. NAB dataset has 6 semantic parts which are "head", "back", "belly", "breast", "tail", and "wing". CUB has one extra part which is "leg". The dimensionalities of the resulting visual features of CUB and NAB are 3584 and 3072, respectively. We also use these extracted visual features input to our model rather that using the images from CUB and NAB directly.

Metric	Top-1 Accuracy (%)				Seen-Unseen AUC (%)			
Dataset	CUB		NAB		CUB		NAB	
Split-mode	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard
CIZSL (Elhoseiny & Elfeki, 2019)	44.6	**14.4**	36.5	**9.3**	39.2	11.9	24.5	6.4
GAZSL (Yizhe Zhu & Elgammal, 2018)	43.7	10.3	35.6	8.6	35.4	8.7	20.4	5.8
Our Model	**45.48**	13.37	**37.60**	9.23	**39.94**	**11.96**	**25.64**	**6.70**

Table 1. Zero-Shot learning results on CUB and NAB datasets

6. Experiments

6.1. metrics

We use two popular ZSL metrics at test time to evaluate ZSL classification performance of our model:

1. **Top-1 unseen class accuracy**: The accuracy of classifying images from unseen classes correctly to one of the unseen class labels. The generator model is used to sample many visual features from unseen classes. Then a classification model such as k nearest neighbor is trained using the generated samples. Later the classification model is used to classify the real images of unseen data. The performance of this classification model in classifying images or visual features of unseen classes to one of the unseen labels is reported as Top-1 unseen class accuracy.

   Since it only uses images (or in our case visual features) of unseen classes at test time, it is considered an incomplete metric. It is more realistic to use data from both seen and unseen classes at test time.

2. **Seen-Unseen Generalized Zero-shot performance with Area under Seen-Unseen curve (Chao et al., 2017)**: This metric, unlike top-1 unseen class accuracy metric, uses data from seen and unseen classes at test time. This metric first plot a curve of seen-unseen accuracy pairs. It plots the curve by sampling from seen and unseen classes using the generator model, classifying the images to the label space, and create test accuracy-pairs of seen and unseen classes. A balancing parameter is used to facilitate the process. After plotting the seen-unseen curve from the accuracy-pairs, we use the method introduced in (Yizhe Zhu & Elgammal, 2018) to calculate the area under this curve which assess the generalizability of our method in class-level text zero-shot recognition.

6.2. ZSL Recognition Results

Table 1 displays the results achieved by our model on CUB and NAB datasets for their easy and hard splits compared to the two baseline models. Our model outperformed other models in majority of benchmarks when top-1 accuracy is used to evaluate the models. Our model displayed an even more impressive performance when see-unseen AUC metric was use managing to outperform other models in all benchmarks showing a great generalization ability compare to two other models.

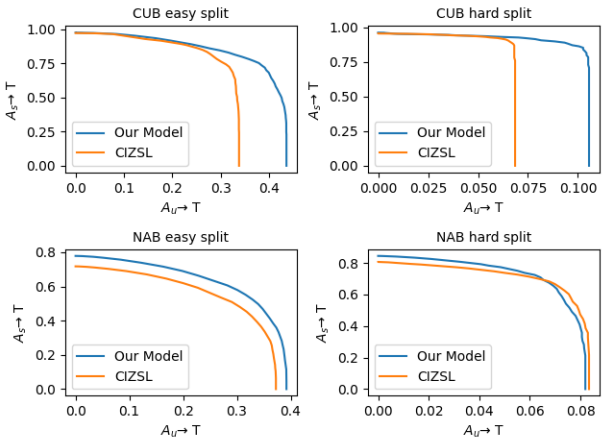

Figure 2. Seen-Unseen accuracy curve with two splits for CUB and NAB datasets

Figure 2 shows the visualization of the seen-unseen curves for our model and CIZSl, which is state-of-the-art model, on all four benchmarks. Our method outperforms CIZSL model and managed to improve the average SU-AUC of easy and hard splits 3.27% and 2.6%, respectively.

6.3. ZSL Convergence speed

CUB and NAB datasets are relatively small datasets compare to other datasets like Imagenet (Deng et al., 2009) that are widely used in computer vision. As a result, the accuracies reported in table 1 for all three models are achieved after training each model for 3000 epochs. However, it is not always possible to train the model for so many epochs if the dataset is too large. Therefore, it is important to understand how fast our model converges in the early epochs. Fig. 3 shows the Seen-Unseen AUC of our model and CIZSL for the first 1500 epochs when trained on CUB easy split. This graph is not to be confused with figure 2. To create Fig. 3, every 100 epochs, we create curves similar to curves

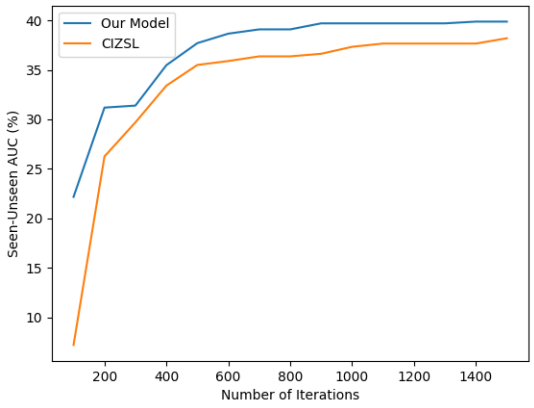

Figure 3. Seen-Unseen accuracies of our model and CIZSL model for the first 1500 training epochs.

in figure 2 and calculate the area under the curves which give us an accuracy. The calculated accuracy is a point in figure 3. This process is repeated 15 times for each model to generate this graph.

Based on the results in table 1, our seen-unseen performance on CUB easy split is 1.9% better than the seen-unseen performance of CIZSL on the same dataset and split. However, based on figure 3, our model achieved the seen-unseen accuracies of 22.17%, 39.1%, and 39.9% after 100, 700, and 1500 iterations versus CIZSL which achieved the seen-unseen accuracies of 7.2%, 36.36%, and 38.19% for the same number of iterations. That is a 207.91%, 7.54%, and 4.48% gap between the accuracies of our model and CIZSL after 100, 700, and 1500 epochs. This shows CIZSL needs more epochs to train. It takes the next 1500 epochs to catch-up and reduce the gap.

6.4. Similarity metrics

There are a few similarity metrics to be uses when comparing the input text description with the description generated by the text generator. We experimented with Cosine similarity and Mean Squared Error (MSE) metrics. Our experiments show that cosine similarity is a better options as the accuracy achieved using this metric is significantly better than the performance achieved by MSE. Additionally, it is shown in literate that cosine similarity is probably best an most commonly used metric for comparing the closeness of two tf-idf vectors (Manning et al., 2008).

7. Conclusion

We introduced a regularizer to encourage the generator to pay close attention to the text data resulting in a better generalisation when tested using SU-AUC metric. paying closer attention to the text resulted in a more realistic ”imagined” or generated visual features why deviating form seen classes. Our experiments shows an improvement on the state-of-the-art model for the task of zero-shot learning and classification. However, more comprehensive evaluation is needed to confirm the abilities of our method. Possible future experiments are:

1. Add our text generator model to other textual Description based ZSL techniques and assess the regularization abilities of our method.

2. Use our generative ZSL model for other closely related tasks such as Zero-Shot Retrieval.

3. Adopt our regularization technique to work with other types of ZSL such as attribute based where classes are accompanied with structured data that describes the classes rather than text data.

References

Akata, Z., Malinowski, M., Fritz, M., and Schiele, B. Multi-cue zero-shot learning with strong supervision, 2016.

Chao, W.-L., Changpinyo, S., Gong, B., and Sha, F. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild, 2017.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Elhoseiny, M. and Elfeki, M. Creativity inspired zero shot learning. In *ICCV*, 2019.

Elhoseiny, M., Zhu, Y., Zhang, H., and Elgammal, A. Link the head to the ”beak”: Zero shot learning from noisy text description at part precision, 2017.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks, 2014.

Guo, Y., Ding, G., Han, J., and Gao, Y. Synthesizing samples for zero-shot learning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 1774–1780, 2017. doi: 10.24963/ijcai.2017/246. URL https://doi.org/10.24963/ijcai.2017/246.

Lampert, C. H., Nickisch, H., and Harmeling, S. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 951–958, 2009. doi: 10.1109/CVPR.2009.5206594.

Long, Y. and Shao, L. Describing unseen classes by exemplars: Zero-shot learning using grouped simile ensemble. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 907–915, 2017. doi: 10.1109/WACV.2017.106.

Manning, C. D., Raghavan, P., and Schütze, H. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK, 2008. ISBN 978-0-521-86571-5. URL http://nlp.stanford.edu/IR-book/information-retrieval-book.html.

Van Horn, G., Branson, S., Farrell, R., Haber, S., Barry, J., Ipeirotis, P., Perona, P., and Belongie, S. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 595–604, 2015. doi: 10.1109/CVPR.2015.7298658.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011.

Yizhe Zhu, Mohamed Elhoseiny, B. L. X. P. and Elgammal, A. A generative adversarial approach for zero-shot learning from noisy texts. In *CVPR*, 2018.

Zhang, H., Xu, T., Elhoseiny, M., Huang, X., Zhang, S., Elgammal, A., and Metaxas, D. Spda-cnn: Unifying semantic part detection and abstraction for fine-grained recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1143–1152, 2016. doi: 10.1109/CVPR.2016.129.