Zero-shot Learning

Udit Sharma
Rochester Institute of Technology
1 Lomb Memorial Drive
us2848@rit.edu

Abstract

Zero-shot learning can be considered a front-runner in pursuit of achieving general artificial intelligence in the object detection space. Zero-shot learning refers to the process in which machine is taught how to recognize objects without any labeled image data shown during training. The model is provided semantic description information which helps it to recognize unseen object classes. This technique is essential in the real world where it is not possible to have annotations of fine-grained classes. This paper explores the domain of zero-shot learning by implementing two different approaches. The first approach uses object attributes as semantic embeddings while the next approach uses word2vec representation of each class as word embeddings. Both approaches use ResNet-50 for image embeddings. The model learns to map attributes found in images to the high-dimensional semantic space and uses these embeddings for prediction during test time.

1. Introduction

Image classification and Object recognition have been attempted in a variety of different ways for more than a decade now. Earlier, people used traditional Machine Learning (ML) and Computer Vision (CV) techniques which did not prove to be as successful. Then, the resurgence of Deep Learning (DL) took the research fraternity by storm in 2012 when AlexNet [12] swept the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) by making a significant reduction in error rate (Figure 1). Since then, DL has dominated the image recognition domain with nearly all entries in the ILSVRC challenge since 2013 using DL based techniques. New DL models keep coming each year who can beat the performance of the previous year. Currently, the best performance is shown by ResNet, a Deep Convolution network with over 100 layers, which has produced a top 5 error rate of 2.25% in the 1000 class classification challenge (Figure 1).

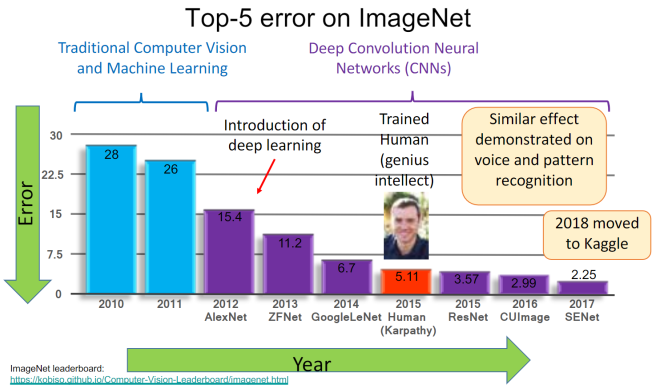

Figure 1. Trend showing decreasing error rate on the ImageNet classification challenge [18]

Most of the classification and detection literature available attempts to perform these tasks with the limitation that the models can only detect classes that it has seen during training. In real-world applications, this limitation makes the models meaningless for practical use as one can't expect the system to just see those 1000 classes in which the model is trained on. This limitation is alleviated with the use of Zero-shot Learning (ZSL). ZSL is a learning approach in which the model learns to identify classes which were unseen in the training set through a connection between semantic and visual representations in seen and unseen classes or description of categories. For example, consider that the class "horse" is present in the training set but the class "Zebra", which closely resembles a horse is not present in training but might be encountered in testing. Now, if we write a description like, “a zebra is like a horse but with white and black stripes”, the model will use its previously learned concepts of stripes, black, white, and horse - and recognize images of zebras from an image database during the test time. Such a situation occurs very frequently in CV applications and will keep occurring as it is practically not possible to have a large amount of training samples for each of the infinite classes that can occur. One approach to Zero-shot Detection (ZSD) is to take the help of attributes of classes and try to link these to other objects which are not present in the training set. This is called attribute-based classification. In this technique, objects are identified based on their high-level description which can be represented as semantic attributes like objects shape and color.

Figure 2. The fact that the Horse and zebra are similar to each other can be exploited using ZSL [20].

Zero-shot learning can also be considered a variation of transfer learning in the sense that the features learned during testing are being extended to classify objects which look familiar to the objects learned during training. Seen and unseen classes are related in a high-dimensional vector space called the semantic space. The basic idea behind Zero-shot learning is to use this semantic space to try to recognize classes that were never seen before.

There are 2 aspects to Zero-shot learning architecture: First is a joint embedding space that consists of visual representations of images that will be mapped to the semantic embeddings. The second aspect is a search from the embedding space for the closest possible attributes to that of the unseen class. The essence remains the same but it is implemented in different ways. This approach is inspired by how humans are able to learn about a new object just by the semantic description of an unseen class. So, this project requires a strong understanding of Computer Vision for feature extraction in the image domain as well as Natural Language Processing.

2. Background

Many attempts have been made to solve the problem of Zero-shot detection. We discuss a few of these works of literature in this section. Among one of the more preliminary attempts to zero-shot detection, Christoph *et al.* [1] focusses heavily on how attributes mapping and transfer of information can be used to recognize an unknown class. They discuss different types of associations that can occur between known and unknown classes and use SVM for learning image features. With the advent of Deep learning techniques later on, it turns out, SVM is not among the best when it comes to object detection or dealing with images in general. Zhang and Saligrama [2] worked on the assumption that both the seen and unseen classes are known and posit that each unseen class can be written as a combination of probabilities of the seen classes. Changpinyo *et al.* [14] uses shallow features like color histograms, PHOG, SIFT and Fisher vectors for image embeddings along with attributes as word embeddings.

Moving on to Deep Learning models, Bernardino Romera-Paredes and Philip H. S. Torr [3] extend the YOLOv2 model for detecting unseen classes calling it the ZS-YOLO to achieve significant improvement in detection accuracies of unseen classes. The motivation of their research was the limitation of current Zero-shot detection techniques to recognize unseen classes even exist in the image. This research also introduces a new dataset called the Animals with Attributes dataset which has proven to be a pioneer when it comes to standardizing research in ZSL. Yang *et al.* [4] developed a fast zero-shot image tagging technique in which they try to draw vectors of each tag in an image and come up with a total principal component for each image. Although this results in inference in constant time, it is limited to the word mapping of 1000 words which was created for this experiment. Zeynep *et al.* [15] train the word embedding from scratch in an unsupervised manner and use state-of-the-art deep features as image embeddings.

Apart from semantic embeddings, knowledge graphs can also be formed and used to map relations between attributes of objects in a graph structure and perform Graph Convolutions in what is known as Graph Convolutional Networks (GCN)[5,6,7].

We demonstrate 2 approaches in this paper, using deep features from the ResNet-50 model, we use attributes and word vectors as embedding space to evaluate which performs better on the AWA2 datasetm[10, 19]

3. Datasets

As per the study conducted for this project till now, We want to consider the 3 datasets discussed below for running our experiments (might not use all of them):

3.1. Animals with Attributes 2 (AWA 2) dataset [10, 19]:

This AWA2 dataset was created in an effort to benchmark all the research occurring in the Zero-shot learning space. This dataset consists of 37322 images from 50 different classes of animals. Apart from these labeled images, the dataset also consists of 85 attributes for each class. These attributes are provided in the form of binary as well as continuous values. Some examples of attributes include the color of the animal, is the animal furry? Is it striped? Does the animal have horns? etc. Figure 3 shows the predicate matrix of each class. Rows represent classes and column the attributes. For example, the first row represents attributes of the class Antelope and some of its attributes depicted in the matrix are that it is furry, has tough skin, is lean in structure while it does not have claws, and does not fly. The dataset also provides a suggested train and test split and we are going to use that suggested split for our experiments. The train split consists of 40 classes and the rest of the 10 classes will be used during training. The split is designed in such a way that the unseen classes are in some way related to any of the seen classes.

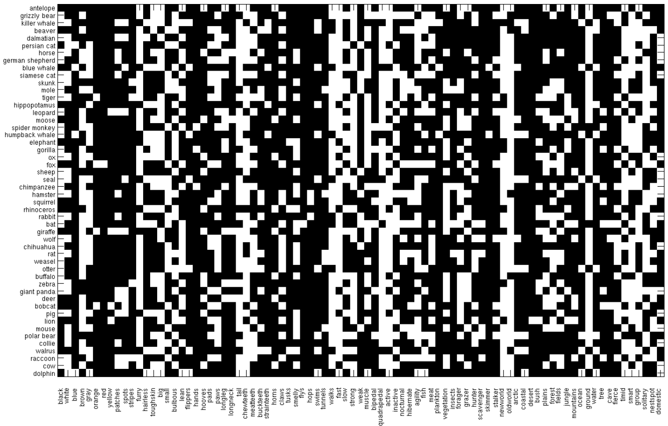

Figure 3. Predicate matrix representing attribute values for each class [10,19]

3.2. Word2Vec:

Everything needs to be converted to numbers for the machine to understand it. Tomas *et al*. [11] found a way around this problem with a technique called Word2Vec. Word2Vec is a 2 layer neural network that takes in a corpus of words as inputs and outputs a vector representation of words as outputs. This made each word comprehensible to the computer. Further, these vectors formed in the way that their semantic similarity was retained. As seen in Figure 4, 'Performance' and 'Success' lie close together as they are correlated with each other in real life. In this paper, we use the pre-trained vectors trained on a part of Google News dataset which is about 100 billion words. This model consists of 300-dimensional vectors from 3 million words and phrases. This pre-trained model uses the skip-gram architecture mode details of which can be found in [11]. We extract the vectors corresponding to our classes and use these vectors as our embedding space. Given the image, the model learns to map the 300-dimensional vector on the embedding space.

4. Proposed Methods

One of the biggest challenges in this approach when it comes to detection is that when the model has not seen those unseen classes during training, how is the model going to recognize that unseen class during testing? How to make sure the model doesn't consider the object as background?

member
national
family
school
war
meeting
wednesday
hold
head
united
leave
office
aid
write
support
london
friday
march
return
child
join
international
uk
act
test
also
role
back
break
appear
man
ahead
club
director
british
book
nation
england
receive
woman
point
side
manager
france
ireland
chelsea
among
film
actor
star
least
rugby
include
award
good
time
come
lose
match
goal
team
within
minute
season
league
chance
show
movie
name
prize
six
race
feature
song
band
winner
end
mark
olympic
william
performance
event
saturday
success
half
past
list
american
lead
city
sunday
album
reach
top
february
single
drop
third
place

Figure 4. Words in the vector space. [20]

To get around this problem, we demonstrate two approaches as discussed later on in this section. These approaches use bottom-up architecture to detect the presence of classes in an image. Traditional neural network-based image classification and object detection models are trained to classify or detect the images as a whole using intensity, color, and textual features. This is not possible for our problem as test classes are not present during training. We posit that an object is made of attributes and the model's responsibility is to look for those attributes in the image instead of the whole object. Therefore the model tries to learn the features in the image to the embedding space.

4.1. Attributes as the embedding space

When working with the problem of zero-shot learning, we need to have an intermediate embedding space which creates a mapping between the seen and unseen classes. In this section, we explore the approach of using attributes present in the AWA2 dataset as the intermediate space as shown in figure 2.

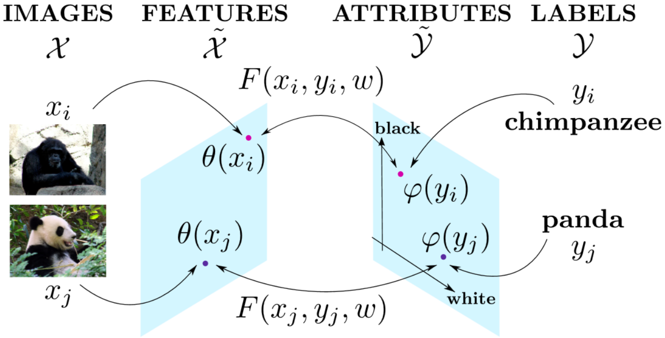

Figure 5. Features extracted from the image beddings are mapped to the attribute space [10]

Resnet-50 is be used as the CNN to extract features which form the image embeddings. This image embedding is then be mapped to the semantic embedding space where each class, be it seen or unseen. will be represented by a high dimensional vector consisting of attributes. Each attribute is a binary value. The learning will take place at an attribute level. During testing time, the classes that CNN sees were not present in training but the attributes were. The attribute values will be determined from the features extracted once the attribute values are obtained, we assign the image to the label corresponding to the closest vector in the high dimensional semantic space. Euclidean distance is used for distance calculation and since the attribute values are binary, binary cross-entropy loss is used for learning. The formula for Binary Cross-Entropy loss is as below.

$$BCELoss = (1-y)log(1-p) - (ylog(p))$$

Here, y is 1 if the predicted class is the correct class and p is the predicted probability of the class.

4.2. Word vectors as the embedding space

The approach in the previous section used attributes as the embedding space. Attribute annotations are also not easily available. Towards the pursuit of generalized Zero-shot learning, this approach uses word vectors as embeddings. The google word2vec [11] vector representations that are being used in this approach consists of 1.5 million word vectors. When trained properly, this model can recognize a lot more classes than the previous approach. Each element in the vector is a floating-point number ranging from -1 to 1. Therefore, contrasting with the previous approach when attribute values were binary, here, the embedding space consists of continuous-valued vectors. We use the Mean Absolute Error (MAE) loss for this approach instead of Binary cross-entropy loss we used for the previous approach. Further, cosine distance will be used as the distance metric as it is a more commonly used metric when dealing with word vectors. For image embeddings, we continue to use the ResNet-50 CNN which we used in the previous approach.

5. Results

One of the challenges in this project is that model needs to be trained from scratch. Using a pre-trained model risks training on classes that are supposed to be unseen for our zero-shot learning model. This could defeat the whole purpose of zero-shot learning. Now, training a model from scratch requires huge computation resources for a long period which are not easy to find. These resources play a significant role in getting state-of-the-art results. Due to the limited resources available to us, we experiment with 2 variations of each approach. One with the full dataset and a limited number of epochs and the other using two-thirds of the data and more number of epochs. None of the classes being used in training will be used in testing and vice versa.

Common preprocessing: As mentioned earlier, we use the Animals with Attributes 2 dataset for our experiments. As part of Data augmentation, We introduce a random rotation of 15 degrees, horizontal flip, vary brightness and contrast of each image randomly by a factor of 0.3 and finally resize the image to 224 x 224 to make it suitable for the ResNet-50 CNN model.

5.1. Attributes as the embedding space

Due to the limited availability of GPU, we were able to run a maximum of 25 epochs when using a full dataset with augmentation. When using the attribute as embedding space, the model is expected to learn 85 values per class. Since each class is created from a combination of these 85 classes, the model potentially learns 4250 values in total. This makes zeros-shot learning models more prone to errors and consequently lower accuracy. Using attributes as the embedding space, We were able to achieve the best results on a batch size of 24 and a learning rate of 0.005. The mean per-class accuracy on the test classes is reported to be 43.4%. These results do beat a few results reported by past researches as can be seen in table 1. The results can be further improved by tuning the attributes that are being used and building a metric to judge the importance of each attribute. Contrary to the experiment with the full dataset, the experiment in which more epochs were run with smaller data did not obtain better results. We can report a top-1 accuracy of 38.7%. In this approach, we set a cap of 400 images randomly chosen from each class. This resulted in a training set of 21670 images down from 30130 in the full dataset. This might have occurred because the model is looking at less variety and more epochs do not let it learn much. Figure 6 shows the confusion matrix for the case when the full dataset was used. The model predicts the class 'rat' more often than others which indicates that the model has not trained properly yet and needs more epoch. Given the nature of the problem, the model performs decently and beats results of much similar research some of which are listed in Table 1. The current approach uses binary attribute values. Better results could be achieved if continuous attribute values are used.

5.2. Word vectors as the embedding space

Google's pre-trained model was used to extract word vectors of class labels. This model provides a 300-dimensional vector consisting of each word. As mentioned earlier, since, these are floating-point values, we can't use the binary cross-entropy loss. We decide to use the L1 loss.

As can be seen from the equation, L1 loss is simply the sum of absolute errors.

Since we are training for a 300-dimensional vector for each class, this is a difficult regression task. We trained this model for 25 epochs and as can be seen in figure 7, the model was still training. After 25 epochs, we can report a testing accuracy of 35.7 %. The model shows promise

Table 1. Summary of results

Model	**Embedding**	**Mean per-class Accuracy**
DAP[13]	Binary attributes	41.4%
SJE[15]	Binary attributes	42.3%
BN[16]	Continuous attributes	60.5 %
Ours (Full data)	**Binary Attributes**	**43.4%**
Ours (More epochs)	Binary Attributes	38.7%
SYNC[14]	Word2Vec (100-D)	42.6%
SYNC	Word2Vec (1000-D)	57.6%
Ours (Full data)	**Word2vec (300-D)**	**35.7%**

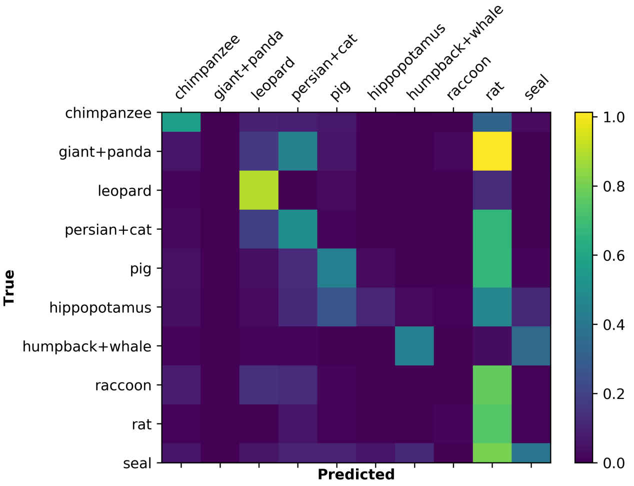

Figure 6. Confusion matrix on test classes with attributes as embeddings and full dataset.

$$L1LossFunction = \sum_{i=1}^{n} |y_{true} - y_{predicted}|$$

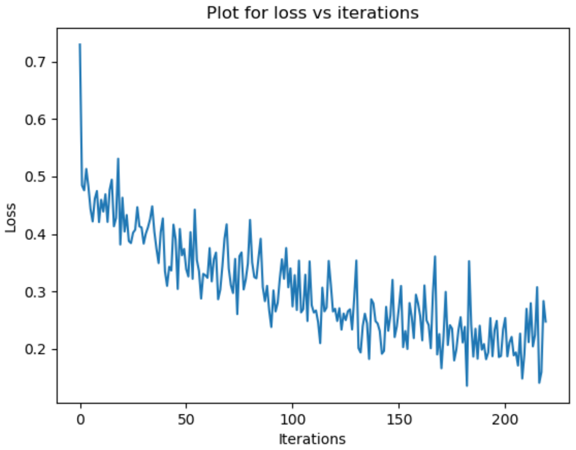

Figure 7. Change in training loss vs. iterations with word vectors as embeddings

and is bound to improve if more epochs are allowed. Another reason for lower performance could be the nature of the dataset combined with the pre-trained embeddings. The dataset consists of all animal classes. The embedding that we use is generic and it places all animal vectors close to each other which make it harder for the model to differentiate between animal classes. The confusion matrix for this approach is shown in figure 8.

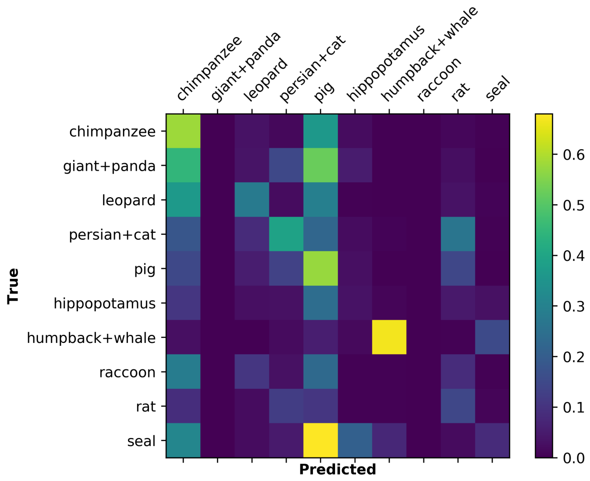

Figure 8. Confusion matrix on test classes with word vectors as embeddings.

6. Conclusion

This paper demonstrates two approaches to achieve zero-shot learning. The first approach involved attributes provided by the AWA2 dataset. This approach achieved state-of-the-art results when compared to other researches using attributes and similar evaluation setting. The other approach which employs word vectors as the embedding space could obtain an accuracy of 35.7%. We attribute this lower accuracy to computation requirements and pre-trained word embeddings which can be considered as part of the future work. Given more computation resources, improvements can also be made by using Generative Adversarial Networks for generating unseen class images from seen classes using semantic representations. Another approach could be to represent semantic space as graphs to perform Graph convolutions on semantic space to improve the robustness of the model. We believe that this project is a stepping stone for us to building a completely novel architecture which achieves state-of-the-art results in the complete ZSL space.

References

[1] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. *”Attribute-Based Classification for Zero-Shot Visual Object Categorization”*. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 36, NO. 3, MARCH 2014.

[2] Z. Zhang and V. Saligrama. *“Zero-shot learning via semantic similarity embedding,”*. ICCV, 2015.

[3] Bernardino Romera-Paredes and Philip H. S. Torr. *”An embarrassingly simple approach to zero-shot learning”*. 32nd International Conference on Machine Learning (ICML), Lille, France, 2015.

[4] Yang Zhang, Boqing Gong and Mubarak Shah. *”Fast Zero-Shot Image Tagging”*. Conference on Computer Vision and Pattern Recognition, 2018.

[5] Xiaolong Wang, Yufei Ye and Abhinav Gupta. *”Zero-shot Recognition via Semantic Embeddings and Knowledge Graphs”*. Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[6] Michael Kampffmeyer, Yinbo Chen, Xiaodan Liang, Hao Wang, Yujia Zhang and Eric P. Xing. *”Rethinking Knowledge Graph Propagation for Zero-Shot Learning”*. Conference on Computer Vision and Pattern Recognition, 2018.

[7] Thomas N. Kipf and Max Welling. *“Semi-Supervised Classification with Graph Convolutional Networks”*. ICLR, 2017.

[8] Pengkai Zhu, Hanxiao Wang and Venkatesh Saligrama. *”Zero Shot Detection”*. IEEE Transactions on Circuits and Systems for Video Technology (TCSVT), 2019.

[9] Li Zhang, Tao Xiang and Shaogang Gong. *“Learning a Deep Embedding Model for Zero-Shot Learning”*. Conference on Computer Vision and Pattern Recognition, 2017.

[10] Yongqin Xian, Christoph H. Lampert, Bernt Schiele and Zeynep Akata. *“Zero-Shot Learning - A Comprehensive Evaluation of the Good, the Bad and the Ugly”*. Conference on Computer Vision and Pattern Recognition, 2017.

[11] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. *Distributed Representations of Words and Phrases and their Compositionality.* In Proceedings of NIPS, 2013.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton *Imagenet classification with deep convolutional neural networks.* In NIPS, 2012.

[13] C. H. Lampert, H. Nickisch, and S. Harmeling. *Attributebased classification for zero-shot visual object categorization.* TPAMI, 36(3):453–465, 2014.

[14] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha. *“Synthesized classifiers for zero-shot learning”* in CVPR, 2016.

[15] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. *Evaluation of output embeddings for fine-grained image classification.* In CVPR, 2015

[16] X. Wang and Q. Ji. *A unified probabilistic approach modeling relationships between attributes and objects.* In ICCV, 2013.

[17] X. Wang and Q. Ji. *A unified probabilistic approach modeling relationships between attributes and objects.* In ICCV, 2013.

[18] ImageNet, `http://image-net.org/index`

[19] Animals with Attributes 2 Dataset, `https://cvml.ist.ac.at/AwA2/`

[20] Reference to image about word vectors, `https://towardsdatascience.com/understanding-wor`

[21] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, Cordelia Schmid. Label-Embedding for AttributeBased Classification. IEEE Computer Vision and Pattern Recognition (CVPR), IEEE, Jun 2013, Portland, United States. ffhal-00815747v1f