

Zero-Shot Object Detection: Learning to Simultaneously Recognize and Localize Novel Concepts

Shafin Rahman^{†*}, Salman Khan^{*†} and Fatih Porikli[†]

[†]Australian National University

^{*}DATA61, CSIRO

Abstract. Current Zero-Shot Learning (ZSL) approaches are restricted to recognition of a single dominant unseen object category in a test image. We hypothesize that this setting is ill-suited for real-world applications where unseen objects appear only as a part of a complex scene, warranting both the ‘recognition’ and ‘localization’ of an unseen category. To address this limitation, we introduce a new ‘Zero-Shot Detection’ (ZSD) problem setting, which aims at simultaneously recognizing and locating object instances belonging to novel categories without any training examples. We also propose a new experimental protocol for ZSD based on the highly challenging ILSVRC dataset, adhering to practical issues, e.g., the rarity of unseen objects. To the best of our knowledge, this is the first end-to-end deep network for ZSD that jointly models the interplay between visual and semantic domain information. To overcome the noise in the automatically derived semantic descriptions, we utilize the concept of meta-classes to design an original loss function that achieves synergy between max-margin class separation and semantic space clustering. Furthermore, we present a baseline approach extended from recognition to detection setting. Our extensive experiments show significant performance boost over the baseline on the imperative yet difficult ZSD problem.

Keywords: Zero-shot learning, Object detection, Zero-shot detection

1 Introduction

Since its inception, zero-shot learning research has been dominated by the object classification problem [2,5,10,18,21,22,29,33,39,47,51,52,53]. Although it still remains as a challenging task, the zero-shot recognition has a number of limitations that render it unusable in real-life scenarios. *First*, it is destined to work for simpler cases where only a single dominant object is present in an image. *Second*, the attributes and semantic descriptions are relevant to individual objects instead of the entire scene composition. *Third*, zero-shot recognition provides an answer to unseen categories in elementary tasks, e.g., classification and retrieval, yet it is unable to scale to advanced tasks such as scene interpretation and contextual modeling, which require a fundamental reasoning about all salient objects

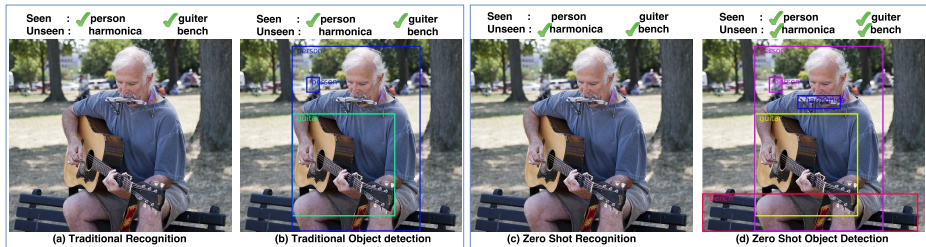


Fig. 1. ZSD deals with a more complex label space (object labels and locations) with considerably less supervision (i.e., no examples of unseen classes). (a) Traditional recognition task only predicts seen class labels. (b) Traditional detection task predicts both seen class labels and bounding boxes. (c) Traditional zero-shot recognition task only predicts unseen class labels. (d) The proposed ZSD predicts both seen and unseen classes and their bounding boxes.

in the scene. *Fourth*, global attributes are more susceptible to background variations, viewpoint, appearance and scale changes and practical factors such as occlusions and clutter. As a result, image-level ZSL fails for the case of complex scenes where a diverse set of competing attributes that do not belong to a single image-level category would exist.

To address these challenges, we introduce a new problem setting called the *zero-shot object detection*. As illustrated in Fig. 1, instead of merely classifying images, our goal is to simultaneously detect and localize each individual instance of new object classes, even in the absence of any visual examples of those classes during the training phase. In this regard, we propose a new zero-shot detection protocol built on top of the ILSVRC - Object Detection Challenge [40]. The resulting dataset is very demanding due to its large scale, diversity, and unconstrained nature, and also unique due to its leveraging on WordNet semantic hierarchy [31]. Taking advantage of semantic relationships between object classes, we use the concept of ‘*meta-classes*’¹ and introduce a novel approach to update the semantic embeddings automatically. Raw semantic embeddings are learned in an unsupervised manner using text mining and therefore they have considerable noise. Our optimization of the class embeddings proves to be an effective way to reduce this noise and learn robust semantic representations.

ZSD has numerous applications in novel object localization, retrieval, tracking, and reasoning about object’s relationships with its environment using only available semantics, e.g., an object name or a natural language description. Although a critical problem, ZSD is remarkably difficult compared to its classification counterpart. While the zero-shot recognition problem assumes only a single primary object in an image and attempts to predict its category, the ZSD task has to predict both the multi-class category label and precise location of each instance in the given image. Since there can be a prohibitively huge number of possible locations for each object in an image and because the semantic class descriptions are noisy, a detection approach is much more susceptible to

¹ Meta-classes are obtained by clustering semantically similar classes.

incorrect predictions compared to classification. Therefore, it would be expected that a ZSD method predicts a class label that might be incorrect but visually and semantically similar to the corresponding true class. For example, wrongly predicting a ‘spider’ as ‘scorpion’ where both are semantically similar because of being invertebrates. To address this issue, we relax the original detection problem to independently study the confusions emanating from the visual and semantic resemblance between closely linked classes. For this purpose, alongside the ZSD, we evaluate on zero-shot meta-class detection, zero-shot tagging, and zero-shot meta class tagging. Notably, the proposed network is trained only ‘once’ for ZSD task and the additional tasks are used during evaluations only.

Although deep network based solutions have been proposed for zero-shot recognition [10,22,51], to the best of our knowledge, we propose the first end-to-end trainable network for the ZSD problem that concurrently relates visual image features with the semantic label information. This network considers semantic embedding vector of classes as a fixed embedding within the network to produce prediction scores for both seen and unseen classes. We propose a novel loss formulation that incorporates max-margin learning [53] and a semantic clustering loss based on class-scores of different meta-classes. While the max-margin loss tries to separate individual classes, semantic clustering loss tries to reduce the noise in semantic vectors by positioning similar classes together and dissimilar classes far apart. Notably, our proposed formulation assumes predefined unseen classes to explore the semantic relationships during model learning phase. This assumption is consistent with recent efforts in the literature which consider class semantics to solve the domain shift problem in ZSL [7,12] and does not constitute transductive setting [8,11,18]. Based on the premise that unseen class semantics may be unknown during training in several practical zero-shot scenarios, we also propose a variant of our approach that can be trained without predefined unseen classes. Finally, we propose a comparison method for ZSD by extending a popular zero-shot recognition framework named ConSE [33] using Faster-RCNN [38].

In summary, this paper reports the following advances:

- We introduce a new problem for zero-shot learning, which aims to jointly recognize and localize novel objects in complex scenes.
- We present a new experimental protocol and design a novel baseline solution extended from conventional recognition to the detection task.
- We propose an end-to-end trainable deep architecture that simultaneously considers both visual and semantic information.
- We design a novel loss function that achieves synergistic effects for max-margin class separation and semantic clustering based on meta-classes. Beside that, our approach can automatically tune noisy semantic embeddings.

2 Problem Description

Given a set of images for seen object categories, ZSD aims at the *recognition* and *localization* of previously unseen object categories. In this section, we formally

describe the ZSD problem and its associated challenges. We also introduce variants of the detection task, which are natural extensions of the original problem. First, we describe the notations used in the following discussion.

Preliminaries: Consider a set of ‘seen’ classes denoted by $\mathcal{S} = \{1, \dots, S\}$, whose examples are available during the training stage and S represents their total number. There exists another set of ‘unseen’ classes $\mathcal{U} = \{S+1, \dots, S+U\}$, whose instances are only available during the test phase. We denote the set of all object classes by $\mathcal{C} = \mathcal{S} \cup \mathcal{U}$, such that $C = S + U$ denote the cardinality of the label space.

We define a set of meta (or super) classes by grouping similar object classes into a single meta category. These meta-classes are denoted by $\mathcal{M} = \{z_m : m \in [1, M]\}$, where M denote the total number of meta-classes and $z_m = \{k \in \mathcal{C} \text{ s.t.}, g(k) = m\}$. Here, $g(k)$ is a mapping function which maps each class k to its corresponding meta-class $z_{g(k)}$. Note that the meta-classes are mutually exclusive i.e., $\cap_{m=1}^M z_m = \phi$ and $\cup_{m=1}^M z_m = \mathcal{C}$.

The set of all training images is denoted by \mathcal{X}^s , which contains examples of all seen object classes. The set of all test images containing samples of unseen object classes is denoted by \mathcal{X}^u . Each test image $\mathbf{x} \in \mathcal{X}^u$ contains at least one instance of an unseen class. Notably, no unseen class object is present in \mathcal{X}^s , but \mathcal{X}^u may contain seen objects.

We define a d dimensional word vector \mathbf{v}_c (word2vec or GloVe) for every class $c \in \mathcal{C}$. The ground-truth label for an i^{th} bounding box is denoted by y_i . The object detection task also involves identifying the background class for negative object proposals, we introduce the extended label sets: $\mathcal{S}' = \mathcal{S} \cup y_{bg}$, $\mathcal{C}' = \mathcal{C} \cup y_{bg}$ and $\mathcal{M}' = \mathcal{M} \cup y_{bg}$, where $y_{bg} = \{C+1\}$ is a singleton set denoting the background label.

Task Definitions: Given the observed space of images $\mathcal{X} = \mathcal{X}^s \cup \mathcal{X}^u$ and the output label space \mathcal{C}' , our goal is to learn a mapping function $f : \mathcal{X} \mapsto \mathcal{C}'$ which gives the minimum regularized empirical risk ($\hat{\mathcal{R}}$) as follows:

$$\arg \min_{f \in \mathcal{F}} \hat{\mathcal{R}}(f(\mathbf{x}; \theta)) + \Omega(\theta), \quad (1)$$

where, $\mathbf{x} \in \mathcal{X}^s$ during training, θ denotes the set of parameters and $\Omega(\theta)$ denotes the regularization on the learned weights. The mapping function has the following form:

$$f(\mathbf{x}; \theta) = \arg \max_{y \in \mathcal{C}'} \max_{b \in \mathcal{B}(\mathbf{x})} \mathcal{F}(\mathbf{x}, y, b; \theta), \quad (2)$$

where $\mathcal{F}(\cdot)$ is a compatibility function, $\mathcal{B}(\mathbf{x})$ is the set of all bounding box proposals in a given image \mathbf{x} . Intuitively, Eq. 2 finds the best scoring bounding boxes for each object category and assigns them the maximum scoring object category. Next, we define the zero-shot learning tasks which go beyond a single unseen category recognition in images. Notably, the training is framed as the challenging ZSD problem, however the remaining task descriptions are used during evaluation to relax the original problem:

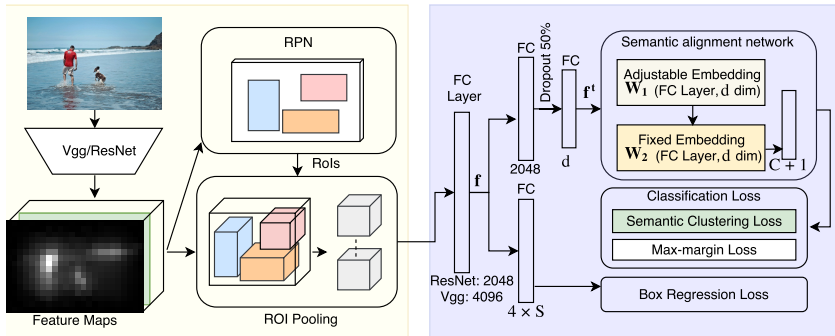


Fig. 2. Network Architecture - *Left*: Image level feature maps are used to propose candidate object boxes and their corresponding features. *Right*: The features are used for classification and localization of new classes by utilizing their semantic concepts.

- T1** *Zero-shot detection (ZSD)*: Given a test image $\mathbf{x} \in \mathcal{X}^u$, the goal is to categorize and localize each instance of an unseen object class $u \in \mathcal{U}$.
- T2** *Zero-shot meta-class detection (ZSMD)*: Given a test image $\mathbf{x} \in \mathcal{X}^u$, the goal is to localize each instance of an unseen object class $u \in \mathcal{U}$ and categorize it into one of the super-classes $m \in \mathcal{M}$.
- T3** *Zero-shot tagging (ZST)*: To recognize one or more unseen classes in a test image $\mathbf{x} \in \mathcal{X}^u$, without identifying their location.
- T4** *Zero-shot meta-class tagging (ZSMT)*: To recognize one or more meta-classes in a test image $\mathbf{x} \in \mathcal{X}^u$, without identifying their location.

Among the above mentioned tasks, the ZSD is the most difficult problem and difficulty level decreases as we go down the list. The goal of the later tasks is to distill the main challenges in ZSD by investigating two ways to relax the original problem: **(a)** The effect of reducing the unseen object classes by clustering similar unseen classes into a single super-class (T2 and T4). **(b)** The effect of removing the localization constraint. To this end we investigate the zero-shot tagging problem, where the goal is to only recognize all object categories in an image (T3 and T4).

The state-of-the-art in zero-shot learning deals with only recognition/tagging. The proposed problem settings add the missing detection task which indirectly encapsulates traditional recognition and tagging task.

3 Zero-Shot Detection

Our proposed model uses Faster-RCNN [38] as a backbone architecture, due to its superior performance among competitive end-to-end detection models [17,28,37]. We first provide an overview of our proposed model architecture and then discuss network learning. Finally, we extend a popular ZSL approach to the detection problem, against which we compare our performance in the experiments.

3.1 Model Architecture

The overall architecture is illustrated in Fig 2. It has two main components marked in color: the first provides object-level feature descriptions and the second integrates visual information with the semantic embeddings to perform zero-shot detection. We explain these in detail next.

Object-level Feature Encoding: For an input image \mathbf{x} , a deep network (VGG or ResNet) is used to obtain the intermediate convolutional activations. These activations are treated as feature maps, which are forwarded to the Region Proposal Network (RPN). The RPN generates a set of candidate object proposals by automatically ranking the anchor boxes at each sliding window location. The high-scoring candidate proposals can be of different sizes, which are mapped to fixed sized representation using a RoI pooling layer which operates on the initial feature maps and the proposals generated by the RPN. The resulting object level features for each candidate are denoted as ‘ \mathbf{f} ’. Note that the RPN generates object proposal based on the objectness measure. Thus, a trained RPN on seen objects can generate proposals for unseen objects also. In the second block of our architecture, these feature representations are used alongside the semantic embeddings to learn useful representations for both the seen and unseen object-categories.

Integrating Visual and Semantic Contexts: The object-level feature \mathbf{f} is forwarded to two branches in the second module. The **top branch** is trained to predict the object category for each candidate box. Note that this can assign a class $c \in \mathcal{C}'$, which can be a seen, unseen or background category. The branch consists of two main sub-networks, which are key to learning the semantic relationships between seen and unseen object classes.

The first component is the ‘*Semantic Alignment Network*’ (SAN), which consist of an adjustable FC layer, whose parameters are denoted as $\mathbf{W}_1 \in \mathbb{R}^{d \times d}$, that projects the input visual feature vectors to a semantic space with d dimensions. The resulting feature maps are then projected onto the **fixed** semantic embeddings, denoted by $\mathbf{W}_2 \in \mathbb{R}^{d \times (C+1)}$, which are obtained in an unsupervised manner by text mining (e.g., Word2vec and GloVe embeddings). Note that, here we consider both seen and unseen semantic vectors which require unseen classes to be predefined. This consideration is inline with a very recent effort [12] which adopt this setting to explore the cluster manifold structure of the semantic embedding space and address domain shift issue. Given a feature representation input to SAN in the top branch, \mathbf{f}^t , the overall operation can be represented as:

$$\mathbf{o} = (\mathbf{W}_1 \mathbf{W}_2)^T \mathbf{f}^t. \quad (3)$$

Here, \mathbf{o} is the output prediction score. The \mathbf{W}_2 is formed by stacking semantic vectors for all classes, including the background class. For background class, we use the mean word vectors $\mathbf{v}_b = \frac{1}{C} \sum_{c=1}^C \mathbf{v}_c$ as its embedding in \mathbf{W}_2 .

Notably, a non-linear activation function is not applied between the adjustable and fixed semantic embeddings in the SAN. Therefore, the two projections can be understood as a single learnable projection on to the semantic

embeddings of object classes. This helps in automatically updating the semantic embeddings to make them compatible with the visual feature domain. It is highly valuable because the original semantic embeddings are often noisy due to the ambiguous nature of closely related semantic concepts and the unsupervised procedure used for their calculation. In Fig. 3, we visualize modified embedding space when different loss functions are applied during training.

The **bottom branch** is for bounding box regression to add suitable offsets to the proposals to align them with the ground-truths such that the precise location of objects can be predicted. This branch is set up in the same manner as in Faster-RCNN [38].

3.2 Training and Inference

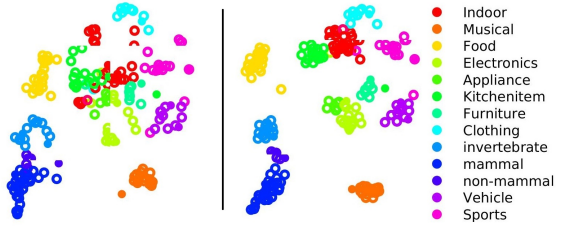
We follow a two step training approach to learn the model parameters. The **first** part involves training the backbone Faster-RCNN for only seen classes using the training set \mathcal{X}^s . This training involves initializing weights of shared layers with a pre-trained Vgg/ResNet model, followed by learning the RPN, classification and detection networks. In the **second** step, we modify the Faster-RCNN model by replacing the last layer of Faster-RCNN classification branch with the proposed semantic alignment network and an updated loss function (see Fig. 2). While rest of the network weights are used from the first step, the weights \mathbf{W}_1 are randomly initialized and the \mathbf{W}_2 are fixed to semantic vectors of the object classes and not updated during training.

While training in second step, we keep the shared layers trainable but fix the layers specific to RPN since the object proposals requirements are not changed from the previous step. The same seen class images \mathcal{X}^s are used for training, consistent with the first step. For each given image, we obtain the output of RPN which consists of a total of ‘R’ ROIs belonging to both positive and negative object proposals. Each proposal has a corresponding ground-truth label given by $y_i \in \mathcal{S}'$. Positive proposals belong to any of the seen class \mathcal{S} and negative proposals contain only background. In our implementation, we use an equal number of positive and negative proposals. Now, when object proposals are passed through ROI-Pooling and subsequent dense layers, a feature representation \mathbf{f}_i is calculated for each ROI. This feature is forwarded to two branches, the classification branch and regression branch. The overall loss is the summation of the respective losses in these two branches, i.e., classification loss and bounding box regression loss.

$$L(\mathbf{o}_i, b_i, y_i, b_i^*) = \arg \min_{\Theta} \frac{1}{T} \sum_i \left(L_{cls}(\mathbf{o}_i, y_i) + L_{reg}(b_i, b_i^*) \right)$$

where Θ denotes the parameters of the network, \mathbf{o}_i is the classification branch output, $T = N \times R$ represents the total number of ROIs in the training set with N images. b_i and b_i^* are parameterized coordinates of predicted and ground-truth bounding boxes respectively and y_i represents the true class label of the i^{th} object proposal.

Fig. 3. The 2D tSNE embedding of modified word vectors $\mathbf{W}_1 \mathbf{W}_2$ using only max-margin loss, L_{mm} (left) and with clustering loss, $L_{mm} + L_{mc}$ (right). Semantically similar classes are embedded more closely in cluster based loss.



Classification loss: This loss deals with both seen and unseen classes. It has two components: a max-margin loss (L_{mm}) and a meta-class clustering loss (L_{mc}).

$$L_{cls}(\mathbf{o}_i, y_i) = \lambda L_{mm}(\mathbf{o}_i, y_i) + (1 - \lambda) L_{mc}(\mathbf{o}_i, g(y_i)), \quad (4)$$

where, λ is a hyper-parameter that controls the trade-off between the two losses. We define,

$$L_{mm}(\mathbf{o}_i, y_i) = \frac{1}{|\mathcal{C}' \setminus y_i|} \sum_{c \in \mathcal{C}' \setminus y_i} \log \left(1 + \exp(o_c - o_{y_i}) \right), \text{ and}$$

$$L_{mc}(\mathbf{o}_i, g(y_i)) = \frac{1}{|\mathcal{M}' \setminus z_{g(y_i)}| |z_{g(y_i)}|} \sum_{c \in \mathcal{M}' \setminus z_{g(y_i)}} \sum_{j \in z_{g(y_i)}} \log \left(1 + \exp(o_c - o_j) \right)$$

where, o_k represents the prediction response of class $k \in \mathcal{S}$. L_{mm} tries to separate the prediction response of true class from rest of the classes. In contrast, L_{mc} tries to cluster together the members of each super-class and pulls further apart the classes belonging to different meta-classes.

We illustrate the effect of clustering loss on the learned embeddings in Fig. 3. The use of L_{mc} enables us to cluster semantically similar classes together which results in improved embeddings in the semantic space. For example, all animals related meta-classes are in close position whereas food and vehicle are far apart. Such a clear separation in semantic space helps in obtaining a better ZSD performance. Moreover, meta-class based clustering loss does not harm fine-grained detection because the hyper-parameter λ is used to put more emphasis on the max-margin loss (L_{mm}) as compared to the clustering part (L_{mc}) of the overall loss (L_{cls}). Still, the clustering loss provides enough guidance to the noisy semantic embeddings (e.g., unsupervised w2v/glove) such that similar classes are clustered together as illustrated in Fig. 3. Note that w2v/glove try to place similar words nearby with respect to millions of text corpus, it is therefore not fine-tuned for just 200 class recognition setting.

Regression loss: This part of the loss is similar to faster-RCNN regression loss which fine-tunes the bounding box for each seen class ROI. For each \mathbf{f}_i , we get $4 \times S$ values representing 4 parameterized co-ordinates of the bounding box of each object instance. The regression loss is calculated based on these co-ordinates and parameterized ground truth co-ordinates. During training, no bounding box

prediction is done for background and unseen classes due to unavailability of visual examples. As an alternate approach, we approximate the bounding box for an unseen object through the box proposal for a closely related seen object that achieves maximum response. This is a reasonable approximation because visual features of unseen classes are related to that of similar seen classes.

Prediction: We normalize each output prediction value of classification branch using $\hat{o}_c = \frac{o_c}{\|\mathbf{v}_c\|_2 \|\mathbf{f}^i\|_2}$. It basically calculates the cosine similarity between modified word vectors and image features. This normalization maps the prediction values within 0 to 1 range. We classify an object proposal as background if maximum responds among \hat{o}_c where $c \in \mathcal{C}'$ belongs to y_{bg} . Otherwise, we detect an object proposal as unseen object if its maximum prediction response among \hat{o}_u where $u \in \mathcal{U}$ is above a threshold α .

$$y_u = \arg \max_{u \in \mathcal{U}} \hat{o}_u \quad s.t., \hat{o}_u > \alpha. \quad (5)$$

The other detection branch finds b_i which is the parameterized co-ordinates of bounding boxes corresponds to S seen classes. Among them, we choose a bounding box corresponding to the class having the maximum prediction response in \hat{o}_s where $s \in \mathcal{S}$ for the classified unseen class y_u . For the tagging tasks, we simply use the mapping function $g(\cdot)$ to assign a meta-class for any unseen label.

3.3 ZSD without Pre-defined Unseen

While applying clustering loss in Sec. 3.2, the meta-class assignment adds high-level supervision in the semantic space. While doing this assignment, we consider both seen and unseen classes. Similarly, the max-margin loss considers the set \mathcal{C}' consisting of both seen and unseen classes. This problem setting helps to identify the clustering structure of the semantic embeddings to address domain adaptation for zero-shot detection. However, in several practical scenarios, unseen classes may not be known during training. Here, we report a simplified variant of our approach to train the proposed network without pre-defined unseen classes.

For this problem setting, we use only seen+bg word vectors (instead of seen+unseen+bg vectors) as the fixed embedding $\mathbf{W}_2 \in \mathbb{R}^{d \times (S+1)}$ to train the whole framework with only the max-margin loss, L'_{mm} , defined as follows: $L'_{mm}(\mathbf{o}_i, y_i) = \frac{1}{|S' \setminus y_i|} \sum_{c \in S' \setminus y_i} \log \left(1 + \exp(o_c - o_{y_i}) \right)$. Since the output classification layer cannot make predictions for unseen classes, we apply a procedure similar to ConSE during the testing phase [33]. The choice of [33] here is made due to two main reasons: **(a)** In contrast to other ZSL methods which train separate models for each class [5,36], ConSE can work on the prediction score of a single end-to-end framework. **(b)** It is straight-forward to extend a single network to ZSD along with ConSE, since [33] uses semantic embeddings only during the test phase.

Suppose, for an object proposal, $\mathbf{o} \in \mathbb{R}^{S+1}$ is the vector containing final probability values of only seen classes and background. As described earlier, we

ignore the object proposal if the background class get highest probability score. For other cases, we sort the vector \mathbf{o} in descending order to compute a list of indices \mathbf{l} and the sorted list $\hat{\mathbf{o}}$:

$$\hat{\mathbf{o}}, \mathbf{l} = \text{sort}(\mathbf{o}) \quad \text{s.t.}, o_j = \hat{o}_{l_j}. \quad (6)$$

Then, top K score values (s.t., $K \leq S$) from $\hat{\mathbf{o}}$ are combined with their corresponding word vectors using the equation: $\mathbf{e}_i = \sum_{k=1}^K \hat{\mathbf{o}}_k \mathbf{v}_{l_k}$. We consider \mathbf{e}_i as a semantic space projection of an object proposal which is a combination of word vectors weighted by top K seen class probabilities. The final prediction is made by finding the maximum cosine similarity among \mathbf{e}_i and all unseen word vectors,

$$y_u = \arg \max_{u \in \mathcal{U}} \cos(\mathbf{e}_i, \mathbf{v}_u).$$

In this paper, we use $K = 10$ as proposed in [33]. For bounding box detection, we choose the box for which corresponding seen class gets maximum score.

4 Experiments

4.1 Dataset and Experiment Protocol

Dataset: We evaluate our approach on the standard ILSVRC-2017 detection dataset [40]. This dataset contains 200 object categories. For training, it includes 456,567 images and 478,807 bounding box annotations around object instances. The validation dataset contains 20,121 images fully annotated with the 200 object categories which include 55,502 object instances. A category hierarchy has been defined in [40], where some objects have multiple parents. Since, we also evaluate our approach on meta-class detection and tagging, we define a single parent for each category (see supplementary material for detail).

Seen/unseen split: Due to lack of an existing ZSD protocol, we propose a challenging seen/unseen split for ILSVRC-2017 detection dataset. Among 200 object categories, we randomly select 23 categories as unseen and rest of the 177 categories are considered as seen. This split is designed to follows the following practical considerations: (a) unseen classes are rare, (b) test categories should be diverse, (c) the unseen classes should be semantically similar with at least some of the seen classes. The details of split are provided in supplementary material.

Train/test set: A zero-shot setting does not allow any visual example of an unseen class during training. Therefore, we customize the training set of ILSVRC such that images containing any unseen instance are removed. This results in a total of 315,731 training images with 449,469 annotated bounding boxes. For testing, the traditional zero-shot recognition setting is used which considers only unseen classes. As the test set annotations are not available to us, we cannot separate unseen classes for evaluation. Therefore, our test set is composed of the left out data from ILSVRC training dataset plus validation images having at least one unseen bounding box. The resulting test set has 19,008 images and 19,931 bounding boxes.

Network	ZSD			ZSMD			ZST			ZSMT		
	Baseline	Ours (L'_{mm})	Ours (L_{cls})	Baseline	Ours (L'_{mm})	Ours (L_{cls})	Baseline	Ours (L'_{mm})	Ours (L_{cls})	Baseline	Ours (L'_{mm})	Ours (L_{cls})
R+w2v	12.7	15.0	16.0	13.7	15.4	15.4	23.3	27.5	30.0	28.8	33.4	39.3
R+glo	12.0	12.3	14.6	12.9	14.1	16.1	22.3	24.5	26.2	29.2	31.5	36.3
V+w2v	10.2	12.7	11.8	11.4	12.5	11.8	23.3	25.6	26.2	29.0	31.3	36.0
V+glo	9.0	10.8	11.6	9.7	11.3	11.8	20.3	22.9	23.9	27.3	29.2	34.2

Table 1. mAP of the unseen classes. Ours (with L'_{mm}) and Ours (with L_{cls}) denote the performance without predefined unseen and with cluster loss respectively (Sec. 3.3 and Sec. 3.2). For cluster case, $\lambda = 0.8$.

Semantic embedding: Traditionally ZSL methods report performance on both supervised attributes and unsupervised word2vec/glove as semantic embeddings. As manually labeled supervised attributes are hard to obtain, only small-scale datasets with these annotations are available [9,20]. ILSVRC-2017 detection dataset used in the current work is quite huge and does not provide attribute annotations. In this paper, we work on ℓ_2 normalized 500 and 300 dimensional unsupervised word2vec [30] and GloVe [35] vector respectively to describe the classes. These word vectors are obtained by training on several billion words from Wikipedia dump corpus.

Evaluation Metric: We report average precision (AP) of individual unseen classes and mean average precision (mAP) for the overall performance of unseen classes.

Implementation Details: Unlike Faster-RCNN, our first step is trained in one step: after initializing shared layer with pre-trained weights, RPN and detection network of Fast-RCNN layers are learned together. Some other settings includes rescaling shorter size of image as 600 pixels, RPN stride = 16, three anchor box scale 128, 256 and 512 pixels, three aspect ratios 1:1, 1:2 and 2:1, non-maximum suppression (NMS) on proposals class probability with IoU threshold = 0.7. Each mini-batch is obtained from a single image having 16 positive and 16 negative (background) proposals. Adam optimizer with learning rate 10^{-5} , $\beta_1 = 0.9$ and $\beta_2 = 0.999$ is used in both state training. First step is trained over 10 million mini-batches without any data augmentation, but data augmentation through repetition of object proposals is used in second step (details in supplementary material). During testing, the prediction score threshold was 0.1 for baseline and Ours (with L'_{mm}) and 0.2 for clustering method (Ours with L_{cls}). We implement our model in *Keras*.

4.2 ZSD Performance

We compare different versions of our method (with loss configurations L'_{mm} and L_{cls} respectively) to a baseline approach. Note that the baseline is a simple extension of Faster-RCNN [38] and ConSE [33]. We apply the inference strategy mentioned in Sec. 3.3 after first step training as we can still get a vector $\mathbf{o} \in \mathbb{R}^{S+1}$ on the classification layer of Faster-RCNN network. We use two different architectures i.e., VGG-16 (V) [42] and ResNet-50 (R) [13] as the backbone of the Faster-RCNN during the first training step. In second step, we experiment

	OVERALL	Similar classes NOT present											Similar classes present											
		p.box	syringe	harmonica	maraca	barrito	pineapple	bowtie	s.trunk	d.washer	canopener	p.rack	bench	e.fan	ipod	scorpion	snail	hamster	tiger	ray	train	unicycle	golfball	h.bar
		ZSD Baseline = 6.3, Ours (L'_{mm}) = 6.5 , Ours (L_{cls}) = 4.4											ZSD Baseline = 18.6, Ours (L'_{mm}) = 22.7, Ours (L_{cls}) = 27.4											
Zero-Shot Detection (ZSD)																								
Baseline	12.7	0.0	3.9	0.5	0.0	36.3	2.7	1.8	1.7	12.2	2.7	7.0	1.0	0.6	22.0	19.0	1.9	40.9	75.3	0.3	28.4	17.9	12.0	4.0
Ours (L'_{mm})	15.0	0.0	8.0	0.2	0.2	39.2	2.3	1.9	3.2	11.7	4.8	0.0	0.0	7.1	23.3	25.7	5.0	50.5	75.3	0.0	44.8	7.8	28.9	4.5
Ours (L_{cls})	16.4	5.6	1.0	0.1	0.0	27.8	1.7	1.5	1.6	7.2	2.2	0.0	4.1	5.3	26.7	65.6	4.0	47.3	71.5	21.5	51.1	3.7	26.2	1.2
Zero-Shot Tagging (ZST)																								
Baseline	23.3	2.9	13.4	9.6	3.1	61.7	20.7	16.3	7.5	29.4	8.6	12.2	8.5	4.9	46.2	30.7	11.0	51.8	77.6	9.0	46.1	39.0	12.7	12.6
Ours (L'_{mm})	27.5	2.9	20.8	10.5	3.3	72.5	27.7	16.7	7.9	22.9	14.3	2.8	6.7	14.5	46.8	42.6	16.0	59.1	80.0	12.9	67.3	34.1	34.0	17.1
Ours (L_{cls})	30.6	12.6	10.2	11.9	4.9	48.9	21.8	17.9	29.1	32.2	10.0	4.1	20.7	10.7	52.2	82.6	12.3	58.5	75.5	48.9	72.2	16.9	33.9	15.5
Zero-Shot Meta Detection (ZSMD)																								
Meta-class		Indoor	Musical	Food	Clothing	Appli.	Kitchen	Furn.	Electronic	Invertebra.	Mammal	Fish	Vehicle	Sport										
Baseline	13.7	3.3	0.3	24.0	4.0	12.2	2.1	1.0	12.1	17.0	70.7	0.3	22.1	8.5										
Ours (L'_{mm})	15.4	8.1	0.1	18.4	2.3	11.7	3.0	0.0	14.3	27.8	73.6	0.0	32.1	9.0										
Ours (L_{cls})	15.6	3.5	0.1	10.0	1.9	7.2	1.2	4.1	15.3	31.4	66.8	21.5	31.2	9.3										
Zero-Shot Meta-class Tagging (ZSMT)																								
Baseline	28.8	15.2	12.0	55.6	25.2	29.4	10.7	8.5	31.5	36.5	75.8	9.0	48.4	17.0										
Ours (L'_{mm})	33.4	24.1	13.6	55.9	31.3	22.9	14.7	6.7	33.0	49.4	82.6	12.9	64.2	23.2										
Ours (L_{cls})	39.9	19.2	15.5	45.6	38.5	32.2	12.4	20.7	40.3	58.2	84.8	48.9	74.7	27.1										

Table 2. Average precision of individual unseen classes using ResNet+w2v and loss configurations L'_{mm} and L_{cls} (cluster based loss with $\lambda = 0.6$). We have grouped unseen classes into two groups based on whether visually similar classes present in the seen class set or not. Our proposed method achieve significant performance improvement for the group where similar classes are present in the seen set.

with both Word2vec and GloVe as the semantic embedding vectors used to define \mathbf{W}_2 . Fig. 4 illustrates some qualitative ZSD examples. More performance results of ZSD on other datasets is provided in the supplementary material.

Overall results: Table 1 reports the mAP for all approaches on four tasks: ZSD, ZSMD, ZST, and ZSMT across different combinations of network architectures. We can make following observations: (1) Our cluster based method outperforms other competitors on all four tasks because its loss utilizes high-level semantic relationships from meta-class definitions which are not present in other methods. (2) Performances get improved from baseline to Ours (with L'_{mm}) across all zero-shot tasks. The reason is baseline method did not consider word vectors during the training. Thus, overall detection could not get enough supervision about the semantic embeddings of classes. In contrast, L'_{mm} loss formulation considers word vectors. (3) Performances get improved from ZST to ZSMT across all methods whereas similar improvement is not common from ZSD to ZSMD. It's not surprising because ZSMD can get some benefit if meta-class of the predicted class is same as the meta-class of true class. If this is violated frequently, we cannot expect significant performance improvement in ZSMD. (4) In comparison of traditional object detection results, ZSD achieved significantly lower performance. Remarkably, even the state-of-the-art zero-shot classification approaches perform quite low e.g., a recent ZSL method [51] reported 11% hit@1 rate on ILSVRC 2010/12. This trend does not undermine to significance of ZSD, rather highlights the underlying challenges.

Individual class detection: Performances of individual unseen classes indicate the challenges for ZSD. In Table 2, we show performances of individual unseen classes across all tasks with our best (R+w2v) network. We observe that the unseen classes for which visually similar classes are present in their meta-

Top1 Accuracy	Network	w2v	glo
Akata'16 [1]	V	33.90	-
DMaP-I'17[24]	G+V	26.38	30.34
SCoRe'17[32]	G	31.51	-
Akata'15 [3]	G	28.40	24.20
LATEM'16 [46]	G	31.80	32.50
DMaP-I'17 [24]	G	26.28	23.69
Ours	R	36.77	36.82

Table 3. Zero shot recognition on CUB using $\lambda = 1$ because no meta-class assignment is done here. For fairness, we only compared our result with the inductive setting of other methods without per image part annotation and description. We refer V=VGG, R=ResNet, G=GoogLeNet.

classes achieve better detection performance (ZSD mAP 18.6, 22.7, 27.4) than those which do not have similar classes (ZSD mAP 6.3, 6.5, 4.4) for the all methods (baseline, our's with L'_{mm} and L_{cls}). Our proposed cluster method with loss L_{cls} outperforms the other versions significantly for the case when visually similar classes are present. For the all classes, our cluster method is still the best (mAP: cluster 16.4 vs. baseline 12.7). However, our's with L'_{mm} method performs better for when case similar classes are not present (mAP 6.5 vs 4.4). For the easier tagging tasks (ZST and ZSMT), the cluster method gets superior performance in most of the cases. This indicates that one potential reason for the failure cases of our cluster method for ZSD might be confusions during localization of objects due to ambiguities in visual appearance of unseen classes.

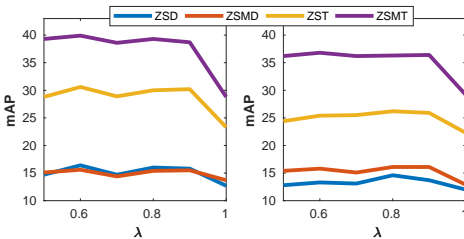


Fig. 4. Effect of varying λ in different zero-shot tasks for ResNet+w2v (left) and ResNet+glo (right).

Varying λ : The hyperparameter λ controls the weight between L_{mm} and L_{mc} in L_{cls} . In Fig. 4, we illustrate the effect of varying λ on four zero-shot tasks for R+w2v and R+glo. It shows that performances has less variation in the range of $\lambda = .5$ to $.9$ than $\lambda = .9$ to 1 . For a larger λ , mAP starts dropping since the impact of L_{mc} decreases significantly.

4.3 Zero Shot Recognition (ZSR)

Being a detection model, the proposed network can also perform traditional ZSR. We evaluate ZSR performance on popular Caltech-UCSD Birds-200-2011 (CUB) dataset [44]. This dataset contains 11,788 images from 200 classes and provides single bounding boxes per image. Following standard train/test split [47], we use 150 seen and 50 unseen classes for experiments. For semantics embedding, we use 400-d word2vec (w2v) and GloVe (glo) vector [46]. Note that, we do not use per image part annotation (like [1]) and descriptions (like [51]) to enrich semantic embedding. For a given test image, our network predicts unseen class bounding boxes. We pick only one label with the highest prediction score per image. In this way, we report the mean Top1 accuracy of all unseen classes in Table 3. One can find our proposed solution achieve significant performance improvement in comparison with state-of-the-art methods.

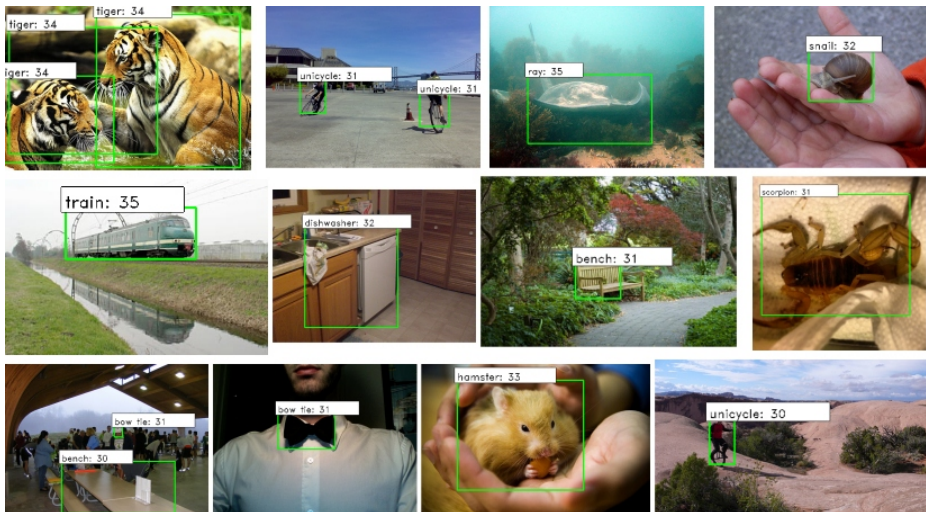


Fig. 5. Selected examples of ZSD of our cluster ($\lambda = .6$) method with R+w2v, using the prediction score threshold = 0.3. (See supplementary material for more examples)

4.4 Challenges and New Directions

ZSD is challenging: Our empirical evaluations show that ZSD needs to deal with the following challenges: (1) Unseen classes are rare compared to seen classes; (2) Small unseen objects are hard to detect and harder to relate with their semantics; (3) The scarcity of similar seen class leads to an inadequate description of an unseen class; (4) As derived in an unsupervised manner, the noise of semantic space affects ZSD. These issues are discussed in detail in supplementary material.

Future challenges: The ZSD problem warrants further investigation. (1) Unlike current work one can consider fine-tuning the bounding box of the both seen and unseen classes based on visual and semantic correspondences. (2) Rather mapping image feature to the semantic space, the reverse mapping may help ZSD similar to ZSR used in [19,51]. (3) One can consider the fusion of different word vectors (word2vec and GloVe) to improve ZSD. (4) Like generalized ZSL [48,47,24], one can extend it to a more realistic generalized ZSD. Moreover, weakly supervised or semi-supervised version of zero shot problems is also possible while performing ZSD/GZSD.

5 Conclusion

While traditional ZSL research focuses on only object recognition, we propose to extend the problem to object detection (ZSD). To this end, we offer a new experimental protocol with ILSVRC-2017 dataset specifying the seen-unseen, train-test split. We also develop an end-to-end trainable CNN model to solve this problem. We show that our solution is better than a strong baseline.

Overall, this research throws some new challenges to ZSL community. To make a long-standing progress in ZSL, the community needs to move forward in the detection setting rather than merely recognition.

References

1. Z. Akata, M. Malinowski, M. Fritz, and B. Schiele. Multi-cue zero-shot learning with strong supervision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
2. Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-Embedding for Image Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7):1425–1438, July 2016.
3. Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 07-12-June-2015, pages 2927–2936, 2015.
4. Z. Al-Halah, M. Tapaswi, and R. Stiefelhagen. Recovering the missing link: Predicting class-attribute associations for unsupervised zero-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
5. Z. Changpinyo, W.-L. Chao, B. Gong, and F. Sha. Synthesized classifiers for zero-shot learning. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-January, pages 5327–5336, 2016.
6. B. Demirel, R. Gokberk Cinbis, and N. Ikozler-Cinbis. Attributes2classname: A discriminative model for attribute-based unsupervised zero-shot learning. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
7. J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam. Large-scale object classification using label relation graphs. In *European conference on computer vision*, pages 48–64. Springer, 2014.
8. S. Deutsch, S. Kolouri, K. Kim, Y. Owechko, and S. Soatto. Zero shot learning via multi-scale manifold regularization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
9. A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1778–1785. IEEE, 2009.
10. A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. A. Ranzato, and T. Mikolov. Devise: A deep visual-semantic embedding model. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2121–2129. Curran Associates, Inc., 2013.
11. Y. Fu, Y. Yang, T. Hospedales, T. Xiang, and S. Gong. Transductive multi-label zero-shot learning. *arXiv preprint arXiv:1503.07790*, 2015.
12. Z. Fu, T. Xiang, E. Kodirov, and S. Gong. Zero-shot learning on semantic class prototype graph. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2017.
13. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. volume 2016-January, pages 770–778, 2016. cited By 107.
14. R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell. Natural language object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4555–4564, 2016.

15. D. Jayaraman and K. Grauman. Zero-shot recognition with unreliable attributes. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3464–3472. Curran Associates, Inc., 2014.
16. S. Jetley, M. Sapienza, S. Golodetz, and P. H. Torr. Straight to shapes: Real-time detection of encoded shapes. *arXiv preprint arXiv:1611.07932*, 2016.
17. K. H. J. S. Jifeng Dai, Yi Li. R-FCN: Object detection via region-based fully convolutional networks. *arXiv preprint arXiv:1605.06409*, 2016.
18. E. Kodirov, T. Xiang, Z. Fu, and S. Gong. Unsupervised domain adaptation for zero-shot learning. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.
19. E. Kodirov, T. Xiang, and S. Gong. Semantic autoencoder for zero-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
20. C. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, pages 951–958, 2009.
21. C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, March 2014.
22. J. Lei Ba, K. Swersky, S. Fidler, et al. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4247–4255, 2015.
23. X. Li, S. Liao, W. Lan, X. Du, and G. Yang. Zero-shot image tagging by hierarchical semantic embedding. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 879–882. ACM, 2015.
24. Y. Li, D. Wang, H. Hu, Y. Lin, and Y. Zhuang. Zero-shot recognition using dual visual-semantic mapping paths. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
25. Z. Li, E. Gavves, T. Mensink, and C. G. Snoek. Attributes make sense on segmented objects. In *European Conference on Computer Vision*, pages 350–365. Springer, 2014.
26. Z. Li, R. Tao, E. Gavves, C. Snoek, and A. Smeulders. Tracking by natural language specification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6495–6503, 2017.
27. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
28. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. *SSD: Single Shot MultiBox Detector*, pages 21–37. Springer International Publishing, Cham, 2016.
29. S. H. Maxime Bucher and F. Jurie. Improving semantic embedding consistency by metric learning for zero-shot classification. In *Proceedings of The 14th European Conference on Computer Vision*, 2016.
30. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

31. G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.
32. P. Morgado and N. Vasconcelos. Semantically consistent regularization for zero-shot recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
33. M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean. Zeroshot learning by convex combination of semantic embeddings. In *In Proceedings of ICLR*, 2014.
34. M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell. Zero-shot learning with semantic output codes. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1410–1418. Curran Associates, Inc., 2009.
35. J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.
36. S. Rahman, S. H. Khan, and F. Porikli. A unified approach for conventional zero-shot, generalized zero-shot and few-shot learning. *arXiv preprint arXiv:1706.08653*, 2017.
37. J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.
38. S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, June 2017.
39. B. Romera-Paredes and P. Torr. An embarrassingly simple approach to zero-shot learning. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 2152–2161, 2015.
40. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.
41. Y. Shigeto, I. Suzuki, K. Hara, M. Shimbo, and Y. Matsumoto. Ridge regression, hubness, and zero-shot learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 135–151. Springer, 2015.
42. K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
43. R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 935–943. Curran Associates, Inc., 2013.
44. C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.
45. X. Wang and Q. Ji. A unified probabilistic approach modeling relationships between attributes and objects. *Proceedings of the IEEE International Conference on Computer Vision*, pages 2120–2127, 2013.
46. Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele. Latent embeddings for zero-shot classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
47. Y. Xian, B. Schiele, and Z. Akata. Zero-shot learning - the good, the bad and the ugly. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2017.
48. X. Xu, F. Shen, Y. Yang, D. Zhang, H. T. Shen, and J. Song. Matrix tri-factorization with manifold regularizations for zero-shot learning. In *Proc. of CVPR*, 2017.

49. M. Ye and Y. Guo. Zero-shot classification with discriminative semantic representation learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
50. F. X. Yu, L. Cao, R. S. Feris, J. R. Smith, and S. F. Chang. Designing category-level attributes for discriminative visual recognition. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 771–778, June 2013.
51. L. Zhang, T. Xiang, and S. Gong. Learning a deep embedding model for zero-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
52. Z. Zhang and V. Saligrama. Zero-shot learning via semantic similarity embedding. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.
53. Z. Zhang and V. Saligrama. Zero-shot learning via joint latent similarity embedding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Supplementary Material

Zero Shot Object Detection

A. Related Work

End-to-end Object detection: Though object detection has been extensively studied in the literature, we can only find a few end-to-end learning pipelines capable of simultaneous object localization and classification. Popular examples of such approaches are Faster R-CNN [38], R-FCN [17], SSD [28] and YOLO [37]. The contribution of these methods relies on object localization process. Methods like Faster R-CNN [38], R-FCN [17] are based on Region Proposal Network (RPN) which provides bounding box proposals of possible objects and then classifying and fine tuning the box prediction in the later layers. In contrast, methods like SSD [28] and YOLO [37] draw bounding box and classify it in a single step. Unlike RPN; these methods predict bounding box offset of some pre-defined anchors instead of the box co-ordinates itself. The later methods are generally faster than the previous ones. However, RPN based methods are better in terms of accuracy. In current work, we prioritize accuracy over speed. Therefore, we build zero-shot object detection model based on the Faster RCNN.

Semantic embedding: Any zero-shot task like recognition or tagging requires semantic information of classes. This semantic information works as a bridge among seen and unseen classes. The common way to preserve the semantic information of a class is by using a one-dimensional vector. The vector space that holds semantic information of classes is called ‘semantic embedding space’. Visually similar classes reside in a close position in this space. The semantic vector of any class can be generated both manually or automatically. The manually generated semantic vectors are often called attributes [44,21]. Although attributes can describe a class with less noise (than other kinds of embeddings), those are very hard to obtain because of manual annotations. As a workaround, automatic semantic embedding can be generated from a large corpus of unannotated text like (Wikipedia, news article, etc.) or hierarchical relationship of classes in WordNet [31]. Some popular examples of such kind of semantic embeddings are word2vec [30], GloVe [35], and hierarchies [46]. As generated by an unsupervised manner, such embeddings become noisy but provide more flexibility and scalability than manual vectors.

Zero-shot learning: Humans can recognize an object by relating known objects, without prior visual experience. Simulating this behavior into an automated machine vision system is called Zero-shot learning (ZSL). ZSL attempts to recognize unseen objects without any visual examples of the unseen category. In recent years, numerous effective methods for ZSL have been proposed. Every ZSL strategy has to relate seen and unseen embedding through semantic embedding vector. Based on how this relation is established, we can categorize ZSL

strategies into three types. The **first** type of methods attempt to predict the semantic vector of classes [34,45,21,50]. An object is classified as an unseen class based on similarity of predicted vector and semantic vectors of unseen classes. Predicting a high dimensional vector is not an efficient way to related seen-unseen classes because it cannot work consistently if the semantic vectors are noisy [15]. This reason provokes this kind of methods to use attributes as semantic embedding as they are less noisy. The **second** kind of methods learn a linear [2,3,39] or non-linear [46,43] compatibility function to relate the seen image feature and corresponding semantic vector. This compatibility function yields high value if visual feature and semantic vector come from the same class and vice versa. A visual feature is classified to an unseen class if it gets the best compatibility score among all possible unseen classes. Such methods work consistently across a wide variety of semantic embedding vectors. The **third** kind of methods describe unseen classes by mixing seen visual features and semantic embedding [33,5,52]. For this mixing purpose, sometimes methods perform per class learning and later combine individual class output to decide outputs for unseen classes. While most of the ZSL approaches convert visual feature to semantic spaces, [19,51] mapped semantic vectors to the visual domain to address the hubness problem during prediction [41]. Irrespective of method types, attributes work better as semantic embeddings compared to unsupervised word2vec, GloVe, and hierarchies because of less noise. To minimize this performance gap, researchers have investigated transductive setting [49,48,24], domain adaptation [8,18] and class-attribute association [4,6] techniques. Usually, all ZSL methods are evaluated on a restricted case of recognition problem where test data only contain unseen images. Few recent efforts performed experiments on generalized version of ZSL [48,47,24]. They found that established ZSL methods perform poorly in such settings. Still, all these methods perform a recognition task in zero-shot settings. In this paper, we extend recognition problem to a more complex detection problem.

Zero-shot image tagging: Instead of assigning one unseen label to an image during recognition task, zero-shot tagging allows to tag multiple unseen tags to an image and/or ranking the array of unseen tags. Very few papers addressed the zero-shot version of this problem [23,11,53]. Li et al. [23] applied the idea of [33] in tagging. They argued that semantic embeddings (like word2vec) of all possible tags may not be available, and therefore, proposed a hierarchical semantic embedding method for those unavailable tags based on its ancestor classes in WordNet hierarchy. [11] considered the power set of fixed unseen tags as the label set to perform transductive multi-label learning. Recently, [53] proposed a fast zero-shot tagging approach that can rank both seen and arbitrary unseen tags during the testing stage. All previous attempts are not end-to-end because they preform training on the top of pre-trained CNN features. In this paper, we propose an end-to-end method for seen detection with zero-shot object tagging.

Object-level attribute reasoning: Object level attribute reasoning has been studied under two themes in the literature. The first theme advocates the use of object-level semantic representations in a traditional ZSL setting. Li et

al. [25] proposed to use local attributes and employed these shared characteristics to obtain zero-shot classification and segmentations. However, they dealt with fine-grained categorization task, where both seen and unseen objects have similar shapes (and segmentation masks), there is a single dominant category in each image and work with only supervised attributes. Another approach aiming at zero-shot segmentation is to learn a shape space shared with the novel objects. This technique, however, can only segment new object shapes that are very similar to the training set [16]. Along the second theme, some efforts have more recently been reported for object localization and tracking using natural language descriptions [14,26]. Different to our problem, they assume an accurate semantic description of the object, use supervised examples of objects during training, and therefore do not tackle the zero-shot detection problem.

B. Dataset and Experiment Protocol

B.1 Meta-class assignment

The classes of ILSVRC detection dataset maintain a defined hierarchy [40]. However, this hierarchy does not follow a tree structure. In this paper, we choose a total of $M = 14$ meta-classes (including person), in which the 200 object classes are divided. Table 1 describes meta-class assignment of all 200 classes. This assignment mostly follows the hierarchy of question prescribed in the original paper [40]. Few notable exceptions are (1) the classes of first-aid/medical items, cosmetics, carpentry items, school supplies and bag are grouped as indoor accessory, (2) liquid container related classes are merged with kitchen items, (3) flower pot is considered as furniture similar to MicroSoft COCO super-categories [27], (4) All living organisms (other than people) related classes are grouped into three different meta-class categories based on their similarity in word vector embedding space: invertebrate, mammal and non-mammal animal. Although one can argue that all invertebrate are non-mammal, this is just an assignment definition we apply in this paper to obtain a uniform distribution of images across super-classes.

B.2 Train/Test Split

Since the unseen classes are rare in real life settings and therefore their images are hard to collect, we assume that the training set only contains frequent classes. For ILSVRC detection dataset, number of instances per class follows a long-tail distribution (Figure 1). For each of our defined meta-class categories, we first plot the instance distribution of the child classes like Figure 3. Then, we randomly select one or two classes (depending on the number of child classes) from the rare second half of the distribution. We choose two unseen classes from the meta-classes which have relatively large (9 or more) number of child classes. In contrast, we choose one class as unseen for the meta-classes having less number of child classes. The only exception is that we do not choose ‘Person’ meta-class as unseen because it has no similar child class.

ID	Meta/Super-class	Categories
1	Indoor Accessory (25)	axe, backpack, band aid, binder, chain saw, cream, crutch, face-powder, hairspray, hammer, lipstick, nail, neck-brace, pencilbox , pencilsharpener, perfume, plastic-bag, power-drill, purse, rubber-eraser, ruler, screwdriver, stethoscope, stretcher, syringe
2	Musical (17)	accordion, banjo, cello, chime, drum, flute, french-horn, guitar, harmonica , harp, maraca , oboe, piano, saxophone, trombone, trumpet, violin
3	Food (21)	apple, artichoke, bagel, banana, bell-pepper, burrito , cucumber, fig, guacamole, hamburger, head-cabbage, hotdog, lemon, mushroom, orange, pineapple , pizza, pomegranate, popsicle, pretzel, strawberry
4	Electronics (16)	computer-keyboard, computer-mouse, digital-clock, electric-fan , hair-dryer, iPod , lamp, laptop, microphone, printer, vacuum, remote-control, tape-player, traffic-light, tv or monitor, washer
5	Appliance (7)	coffee-maker, dishwasher , microwave, refrigerator, stove, toaster, waffle-iron
6	Kitchen item (17)	beaker, bowl, can-opener , cocktail-shaker, corkscrew, cup or mug, frying-pan, ladle, milk-can, pitcher, plate-rack , salt or pepper shaker, soap-dispenser, spatula, strainer, water-bottle, wine-bottle
7	Furniture (8)	baby-bed, bench , bookshelf, chair, filing-cabinet, flower-pot, sofa, table
8	Clothing (11)	bathing-cap, bow-tie , brassiere, diaper, hat with a wide brim, helmet, maillot, miniskirt, sunglasses, swimming-trunks , tie
9	Invertebrate animal (14)	ant, bee, butterfly, centipede, dragonfly, goldfish, isopod, jellyfish, ladybug, lobster, scorpion , snail , starfish, tick
10	mammal animal (28)	antelope, armadillo, bear, camel, cattle, dog, domestic-cat, elephant, fox, giant-panda, hamster , hippopotamus, horse, koala-bear, lion, monkey, otter, porcupine, rabbit, red-panda, seal, sheep, skunk, squirrel, swine, tiger , whale, zebra
11	non-mammal animal (6)	bird, frog, lizard, ray , snake, turtle
12	Vehicle (12)	airplane, bicycle, bus, car, cart, golfcart, motorcycle, snowmobile, snowplow, train , unicycle , watercraft
13	Sports (17)	balance-beam, baseball, basketball, bow, croquet-ball, dumbbell, golf-ball , horizontal-bar , ping-pong-ball, puck, punching-bag, racket, rugby-ball, ski, soccer-ball, tennis-ball, volleyball
14	Person (1)	person

Table 1. Assigned meta-class to each of the 200 object categories. The unseen classes are presented as bold.

This random selection procedure avoids biasness, ensures diversity (due to selection from all meta-classes) and conforms to the observation that unseen classes are not frequent.

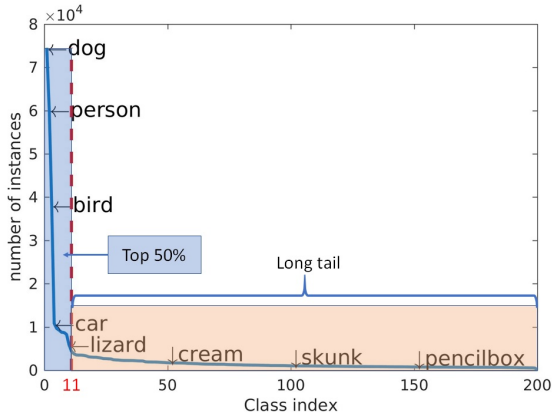


Fig. 1. Long-tail distribution of imageNet dataset

mAP	Network	w2v	glo
Baseline	R	31.0	26.7
Our (L_{cls})	R	33.5	32.3
Baseline	V	30.3	27.9
Our (L_{cls})	V	30.4	28.4

Table 2. ZSD on CUB using $\lambda = 1$. We refer V=VGG and R=ResNet

B.3 Data Augmentation

We visualize the long-tail distribution of ILSVRC detection classes in Figure 1. One can find that only 11 highly frequent classes (out of 200) cover top 50% of the distribution. This distribution creates a significant impact on ZSD. To address this problem, in the second step of training, we augment the less frequent data to make a balance among similar seen classes for each unseen category. From the 10 million mini-batches used at the first stage of training, we create a set of over 2.8 million mini-batches for the second stage training. While creating this set, we make sure that every unseen class gets at least 10K similar (positive) instances from classes whose meta-class category is common to that of unseen class. In doing so, for some unseen classes like ‘ray’, we need to randomly augment data by repetition because the total instances of classes in the meta-class ‘non-mammal animal’ are not more than 10K. In contrast, the unseen class like ‘tiger’ has more than 10K similar instances in ‘mammal animal’ meta-class. Therefore, we randomly pick 10K among those to balance the training set. After this, the rest of instances of 2.8 million mini-batches are chosen as the background.

C. ZSD on CUB

We evaluate the ZSD performance of the baseline and our proposed method based on single bounding box per image provided in CUB dataset [44]. Table 2 describes the performance comparison between the baseline and our basic method. Our overall loss (L_{cls}) based method outperforms the baseline in the different network and semantic settings. Note that, we do not define any meta-class for the CUB classes. Therefore, we use $\lambda = 1$ for CUB related experiments.

D. Further Analysis

ZSD Challenges: In general, detection is a harder task than recognition/tagging because of locating the bounding box at the same time. The strict requirement of not using any unseen class images during training of zero-shot setting is itself a tough condition for recognition/tagging task which gets intensified to a high degree for detection task. We have used ILSVRC-2017 detection dataset to evaluate some baseline performances of the proposed problem. This dataset has 200 classes including a total 478,807 object instances of different shapes/size and distribution (See Figure 2). Within those, we define $M = 14$ meta classes which contain one or more specific classes. Figure 3 describes the normalized number of instances per classes within meta class. Considering this challenging dataset, here we describe some other difficulties of the zero shot detection task:

Rarity: ILSVRC dataset contains a long-tail distribution issue, i.e., many rare classes get less number of instances. It is apparent that an unseen class should be within the set of rare classes. To address this fact, we randomly choose unseen classes from each meta-class z_j which lies in the rarest 50% in the distribution. It affects the zero-shot version of the problem also.

Object size: Some rare object classes like syringe, ladybug etc. usually have a small size. Smaller objects are difficult to detect as well as recognize.

High Diversity: Every meta-class gets a different number of classes and there exists a high visual diversity in each meta-class images. Since, being in a same meta-class does not guarantee of the visual similarity, it is difficult to learn relationships for the unseen categories which are quite different from the seen categories in the same super-class. As an example, ‘tiger’ has many similar classes compared to ‘ray’. The scarcity of similar class enables an inadequate description of the unseen class which eventually affect the zero shot detection performance.

Noise in semantic space: We use unsupervised semantic embedding vectors word2vec/GloVe as the class description. Such embeddings are noisy in general as they are generated automatically from unannotated text mining. It also affects the zero-shot detection performance significantly.

Seen vs. Unseen Class Performance: The overall performance of ZSD is depended on the learning of seen classes. Therefore, the performance of seen detection can be an indication of how possibly ZSD works. To this end, we also study the detection performance on seen classes of ILSVRC validation dataset after the first step of faster-RCNN training (Table 3). It indicates the baseline

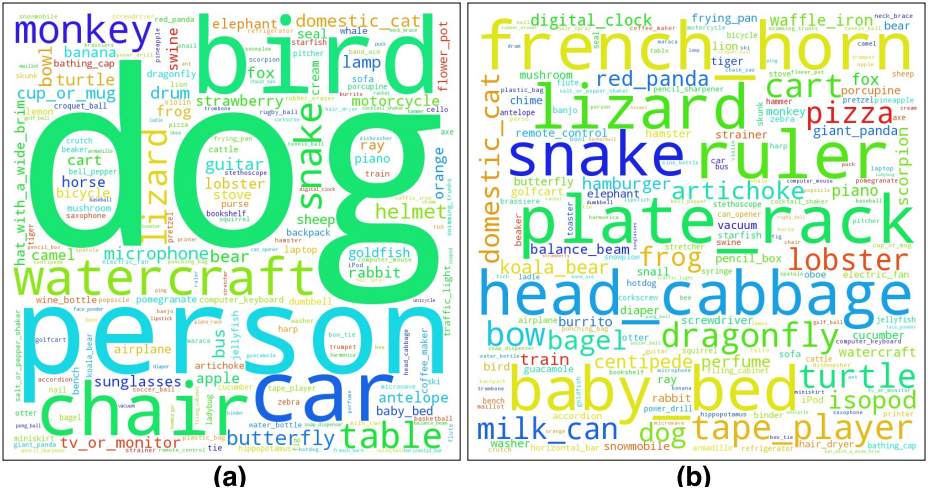


Fig. 2. Word cloud based on (a) number of object instance (b) Mean object size in pixel

performance of seen classes necessary to achieve the ZSD performance reported in the paper. The baseline method result is better than our proposed approaches. It is justifiable as both of our proposed methods can generate prediction for both seen and unseen class together which sacrifices the seen performance a bit to achieve distinction among all seen and unseen classes. The Table 3 also compares the seen result with the unseen performance. One can find that performance of selected unseen classes is similar to that of seen classes for our (I_{cls}) method. It indicates a balanced generalization of ZSD in both seen and unseen classes.

Learning without meta-class: For some applications, the meta-class based supervision may not be available. In such case, one can define meta-class in an unsupervised manner by applying a clustering mechanism on original semantic embedding.

ZSL vs ZSD loss: Many traditional non-end-to-end trainable ZSR methods consider different aspects of regularization [32], transductive setting [24], metric learning [29], domain adaptation [18] and class attribute association [4] etc. Similarly, the end-to-end trainable ZSR methods [51,22] employ different non-linearity in feature and semantic pipeline. But, those traditional loss formulations need to be redesigned in ZSD case to be compatible for both classification and box detection losses.

E. Qualitative results

We provide more examples of ZSD in Fig. 4. One can find that the prediction score threshold is lower (0.3 used in the examples) than the value (greater than

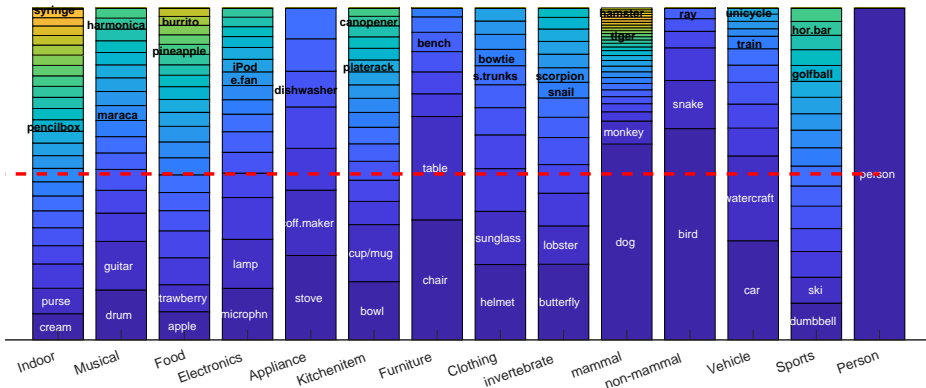


Fig. 3. Distribution of instances per classes within each meta class. Two most common (frequent) seen classes and unseen classes are marked in white and black color text respectively. Red dashed line indicates 50 percentile boundary. All unseen classes lie within the rarest half of the instance distribution.

mAP	Step 1	Baseline	Ours (L'_{mm})	Our (L_{cls})
Seen	33.7	33.4	27.7	26.1
Unseen (all)	-	12.7	15.0	16.4
Unseen (selected)	-	18.6	22.7	27.4

Table 3. Comparison of seen and unseen class performance using ResNet as convolution layers. word2vec is used for baseline, our (L'_{mm}) and our (L_{cls}). Best performance in each row are shown as bold. We refer Unseen (all): mAP of all unseen classes, Unseen (selected): mAP of selected classes for which visually similar classes are present.

0.5) used in traditional object detection like faster-RCNN [38]. It indicates that the prediction of ZSD has less confidence than that of traditionally seen detection. As zero-shot method does not observe any training instances of unseen classes during the whole learning process, the confidence of prediction cannot be as strong as the seen counterpart. Moreover, a ZSD method needs to correspond visual features with semantic word vectors which are noisy in general. It degrades the overall confidence for ZSD.

In the last layer of the box regression branch, our method does not have specified bounding boxes for un-seen classes. Instead, bounding box corresponding to a closely related seen class that has the maximum score is used for un-seen localization. Therefore, a correct unseen class prediction sometimes cannot get very accurate localizations as illustrated in Fig. 5.

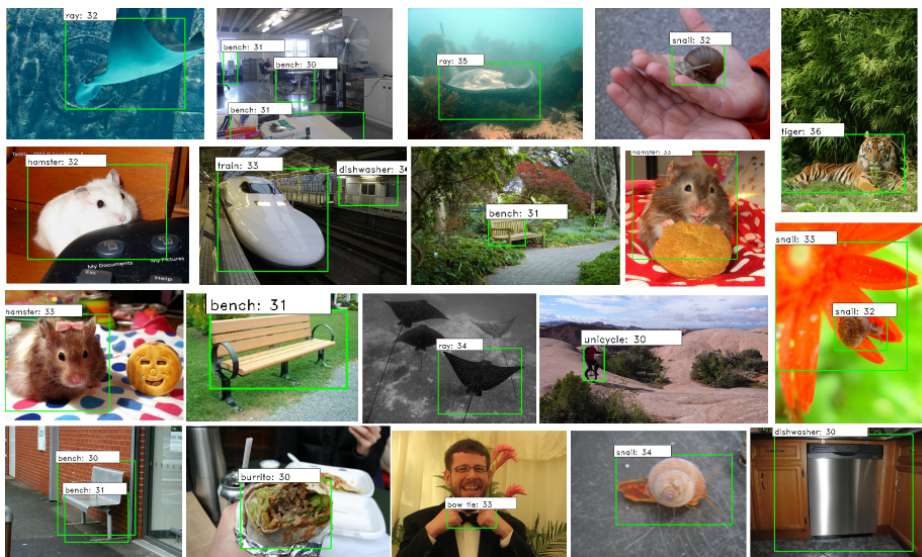


Fig. 4. Selected examples of ZSD of our (L_{cls}) with $\lambda = .6$ and $R+w2v$, using the prediction score threshold = 0.3.

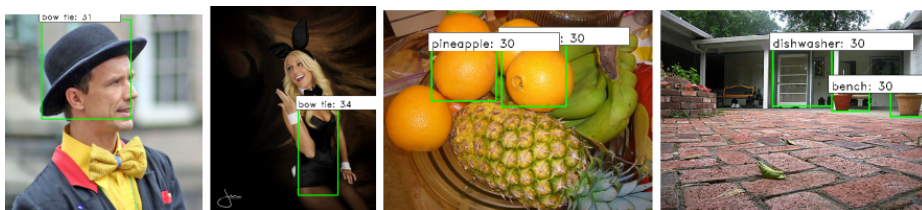


Fig. 5. Examples of incorrect detection but correct classification. The unseen class ‘bow-tie’, ‘pineapple’ and ‘bench’ are incorrectly localized in these images.