2016 23rd International Conference on Pattern Recognition (ICPR)
Cancún Center, Cancún, México, December 4-8, 2016

Object Figure-Ground Segmentation Using Zero-Shot Learning

Shujon Naha
Department of Computer Science
University of Manitoba
Winnipeg, MB, Canada
Email: shujon@cs.umanitoba.ca

Yang Wang
Department of Computer Science
University of Manitoba
Winnipeg, MB, Canada
Email: ywang@cs.umanitoba.ca

***Abstract*—We consider the problem of object figure-ground segmentation when the object categories are not available during training (i.e. zero-shot). During training, we learn standard segmentation models for a handful of object categories (called "source objects") using existing semantic segmentation datasets. During testing, we are given images of objects (called "target objects") that are unseen during training. Our goal is to segment the target objects from the background. Our method learns to transfer the knowledge from the source objects to the target objects. Our experimental results demonstrate the effectiveness of our approach.**

I. Introduction

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 [1]) 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. However, 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 [2] 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 resource and has enabled the recent deep learning revolution [3] 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 [4] 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 goal is to transfer the segmentation models from the source objects to the target objects.

Our problem setup is illustrated in Fig 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.

Previous zero-shot learning work in computer vision mainly focuses on object classification. The main contribution and novelty of our work is that we apply zero-shot learning to object segmentation, which arguably is a more challenging problem.

II. Related Works

Previous work has explored both interactive and fully automatic methods for object segmentation. GrabCut [1] is an example of the interactive object segmentation. It requires the user to provide an initial bounding box of the object of interest in the image. Fully automatic object segmentation typically requires learning segmentation models from annotated training data. Early work (e.g. [5]) focuses on single object segmentation. The goal is to generate a binary mask that separates the object of interest from the background. Recent work in semantic segmentation (e.g. [6]) focuses on multi-class object segmentation. The goal is to assign a label from a predefined set of object classes to each pixel in an image.

978-1-5090-4846-5/16/$31.00 ©2016 IEEE

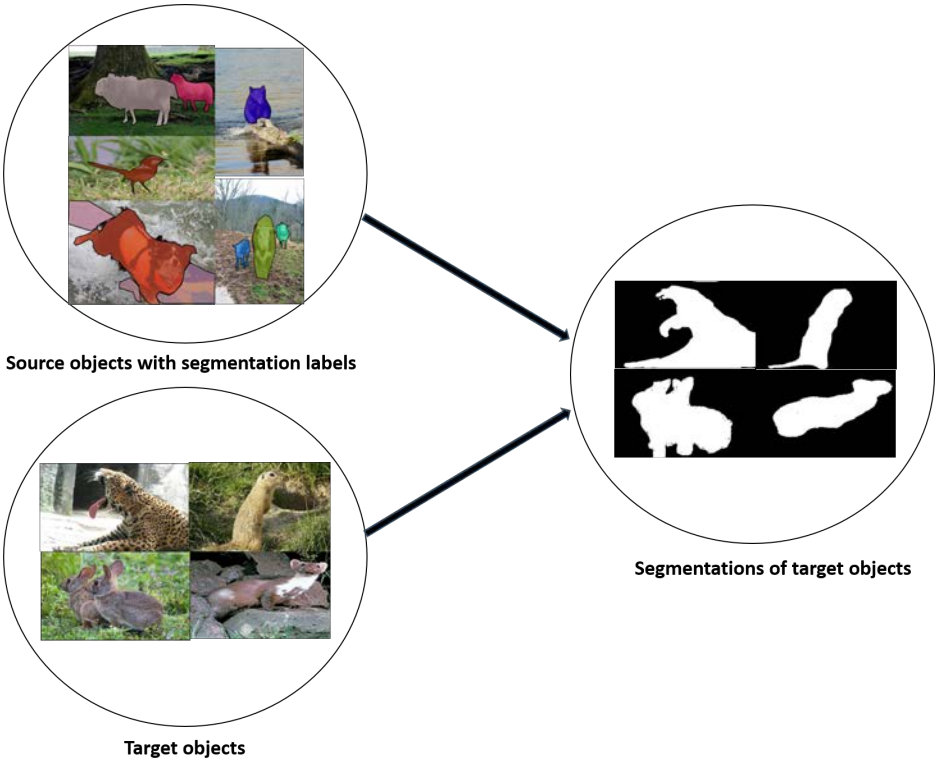


Fig. 1. An illustration of our problem setup. Our training data consist of images of source objects. The pixel-level semantic labels are available on the training data. Our test data consist of images of target objects, where we know the object label of each image, but we do not have the pixel-level 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.

In computer vision, there is a line of work on learning new object classes by transferring knowledge from related object classes. Most work in this area focuses on object classification. For example, there has been work on knowledge transfer for object recognition by sharing parameters [7], learning intermediate attributes [8], etc. Some recent work [9] uses word vectors to transfer the knowledge among related objects. There is also some work on using transfer learning for detection, e.g. by sharing parts [10] or domain adaptation [11].

The closest work to ours is the work on segmentation propagation in ImageNet [12]. It populates the ImageNet with pixel-level segmentations by exploiting existing annotations in the form of class labels and bounding-boxes. In contrast, our work does not require bounding boxes for any unknown classes.

III. Our Approach

Our approach consists of several steps. First, we build segmentation models for source object classes by learning standard semantic segmentation models (Sec. III-A). For a target object, we propose two approaches for measuring the semantic distances between this target object and all the source objects (Sec. III-B). 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. III-C). Finally, we use the transferred scores to obtain the figure-ground segmentation of the target object in this image (Sec. III-D).

A. Segmentation Models for Source Objects

The first step of our approach is to build segmentation models for source objects. We use the approach in [13] to train the segmentation models using images 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 scores for each pixel being one of the source object classes.

B. 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 paper, 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 [9], [14]. 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 on the list.

C. Knowledge Transfer

During testing, we are given an image of one of the target object classes. We assume that we known 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. III-A) 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. III-A to segment the target object.

Figure 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. III-A on this image. For each pixel p in the image x, we will get a 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. 2(a)). The top-5 ranked source objects to "dolphin" might be "bear", "mouse", "bird", "horse", "zebra". Figure 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:

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

where M is a free parameter and $M \leq K$. The intuition of Eq. 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 2(c) shows the visualization of $s_p^u(x)$.

D. Segmenting Target Objects

After the knowledge transfer in Sec. III-C, 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 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 [15] 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. 3(c).

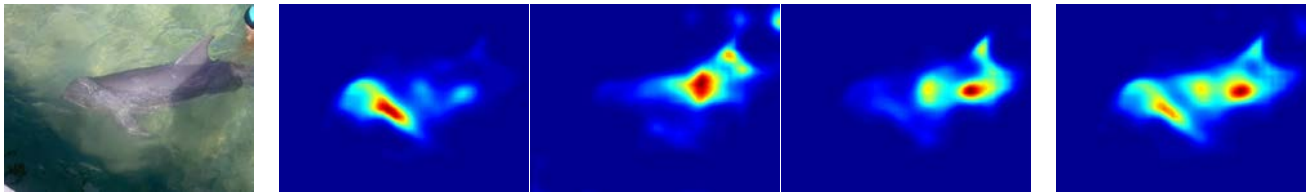
Finally we use GraphCut to further improve the results. The GraphCut algorithm needs the color histograms of the foreground/background of an image in order to define the unary potentials. We take the pixels labeled as foreground (background) by the logistic regression classifier and build the color histograms. Examples of final segmentation results obtained from GraphCut are shown in Fig. 3(d).

IV. Experiments

We use the Microsoft Common Objects in Context (MS COCO) dataset [4] as the source object dataset and consider two different target object datasets: the ImageNet-445 dataset used in [16] and the Cross-Category Object Recognition (CORE) dataset used in [17], [18]. In the following, we first describe the experiment setup in Sec. IV-A, then present the results in Sec. IV-B.

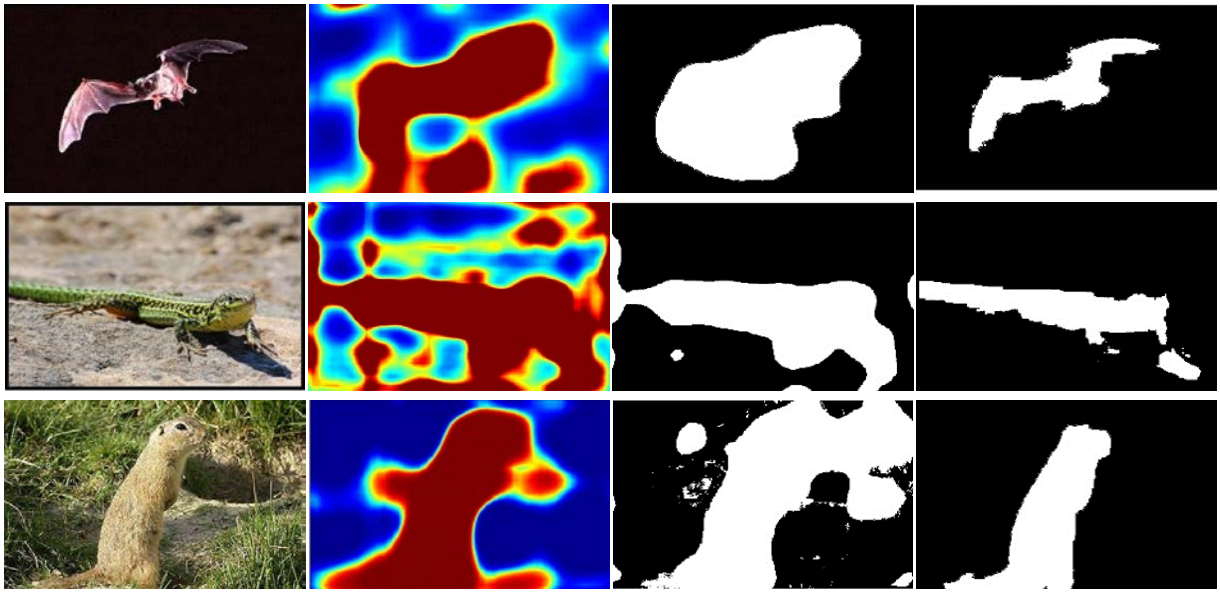
A. Experiment Setup

The MS COCO dataset contains images of 80 object categories. All images are annotated with ground-truth semantic



(a) (b) (c)

Fig. 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 $s_p^u(x)$.



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

Fig. 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.

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 the MS COCO dataset to train the "Deeplab COCO LargeFOV" model from [13]. We use the default parameters in [13] 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) [19] 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 also been used in [12].

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. III-D. 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.

B. Results

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

ImageNet-445: The ImageNet-445 dataset [16] 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

TABLE I

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.

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
ours	Word vector	**53.63**
	ImageNet hierarchy	**55.65**

TABLE II

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.

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
ours	Word vector	**44.94**
	ImageNet hierarchy	**44.24**

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 [20]. The results on this dataset are shown in Table I. We can see that our approach outperforms other baseline methods. Figure 4 shows some qualitative examples on this dataset.

CORE: We also apply our approach on the CORE dataset [17], [18]. 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 II. Again, our proposed approach outperforms other baseline methods. Figure 5 shows some qualitative results on this dataset.

V. CONCLUSION

In this paper, 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.

ACKNOWLEDGMENT

This work is supported by NSERC. We thank NVIDIA for donating the GPUs used in this work.

REFERENCES

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

[2] 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.

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

[4] 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.

[5] E. Borenstein, E. Sharon, and S. Ullman, "Combining top-down and bottom-up segmentation," in *CVPR Workshop*, 2004.

[6] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation," in *European Conference on Computer Vision*, 2006.

[7] 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.

[8] 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.

[9] 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.

[10] P. Ott and M. Everingham, "Shared parts for deformable part-based models," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

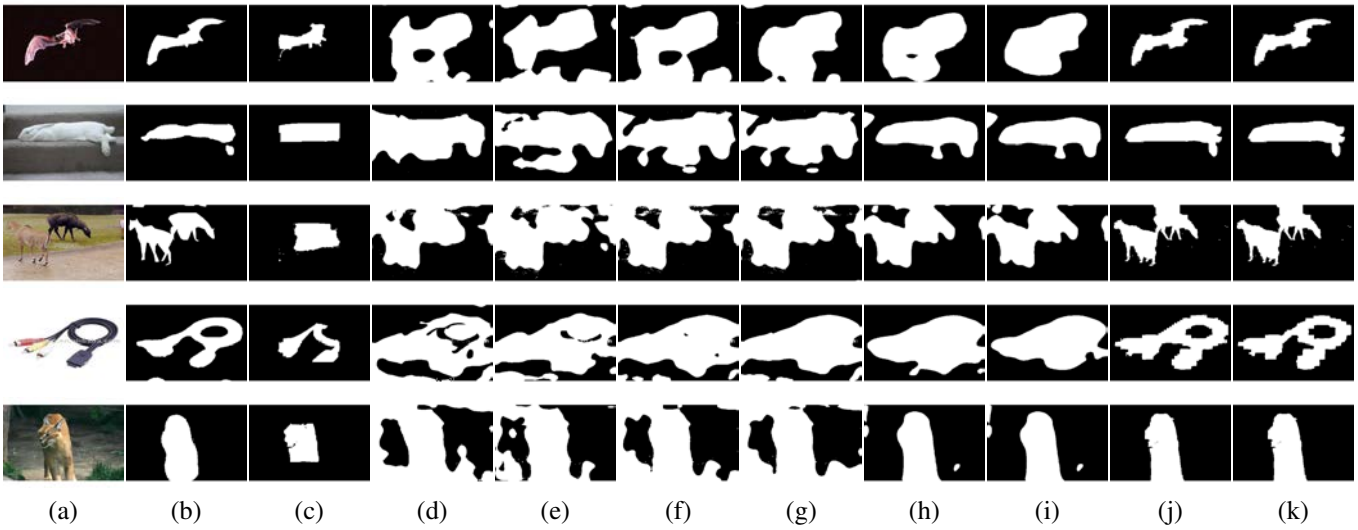


Fig. 4. Quantitative results on the ImageNet-445 dataset. (a) input image; (b) ground truth object segmentation; (c) GrabCut image center; (d) distance (word vector); (e) distance (ImageNet hierarchy); (f) transfer only (word vector); (g) transfer only (ImageNet hierarchy); (h) transfer + refinement (word vector); (i) transfer + refinement (ImageNet hierarchy); (j) ours (word vector); (k) ours (ImageNet hierarchy).

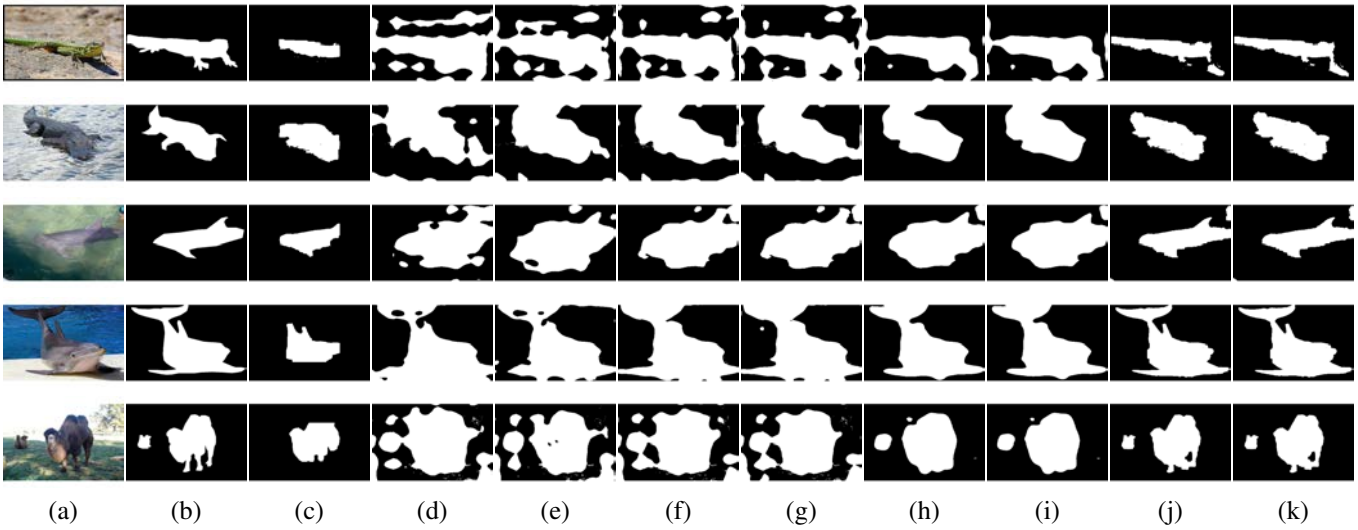


Fig. 5. Quantitative results on the CORE dataset. (a) input image; (b) ground truth object segmentation; (c) GrabCut image center; (d) distance (word vector); (e) distance (ImageNet hierarchy); (f) transfer only (word vector); (g) transfer only (ImageNet hierarchy); (h) transfer + refinement (word vector); (i) transfer + refinement (ImageNet hierarchy); (j) ours (word vector); (k) ours (ImageNet hierarchy).

[11] J. Hoffman, S. Guadarrama, E. S. Tzeng, J. Donahue, T. Darrell, K. Saenko, and R. B. Girshick, "LSDA: Large scale detection through adaptation," in *Advances in Neural Information Processing Systems*. MIT Press, 2014.

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

[13] 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.

[14] S. Naha and Y. Wang, "Zero-shot object recognition using semantic label vectors," in *Conference on Computer and Robot Vision*, 2015.

[15] D. Ramanan, D. A. Forsyth, and A. Zisserman, "Tracking people by learning their appearance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 65–81, January 2007.

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

[17] 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.

[18] 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.

[19] 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*, vol. 88, no. 2, pp. 303–338, 2010.

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