Zero-Shot Object Recognition Using Semantic Label Vectors

Shujon Naha and Yang Wang
Department of Computer Science
University of Manitoba
Winnipeg, MB, Canada
{*shujon, ywang*}*@cs.umanitoba.ca*

***Abstract*—We consider the problem of zero-shot recognition of object categories from images. Given a set of object categories (called "known classes") with training images, our goal is to learn a system to recognize another non-overlapping set of object categories (called "unknown classes") for which there are no training images. Our proposed approach exploits the recent work in natural language processing which has produced vector representations of words. Using the vector representations of object classes, we develop a method for transferring the appearance models from known object classes to unknown object classes. Our experimental results on three benchmark datasets show that our proposed method outperforms other competing approaches.**

***Keywords*-object recognition; zero-shot learning; transfer learning**

I. Introduction

Visual object recognition is a cornerstone problem in computer vision. The standard approach is to formulate the object recognition as a classification problem. Given an input image, the goal is to predict the label of this image from a set of predefined category labels. Object recognition systems are usually trained using machine learning techniques. In order to achieve good classification performance, they usually require a large amount of labeled training data.

It has been estimated that humans can recognize between 5,000 and 30,000 object categories [1]. Collecting training images for all these object categories is tedious and expensive. Therefore, various techniques for reducing the number of training images have been proposed. Humans are known to be capable to learn a new object category from a small number (2 or 3) images [2]. They can learn completely unseen classes purely from a high-level description without any training images. This is known as *zero-shot* object recognition. In computer vision, there has been work on using "attributes" as an intermediate layer for zero-shot object recognition. These work first learn classifiers to predict the attribute labels using the training images from known classes. Then these attribute classifiers can be used to recognize completely unseen object categories [3], [4].

The limitation of attribute-based approaches is that the attributes have to be manually defined. Farhadi et al. [3] manually define 64 visual attributes and use crowd-sourcing to obtain the ground-truth attribute annotations for images. Lampert et al. [4] use the data collected in the cognitive science literature [5] to define 85 attributes for 50 animal classes. Rohrbach et al. [6] try to extract class-attribute relations by mining online resources. But the attributes are limited to "part attributes" and it is not clear how to generalize their approach to other attributes or object classes.

In this paper, we propose a new approach for zero-shot object recognition. We assume that each object class (either known or unknown) can be represented as a fixed-length vector, which we call the *semantic label vector*. If two objects (e.g. "cat" and "dog") are semantically close, their corresponding semantic label vectors tend to be close as well. Attribute-based representation can be considered as a special case of the semantic label vector. However, our approach is not limited to attribute vectors. In the natural language processing community has produced vector representations of words by analyzing large collections of text documents. Our approach can be used together with these word vectors as well. The advantage of using word vectors as the semantic label vectors is that these word vectors can be obtained automatically from large collections of text documents, so we do not have to define them manually. In computer vision, these word vectors have been used in object recognition [7], image-sentence mapping [8], etc.

Problem Statement: We assume that there are K known object classes and L unknown object classes. There is no overlap between known and unknown object classes. We have training images only for the K known object classes. Each object class (either known or unknown) is associated with a semantic label vector. If two object classes are semantically close, their semantic label vectors tend to be close. We will discuss how to get the semantic label vectors in II-A. During testing, we are given an image from one of the L unknown classes. Our goal is to predict the class label of this image. Note that since we do not have training images for unknown classes, this problem cannot be solved using traditional supervised learning approaches.

II. Our Approach

The overview of our approach is summarized in Fig. 1. For each object category (either known or unknown), we assume that we have a vector representation of this category. This vector representation can be obtained via crowd-sourcing, or automatically from linguistic data. In

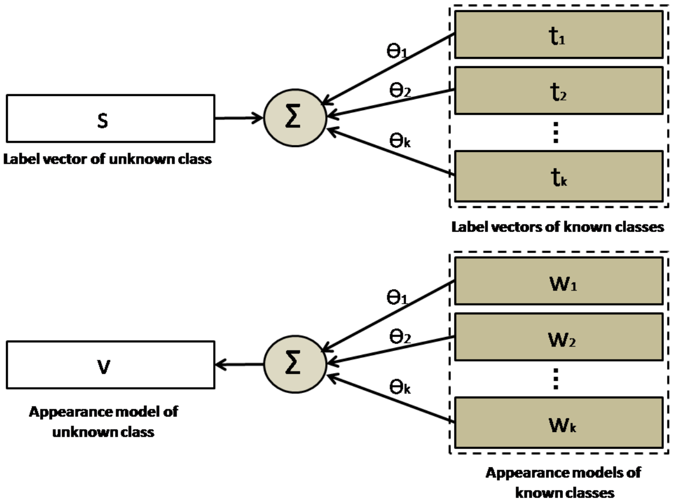

θ1
θ2
θk
t1
t2
⋮
tk
s
Σ
Label vector of unknown class
Label vectors of known classes
θ1
θ2
θk
w1
w2
⋮
wk
v
Σ
Appearance model of unknown class
Appearance models of known classes

Figure 1. An overview of our approach. (Top) We represent the word vector $\mathbf{s}$ of an unknown object class as a sparse linear combination of the word vectors of known objects $\mathbf{t}_1$, $\mathbf{t}_2$,...,$\mathbf{t}_K$. The coefficients of this linear combination are θ_1, θ_2,...,θ_K. (Bottom) We use the same coefficients to represent the appearance model $\mathbf{v}$ of the unknown object as the linear combination of the appearance models of known objects $\mathbf{w}_1$, $\mathbf{w}_2$,...,$\mathbf{w}_K$. Then we can use the appearance model $\mathbf{v}$ for recognition.

skunk:
black, vegetation, paws, furry, stripes, tail
buffalo:
black, vegetation, strong, furry, tough skin, big, hooves
lion:
brown, furry, big, strong, furry, paws, smelly

Figure 2. Examples of attributes for three animal classes: skunk, buffalo, and lion.

Section II-A, we give details of how to obtain the word vectors. For an unknown object category, we use the vector representation to capture the semantic relatedness of this object category to all the known object classes. In this paper, we choose to represent the unknown object as a sparse linear combination of known objects. For each known object, we can learn its appearance model since we have the training data. Then we transfer the appearance of the known objects to the unknown object based on their semantic relatedness. Finally, we use the transferred appearance models for the unknown objects for prediction. Our approach is closely related to [9]. The method in [9] deals with localizing unseen objects in weakly labeled images or videos, while our work focuses on recognizing unseen objects.

A. *Semantic Label Vector*

We assume that we have access to a vector representation of an object class, which we call “semantic label vector”. The label vectors capture the semantic knowledge about objects. Ideally, if two objects (e.g. “cat” and “dog”) are semantically similar, the corresponding label vectors will be close. In this paper, we consider two different types of semantic label vectors.

1) Attribute vectors: In computer vision, attributes have been proposed to capture high-level concepts related to objects. For example, Fig. 2 shows examples of attributes of some object classes. The attributes can be defined either per-image (e.g. [3]) or per-class (e.g. [4]). In this paper, we consider attributes on a per-class basis. In other words, each object class is associated with a vector describing the presence/absence of each attribute in the object category. The attribute vector for an object category can be manually defined. In some cases, they can be obtained from other sources. For example, Lampert et al. [4] use the data collected in cognitive science research [5] to define the attribute vectors for animals.

2) Word vectors: The limitation of attribute vectors is that they are available only for certain object classes provided by some datasets. An alternative is to use the word semantic knowledge available from the natural language processing (NLP) community. Recent work in NLP has produced valuable resources on word semantic by analyzing large collections of text documents. For example, a word is represented as a fixed length vector in [10]. If two words (e.g. “cat” and “dog”) are semantically close, the distance of their word vectors tend to be small. Figure 3 shows a visualization of the word vectors by projecting them on 2D using t-SNE [11].

B. *Unknown Object as Sparse Reconstruction*

Now we have a vector representation for each object class. In this section, we will describe how to represent an unknown object as a linear combination of known objects based on the label vectors. This will give us the semantic relatedness of the unknown object and known objects. In II-C, we will use this semantic relatedness to transfer the appearance model from known objects to an unknown object.

We denote the label vectors of the K known objects as $\mathbf{t}_k$ ($k = 1, 2, ..., K$). Let $\mathbf{s}$ be the label vector of an unknown object class, we assume that $\mathbf{s}$ be approximated by a convex combination of the label vectors of the K known objects,

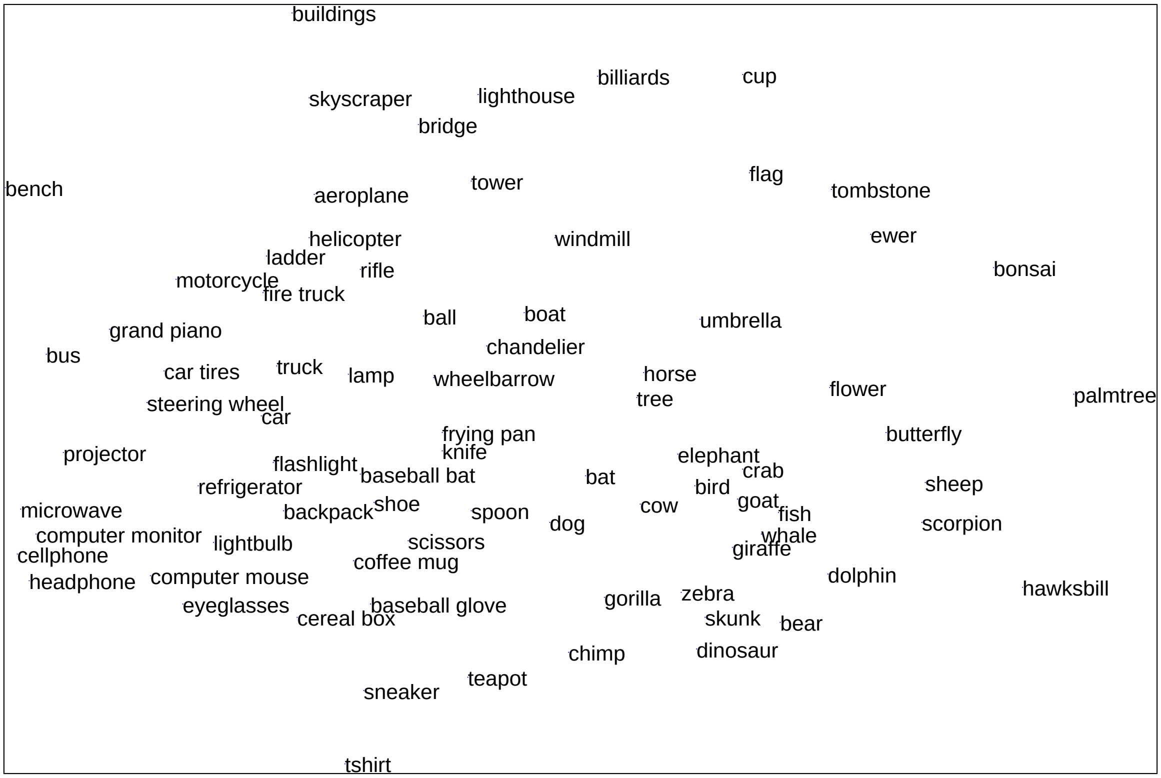


Figure 3. Visualization of word vectors in 2D. The 2D embedding of the word vectors is obtained using the t-SNE algorithm [11]. From the visualization, we can see that semantically similar words tend to be close in terms of their word vectors.

i.e.

$$\mathbf{s} \approx \theta_1 \mathbf{t}_1 + \theta_2 \mathbf{t}_2 + ... + \theta_K \mathbf{t}_K \tag{1}$$

$$\text{where} \quad \theta_k \geq 0, \quad k = 1, 2, ..., K \tag{2}$$

We can estimate the coefficients $\Theta = [\theta_1, \theta_2, ..., \theta_K]$ by solving an optimization problem similar to sparse coding [12].

$$\min_{\Theta \geq 0} || \sum_{k=1}^{K} \theta_k \mathbf{t}_k - \mathbf{s} ||_2^2 + \lambda ||\Theta||_1 \tag{3}$$

The first term in Eq. 3 is the reconstruction error. The second term in Eq. 3 is a L_1 regularization that encourages the solution to be sparse, i.e. we would like to approximate an unknown object with only a small number of known objects. The parameter λ controls the trade-off between the reconstruction error and the regularization.

C. Appearance Transfer

We now describe how to use the coefficients Θ obtained in Eq. 3 to transfer the appearance models from the K known object classes to an unknown object class.

Let $\mathbf{w}_k$ represent the appearance model of the k-th familiar object. Given the feature vector $\mathbf{x}$ of an image, we use the linear model $f_k(\mathbf{x}) = \mathbf{w}_k^\top \mathbf{x}$ as the score of predicting the the image as the k-th known object. Since we have training data for the known objects, we can obtain their appearance models $\mathbf{w}_k$ $(k = 1, 2, ..., K)$ using standard supervised learning approaches. In this paper, we use a linear SVM to learn the appearance models $\mathbf{w}_k$ $(k = 1, 2, ..., K)$.

Since we do not have training data for any unknown object class, we cannot directly learn its appearance model using standard supervised learning techniques. Instead, we will construct the appearance model of an unknown object class by transferring the appearance models of known object classes. Our main assumption is that the label vectors and appearance models of objects are related in similar ways. In other words, we can use the coefficients Θ in Eq. 3 to represent the appearance model $\mathbf{v}$ of an unknown object class as:

$$\mathbf{v} \approx \sum_{k=1}^{K} \theta_k \mathbf{w}_k \tag{4}$$

In Eq. 4, the coefficients θ_k $(k = 1, 2, ..., K)$ are obtained using the label vectors (see Eq. 3). So as long as we have a vector representation of object classes, we can use Eq. 4 to transfer appearance models from known objects to an unknown object.

D. Recognizing Novel Objects

Suppose we have L unknown object classes. For each unknown object class, we obtain its appearance model $\mathbf{v}_i$

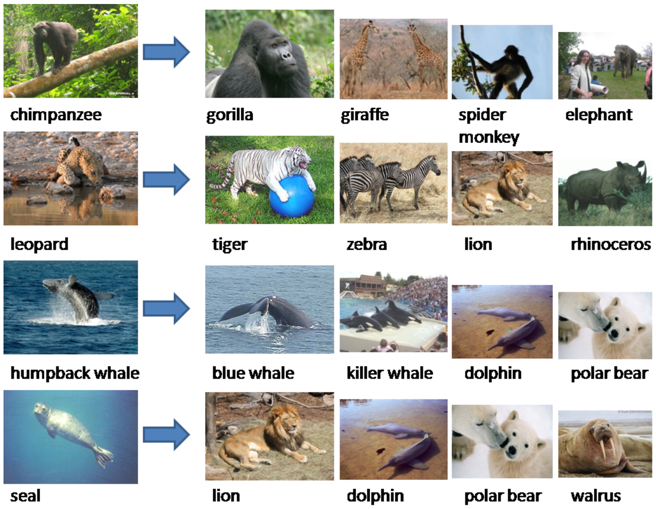


Figure 4. Examples of word vector distances. In each row, we show an object class and the most similar four object classes according to the word vector distances.

($i = 1, 2, ..., L$) using the method in Section II-C. Given an image $\mathbf{x}$ that belongs to one of the L unknown object classes, we can simply predict the label y for this image by choosing the appearance model that gives the maximum score, i.e.:

$$y = \arg\max_i \mathbf{v}_i^\top \mathbf{x} \tag{5}$$

III. Experiments

We evaluate our approach on three benchmark datasets: animal dataset [4], object attribute dataset [3], and a subset (112 object classes) of the ImageNet [13]. Since the name of an object class can be a phrase (e.g. "giant panda"), we use Google's word2phrase tool [10] to pre-process the training text data when generating the word vectors. It allows to generate vectors for phrases like "giant panda". In the end, we generate the word vectors for all object classes.

For comparison, we define several baseline approaches. Since we have training images for the known classes, we can learn a multiclass SVM classifier to predict the label as one of the known classes. For a given image from one of the unknown classes, we first use the learned SVM classifier to predict one of the known classes. We then pick the unknown class that is most similar to the predicted known class. We define the following three different ways of measuring the similarity of two object classes. Each of them gives a baseline approach.

Word vector distance: this method measures the distance of two object classes using the L_2 distance of their corresponding word vectors. A smaller distance means that the two object classes are more similar. Fig. 4 shows some examples of several object classes and the most similar object classes according to the word vector distances.

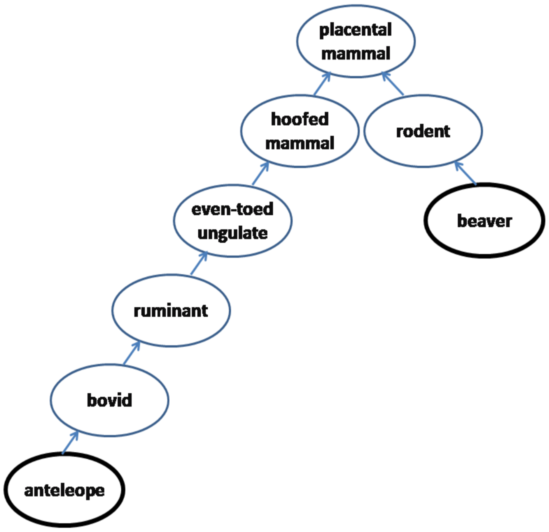


Figure 5. An illustration of how to compute the WordNet distance. For two objects ("anteleope" and "beaver") in the WordNet hierarchy, we use the length of the path between them as the distance measurement. In this case, the distance between these two objects is 7.

WordNet distance: this method measures the distance of two object classes by considering their distance in the WordNet hierarchy [14]. Fig. 5 illustrates how to compute the WordNet distance of two object classes ("anteleope" and "beaver"). Note that the distance is always an integer value in this case. For a given object class, there might be multiple unknown classes that have the minimum distance according to this distance measurement. In this case, we simply assume that the final prediction is achieved by randomly picking an unknown class that has the minimum distance. Given a test image, if there are T unknown classes that have the minimum distance and the ground-truth class is one of them, we consider this test image to be $1/T$ correct.

Attribute distance: this method measures the distance of two object classes using the L_2 distance of the attribute vectors of these two classes. Fig. 6 shows some examples of the object classes and the most similar object classes according to the attribute vector distances.

A. Animal dataset

This dataset contains over 30,000 animal images of 50 classes. Each class is associated with 85 binary attributes. These attributes are obtained from the cognitive science literature [5]. Figure 7 visualizes the resulting 50×85 class-attribute matrix.

Following [4], 40 animal classes are used as the known classes and the remaining 10 used as the unknown classes. We use Caffe [15] to extract the image features on this dataset. The Caffe feature representation has been shown to be effective in many object recognition tasks.

In Table I, we compare our approach (using both attribute vector and word vector) with the three baselines. We also

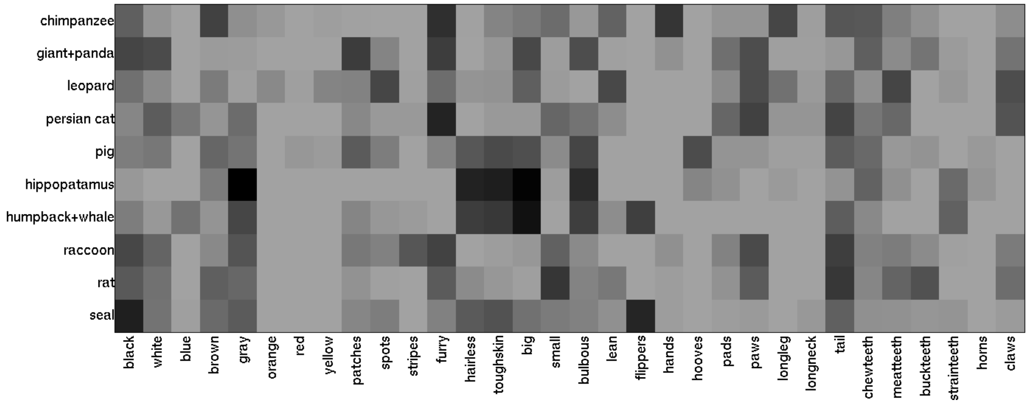


Figure 7. Visualization of the class-attribute matrix on the animal dataset. Darker boxes mean stronger associated between an attribute and a class. Binary attributes are obtained by thresholding the values in the matrix.

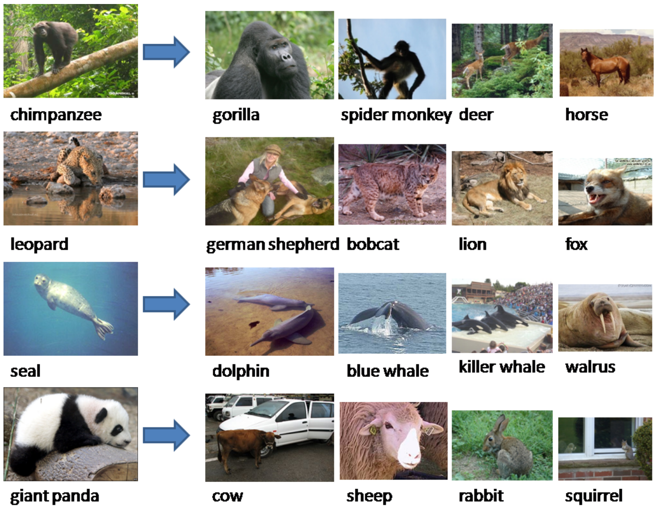
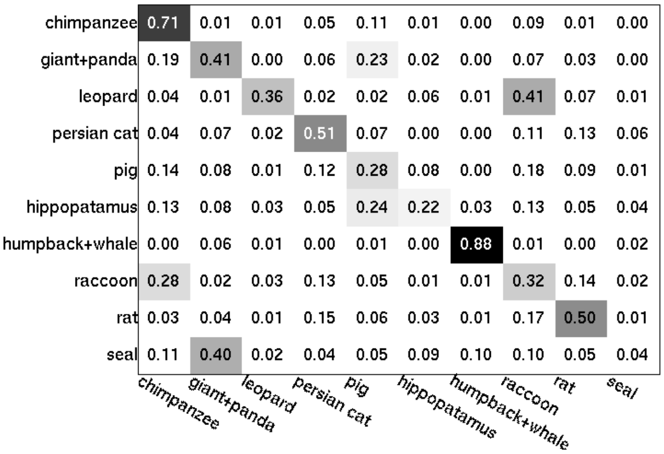


Figure 6. Examples of attribute vector distances. In each row, we show an object class and the most similar four object classes according to the attribute vector distances.

Table I
COMPARISON OF OUR APPROACH WITH SEVERAL BASELINE METHODS ON THE ANIMAL DATASET.

method	accuracy(%)
our approach (attribute vector)	**46.21**
our approach (word vector)	**43.88**
word vector distance	38.38
WordNet distance	35.6
attribute distance	35.59
Lampert et al. [4]	42.2
Rohrbach et al. [16]	42.7

compare with previous published work in [4], [16]. The comparison shows that our proposed method outperforms the competing approaches. In Fig. 8, we visualize the confusion matrix of our approach with word vectors on this dataset.



	chimpanzee	giant+panda	leopard	persian cat	pig	hippopatamus	humpback+whale	raccoon	rat	seal
chimpanzee	0.71	0.01	0.01	0.05	0.11	0.01	0.00	0.09	0.01	0.00
giant+panda	0.19	0.41	0.00	0.06	0.23	0.02	0.00	0.07	0.03	0.00
leopard	0.04	0.01	0.36	0.02	0.02	0.06	0.01	0.41	0.07	0.01
persian cat	0.04	0.07	0.02	0.51	0.07	0.00	0.00	0.11	0.13	0.06
pig	0.14	0.08	0.01	0.12	0.28	0.08	0.00	0.18	0.09	0.01
hippopatamus	0.13	0.08	0.03	0.05	0.24	0.22	0.03	0.13	0.05	0.04
humpback+whale	0.00	0.06	0.01	0.00	0.01	0.00	0.88	0.01	0.00	0.02
raccoon	0.28	0.02	0.03	0.13	0.05	0.01	0.01	0.32	0.14	0.02
rat	0.03	0.04	0.01	0.15	0.06	0.03	0.01	0.17	0.50	0.01
seal	0.11	0.40	0.02	0.04	0.05	0.09	0.10	0.10	0.05	0.04

Figure 8. Confusion matrix of our approach with word vectors on the animal dataset. Each row corresponds to a ground-truth label and each column corresponds to a prediction. The (i, j) entry of the confusion matrix shows the percentage of images from class i that are classified as class j.

B. *Object attribute dataset*

This dataset contains images of 20 known object classes and 12 unknown classes. On this dataset, the attributes are annotated on the per-image level. Each image is annotated with 64 binary attributes. In order to obtain the attribute annotation for a class, we simply take the average of the attribute vectors of all images in this class. Figure 9 visualizes the class-attribute matrix on this dataset.

The comparison of our approach and the baselines is shown in Table II. Again, our method outperforms the baseline approaches. We visualize the confusion matrix of our approach with word vectors on this dataset in Fig. 10.

C. *ImageNet-112 dataset*

This dataset is collected in [13] and is a subset of the ImageNet [14]. It contains images from 112 object classes. We consider 76 of them as the known classes and the

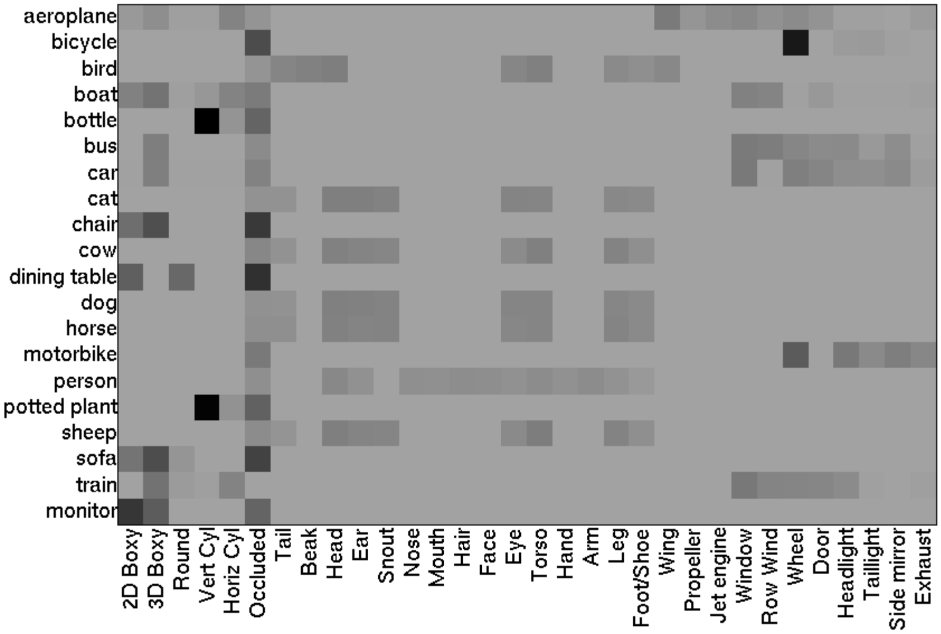


Figure 9. Visualization of the class-attribute matrix of the object attribute dataset.

Table II
COMPARISON OF OUR APPROACH WITH SEVERAL BASELINE METHODS ON THE OBJECT ATTRIBUTE DATASET.

method	accuracy(%)
our approach (attribute vector)	**30**
our approach (word vector)	**25**
word vector distance	23.84
WordNet distance	18.21
attribute distance	18.79

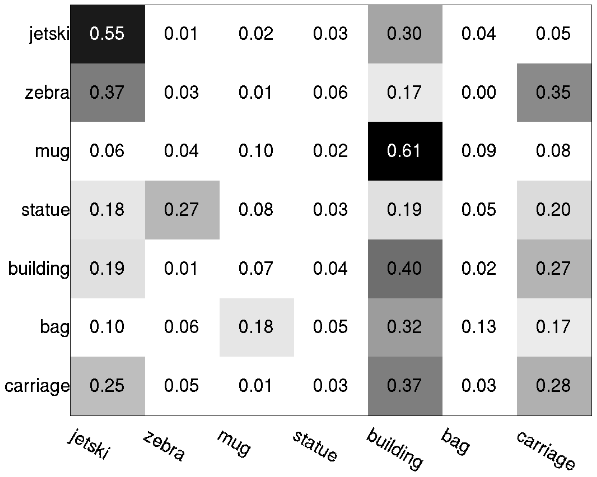


Figure 10. Confusion matrix of our approach with word vectors on the object attribute dataset.

Table III
COMPARISON OF OUR APPROACH WITH SEVERAL BASELINE METHODS ON THE IMAGENET-112 DATASET.

method	accuracy(%)
our approach (word vector)	**28.23**
word vector distance	24.36
WordNet distance	19.2

remaining 36 classes as the unknown classes. Similarly, we use the Caffe [15] feature to represent each image in this dataset.

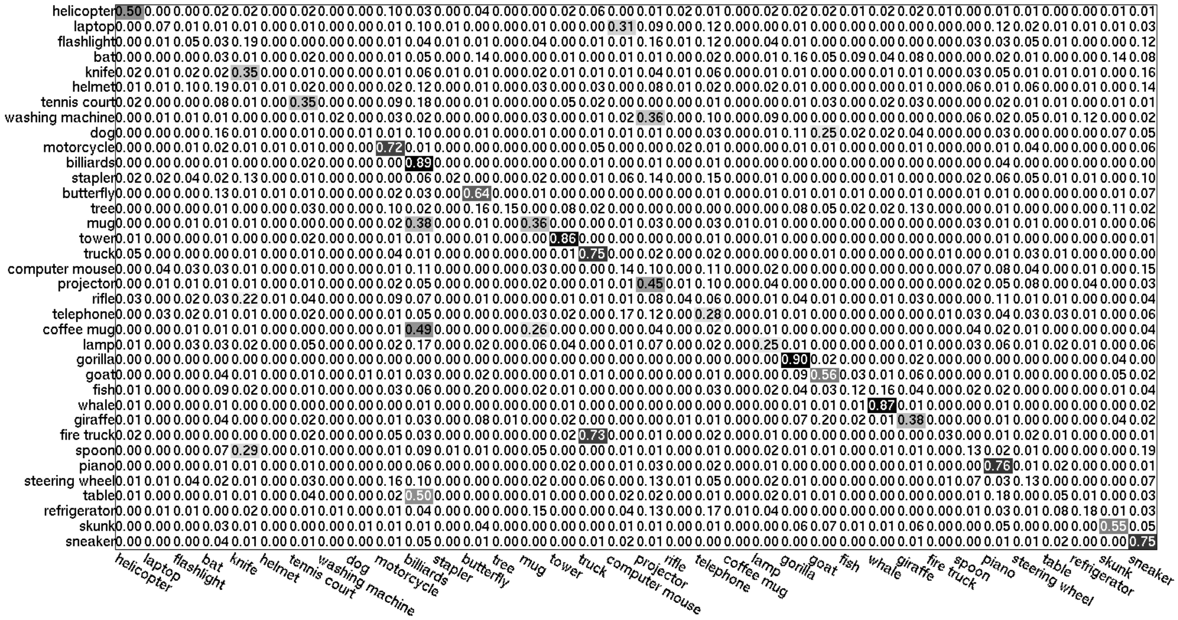
Since this dataset does not have attribute annotations, we only apply our approach with word vectors on this dataset. The comparison of our approach and the baselines is shown in Table III. Again, our method outperforms the baseline approaches. We visualize the confusion matrix of our approach on this dataset in Fig. 11.

IV. CONCLUSION

We have proposed an approach for zero-shot object recognition. The novelty of our approach is that we use the semantic label vectors of object classes to define how an unknown class is related to known classes. In this paper, we have considered both attribute vectors and word vectors of object classes as the label vectors. Our experimental results on three benchmark datasets have demonstrated that our proposed method outperforms other baseline approaches.

ACKNOWLEDGEMENT

This work was supported by NSERC and the University of Manitoba Research Grants Program (URGP). We gratefully



	helicopter	laptop	flashlight	bat	knife	helmet	tennis court	washing machine	dog	motorcycle	billiards	stapler	butterfly	tree	mug	tower	truck	computer mouse	projector	rifle	telephone	coffee mug	lamp	gorilla	goat	fish	whale	giraffe	fire truck	spoon	piano	steering wheel	table	refrigerator	skunk	sneaker
helicopter	0.50	0.00	0.00	0.02	0.02	0.00	0.02	0.00	0.00	0.10	0.03	0.00	0.04	0.00	0.00	0.02	0.06	0.00	0.01	0.02	0.01	0.00	0.02	0.02	0.02	0.01	0.02	0.02	0.01	0.00	0.01	0.01	0.00	0.00	0.01	0.01
laptop	0.00	0.07	0.01	0.01	0.01	0.00	0.01	0.00	0.00	0.01	0.10	0.01	0.00	0.00	0.01	0.00	0.00	0.31	0.09	0.00	0.12	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.12	0.02	0.01	0.01	0.01	0.03
flashlight	0.00	0.01	0.05	0.03	0.19	0.00	0.00	0.00	0.00	0.01	0.04	0.01	0.01	0.00	0.04	0.00	0.01	0.01	0.16	0.01	0.12	0.00	0.04	0.01	0.00	0.00	0.00	0.00	0.00	0.03	0.03	0.05	0.01	0.00	0.00	0.12
bat	0.00	0.00	0.00	0.03	0.01	0.00	0.02	0.00	0.00	0.01	0.05	0.00	0.14	0.00	0.00	0.01	0.00	0.01	0.01	0.00	0.02	0.00	0.01	0.16	0.05	0.09	0.04	0.08	0.00	0.00	0.02	0.01	0.00	0.00	0.14	0.08
knife	0.02	0.01	0.02	0.02	0.35	0.00	0.01	0.00	0.00	0.01	0.06	0.01	0.01	0.00	0.02	0.01	0.01	0.01	0.04	0.01	0.06	0.00	0.01	0.01	0.00	0.00	0.01	0.01	0.00	0.03	0.05	0.01	0.01	0.01	0.00	0.16
helmet	0.01	0.01	0.10	0.19	0.01	0.01	0.02	0.00	0.00	0.02	0.12	0.00	0.01	0.00	0.03	0.00	0.03	0.00	0.08	0.01	0.02	0.00	0.02	0.01	0.00	0.00	0.00	0.01	0.00	0.06	0.01	0.06	0.00	0.01	0.00	0.14
tennis court	0.02	0.00	0.00	0.08	0.01	0.00	0.35	0.00	0.00	0.09	0.18	0.00	0.01	0.00	0.00	0.05	0.02	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.03	0.00	0.02	0.03	0.00	0.00	0.02	0.01	0.01	0.00	0.01	0.01
washing machine	0.00	0.01	0.01	0.01	0.00	0.00	0.01	0.02	0.00	0.03	0.02	0.00	0.00	0.00	0.03	0.00	0.01	0.02	0.36	0.00	0.10	0.00	0.09	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.02	0.05	0.01	0.12	0.00	0.02
dog	0.00	0.00	0.00	0.16	0.01	0.00	0.01	0.00	0.01	0.01	0.10	0.00	0.01	0.00	0.01	0.00	0.01	0.01	0.01	0.00	0.03	0.00	0.01	0.11	0.25	0.02	0.02	0.04	0.00	0.00	0.03	0.00	0.00	0.00	0.07	0.05
motorcycle	0.00	0.00	0.01	0.02	0.01	0.01	0.01	0.00	0.00	0.72	0.01	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.00	0.02	0.01	0.00	0.00	0.01	0.01	0.00	0.00	0.01	0.00	0.00	0.01	0.04	0.00	0.00	0.00	0.06
billiards	0.00	0.00	0.00	0.01	0.00	0.00	0.02	0.00	0.00	0.00	0.89	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.00
stapler	0.02	0.02	0.04	0.02	0.13	0.00	0.01	0.00	0.00	0.00	0.06	0.02	0.00	0.00	0.02	0.00	0.01	0.06	0.14	0.00	0.15	0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.02	0.06	0.05	0.01	0.01	0.00	0.10
butterfly	0.00	0.00	0.00	0.13	0.01	0.01	0.01	0.00	0.00	0.02	0.03	0.00	0.64	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.01	0.01	0.01	0.00	0.01	0.00	0.01	0.01	0.00	0.00	0.00	0.01	0.07
tree	0.00	0.00	0.00	0.01	0.00	0.00	0.03	0.00	0.00	0.10	0.02	0.00	0.16	0.15	0.00	0.08	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.08	0.05	0.02	0.02	0.13	0.00	0.00	0.01	0.00	0.00	0.00	0.11	0.02
mug	0.00	0.00	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.02	0.38	0.00	0.01	0.00	0.36	0.00	0.00	0.01	0.03	0.00	0.03	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.01	0.01	0.00	0.01	0.00	0.06
tower	0.01	0.00	0.00	0.01	0.00	0.00	0.02	0.00	0.00	0.01	0.01	0.00	0.01	0.00	0.00	0.86	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.01
truck	0.05	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.04	0.01	0.00	0.00	0.00	0.00	0.01	0.75	0.00	0.02	0.00	0.02	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.01	0.03	0.01	0.00	0.00	0.00
computer mouse	0.00	0.04	0.03	0.03	0.01	0.00	0.01	0.00	0.00	0.01	0.11	0.00	0.00	0.00	0.03	0.00	0.00	0.14	0.10	0.00	0.11	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.08	0.04	0.00	0.01	0.00	0.15
projector	0.00	0.01	0.01	0.01	0.01	0.00	0.01	0.00	0.00	0.02	0.05	0.00	0.00	0.00	0.02	0.00	0.01	0.01	0.45	0.01	0.10	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.05	0.08	0.00	0.04	0.00	0.03
rifle	0.03	0.00	0.02	0.03	0.22	0.01	0.04	0.00	0.00	0.09	0.07	0.00	0.01	0.00	0.00	0.01	0.01	0.01	0.08	0.04	0.06	0.00	0.01	0.04	0.01	0.00	0.01	0.03	0.00	0.00	0.11	0.01	0.01	0.00	0.00	0.04
telephone	0.00	0.03	0.02	0.01	0.01	0.00	0.02	0.00	0.00	0.01	0.05	0.00	0.00	0.00	0.03	0.02	0.00	0.17	0.12	0.00	0.28	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.01	0.03	0.04	0.03	0.03	0.01	0.00	0.06
coffee mug	0.00	0.00	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.01	0.49	0.00	0.00	0.00	0.26	0.00	0.00	0.00	0.04	0.00	0.02	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.02	0.01	0.00	0.00	0.00	0.04
lamp	0.01	0.00	0.03	0.03	0.02	0.00	0.05	0.00	0.00	0.02	0.17	0.00	0.02	0.00	0.06	0.04	0.00	0.01	0.07	0.00	0.02	0.00	0.25	0.01	0.00	0.00	0.00	0.01	0.00	0.03	0.06	0.01	0.02	0.01	0.00	0.06
gorilla	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.90	0.02	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00
goat	0.00	0.00	0.00	0.04	0.01	0.00	0.01	0.00	0.01	0.01	0.03	0.00	0.02	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.09	0.56	0.03	0.01	0.06	0.00	0.00	0.01	0.00	0.00	0.00	0.05	0.02
fish	0.01	0.00	0.00	0.09	0.02	0.00	0.01	0.00	0.00	0.03	0.06	0.00	0.20	0.00	0.02	0.01	0.00	0.00	0.00	0.00	0.03	0.00	0.02	0.04	0.03	0.12	0.16	0.04	0.00	0.02	0.02	0.00	0.00	0.00	0.01	0.04
whale	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.87	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.02
giraffe	0.01	0.00	0.00	0.04	0.00	0.00	0.02	0.00	0.00	0.01	0.03	0.00	0.08	0.01	0.00	0.02	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.07	0.20	0.02	0.01	0.38	0.00	0.00	0.01	0.00	0.00	0.00	0.04	0.02
fire truck	0.02	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.05	0.03	0.00	0.00	0.00	0.00	0.02	0.73	0.00	0.01	0.00	0.02	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.01	0.02	0.01	0.00	0.00	0.01
spoon	0.00	0.00	0.00	0.07	0.29	0.00	0.01	0.00	0.00	0.01	0.09	0.01	0.01	0.00	0.05	0.00	0.00	0.01	0.01	0.00	0.01	0.00	0.02	0.01	0.01	0.00	0.01	0.01	0.00	0.13	0.02	0.01	0.00	0.00	0.00	0.19
piano	0.00	0.00	0.00	0.01	0.01	0.00	0.01	0.00	0.00	0.00	0.06	0.00	0.00	0.00	0.00	0.02	0.00	0.01	0.03	0.00	0.02	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.76	0.01	0.02	0.00	0.00	0.01
steering wheel	0.01	0.01	0.04	0.02	0.01	0.00	0.03	0.00	0.00	0.16	0.10	0.00	0.00	0.00	0.02	0.00	0.06	0.00	0.13	0.01	0.05	0.00	0.02	0.01	0.00	0.00	0.00	0.00	0.01	0.07	0.03	0.13	0.00	0.00	0.00	0.07
table	0.01	0.00	0.00	0.01	0.01	0.00	0.04	0.00	0.00	0.02	0.50	0.00	0.00	0.00	0.01	0.00	0.00	0.02	0.02	0.00	0.01	0.00	0.02	0.01	0.01	0.00	0.00	0.00	0.00	0.01	0.18	0.00	0.05	0.01	0.00	0.03
refrigerator	0.00	0.01	0.01	0.00	0.02	0.00	0.01	0.01	0.00	0.01	0.04	0.00	0.00	0.00	0.15	0.00	0.00	0.04	0.13	0.00	0.17	0.01	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.03	0.01	0.08	0.18	0.01	0.03
skunk	0.00	0.00	0.00	0.03	0.01	0.00	0.00	0.00	0.01	0.01	0.01	0.00	0.04	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.01	0.00	0.00	0.06	0.07	0.01	0.01	0.06	0.00	0.00	0.05	0.00	0.00	0.00	0.55	0.05
sneaker	0.00	0.00	0.00	0.04	0.01	0.00	0.01	0.00	0.00	0.01	0.05	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.02	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.01	0.02	0.00	0.00	0.75

Figure 11. Confusion matrix of our approach with word vectors on the ImageNet-112 dataset.

acknowledge the support of NVIDIA Corporation with the GPU donation used in this research.

References

[1] I. Biederman, "Recognition by components: A theory of human image understanding," *Psychological Review*, vol. 94, no. 2, pp. 115–147, 1987.

[2] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 594–611, April 2006.

[3] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2009.

[4] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2009.

[5] D. Osherson, J. Stern, O. Wilkie, M. Stob, and E. E. Smith, "Default probability," *Cognitive Science*, vol. 15, no. 2, 1001.

[6] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, and B. Schiele, "What helps where - and why? semantic relatedness for knowledge transfer," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.

[7] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov, "DeViSE: A deep visual-semantic embedding model," in *Advances in Neural Information Processing Systems*, 2013.

[8] A. Karpathy, A. Joulin, and L. Fei-Fei, "Deep fragment embeddings for bidirectional image-sentence mapping," in *Advances in Neural Information Processing Systems*, 2014.

[9] M. Rochan and Y. Wang, "Weakly supervised localization of novel objects using appearance transfer," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[10] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*. MIT Press, 2013.

[11] L. van der Maaten and G. E. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[12] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Advances in Neural Information Processing Systems*, 2007.

[13] T. Tommasi, T. Tuytelaars, and B. Caputo, "A testbed for cross-dataset analysis," arXiv: 1402.5923, Tech. Rep., 2014.

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2009.

[15] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv:1408.5093*, 2014.

[16] M. Rohrbach, S. Ebert, and B. Schiele, "Transfer learning in a transductive setting," in *Advances in Neural Information Processing Systems*, 2013.