Zero-shot Learning for Visual Recognition Problems

by

Shujon Naha

A thesis submitted to The Faculty of Graduate Studies of
The University of Manitoba
in partial fulfilment of the requirements of the degree of

MASTER OF SCIENCE

Department of Computer Science
University of Manitoba
Winnipeg

Thesis Supervisor

Dr. Yang Wang

Author

Shujon Naha

Zero-shot Learning for Visual Recognition Problems

Abstract

In this thesis we discuss different aspects of zero-shot learning and propose solutions for three challenging visual recognition problems: 1) unknown object recognition from images 2) novel action recognition from videos and 3) unseen object segmentation. In all of these three problems, we have two different sets of classes, the "known classes", which are used in the training phase and the "unknown classes" for which there is no training instance. Our proposed approach exploits the available semantic relationships between known and unknown object classes and use them to transfer the appearance models from known object classes to unknown object classes to recognize unknown objects. We also propose an approach to recognize novel actions from videos by learning a joint model that links videos and text. Finally, we present a ranking based approach for zero-shot object segmentation. We represent each unknown object class as a semantic ranking of all the known classes and use this semantic relationship to extend the segmentation model of known classes to segment unknown class objects.

Contents

List of Figures

List of Tables

List of My Publications

[1] Naha, S., and Wang, Y. Zero-Shot Object Recognition Using Semantic Label Vectors. In *The 12th Conference on Computer and Robot Vision (CRV)* (June 2015).

[2] Naha, S., and Wang, Y. Beyond Verbs: Understanding Actions in Videos with Text. In *The 23rd International Conference on Pattern Recognition (ICPR)* (December 2016).

[3] Naha, S., and Wang, Y. Object Figure-Ground Segmentation Using Zero-Shot Learning. In *The 23rd International Conference on Pattern Recognition (ICPR)* (December 2016).

Acknowledgments

I would like to thank my supervisor Dr. Yang Wang for his constant support and endless patience in completing my masters thesis. I would not be able to complete my thesis without his guidance in each step of my thesis from beginning to the end. I would also like to thank Department of Computer Science of University of Manitoba and Dr. Wang for the Guaranteed Funded Package and assistance respectively.

I would also like to thank my committee members, Dr. Neil Bruce and Dr. Ekram Hossain for their valuable times and efforts in perfecting my thesis through suggestions and feedback.

Last but not the least, I would like to acknowledge the unconditional love and support of my family staying overseas and dedicate this thesis to my parents Swapan Naha, Beauty Naha, my brother Dipankar Naha and my wife Prianka Banik.

If intelligence was a cake, unsupervised learning would be the cake, supervised learning would be the icing on the cake, and reinforcement learning would be the cherry on the cake. We know how to make the icing and the cherry, but we don't know how to make the cake.

Yann LeCun

Chapter 1

Introduction

Zero-shot learning extends the standard supervised learning to a setting where labelled training examples are not available for all classes. Humans are capable of recognizing new objects with minimal supervision. For example, humans are able to differentiate between different breeds of dog(fine grained classification) from cat or similar looking animals(normal classification) with a few number of images of each type of dog. But this is challenging for supervised learning models which need large amount of images to learn the variances between different object categories. Zero-shot learning methods aim to learn from scarce data just like humans. Human can even analyse the attributes of a completely unseen object and come up with a prediction based on the knowledge of previously learned objects. This transfer of knowledge from known classes to unknown classes is also known as "transfer learning". In this thesis , we focus on three challenging zero-shot visual recognition problems. The first problem is recognizing completely unseen object categories from images. The second problem deals with recognizing previously unseen actions from videos. The

third problem is on segmenting novel object classes from images.

1.1 Unknown Object Recognition from Images

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 labelled training data.

It has been estimated that humans can recognize between 5,000 and 30,000 object categories [3]. 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 able to learn a new object category from a small number (2 or 3) of images [13]. They can even 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 [12; 26].

The limitation of attribute-based approaches is that the attributes have to be manually defined. Farhadi et al. [12] manually define 64 visual attributes and use crowd-sourcing to obtain the ground-truth attribute annotations for images. Lam-

pert et al. [26] use the data collected in the cognitive science literature [38] to define 85 attributes for 50 animal classes. Rohrbach et al. [46] 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 thesis, we propose a zero-shot learning approach which considers each object class as a vector representation of its visual attributes. We also exploit the recent work in natural language processing which has produced vector representations of words. This gives us a semantic vector representation of the object class name. 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.

1.2 Learning Semantic Relationship Between Videos and Text for Novel Action Recognition

Video understanding is one of the key research problems in computer vision. Most previous research in this area focuses on action classification, where the goal is to assign a video into one of the several predefined action categories. Recognizing novel actions from videos is still an unexplored area. But similar to objects, different actions also contain semantic relationships such as, 'playing a violin' and 'playing a guitar' are much semantically closer than the action 'riding a horse'. These relationships can be

utilized to recognize unseen actions based on the knowledge learned on known actions. In the literature, action classification is often performed by using standard classifiers (e.g. SVM) together with various video-based features, such as spatial temporal interest points [27], motion trajectories [55], deep learning based features [53]. In this thesis, our goal is to develop a classifier which will be able to recognize unseen actions based on the knowledge of the previously seen actions.

Early works on action recognition typically use video datasets that are collected in controlled environments (e.g. KTH dataset [49], Weizmann dataset [4]). These videos tend to have simple actions and clean background. Over the year, researchers have introduced more challenging and diverse datasets, e.g. Hollywood-2 [35], UCF actions [51], HMDB [24].

Most previous works in action recognition assume a fixed set of action labels. However, videos in the real world tend to have much more diversity. Given a video, it is possible to describe the video in several different ways. Fig. 1.1 shows some example videos from the YouTube dataset [16]. Each video is associated with multiple sentence descriptions. We can see that people often use different words to describe the same event in the video. So it is difficult to pre-define a set of discrete category labels that encompass the entire semantic space of videos in the wild.

Our training data consist of videos and corresponding textual annotations. Previous works in video-based action recognition only consider annotations in the form of single verbs (e.g. "running", "walking", etc). In this thesis, we move beyond single verbs and consider textual annotations in richer forms, such as short phrases (e.g. "playing guitar"). Fig 1.2 shows examples of the types of videos and textual

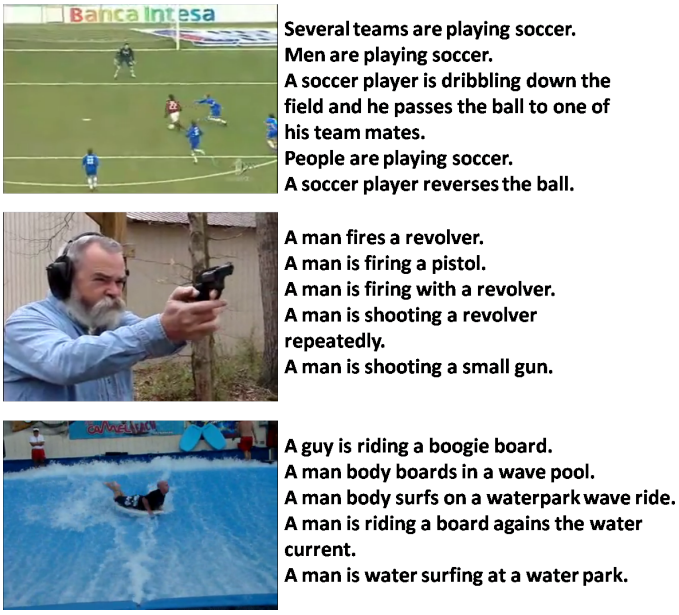


Figure 1.1: Examples from the Youtube dataset [16]. Each video is associated with several sentence descriptions. Notice that different verbs can be used to describe the same video (e.g. "play" versus "pass", "fire" versus "shoot", "ride" versus "board" in these examples).

annotations considered in this thesis.

Our proposed approach solves the problem of joint modelling of videos and corresponding textual descriptions. It consists of three components: the video representation, the textual representation, and a joint model that links videos and text. Our video representation uses the state-of-the-art deep 3D ConvNet to captures the semantic information in the video. Our textual representation uses the recent advancement in learning word and sentence vectors from large text corpus. The joint model is learned to score the correct (video, text) pairs higher than the incorrect ones. We demonstrate our approach in several applications: 1) retrieving sentences given a video; 2) retrieving videos given a sentence; 3) zero-shot action recognition in videos.

Figure 1.2: Examples from the UCF-101 dataset [51]. Each video is associated with one of the 101 short phrases.

1.3 Unknown Object Segmentation

The third problem in this thesis is on segmenting objects from background in images. Object segmentation is a fundamental task in image understanding. If there is a single object of interest, this problem is often known as figure-ground segmentation, where the goal is to produce a binary mask of an image that separates the foreground object from the background. If there are multiple objects of interest, this problem is also referred to as semantic segmentation, where the goal is to assign each pixel in the image a label indicating its object class.

Interactive segmentation (e.g. GrabCut [48]) has been successfully applied for object segmentation. But it requires user input, e.g. in the form of a bounding box around the object of interest. Fully automatic object segmentation approaches typically involve learning the segmentation model from images with ground-truth pixel-level segment annotations. HoIver, manually annotating images with segmentations is very time consuming. Compared with datasets for other visual recognition tasks, current object segmentation datasets are often limited in terms of the number

of object classes and the number of images. For example, ImageNet [8] contains millions of images. Each image is annotated with the class label of the main object in the image. ImageNet has proven to be a valuable resmyce and has enabled the recent deep learning revolution [23] in computer vision. However, none of the ImageNet images is annotated with the object segmentation mask.

To bridge this gap, we propose a zero-shot learning approach for object figure-ground segmentation. Our work is motivated by the following observation. For certain object classes (which we call "source objects"), we have reasonably large datasets with segmentation annotations. For example, the MS COCO dataset [30] contains images with segmentation annotations for about 80 objects. For these 80 objects, we can learn standard segmentation models. But for many other object classes (which we call "target objects"), we do not have training images with segmentation annotations. So we cannot directly learn segmentation models for these target objects. Our method learns to transfer the knowledge from the source objects to the target objects. Our experimental results demonstrate the effectiveness of our approach to segment the target objects.

Chapter 2

Related Works

2.1 Unknown Object Recognition from Image

The goal of zero-shot learning is to recognize classes without training examples. In computer vision, attribute-based representation is a popular approach for zero-shot recognition. Farhadi et al. [12] and Lampert et al. [26] use attributes as intermediate representation shared by object classes.

Some recent work use semantic word embedding (i.e. word vectors) for zero-shot learning in computer vision. Word2vec [36] and Glove [41] are two popular methods for learning word vectors from large corpus. The learned word vectors can be used to measure the semantic distance between two words. If two words (e.g. "dog" and "cat") are semantically close, their word vectors tend to be similar as well. Word vectors have been used for zero-shot recognition of object classes in images [50; 37; 1].

2.2 Action Recognition from Videos

Action Recognition

Action recognition in videos has been widely studied in computer vision. Most works focus on classifying videos into a pre-defined set of action categories, e.g. [4; 21; 27; 55]. Early work in action recognition usually use hand-design features, such as STIP [27], dense trajectories [55]. Inspired by recent success of deep learning in object recognition, some recent work [53; 21] use convolutional neural network to learn the features. Liu et al. [31] use attribute-based representation for action recognition in videos. There is also work on zero-shot event detection [56; 15].

Vision and Text

My work is related to a line of research on connecting video and text. Guadarrama et al. [16] learn to describe arbitrary activities in videos. Their method exploits the semantic hierarchies of subjects, verbs and objects. Yu et al. [58] learn word meanings with the help of video clips. The closest work to my work is Xu et al. [57] which learns to embed videos and sentences in to a common semantic space.

2.3 Unknown Object Segmentation

Supervised Object Segmentation

Most recent works on object segmentation are utilizing Deep Convolutional Neural Network (DCNN) for object segmentation [6; 2; 32; 9]. A widely used approach is to train a DCNN model and use the scores from DCNN as unary potentials for a

Conditional Random field (CRF) [6; 2]. Another DCNN based approach for semantic segmentation uses convolutional layers instead of the last fully connected layers of a DCNN and then apply the DCNN in a sliding window manner on all over the image. Some works use the scores from intermediate feature maps and upsampling to generate a output same size as the input image to resolve the issues of spatial localization for this approach [32]. Others [9] used an extra DCNN to refine the output of the first DCNN model. All of these approaches require fully labelled segmentation ground truth data to train their models.

Weakly Supervised Object Segmentation

Research has been done to reduce the requirement of supervision by learning from weakly labelled data. Weakly labelled segmentation data contains only the name of the objects present in the image but not any pixel level labels. Multiple Instance Learning (MIL) algorithms have been used [40; 42] to train a segmentation model using weakly labelled data. Others [33; 39] have applied Expectation Maximization (EM) algorithms to utilize the weak labels when inferring the latent image segmentations. But still these approaches need a lot of training images per class to train the model where my approach does not have any such restriction.

Segmentation Using Transfer Learning

My thesis is closely related to the work of [18] which propose to use appearance models of previously segmented classes to help segmenting a new class. They have started with training a segmentation model using images of a few fully labelled known classes and then propagated the knowledge to learn the segmentation model of the

next semantically closest object class. Instead of simply measuring appearance similarity, [34] proposes to employ per-exempler SVMs to find neighbours for transferring the segmentation knowledge. [47] proposed a solution to the figure-ground segmentation problem by transferring segmentation masks between images based on their global similarity.

Chapter 3

Zero-Shot Object Recognition

In this chapter, 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. 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 [14], image-sentence mapping [20], etc.

3.1 Unknown Object Recognition from Image

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. During testing, the system will be 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.

The overview of our approach is summarized in Fig. 3.1. For each object category (either known or unknown), we assume that we have a vector representation of this category. 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 thesis, we choose to represent the unknown object class as a sparse linear combination of known object classes. 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 [44]. The method in [44] deals with localizing unseen objects in weakly labelled images or videos, while our work focuses on recognizing unseen objects.

In this thesis, we consider two different types of semantic label vectors to find the semantic relationship between known and unknown classes.

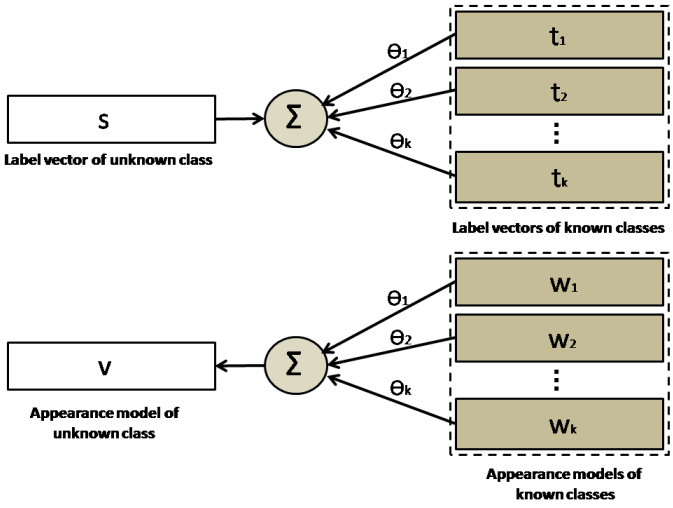


Figure 3.1: An overview of our approach. (Top) we represent the word vector $\mathbf{s}$ of an unknown object class as a sparse linear combination of the word vectors of known objects $\mathbf{t}_1$, $\mathbf{t}_2$,...,$\mathbf{t}_K$. The coefficients of this linear combination are θ_1, θ_2,...,θ_K. (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.

3.1.1 Attribute Vectors

In computer vision, attributes have been proposed to capture high-level concepts related to objects. For example, Fig. 3.2 shows examples of attributes of some object classes. In this thesis, we consider 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. [26] use the data collected in cognitive science research [38] to define the attribute vectors for animals.

Figure 3.2: Examples of attributes for three animal classes: skunk, buffalo, and lion.

3.1.2 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 [36]. If two words (e.g. "cat" and "dog") are semantically close, the distance of their word vectors tend to be small. Figure 3.3 shows a visualization of the word vectors by projecting them on 2D using t-SNE [54].

3.1.3 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

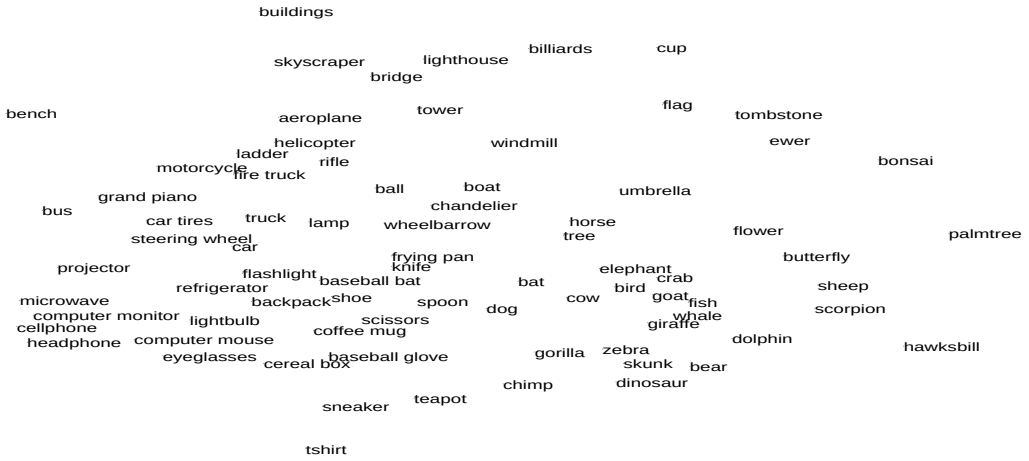


Figure 3.3: Visualization of word vectors in 2D. The 2D embedding of the word vectors is obtained using the t-SNE algorithm [54]. From the visualization, we can see that semantically similar words tend to be close in terms of their word vectors.

unknown object and known objects. In 3.1.4, 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, i.e.

$$\mathbf{s} \approx \theta_1 \mathbf{t}_1 + \theta_2 \mathbf{t}_2 + ... + \theta_K \mathbf{t}_K \tag{3.1}$$

$$\text{where} \quad \theta_k \geq 0, \quad k = 1, 2, ..., K \tag{3.2}$$

We can estimate the coefficients $\Theta = [\theta_1, \theta_2, ..., \theta_K]$ by solving an optimization problem similar to sparse coding [29].

$$\min_{\Theta \geq 0} || \sum_{k=1}^{K} \theta_k \mathbf{t}_k - \mathbf{s}||_2^2 + \lambda ||\Theta||_1 \tag{3.3}$$

The first term in Eq. 3.3 is the reconstruction error. The second term in Eq. 3.3 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.

3.1.4 Appearance Transfer

we now describe how to use the coefficients Θ obtained in Eq. 3.3 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 thesis, 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. 3.3 to represent the appearance model $\mathbf{v}$ of an unknown object class as:

$$\mathbf{v} \approx \sum_{k=1}^{K} \theta_k \mathbf{w}_k \tag{3.4}$$

In Eq. 3.4, the coefficients θ_k $(k = 1, 2, ..., K)$ are obtained using the label vectors (see Eq. 3.3). So as long as we have a vector representation of object classes, we can use Eq. 3.4 to transfer appearance models from known objects to an unknown object.

3.1.5 Recognizing Novel Objects

Suppose we have L unknown object classes. For each unknown object class, we obtain its appearance model $\mathbf{v}_i$ $(i = 1, 2, ..., L)$ using the method in Section 3.1.4. 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.:

$$y = \arg\max_i \mathbf{v}_i^\top \mathbf{x} \tag{3.5}$$

3.2 Experiments

We evaluate our approach on three benchmark datasets: animal dataset [26], object attribute dataset [12], and a subset (112 object classes) of the ImageNet [52]. Since the name of an object class can be a phrase (e.g. "giant panda"), we use Google's word2phrase tool [36] 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. 3.4 shows some examples of several object classes and the most similar object classes according to the word vector distances.

WordNet distance: this method measures the distance of two object classes by considering their distance in the WordNet hierarchy [8]. Fig. 3.5 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,

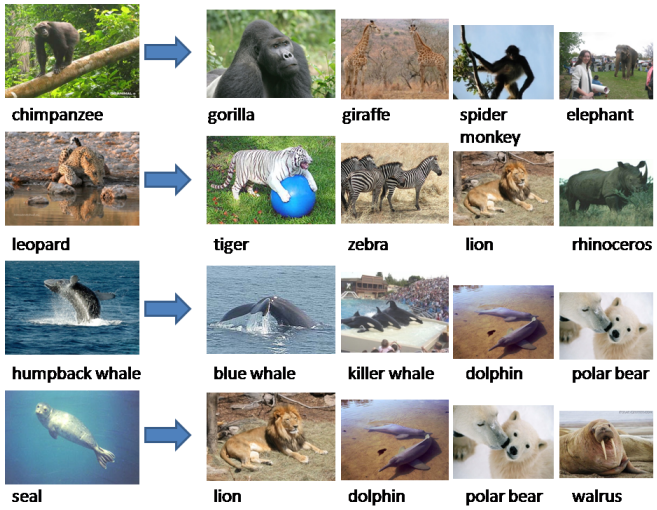


Figure 3.4: Examples of word vector distances. In each row, we show an object class and the most similar four object classes according to the word vector distances.

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. 3.6 shows some examples of the object classes and the most similar object classes according to the attribute vector distances.

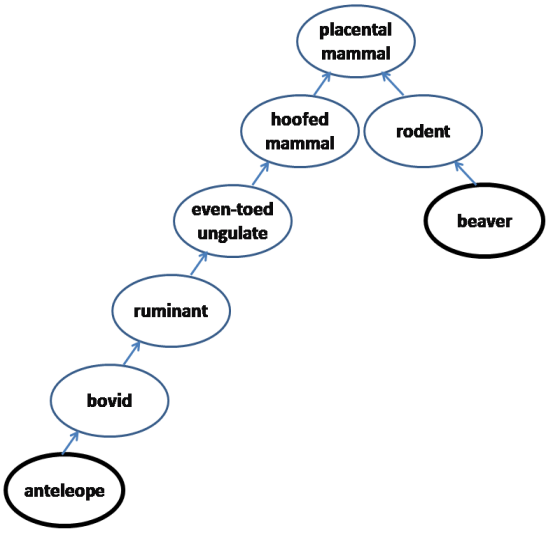


Figure 3.5: An illustration of how to compute the WordNet distance. For two objects ("anteleope" and "beaver") in the WordNet hierarchy, we use the length of the path between them as the distance measurement. In this case, the distance between these two objects is 7.

3.2.1 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 [38]. Figure 3.7 visualizes the resulting 50×85 class-attribute matrix.

Following [26], 40 animal classes are used as the known classes and the remaining 10 used as the unknown classes. We use Caffe [19] 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 3.1, we compare our approach (using both attribute vector and word vector) with the three baselines. We also compare with previous published work in [26; 45]. The comparison shows that our proposed method outperforms the competing approaches. In Fig. 3.8, we visualize the confusion matrix of our approach with word

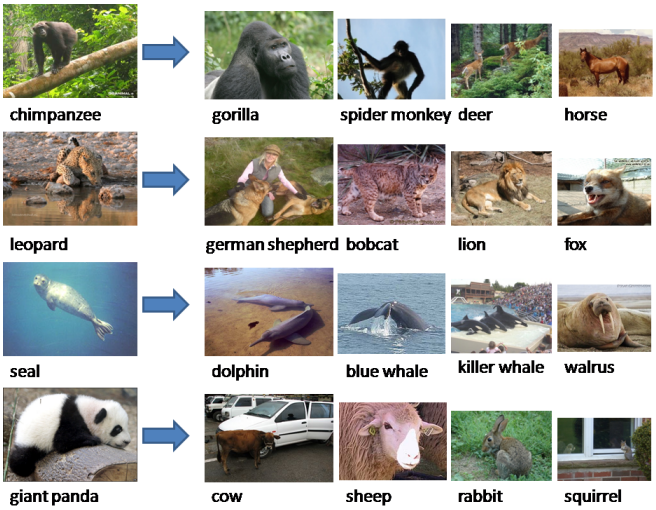


Figure 3.6: Examples of attribute vector distances. In each row, we show an object class and the most similar four object classes according to the attribute vector distances.

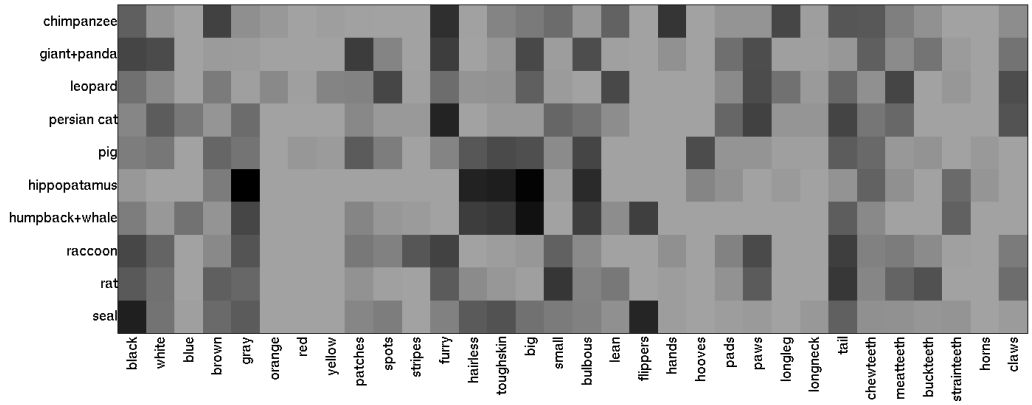


Figure 3.7: Visualization of the class-attribute matrix on the animal dataset. Darker boxes mean stronger associated between an attribute and a class. Binary attributes are obtained by thresholding the values in the matrix.

vectors on this dataset.

Table 3.1: 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. [26]	42.2
Rohrbach et al. [45]	42.7

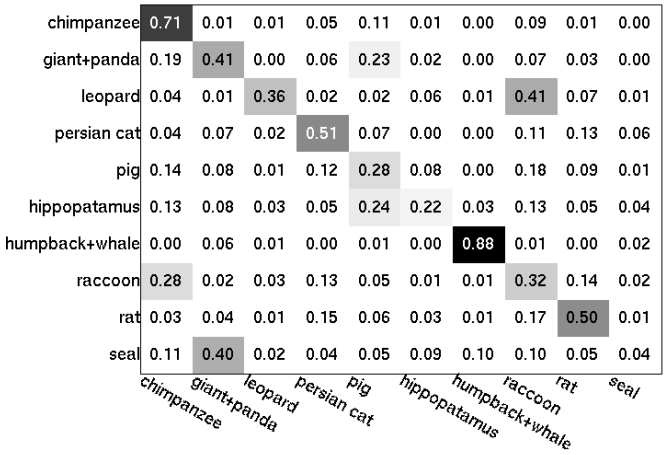


Figure 3.8: Confusion matrix of our approach with word vectors on the animal dataset. Each row corresponds to a ground-truth label and each column corresponds to a prediction. The (i, j) entry of the confusion matrix shows the percentage of images from class i that are classified as class j.

3.2.2 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 3.9 visualizes the class-attribute matrix on this dataset.

The comparison of our approach and the baselines is shown in Table 3.2. Again,

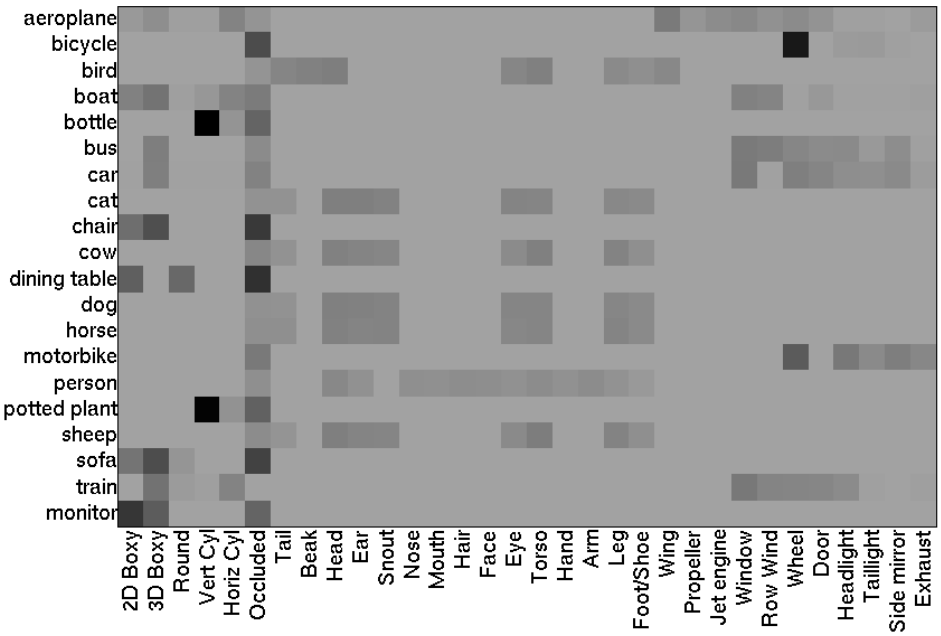


Figure 3.9: Visualization of the class-attribute matrix of the object attribute dataset.

Table 3.2: 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

our method outperforms the baseline approaches. We visualize the confusion matrix of our approach with word vectors on this dataset in Fig. 3.10.

3.2.3 ImageNet-112 dataset

This dataset is collected in [52] and is a subset of the ImageNet [8]. It contains images from 112 object classes. We consider 76 of them as the known classes and the remaining 36 classes as the unknown classes. Similarly, we use the Caffe [19] feature

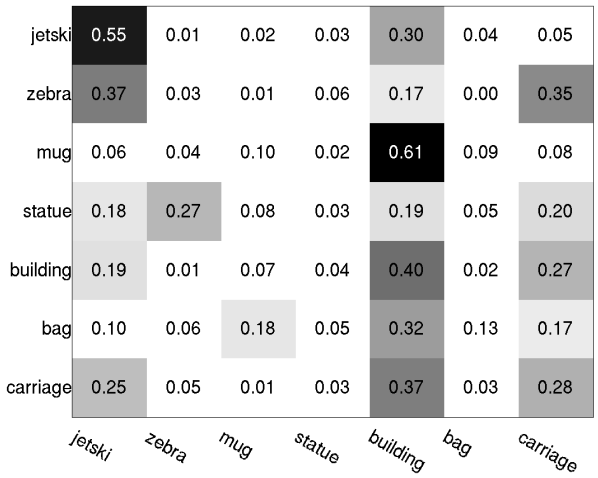


Figure 3.10: Confusion matrix of our approach with word vectors on the object attribute dataset.

Table 3.3: 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

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 3.3. Again, our method outperforms the baseline approaches. We visualize the confusion matrix of our approach on this dataset in Fig. 3.11.

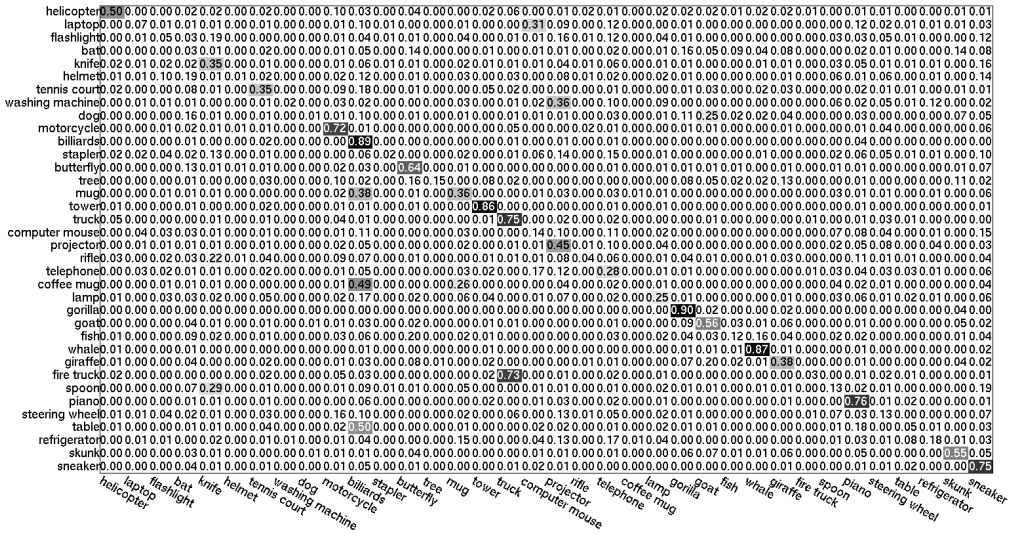


Figure 3.11: Confusion matrix of our approach with word vectors on the ImageNet-112 dataset.

Chapter 4

Understanding Actions in Videos with Text

In this chapter, we learn a scoring function to measure the compatibility between a video and a textual description (phrase or sentence) from the training data. During testing, we can use the learned scoring function to measure the compatibility of an unseen video and an unseen textual annotation.

We demonstrate our approach in several applications. One application is sentence ranking/retrieval for videos. During testing, we are given an unseen video as the query. Our goal is to rank textual descriptions (e.g. sentences) according to their relevance to the query video. Compared with standard video classification, our approach has several advantages. First, we do not assume a fixed set of class labels. This allows our approach to potentially handle a very large semantic space. Second, the words use in the textual descriptions do not necessarily have to appear during training. Third, our method returns a ranked list of textual annotations, so we can

handle the case that a video can be explained by two different annotations. Similarly, our approach can also be used to rank/retrieve videos given a query sentence. Another application is zero-shot video classification. In this case, we want to classify a video into one of the pre-defined categories. Each category is associated with a short phrase, e.g. "riding horse" "playing violin", etc. But in our case, we assume the category labels in training and testing data are disjoint.

4.1 Learning Semantic Relationship Between Videos and Text

Let x be a video and y be the textual description (e.g. a sentence). We use $f(x) \in \mathbb{R}^V$ to denote the V-dimensional feature vector extracted from x and $g(y) \in \mathbb{R}^T$ to denote the T-dimensional feature vector extracted from y. Our goal is to learn a matrix $\mathbf{W} \in \mathbb{R}^{V \times T}$ of model parameters, so that we can use the following scoring function $S_{\mathbf{W}}(x, y)$ to measure the compatibility of $f(x)$ and $g(y)$:

$$S_W(x, y) = f(x)^\top \mathbf{W} g(y) \tag{4.1}$$

In the following, we describe how to define the video representation $f(x)$, the textual representation $g(y)$, and how to learn the model parameters $\mathbf{W}$.

4.1.1 Video Representation

We use the convolutional 3D (C3D) feature proposed in [53] to represent the video. This feature is learned using deep 3-D convolutional network trained on a large scale

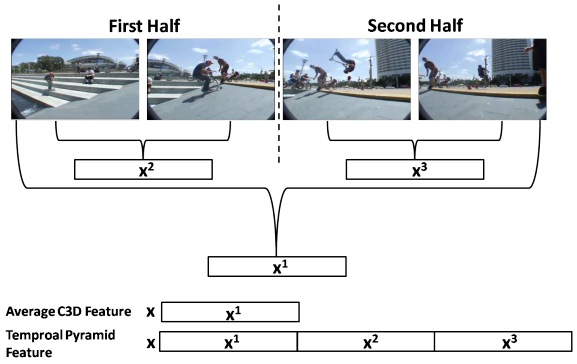


Figure 4.1: Illustration of the average C3D feature and the temporal pyramid feature.

video dataset. It has been shown to be a state-of-the-art feature representation for video analysis. Given a video, the C3D features are extracted by passing the video through the learned C3D network and take the output from the $fc7$ layer. C3D extracts a 4096 dimensional feature vector for every 20 frames in the video.

In order to generate a feature vector for the entire video, we have used the following two approaches. Figure. 4.1 illustrates these two features.

Average C3D feature: In this approach, we simply split a whole video x into multiple shots, where each shot consists of 20 frames. The C3D feature is extracted from each shot. We then take the average of the C3D features from all the shots as the feature vector $f(x)$ for the whole video.

Temporal pyramid: The average C3D feature does not take into account the temporal ordering of the shots. For certain events, the temporal ordering might be useful. For example, if a video is described with the sentence description "A young man skateboards along a step, up a ramp, and then does a flip onto the pavement above.", different temporal segments of the video might contain visual information for different actions (i.e. skateboards along a step, flip onto the pavement) at different

temporal locations (Fig. 4.1). Simply taking the average of the video features from different frame segments will not be very meaningful to relate the complete video with the description sentence. To address this issue, we propose another approach that captures some temporal information. We call it *temporal pyramid*. This is inspired by spatial pyramid matching [28]. We first divide the input video into two equal parts. We then compute the average C3D features of the first part, the second part, and the whole video. We then concatenate these three average C3D vectors as the final representation $f(x)$ of the whole video. The dimension of the final feature representation is $4096 \times 3 = 12288$.

4.1.2 Textual Representation

Given a textual description y, we define a feature vector $g(y)$ to represent the text. For ease of presentation, we will assume the text description is in the form of a sentence.

We would like the textual feature vector $g(y)$ to have the following property. If two sentences y and y' are similar in terms of their semantic meanings, the distance between the two vectors $g(y)$ and $g(y')$ should be small. If the semantic meanings of y and y' are very different, we would like the distance between $g(y)$ and $g(y')$ to be large.

In this thesis, we use two different approaches for the textual representation $g(y)$.

Mean word vectors: Word vectors have recently become a popular representation for capturing semantic meanings. These word vectors are learned from large collections of text documents. In the end, each word is represented as a fixed length

vector (also known as *word embedding*). Similar words (e.g. "dog" and "cat") tend to be mapping closer in this embedding space. Word2vec [36] and GloVe [41] are two popular tools for learning word vectors.

Most tools (e.g. word2vec [36] or GloVe [41]) for extracting word vectors only provide the vectors for individual words. But in our case, we need a vector representation for the whole sentence. Our first approach is to simply take the average of the word vectors of all the words in a sentence. In the experiments, we will show that this simple strategy works surprisingly Ill.

Skip-thought vectors: The skip-thought vector [22] is a very recent work on extracting the vector representation at the sentence level. It was trained from a large collection of more than 11K books. Given a sentence, the skip-thought vector can produce a vector representation that capture the semantic meaning of the sentence. If two sentences are semantically similar, their vector representations will be similar as Ill.

4.1.3 Model Learning

The goal of this section to learn the parameter matrix W. Given the visual representation $f(x)$ and the textual representation $g(y)$, we can score the compatibility of x and y as $S_W(x, y) = f(x)^\top W g(y)$. If the textual description y is a good description of the video x, we would like this score to be high. Otherwise, the score should be low.

We assume a training set with N videos $\{x_1, x_2, ..., x_N\}$. Each training video x_i is associated with one or more textual descriptions (e.g. sentences). We consider these

to be the positive descriptions since they are correct textual descriptions of x_i. We use $\mathcal{P}(x_i)$ to denote the set of positive textual descriptions of x_i. We also assume a set of negative textual descriptions for each x_i. A textual description is negative if it is not a good description of the video. We use $\mathcal{N}(x_i)$ to denote the set of negative textual descriptions of x_i. In Sec. 4.2, we will describe how to construct $\mathcal{P}(x_i)$ and $\mathcal{N}(x_i)$ on the datasets used in the experiments.

Inspired by the large margin criterion in the standard SVM learning, we learn the parameters W by solving the following optimization problem.

$$\min_{\mathbf{W},\xi} \frac{1}{2}||\mathbf{W}||_2^2 + \sum_{i=1}^{N} \sum_{j=1}^{|\mathcal{P}(x_i)|} \xi_{ij} \tag{4.2a}$$

$$\text{s.t.} \quad f(x_i)^\top \mathbf{W} g(y_j) \geq f(x_i)^\top \mathbf{W} g(y_k) + 1 - \xi_{i,j} \tag{4.2b}$$

$$\xi_{i,j} \geq 0, \quad \forall i \in \{1, 2, ..., N\}, \tag{4.2c}$$

$$\forall y_j \in \mathcal{P}(x_i), \forall y_k \in \mathcal{N}(x_i) \tag{4.2d}$$

For a video x_i, the constraint in Eq. 4.2b enforces the score of a positive textual description y_j to be higher than that of a negative description y_k by a margin of 1. Similar to standard SVM, the slack variables ξ_{ij}'s are used to allow soft margins. We use stochastic gradient descent to optimize Eq. 4.2.

4.2 Experiments

We evaluate our approach on two datasets: the YouTube dataset [16] and the UCF-101 dataset [51]. On the YouTube dataset, we consider two applications: retrieving sentences given a video, and retrieving videos given a sentence. On the UCF-101 dataset, we consider zero-shot action recognition.

4.2.1 YouTube dataset

The YouTube dataset contains 1970 videos. Each video is associated with multiple sentence descriptions. See Fig. 1.1 for some examples of this dataset. We only use the sentence descriptions marked as clean in the dataset and ignore the videos that do not have any clean sentence descriptions. We also ignore sentence that do not have a verb. We then split the videos into training and test sets. In the end, the training dataset contains 1300 videos and the test dataset contains 610 videos. On average, each video is associated with 15 sentences.

We learn the model parameters W using the training videos and their associated sentences. For each video, we consider the sentences associated with this video as "positive" textual descriptions. We consider the sentence associated with any other training video as "negative" textual descriptions.

We perform the following two applications on testing videos.

Sentence retrieval: In this task, we are given a query video and try to find good sentence descriptions for this video. For each video in the test set, we rank all the sentences of all testing videos using the scoring function in Eq. 4.1. Since a video might be associated with multiple correct sentences, we use the Normalized Discounted Cumulative Gains (NDCG) [5] to measure the ranking performance. If all the correct sentences are ranked higher than the incorrect ones, the NDCG will be high. We report NDCG values at five different truncation levels.

We consider two baselines for comparison. The first baseline (called "verb only") only uses the word vector corresponding to the verb in the sentence as the textual description. In the second baseline (called "bag-of-words"), we create a dictionary

method		NDCG (%) at				
		20%	40%	60%	80%	100%
skip-thought vector	temporal pyramid feature	**31.07**	**32.94**	**33.29**	**33.40**	**33.44**
	average C3D feature	30.63	32.70	33.08	33.26	33.26
mean word vector	temporal pyramid feature	27.66	30.25	30.95	31.13	31.18
	average C3D feature	27.5	29.81	30.49	30.68	30.71
verb-only vector	temporal pyramid feature	21.63	25.25	26.66	27.51	27.74
	average C3D feature	21.12	24.63	26.18	27.09	27.34
bag-of-words	temporal pyramid feature	22.8	26.06	27.48	28.18	28.43
	average C3D feature	21.79	25.26	26.84	27.57	27.83

Table 4.1: Comparison of mean NDCG at different truncation levels for different approaches to rank sentences for test videos on the YouTube dataset. Our approaches (skip-though vector, mean word vector) performs much better than the two baselines (verb-only vector, bag-of-words). In each approach, we consider both temporal pyramid and average C3D features. The temporal pyramid performs slightly better.

of words from the sentences in the YouTube dataset and represent each sentence using the standard bag-of-words representation. In other words, each sentence is represented as a vector of word frequencies in the sentence.

Table 4.1 shows the comparison. We consider both the average C3D feature and the temporal pyramid as video representations. We can see that our approaches perform much better than the two baselines. This demonstrates the advantage of using a semantic representation of sentences for video understanding. Table 4.1 also shows that the temporal pyramid performs slightly better than the average C3D feature.

We also measure the ranking performance using mean rank defined in [57]. For each testing video, we record the rank of the first correct sentence that describes the query video. We then take the mean of the rank over all testing videos to measure the performance.

method		mean rank
skip-thought vector	temporal pyramid feature	**224.80**
	average C3D feature	231.52
mean word vector	temporal pyramid feature	264.55
	average C3D feature	267.71
verb-only vector	temporal pyramid feature	472.62
	average C3D feature	498.38
bag-of-words	temporal pyramid feature	341.79
	average C3D feature	346.94

Table 4.2: Comparison of the mean rank (lower is better) of different approaches in the sentence retrieval application on the Youtube dataset.

Figure 4.2 shows some retrieved sentences for some sample videos.

Video retrieval: In this task, we consider a sentence as the query and retrieve videos described by this query sentence. Our experiment setting is similar to [57]. We randomly select 5 sentences from each of the testing videos. So in total we have 3050 sentences and 610 videos. For each sentence, we rank all the testing videos according to the score defined in Eq. 4.1. We record the correct video position for each query. Then we calculate the mean rank over all query sentences to measure the performance of the video retrieval. Table 4.3 shows the comparison of different approaches on the YouTube dataset. Again, our approach (skip-though vector, mean word vector) performs much better than the two baselines (verb-only vector, bag-of-words). Similarly, the temporal pyramid feature perform slightly better than the average C3D feature. Figure 4.3 shows some retrieved videos for some sample sentences.

From the above results, we can see that the skip-thought vector performs the best. If we only use the verb, the performance is very poor. Although verbs can identify the action in the video, it fails to relate the subject and object of the video in the corresponding sentence. For example, given the query sentence “A car is making

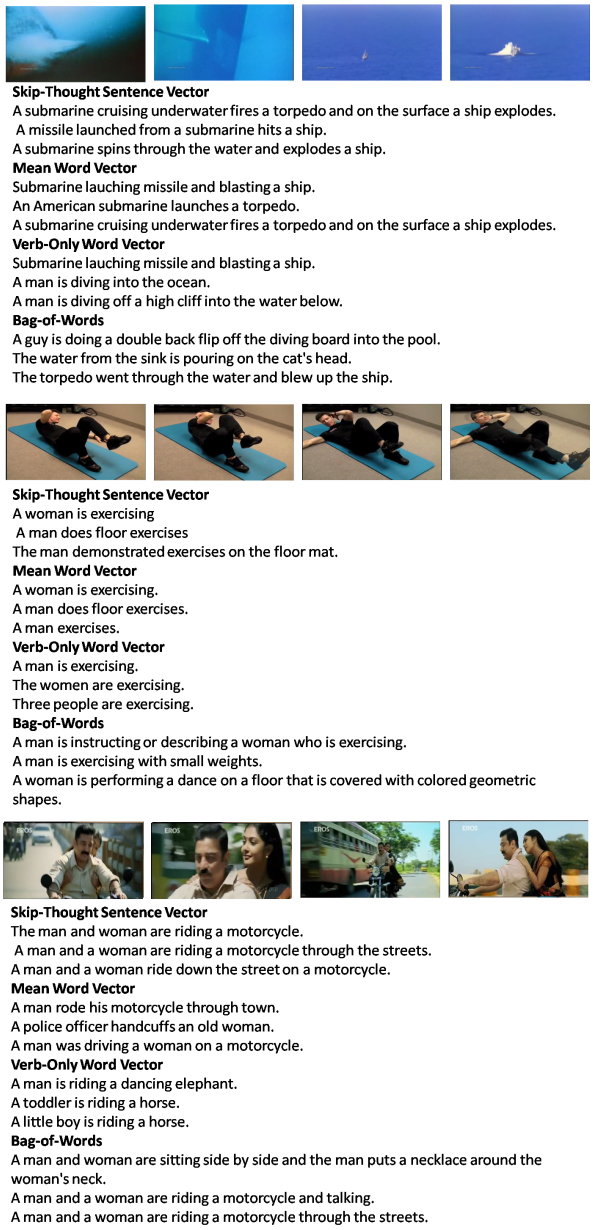


Figure 4.2: Top ranked sentences for sample testing videos on the Youtube dataset for different approaches.

sharp sIrves" (see Fig. 4.3), the verb-only approach retrieves videos for people and electric spark swerving – the action "swerving" is present, but the subject and the object are totally wrong. Similarly for the query sentence "Two teams are playing

method		mean rank
skip-thought vector	temporal pyramid feature	**59.54**
	average C3D feature	61.63
mean word vector	temporal pyramid feature	81.73
	average C3D feature	83.66
verb-only vector	temporal pyramid feature	112.94
	average C3D feature	113.64
bag-of-words	temporal pyramid feature	117.5
	average C3D feature	120.8

Table 4.3: Comparison of mean rank (loIr is better) of different approaches in the video retrieval application on the Youtube dataset.

Method	Video Retrieval	Text Retrieval
Mys	**149.84**	**83.0**
*Xu et al. [57]	236.27	224.10

Table 4.4: Comparison of our results with [57] in term of the mean rank (loIr is better) in video retrieval and text retrieval applications on the YouTube dataset. *These numbers are directly taken from [57]. The numbers are not directly comparable due to variations in features, and dataset constructions. See the text for details.

soccer", the verb-only approach retrieves video where the "playing" action is present but the subject and the object are completely wrong.

We also compare our results on sentence retrieval and video retrieval with [57]. We have folloId the experiment setup in [57] as close as we can, but a direct comparison is difficult since [57] randomly chooses five sentences for each video in the test set. Following [57], we also randomly choose five sentences for each video in the test data. But due to the randomness, the sentences we have chosen are different from those in [57]. With this caveat, Table 4.4 compares our results with [57].

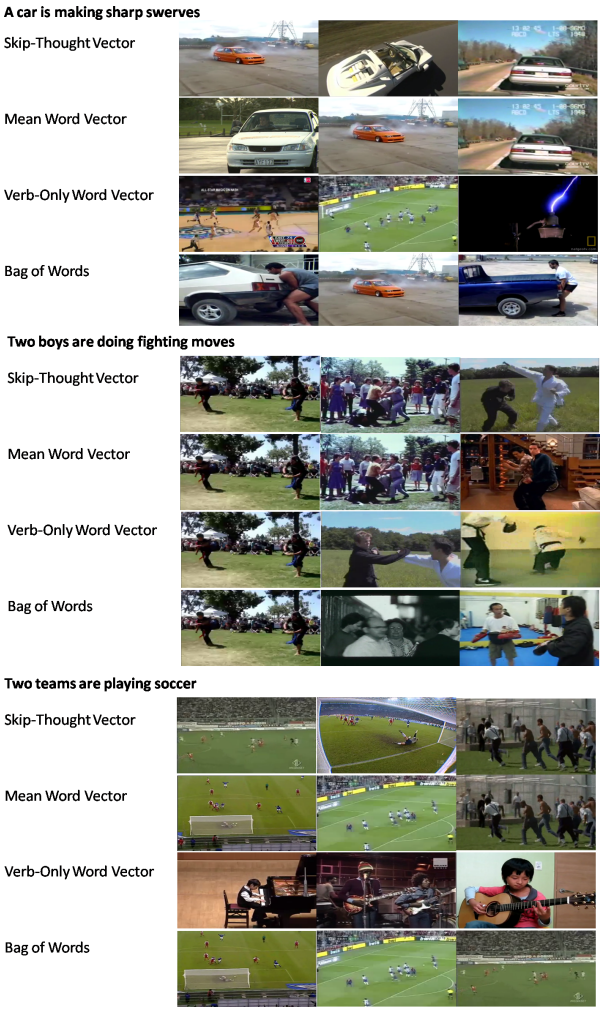


Figure 4.3: Top ranked videos for sample sentences on the YouTube dataset for different approaches.

4.2.2 UCF-101 dataset

The UCF-101 dataset contains 101 action classes and 13320 videos. Each video is associated with a short frame, e.g. "play violin". Most previous work in action recognition uses this dataset for standard action classification.

We perform zero-shot action recognition on this dataset. We split the 101 classes into training and test datasets. The training set consists of 91 classes (we call them

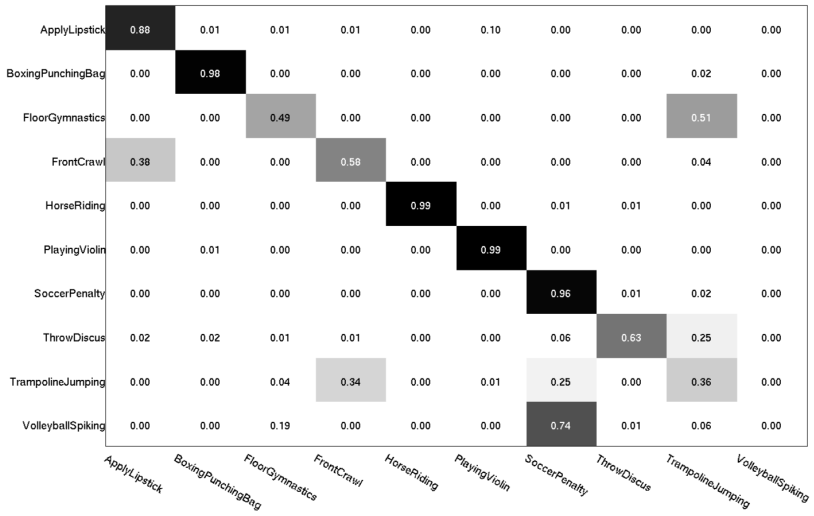


Figure 4.4: Confusion matrix of zero-shot recognition on the UCF-101 dataset.

method		accuracy(%)
our approach	temporal pyramid feature	**68.50**
	average C3D feature	67.14
Lampert et al. [26]	temporal pyramid feature	27.6
	average C3D feature	27.43

Table 4.5: Comparison of our approach with zero-shot learning approach in [26] on the UCF-101 dataset.

"known" classes) and the test set consists of the remaining 10 classes (we call them "unknown" classes). Since the textual description of a class is a phrase (not a sentence), we cannot apply the skip-thought vectors. So we only apply the mean word vector as the representation of the textual description. Each of the 101 classes is represented as the mean of word vectors in the corresponding phrase. For example, the class "play violin" is represented as the mean of the word vectors of "play" and "violin". For each video, we consider the phrase vector corresponding to its class label as the "positive" textual description. We consider the phrase vectors corresponding to the other 100 class labels as the "negative" textual descriptions. We then learn

the model parameters W from the 91 classes in training set.

For a video in the test set, we use Eq. 4.1 to calculate the score corresponding to each of the 10 unknown classes. The predicted class label of this video is the one with the maximum score. Figure 4.4 shows the confusion matrix of recognizing the 10 unknown classes. We can see that even though we do not have any training data for the 10 unknown classes, we can still correctly recognize most of the classes quite Ill. The reason is that the model parameters W capture the semantic meaning of words and phrases.

We also compare our zero-shot learning with the IAP method in [26]. The method in [26] uses the attribute-vector associated with each class for zero-shot learning. Since we do not have attributes on the UCF-101 action dataset, we use the mean word vector in [26]. Table 4.5 shows the comparison. Our approach outperforms [26] by a large margin.

Chapter 5

Zero-Shot Object Figure-Ground Segmentation

The problem discussed in this chapter is illustrated in Fig 5.1. We use a standard semantic segmentation dataset (e.g. MS COCO) as the training dataset. We consider the object classes in the training data as the source objects and learn segmentation models for these object classes. During testing, We are given an image where We know the label of the main object in the image, but the object is not one of the source object classes on the training dataset. Our goal is to segment the object in the image from the background, even though We have never seen images of this object during training. A reliable solution to this problem will allow us to automatically populate large-scale object recognition datasets (e.g. ImageNet) with object segmentation annotations. These segmentation annotations can then be used to learn segmentation models for a large number of object classes.

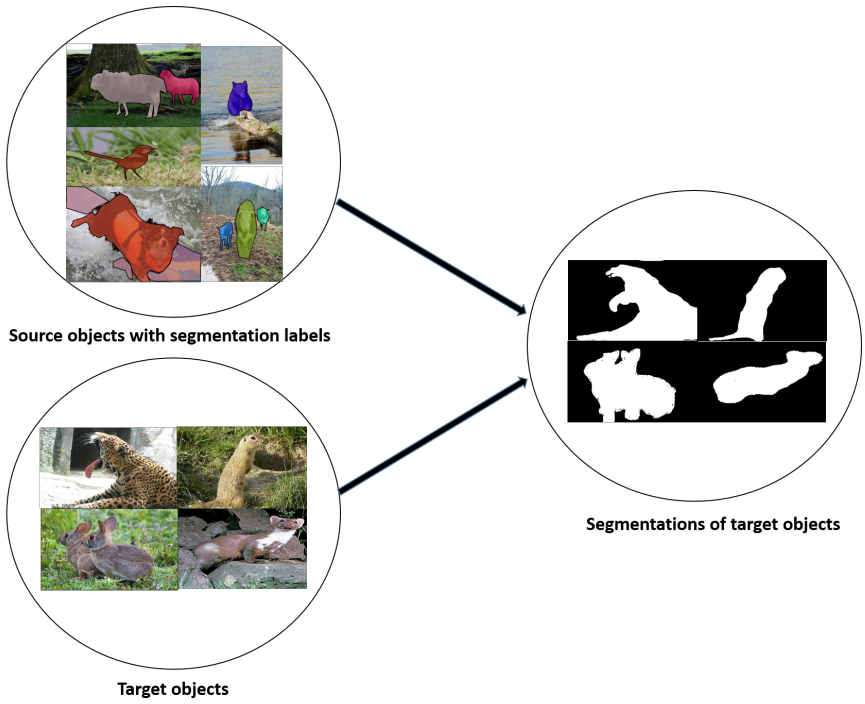


Figure 5.1: An illustration of our problem setup. Our training data consist of Chapter3 of source objects. The pixel-level semantic labels are available on the training data. Our test data consist of images of target objects, where we know the object label of each image, but we do not have the pixel-level segmentations. Note that there is no overlap between the source and target object classes. Our goal is to transfer the knowledge from source objects to target objects, so that we can segment the target objects in the test data.

5.1 Unknown Object Segmentation

Our approach consists of several steps. First, we build segmentation models for source object classes by learning standard semantic segmentation models (Sec. 5.1.1). For a target object, we propose two approaches for measuring the semantic distance between this target object and all the source objects (Sec. 5.1.2). Given an image of the target object, we transfer the segmentation scores from the source objects that are semantically close to the target object (Sec. 5.1.3). Finally, we use the transferred scores to obtain the figure-ground segmentation of the target object in

this image (Sec. 5.1.4).

5.1.1 Segmentation Models for Source Objects

The first step of our approach is to build segmentation models for source objects. We use the approach in [7] to train the segmentation models using Chapter3 of source objects with segmentation annotations. This method uses the deep convolutional neural network (DCNN) to generate an initial segmentation result and then refines the result using a fully connected conditional random field (CRF) for semantic segmentation. Given a test image, we can use the learned DCNN-CRF model to generate score for each pixel being one of the source object classes.

5.1.2 Object Semantic Relationship

In order to do the knowledge transfer, we need to establish the semantic relationship (i.e. distance) between two objects. In this thesis, we consider two different knowledge sources for measuring the distance between objects.

Word vectors: In natural language processing, there has been work on learning word embedding from large collections of text corpus. The goal is to learn to represent each word as a fixed length vector. If two words (e.g. "dog" and "cat") are semantically close, their corresponding word vectors will tend to be similar. Word vectors have been used in various computer vision applications, e.g. zero-shot object recognition [37]. Given two object classes i and j, let v_i and v_j be the word vectors corresponding to the names of these two objects. We can use the Euclidean distance between v_i and v_j to measure the distance between these two object classes.

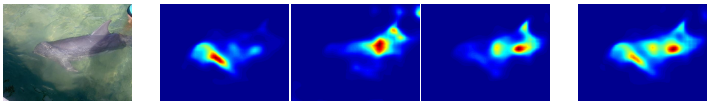
ImageNet hierarchy: We can also use the ImageNet hierarchy to define the distance between two objects. Object classes (known as "synset") in ImageNet are organized in a hierarchy. To find the distance between two objects, we calculate the distance between the two corresponding nodes in the ImageNet hierarchy.

A target object can then be represented as a ranked list of all the source objects. If a source object is closer (in terms of the distance based on either word vectors or ImageNet hierarchy) to the target, this source object will be ranked higher in the list.

5.1.3 Knowledge Transfer

During testing, we am given an image of one of the target object classes. We assume that we know the class label of the image during testing. Our goal is to perform figure-ground segmentation on this image to separate this target object from the background. Since the source objects and target objects are disjoint, we cannot directly use the segmentation models trained for the source objects (Sec. 5.1.1) to segment this image. Our next step is to transfer the knowledge from source objects to target objects, so that we can apply the segmentation models learned in Sec. 5.1.1 to segment the target object.

Figure 5.2 illustrates the knowledge transfer. Let K be the number of source objects. For a target object u, we use $r_u = [r_u^1, r_u^2, ..., r_u^K]$ to denote the ranked list of all source objects. In other words, r_u^1 is the source object most similar (in term of the distance based on word vectors or ImageNet hierarchy) to the target object u, while r_u^K is the most dissimilar one. For a given image x, we apply the segmentation model in Sec. 5.1.1 on this image. For each pixel p in the image x, we will get a



(a) (b) (c)

Figure 5.2: Illustration of the knowledge transfer: (a) a test image of a target class "dolphin"; (b) visualization of $C_p^j(x)$ for each of top-3 ranked source objects (bear, mouse, bird); (c) visualization of the transferred score $s_p^u(x)$.

K-dimensional vector indicating the score of this pixel being one of the K source objects. We use $C_p^j(x)$ to denote the score of the source object j for a particular pixel p in the image x. For example, if the target object is "dolphin" (Fig. 5.2(a)). The top-5 ranked source objects to "dolphin" might be "bear", "mouse", "bird", "horse", "zebra". Figure 5.2(b) shows visualizations of the scores corresponding to these source objects.

Now we would like to use the scores of source objects to estimate the score of the target object on each pixel. Let $s_p^u(x)$ denote the score of the pixel p of the image x being a foreground pixel of the target object u. Let d_u^k be the semantic distance between the target object u and the source object k. We define $s_p^u(x)$ as follows:

$$s_p^u(x) = \sum_{i=1}^{M} \frac{C_p^{r_u^i}(x)}{d_u^{r_u^i}} \tag{5.1}$$

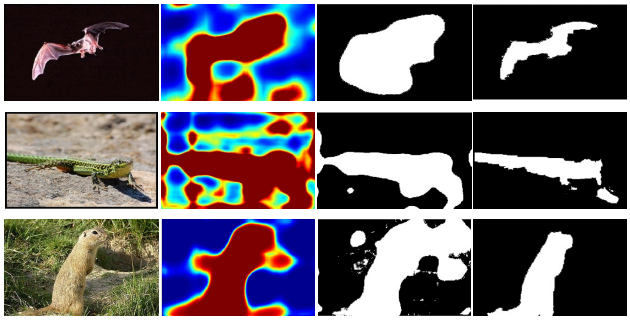
where M is a free parameter and $M \leq K$. The intuition of Eq. 5.1 is to approximate the score of the target object using scores of source objects weighted by their semantic distances to the target object. Note that if a source object is very different from the target object, the scores of the source object are unlikely to be transferable to the target object. The parameter M allows us to only consider the source objects that are similar enough to the target object. By choosing M appropriately, we can effectively ignore those source objects that are very different from the target object. Figure 5.2(c)

shows a visualization of $s_p^u(x)$.

5.1.4 Segmenting Target Objects

After the knowledge transfer in Sec. 5.1.3, we will have a score $s_p^u(x)$ for each pixel p in the image x indicating how likely this pixel belongs to the target object u. A straightforward way of getting the object segmentation is to assign a binary label for each pixel (foreground or background) depending on whether $s_p^u(x)$ is greater than some threshold. In this section, we propose two post-processing techniques to further refine the segmentation output. Figure 5.3 shows some examples of these two post-processing techniques.

We can consider $s_p^u(x)$ as a rough estimate of the foreground/background for each pixel in the image x. This suggests that we can use $s_p^u(x)$ to build a discriminative appearance model for the target object in this *specific image* x. Similar ideas have been used in [43] for people tracking. To build the appearance model, we take the output from the fully connected layer "fc7" of the trained Deep Convolutional Neural Network and use interpolation to resize the output to have the same size as the test image x. After the interpolation, we get a 1024-dimensional feature vector for each pixel of the image. We consider the top 20% pixels in terms of their $s_p^u(x)$ values as foreground pixels and the bottom 20% as background pixels. We then train a logistic regression classifier by using the "fc7" features of the foreground and background pixels. Then we use the trained classifier to label each pixel of test image x as foreground or background. Examples of applying the trained classifiers are shown in Fig. 5.3(c).



(a) (b) (c) (d)

Figure 5.3: Illustration of refinement and GraphCut steps: (a) original image; (b) visualization of $s_p^u(x)$ obtained by knowledge transfer; (c) refined segmentation using the image-specific discriminative appearance model; (d) final segmentation obtained from GraphCut.

Finally, we use GraphCut to further improve the results. The GraphCut algorithm needs the color histograms of the foreground/background of an image in order to define the unary potentials. We take the pixels labeled as foreground (background) by the logistic regression classifier and build the color histograms. Examples of final segmentation results obtained from GraphCut are shown in Fig. 5.3(d).

5.2 Experiments

We use the Microsoft Common Objects in Context (MS COCO) dataset [30] as the source object dataset and consider two different target object datasets: the ImageNet-445 dataset used in [17] and the Cross-Category Object Recognition (CORE) dataset used in [11; 59]. In the following, we first describe the experiment setup in Sec. 5.2.1, then present the results in Sec. 5.2.2.

5.2.1 Experiment Setup

The MS COCO dataset contains images of 80 object categories. All images are annotated with ground-truth semantic segmentation labels. We consider these 80 objects as the source objects and train segmentation models using the MS COCO dataset. We use the training images from MS COCO dataset to train the "Deeplab COCO LargeFOV" model from [7]. We use the default parameters in [7] for the learning.

We then transfer the segmentation models to segment images of target objects using either the word vector or ImageNet hierarchy based distance between objects. The free parameter M is set to be $M = 15$ in our experiments. We use the average interaction-over-union (IoU) [10] to measure the performance and compare our approach with the following baseline methods.

GrabCut image center: This baseline considers an initial window with a rectangle of 25% area of the whole image centered at the image center. Based on this initial window, it then uses GrabCut to segment an image into foreground and background. This baseline method has been also used in [25].

Distance: Given an image x of a target class u, this baseline first finds the closest source object class (based on either word vector or ImageNet hierarchy distance). In other words, this baseline considers $s_p^u(x)$ as $s_p^u(x) = C_p^{r_u^1}(x)$. Then we use the median of the scores of pixels in the image as a threshold and mark a pixel as foreground if its score is great than the threshold and mark it as background otherwise.

We also consider two baselines that are stripped down versions of our approach.

Transfer only: This baseline is similar to our approach, but without the post-

processing in Sec. 5.1.4. After getting the score $s_p^u(x)$ for each pixel p in the image x indicating how likely it belongs to the target object u, we simply take the median of the scores of all pixels in the image as the threshold. A pixel is marked as foreground if its score is greater than the threshold.

Transfer + refinement: This is similar to our approach, but without the final GraphCut step.

5.2.2 Results

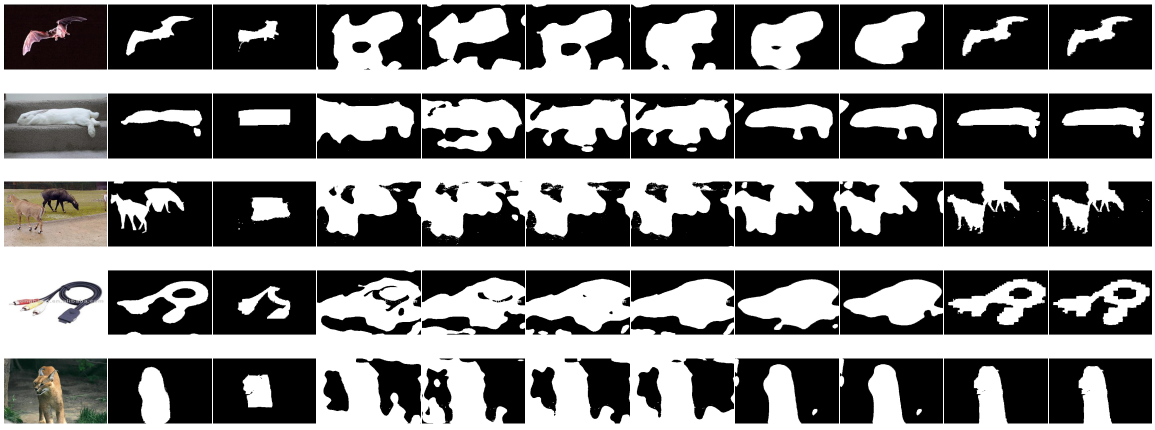
We consider two datasets as target objects and present results on them.

ImageNet-445: The ImageNet-445 dataset [17] contains 4276 images of 445 classes from ImageNet. There are two overlapping classes (cow and tennis racket) between these 445 object classes and the 80 object classes in the MS COCO dataset. We remove these two classes from the target object set. We have used both the word vectors and the ImageNet hierarchy to represent the semantic distance between object classes. For the word vectors, we extract a 300-dimensional vector corresponding to the name of each object class using GloVe [41]. The results on this dataset are shown in Table 5.1. We can see that our approach outperforms other baseline methods. Figure 5.4 shows some qualitative examples on this dataset.

CORE: We also apply our approach on the CORE dataset [11; 59]. The dataset contains 1049 images of 27 object classes. Ten of these object classes also appear in the MS COCO dataset. We remove these ten object classes from the set of source objects when doing the knowledge transfer. The results on this dataset are shown in Table 5.2. Again, our proposed approach outperforms other baseline methods.

Approach		Avg IoU (%)
GrabCut image center		35.04
distance	Word vector	42.46
	ImageNet hierarchy	43.89
transfer only	Word vector	45.73
	ImageNet hierarchy	47.52
transfer + refinement	Word vector	49.50
	ImageNet hierarchy	51.61
our	Word vector	**53.63**
	ImageNet hierarchy	**55.65**

Table 5.1: Segmentation results on the ImageNet-445 dataset. We compare our approach with several baselines in terms of the average interaction-over-union (average IoU). We consider both word vector and ImageNet hierarchy distances in our approach and the baseline approaches.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k)

Figure 5.4: Quantitative results on the ImageNet-445 dataset. (a) input image; (b) ground truth object segmentation; (c) GrabCut image center; (d) distance (word vector); (e) distance (ImageNet hierarchy); (f) transfer only (word vector); (g) transfer only (ImageNet hierarchy); (h) transfer + refinement (word vector); (i) transfer + refinement (ImageNet hierarchy); (j) our (word vector); (k) our (ImageNet hierarchy).

Figure 5.5 shows some qualitative results on this dataset.

Approach		Avg IoU (%)
GrabCut image center		38.77
distance	Word vector	31.44
	ImageNet hierarchy	33.20
transfer only	Word vector	33.72
	ImageNet hierarchy	33.56
transfer + refinement	Word vector	37.98
	ImageNet hierarchy	37.82
our	Word vector	**44.94**
	ImageNet hierarchy	**44.24**

Table 5.2: Segmentation results on the CORE dataset. We compare our approach with several baselines in terms of the average interaction-over-union (average IoU). We consider both word vector and ImageNet hierarchy distances in our approach and the baseline approaches.

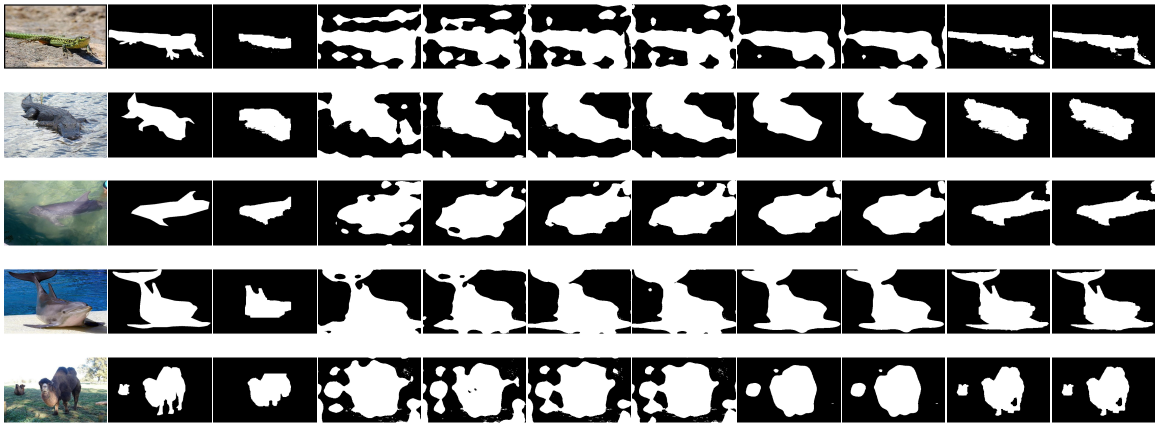


Figure 5.5: Qualitative results on the CORE dataset. (a) input image; (b) ground truth object segmentation; (c) GrabCut image center; (d) distance (word vector); (e) distance (ImageNet hierarchy); (f) transfer only (word vector); (g) transfer only (ImageNet hierarchy); (h) transfer + refinement (word vector); (i) transfer + refinement (ImageNet hierarchy); (j) our (word vector); (k) our (ImageNet hierarchy).

Chapter 6

Conclusion

In this thesis, we discuss a relatively new paradigm of machine learning called "zero-shot learning". We explore three different applications of zero-shot learning in the area of computer vision. We analyse the problems and propose novel solutions for them.

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 thesis, 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.

Videos provide a rich source of visual data. Most previous work in video understanding focuses on simple action classification. In this thesis, we have proposed an approach for learning a joint model of videos and textual descriptions. We have considered textual descriptions in the forms of sentences or phrases.We have demon-

strated our approach in several applications: sentence retrieval given a video, video retrieval given a sentence, zero-shot action recognition in videos. Our experimental results demonstrate that the proposed method provides an effective way of capturing the relationship of videos and their corresponding linguistic descriptions.

Finally, we have proposed a zero-shot learning approach for object figure-ground segmentation. Our approach learns the segmentation models for a set of source objects, then transfers the knowledge from source objects to target objects. This transfer learning allows us to segment target objects even when we have never seen images of target objects during training. Our experimental results demonstrate that our approach outperforms other alternative methods.

We believe our research will pave the way to formalize higher level human perception capabilities and help us to process numerous amount of existing visual data with minimal human supervision.

Bibliography

[1] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[2] S. Bell, P. Upchurch, N. Snavely, and K. Bala. Material recognition in the wild with the materials in context database. *arXiv preprint arXiv:1412.0623*, 2014.

[3] I. Biederman. Recognition by components: A theory of human image understanding. *Psychological Review*, 94(2):115–147, 1987.

[4] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *IEEE International Conference on Computer Vision*, 2005.

[5] O. Chapelle, Q. Le, and A. Smola. Large margin optimization of ranking measures. In *NIPS Workshop on Learning to Rank*, 2007.

[6] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[7] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic

image segmentation with deep convolutional nets and fully connected CRFs. In *International Confernece on Learning Representations*, 2015.

[8] 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.

[9] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *arXiv preprint arXiv:1411.4734*, 2014.

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[11] A. Farhadi, I. Endres, and D. Hoiem. Attribute-centric recognition for cross-category generalization. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.

[12] 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.

[13] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, April 2006.

[14] 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.

[15] C. Gan, M. Lin, Y. Yang, Y. Zhuang, and A. G. Hauptmann. Exploring semantic inter-class relationships (sir) for zero-shot action recognition. In *AAAI Conference on Artificial Intelligence*, 2015.

[16] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, and K. Saenko. YouTube2Text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *IEEE International Conference on Computer Vision*, 2013.

[17] M. Guillaumin, D. Kuettel, and V. Ferrari. Imagenet auto-annotation with segmentation propagation. *International Journal of Computer Vision*, 2014.

[18] M. Guillaumin, D. Küttel, and V. Ferrari. Imagenet auto-annotation with segmentation propagation. *International Journal of Computer Vision*, 110(3):328–348, 2014.

[19] 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.

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

[21] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei.

Large-scale video classification with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[22] R. Kiros, Y. Zhu, R. Salakhutdinov, R. Zemel, A. Torralba, R. Urtasun, and S. Fidler. Skip-through vectors. In *Arxiv*, 2015.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

[24] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A large video database for human motion recognition. In *IEEE International Conference on Computer Vision*, 2011.

[25] D. Kuettel, M. Guillaumin, and V. Ferrari. Segmentation propagation in imagenet. In *European Conference on Computer Vision*, 2012.

[26] 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.

[27] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

[28] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognition natural scene categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

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

[30] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hayes, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, 2014.

[31] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos"in the wild". In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2009.

[32] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *arXiv preprint arXiv:1411.4038*, 2014.

[33] W.-L. Lu, J.-A. Ting, J. J. Little, and K. P. Murphy. Learning to track and identify players from broadcast sports videos. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(7):1704–1716, 2013.

[34] T. Malisiewicz, A. Gupta, A. Efros, et al. Ensemble of exemplar-svms for object detection and beyond. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 89–96. IEEE, 2011.

[35] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2009.

[36] 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.

[37] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. In *International Conference on Learning Representations*, 2014.

[38] D. Osherson, J. Stern, O. Wilkie, M. Stob, and E. E. Smith. Default probability. *Cognitive Science*, 15(2), 1001.

[39] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation.

[40] D. Pathak, E. Shelhamer, J. Long, and T. Darrell. Fully convolutional multi-class multiple instance learning. *arXiv preprint arXiv:1412.7144*, 2014.

[41] J. Pennington, R. Socher, and C. D. Manning. GloVe: Global vector for word representation. In *Conference on Empirical Methods in Natural Language Processing*, 2014.

[42] P. O. Pinheiro and R. Collobert. From image-level to pixel-level labeling with convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1713–1721, 2015.

[43] D. Ramanan, D. A. Forsyth, and A. Zisserman. Tracking people by learning their appearance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):65–81, January 2007.

[44] M. Rochan and Y. Wang. Weakly supervised localization of novel objects us-

ing appearance transfer. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

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

[46] 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.

[47] A. Rosenfeld and D. Weinshall. Extracting foreground masks towards object recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1371–1378. IEEE, 2011.

[48] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterative graph cuts. In *SIGGRAPH*, 2004.

[49] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local SVM approach. In *IEEE International Conference on Pattern Recognition*, volume 3, pages 32–36, 2004.

[50] R. Socher, D. Chen, C. D. Manning, and A. Y. Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*. MIT Press, 2013.

[51] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human action classes from videos in the wild. Technical Report CRCV-TR-12-01, UCF, 2012.

[52] T. Tommasi, T. Tuytelaars, and B. Caputo. A testbed for cross-dataset analysis. Technical report, arXiv: 1402.5923, 2014.

[53] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatial-temporal features with 3d convolutional networks. Arxiv, 2015.

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

[55] H. Wang and C. Schmid. Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision*, 2013.

[56] S. Wu, S. Bondugula, F. Luisier, X. Zhuang, and P. Natarajan. Zero-shot event detection using multi-modal fusion of weakly supervised concepts. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[57] R. Xu, C. Xiong, W. Chen, and J. J. Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *AAAI*, 2015.

[58] H. Yu and J. M. Siskind. Grounded language learning from video described with sentences. In *Proceedings of ACL*, 2013.

[59] S. Zheng, M.-M. Cheng, J. Warrell, P. Sturgess, V. Vineet, C. Rother, and P. H. S. Torr. Dense semantic image segmentation with objects and attributes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.