Noname manuscript No.
(will be inserted by the editor)

Zero-Shot Object Detection: Joint Recognition and Localization of Novel Concepts

Shafin Rahman · Salman H. Khan · Fatih Porikli

Received: date / Accepted: date

Abstract Zero shot learning (ZSL) identifies unseen objects for which no training images are available. Conventional ZSL approaches are restricted to a recognition setting where each test image is categorized into one of several unseen object classes. We posit that this setting is ill-suited for real-world applications where unseen objects appear only as a part of a complete scene, warranting both 'recognition' and 'localization' of the 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 samples. We introduce an integrated solution to the ZSD problem that jointly models the complex interplay between visual and semantic domain information. Ours is an end-to-end trainable deep network for ZSD that effectively overcomes the noise in the unsupervised semantic descriptions. To this end, we utilize the concept of meta-classes to design an original loss function that achieves synergy between max-margin class separation and semantic domain clustering. In order to set a benchmark for ZSD, we propose an experimental protocol for the large-scale ILSVRC dataset that adheres to practical challenges, e.g., rare classes are more likely to be the unseen ones. Furthermore, we present a baseline approach extended from conventional recognition to the ZSD setting. Our extensive experiments show a significant boost in performance (in terms of mAP and Recall) on the imperative yet difficult ZSD problem on ImageNet detection, MSCOCO and FashionZSD datasets.[1]

Keywords Zero-shot learning · Zero-shot object detection · Deep learning · Loss function

Shafin Rahman
North South University, Dhaka, Bangladesh
Data61, CSIRO, ACT 2601, AU
Australian National University, Canberra ACT 0200 AU
E-mail: shafin.rahman@northsouth.edu

Salman H. Khan
Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE
E-mail: salman.khan@mbzuai.ac.ae

Fatih Porikli
Australian National University, Canberra ACT 0200 AU
Huawei, San Diego, CA, USA
E-mail: fatih.porikli@anu.edu.au

1 Introduction

Humans have the amazing ability to develop a generalizable knowledge-base that compiles our sensorimotor experiences over time and relates them to abstract concepts. For instance, if we have seen visual examples of '*horse*' and '*donkey*', we can easily recognize their distinctive individual characteristics, such as horses have short ears, long tails and thin coats, while donkeys are shorter in height, have thick coats, long ears and shorter tails. These associations between visual and semantic content enable us to make inferences about unobserved content based on our previous knowledge. As an example, if we are described an animal that has close resemblance to both a horse and a donkey and which is smaller than a horse but bigger than a donkey, we can imagine what a '*mule*' looks like. Such an intelligent reasoning ability regarding the unobserved world would be highly valuable for life-long and self-learning machines.

Since its inception, the main focus of zero-shot learning research has been object classification [2, 6, 13, 20,

[1] The codes and dataset split are available at: `https://github.com/salman-h-khan/ZSD_Release`

24,25,32,36,47,56,64,66,67]. Although zero-shot recognition is still an open research problem, we hypothesize that this setting has a number of limitations that render it unsuitable for real-life applications. *First*, it assumes a simple case where only a single dominant category is present in an image. *Second*, the predictions are made for a complete scene, while in practice, the attributes and semantic descriptions are generally relevant to individual objects rather than the entire scene. *Third*, zero-shot recognition provides an answer to unseen categories in elementary tasks, e.g., classification and retrieval, but it cannot be scaled to advanced tasks, such as scene interpretation and contextual modeling, which require a fundamental reasoning for all salient objects in the scene. *Fourth*, global attributes are more susceptible to background variations, viewpoint, appearance and scale changes and practical challenges, such as occlusions and clutter. As a result, image-level ZSL fails for complex scenes where a diverse set of competing attributes that belong to multiple object categories exists.

Zero-shot Object Detection: To address the above-mentioned challenges, we introduce a new problem setting called *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 [48]. The resulting dataset is very demanding because of its large-scale, diversity, and unconstrained nature, and also unique due to its leveraging of WordNet semantic hierarchy [34]. Taking advantage of the semantic relationships among object classes, we use the concept of '*meta-classes*'[2] 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 and tracking, and determining an object's relationships with its environment using only the 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 there is 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 a given image. Since there can be a prohibitively large 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 incorrect predictions compared to classification. Therefore, a ZSD method is likely to predict a class label that might be incorrect but is visually and semantically similar to the corresponding true class. For example, wrongly predicting a 'spider' as 'scorpion', where both are semantically similar because they are 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 our model under zero-shot meta-class detection, zero-shot tagging, and zero-shot meta class tagging settings. Notably, the proposed network is trained only 'once' for the ZSD task and the additional tasks are used during evaluations only.

Our Contributions: Apart from a new large-scale protocol for ZSD, we propose an end-to-end trainable network for the ZSD problem that concurrently relates visual image features with semantic label information. This network uses a semantic embedding vector of classes within the network to produce prediction scores for both seen and unseen classes. We propose a novel loss formulation that incorporates max-margin learning [67, 65] and a semantic clustering loss based on the class-scores of different meta-classes. While the max-margin loss attempts to separate individual classes, the 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 when exploring the semantic relationships during the model learning phase. This assumption is consistent with recent efforts in the literature, which adopt class semantics to solve the domain shift problem in ZSL [10,15], and does not constitute a transductive setting [11,14,20]. Based on the premise that, in practice, unseen class semantics are sometimes unknown during training for 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 [36], using Faster-RCNN [46]. In summary, this paper reports the following advances:

– We introduce a new problem setting for zero-shot learning, which aims to jointly recognize and localize novel objects in complex scenes.

[2] Meta-classes are obtained by clustering semantically similar classes.

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)** The traditional recognition task only predicts seen class labels. **(b)** The traditional detection task predicts both seen class labels and bounding boxes. **(c)** The traditional zero-shot recognition task only predicts unseen class labels. **(d)** The proposed ZSD predicts both seen and unseen classes and their bounding boxes.

- 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. Additionally, our approach can automatically tune noisy semantic embeddings.

A preliminary version of this work appeared in [44]. The current version extends [44] in the following aspects: (a) a comprehensive description of the experimental protocol for the ImageNet dataset is provided in Sec. 5.1, (b) new ZSD experiments on both the small-scale CUB dataset and large-scale MS-COCO dataset are reported in Sec. 5, (c) a description of closely related works and comparison with our approach is included in Sec. 2, and (d) an elaborate qualitative result analysis is performed in Sec. 5.8.

2 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 [46], R-FCN [7], SSD [31] and YOLO [45]. The contribution of these methods pertains to object localization. Methods like Faster R-CNN [46], R-FCN [7] are based on two-stage training, where a Region Proposal Network (RPN) first provides bounding box proposals for possible objects and then the network performs box-classification and box-regression in the later layers. In contrast, methods like SSD [31] and YOLO [45] draw bounding boxes and classify them in a single step. Unlike RPN, these methods predict the bounding box offset of pre-defined anchors rather than the box co-ordinates themselves. The later methods are generally faster than the previous ones. However, RPN based methods are more accurate. All these object detectors are only capable of detecting objects whose training samples were available. In our current work, we focus on zero-shot object detection, which aims at detecting previously unseen object classes during inference. We build our model on top of a two-stage object detector (Faster RCNN), chosen due to its excellent performance for the regular object detection task.

Semantic embedding: Semantic information about object classes is critical for any zero-shot learning problem, such as recognition or tagging. This semantic information works as a bridge between seen and unseen classes. A common way to encode the semantic information of a class is by using a vector represented in the '*semantic embedding space*'. Visually similar classes reside in close proximity in this space. The semantic vector of any class can be generated either manually or automatically. Manually generated semantic vectors are often called '*attributes*' [53, 24]. Although attributes can describe a class with less noise (than other kinds of embeddings), they require considerable human effort to acquire manual annotations. As a workaround, automatic semantic embeddings can be generated from a large corpus of unannotated text (e.g., Wikipedia, news articles etc.) or the hierarchical relationship of classes in WordNet corpus [34]. Some popular examples of such semantic embeddings are word2vec [33], GloVe [38], and hierarchies [55]. Since these embeddings are generated in an unsupervised manner, they are relatively noisy but provide more flexibility and scalability compared to manually acquired attributes.

Zero-shot learning: Humans can recognize a new object by relating it to known concepts, without need

for prior visual experience. Simulating this behavior in an automated machine vision system is called Zero-shot learning (ZSL). In recent years, numerous methods for ZSL have been proposed. A common thread in all ZSL strategies is that they relate seen and unseen classes through semantic embeddings. Based on how this relation is established, ZSL strategies can be categorized into four types.

a) The **first** type of methods attempt to predict class-specific semantic embeddings [37,54,24,63]. An object is classified into an unseen class based on the similarity between the predicted and ground-truth semantics of unseen classes. This approach does not work consistently if the semantic vectors are noisy [18]. This leads such methods to use manually obtained attributes as the semantic embedding.
b) The **second** kind of methods learn a linear [2,3,47] or non-linear [55,52] compatibility function to relate the seen image feature and corresponding semantic vector. This compatibility function yields a high score if the visual feature and semantic vector come from the same class and vice versa. Visual features with the highest compatibility score are classified as unseen. Such methods work consistently across a wide variety of semantic embedding vectors.
c) The **third** kind of methods determine unseen classes by mixing seen visual features and semantic embeddings [36,6,66]. For this purpose, some methods perform per class learning and later combine individual class outputs to make unseen class predictions. The main difference from (b) is that they do not use class semantics during training with seen classes. After the seen training, they relate seen and unseen through mixing class semantics. While most of the ZSL approaches convert visual features to semantic space, [21,64] mapped semantic vectors to the visual domain to address the hubness problem during prediction [50].
d) The **fourth** kind of approaches use synthesized features to improve ZSL and GZSL performance [58, 49,59]. The synthesized features are generated using a Generative Adversarial Network (GAN), a Variational Auto-encoder (VAE) or their combination. After synthesizing features of unseen classes, they train unseen classes in a similar way as supervised learning. In recent years, these generative approaches have achieved state-of-the-art ZSL performance. However, they are dependent on attribute semantics that require hard manual labeling. Moreover, adding a new unseen class can be costly because it requires retraining based on new synthesized unseen data.

To minimize the difficulty level of the ZSL problem, researchers have investigated transductive setting [62, 61,27], domain adaptation [11,20] and class-attribute association [4,9] techniques. Usually, ZSL methods are evaluated on a restricted case of the recognition problem where test data only contain unseen images. Few recent studies performed experiments on generalized version of ZSL [61,57,27]. They found that the established ZSL methods perform poorly in such settings. Still, all these methods are restricted to the recognition task. In this paper, we extend recognition problem to a more complex detection problem, where both recognition and localization are required.

Zero-shot image tagging: Rather than assigning one unseen label to each image, as done in the recognition task, zero-shot tagging allows multiple unseen tags be allocated to an image and/or the array of unseen tags to be ranked. Very few papers have addressed the zero-shot version of this problem [26,14,67]. Li et al. [26] applied the ZSL approach proposed in [36] to image tagging. They argued that semantic embeddings of all possible tags may not be available, and therefore, proposed a hierarchical semantic embedding method for the unavailable tags based on its ancestor classes in WordNet hierarchy. [14] considered the power set of fixed unseen tags as the label set to perform transductive multi-label learning. Recently, [67] 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 top of pre-trained CNN features. In this paper, we propose an end-to-end method for zero-shot detection and also report performance on relatively simpler zero-shot object tagging task which does not require precise localization.

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. [28] proposed to use local attributes and employed these shared characteristics to obtain zero-shot classification and segmentation. 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 [19]. Along the second theme, some efforts have more recently been reported for object localization and tracking using natural language descriptions [17,29]. 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.

Recent efforts on Zero-shot object detection: In parallel to the preliminary version of this work [44], several concurrent but independent efforts on ZSD have been reported [5,8,68]. Bansal et al. [5] presents a background aware approach for ZSD. It works on pre-computed object proposals from Edgebox method. The main contribution is to treat the background class such that the model does not classifies an unseen object as background. Because of the dependence of offline object proposals, this method is not end-to-end trainable. Demirel et al. [8] proposed a YOLO detector based method for ZSD. This method mainly focuses on how to score an unseen region using unseen word vectors and prediction scores. However, their experimental evaluations are performed on relatively small-scale datasets, Fashion MNIST and Pascal VOC. Zhu et al. [68] also proposed a YOLO based architecture for ZSD. But, their work only localizes unseen objects without categorizing it to a particular unseen class. Recently, a new polarity loss for zero-shot detection has been proposed in [39,41] and achieved significant performance boost on MSCOCO-2014 and Pascal VOC-07 datasets. Further, [40] proposed a transductive learning framework for ZSD. However, none of the works mentioned above deal with the challenging ImageNet dataset. Apart from proposing an end-to-end model for ZSD, we provide a new protocol alongwith seen/unseen split to test ZSD on ImageNet data. Moreover, we test our method on other large-scale datasets such as MSCOCO-2014 [30].

3 Problem Description

For a given set of images from seen object categories, ZSD aims to *recognize* and *localize* 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.

3.1 Preliminaries

Consider a set of 'seen' classes denoted by $\mathcal{S} = \{1, \ldots, \mathrm{S}\}$, whose examples are available during the training stage and S represents their total number. There exists another set of 'unseen' classes $\mathcal{U} = \{\mathrm{S}+1, \ldots, \mathrm{S}+\mathrm{U}\}$, 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 $\mathrm{C} = \mathrm{S} + \mathrm{U}$ 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 $\mathcal{M} = \{z_m : m \in [1, \mathrm{M}]\}$, where M denote the total number of meta-classes and $z_m = \{k \in \mathcal{C}\ s.t.,\ g(k) = m\}$. 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., $\cap_{m=1}^{\mathrm{M}} z_m = \phi$ and $\cup_{m=1}^{\mathrm{M}} z_m = \mathcal{C}$.

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. Since 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 $y_{bg} = \{\mathrm{C}+1\}$ is a singleton set denoting the background label.

3.2 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 $f : \mathcal{X} \mapsto \mathcal{C}'$ that gives the minimum regularized empirical risk ($\hat{\mathcal{R}}$), as follows:

$$\arg\min_{\Theta} \hat{\mathcal{R}}(f(\mathbf{x}; \Theta)) + \Omega(\Theta), \tag{1}$$

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:

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

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 based on an objectness measure 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:

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 of relaxing the original problem: **(a)** Reducing the unseen object classes by clustering similar unseen classes into a single super-class (T2 and T4). **(b)** Removing the localization constraint. To this end, we investigate the zero-shot tagging problem, where the goal is only to recognize all object categories in an image (T3 and T4).

Current state-of-the-art methods for zero-shot learning only deal with recognition/tagging. The proposed problem settings add the missing detection task, which indirectly encapsulates traditional recognition and tagging tasks.

4 Zero-Shot Object Detection

Our proposed model uses Faster-RCNN [46] as the backbone architecture, due to its superior performance among competitive end-to-end object detection models [7,31, 45]. 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.

4.1 Model Architecture

The overall architecture is illustrated in Fig 2. It has two main components enclosed in boxes: 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/ ResNet) is used to obtain the intermediate convolutional activations. These activations are treated as feature maps, which are forwarded to a 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 proposals can be of different sizes, which are mapped to fixed sized representation using a RoI pooling layer that 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 training does not use object class information. It only predicts an objectness score and bounding box parameters to each anchor. As RPN learns what qualifies an object, a RPN trained on seen objects can generate proposals for unseen objects also. We validate this argument through experiments reported in Sec. 5.2. In the second block of our architecture, the object-specific 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 $\mathbf{W}_2 \in \mathbb{R}^{d \times (\mathrm{C}+1)}$, 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 [15], which adopted this setting to explore the cluster manifold structure of the semantic embedding space and address the domain shift issue. Given a feature representation input ($\mathbf{f}^t$) to SAN in the top branch the overall operation can be represented as:

$$\mathbf{o} = (\mathbf{W_1}\mathbf{W_2})^T \mathbf{f}^t. \tag{3}$$

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 $\mathbf{v}_b = \frac{1}{\mathrm{C}} \sum_{c=1}^{\mathrm{C}} \mathbf{v}_c$ as its embedding in $\mathbf{W_2}$. The reason for using such an embedding for the background class is two-fold. (1) Since a background

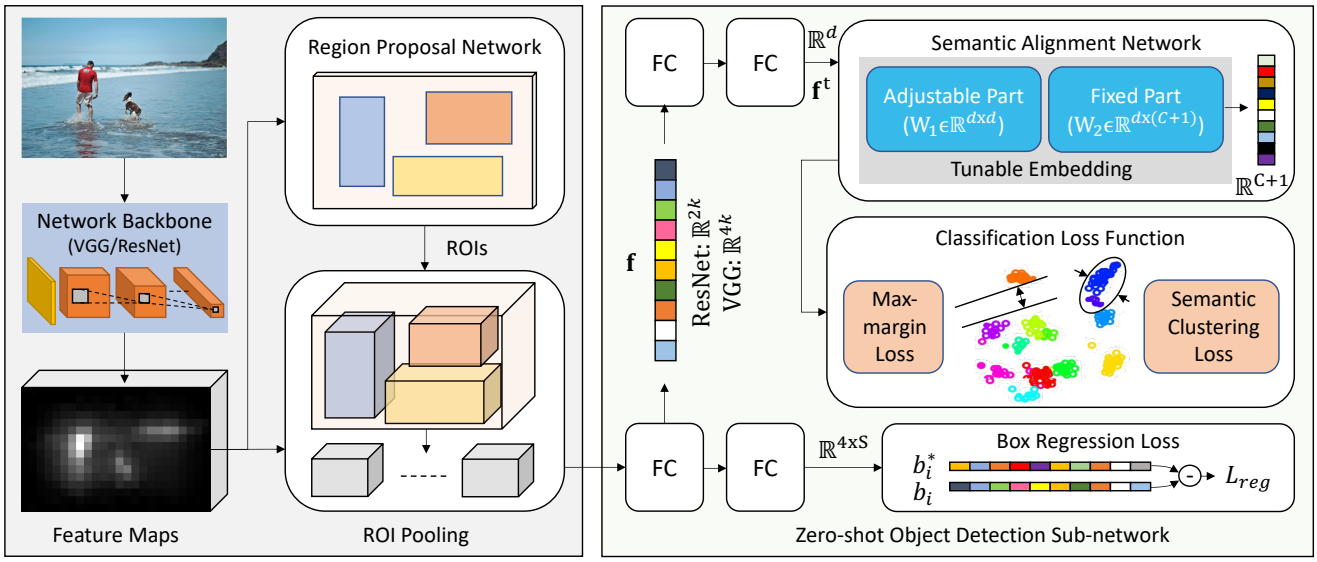

Fig. 2: Network Architecture - *Left:* Image level feature maps are used to propose candidate object boxes and their corresponding features. *Right:* The features are used for classification and localization of new classes by utilizing their semantic concepts.

box can contain parts of objects (with IoU < 0.5), an average embedding adequately models the semantics that could appear in the background category. (2) It keeps the relationship between word vectors consistent which is not possible otherwise. To test this hypothesis, we replace the background embedding with an all one vector, that results in a very low performance mark (3.2 mAP) for ZSD.

The projection defined by $\mathbf{W_1}$ is tunable while $\mathbf{W_2}$ defines a fixed embedding. Notably, a non-linear activation function is not applied between the tunable 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** enables bounding box regression to add suitable offsets to align the proposals with the ground-truths for precise predictions of object locations. This branch is set up similar to Faster-RCNN [46].

4.2 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 again used for training. 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

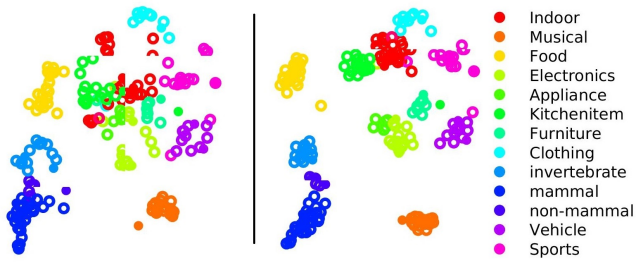

Fig. 3: The 2D tSNE embedding of modified word vectors $\mathbf{W_1 W_2}$ using only max-margin loss, L_{mm} (left) and with clustering loss, $L_{mm} + L_{mc}$ (right). Semantically similar classes are embedded more closely in cluster based loss.

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.

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

where Θ denotes the parameters of the network, $\mathbf{o}_i$ is the classification branch output, $\mathrm{T} = \mathrm{N} \times \mathrm{R}$ 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.

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

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

where, hyper-parameter λ controls the trade-off between two losses. We fix it by performing traditionally seen object detection task. We have used the validation set of ILSVRC detection dataset for this. We define,

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

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

where, sets $\mathcal{M}'' = \{\mathcal{M}' \setminus z_{g(y_i)}\}$ and $\mathcal{Z} = \{z_{g(y_i)}\}$, o_k represent the prediction response of class $k \in \mathcal{S}$. L_{mm} tries to separate the prediction response of the true class from rest of the classes. In contrast, L_{mc} pulls the classes belonging to different meta-classes further apart and (implicitly) tries to cluster together the members of each super-class. The benefit of using super-classes in our approach is two-fold. First, our L_{mc} loss utilizes the super-class definition to cluster similar classes together. This helps in identifying visual instances of unseen classes by relating them with the similar seen classes. In this way, the super-class definition is useful specifically for ZSD, where semantic relationships are very helpful to make sense of the unseen classes. Second, the super-class definition helps us define additional auxiliary tasks such as ZSMT and ZSMD that can shed light on which particular aspects of the ZSD problem are more challenging (i.e., localization or recognition).

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 animal-related meta-classes are close together, 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 fine-tunes the bounding box for each seen class ROI. For each $\mathbf{f}_i$, we get $4 \times \mathrm{S}$ 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 $\hat{o_c} = \frac{o_c}{\|\mathbf{v}_c\|_2 \|\mathbf{f}^t\|_2}$. 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.

$$y_u = \arg\max_{u \in \mathcal{U}} \hat{o_u} \quad s.t., \ \hat{o_u} > \alpha. \tag{5}$$

The other detection branch finds b_i which is the set of parameterized co-ordinates of bounding boxes for 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 $g(.)$ to assign a meta-class for any unseen label.

4.3 ZSD without Pre-defined Unseen

While applying clustering loss in Sec. 4.2, 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 $\mathcal{C}'$ that has 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 $\mathbf{W}_2 \in \mathbb{R}^{d \times (\mathrm{S}+1)}$ to train the whole framework with only the max-margin loss, L'_{mm}, defined as follows:

$$L'_{mm}(\mathbf{o}_i, y_i) = \frac{1}{|\mathcal{S}' \setminus y_i|} \sum_{c \in \mathcal{S}' \setminus y_i} \log\Big(1 + \exp(o_c - o_{y_i})\Big).$$

Since the output classification layer cannot make predictions for unseen classes, we apply a procedure similar to ConSE during the testing phase [36]. Here, the choice of [36] is made due to two main reasons: **(a)** In contrast to other ZSL methods which train separate models for each class [6,43], ConSE can work on the prediction score of a single model. **(b)** It is straight-forward to extend a single network to ZSD using ConSE, since [36] uses semantic embeddings only during the test phase.

Suppose, for an object proposal, vector $\mathbf{o} \in \mathbb{R}^{\mathrm{S}+1}$ contains final probability values of only seen classes and background. As described earlier, we ignore an object proposal if the background gets highest 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 $\hat{\mathbf{o}}$:

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

Then, top K score values (s.t., $K \leq \mathrm{S}$) 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 $\mathbf{e}_i$ to be a semantic space projection of an object proposal that 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 [36]. For bounding box detection, we choose the box for which corresponding seen class gets maximum score.

5 Experiments

5.1 Dataset and Experiment Protocol

Dataset: We evaluate our approach on the standard ILSVRC-2017 detection dataset [48]. 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 [48], 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.

Seen/unseen split: We propose a challenging ZSD protocol (seen/unseen splits) 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 follow 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 on how we meet these considerations are provided below.

Meta-class assignment: The classes of ILSVRC detection dataset maintain a defined hierarchy [48]. 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 [48]. 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

ID	Metaclass	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, remote-control, tape-player, traffic-light, tv or monitor, vacuum, 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**.

items, (3) flower pot is considered as furniture similar to MicroSoft COCO super-categories [30], (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.

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.

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 5). For each of our defined meta-class categories, we first plot the instance distribution of the child classes like Figure 4. 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. This random selection procedure avoids biasness, ensures diversity (due to selection from all meta-classes), semantic similarity with seen (due to presence of multiple seen classes in each meta-category) and conforms to the fact that unseen classes are not the frequent ones.

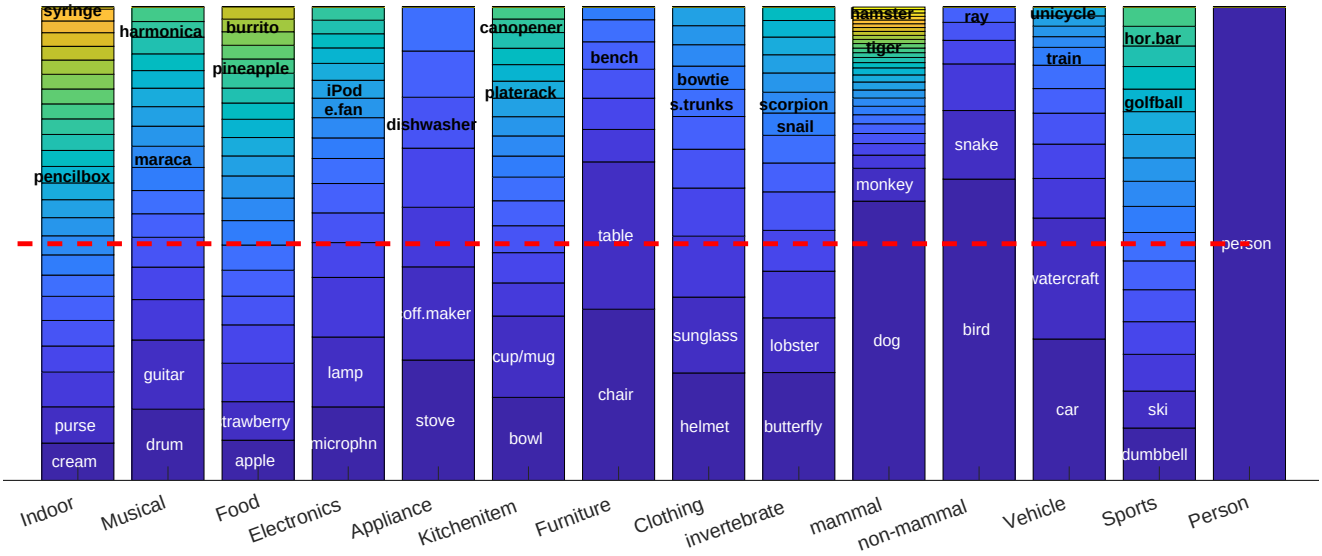

Fig. 4: Distribution of instances per classes within each meta class. Two most common (frequent) seen classes and unseen classes are marked in white and black color text respectively. Red dashed line indicates 50 percentile boundary. All unseen classes lie within the rarest half of the instance distribution.

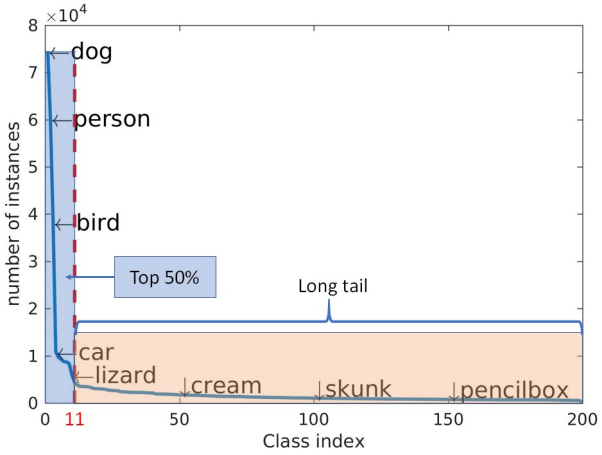

Fig. 5: Long-tail distribution of imageNet dataset

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 [12, 23]. 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 [33] and GloVe [38] 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 stages of 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. 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*.

Data Augmentation: We visualize the long-tail distribution of ILSVRC detection classes in Figure 5. 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 simi-

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 2: mAP of the unseen classes of ILSVRC-2017 detection dataset. Ours (with L'_{mm}) and Ours (with L_{cls}) denote the performance without predefined unseen and with cluster loss respectively (Sec. 4.3 and Sec. 4.2) . For cluster case, $\lambda = 0.8$.

	OVERALL	p.box	syringe	harmonica	maraca	burrito	pineapple	bowtie	s.trunk	d.washer	canopener	p.rack	bench	e.fan	iPod	scorpion	snail	hamster	tiger	ray	train	unicycle	golfball	h.bar
		Similar classes NOT present ZSD Baseline = 6.3, Ours (L'_{mm}) = **6.5**, Ours (L_{cls}) = 4.4											Similar classes present 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

Meta-class		Indoor	Musical	Food	Clothing	Appli.	Kitchen	Furn.	Electronic	Invertebra.	Mammal	Fish	Vehicle	Sport
Zero-Shot Meta Detection (ZSMD)														
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 3: Average precision of individual unseen classes of ILSVRC-2017 detection dataset 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.

lar 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.

Comparison methods: Here, we discuss different variants of our approach used for comparison in this paper. For all methods, we use the same inference strategy mentioned in Sec. 4.2.

- Baseline: We train an original Faster-RCNN [46] architecture with all seen data but without any word vectors. In this approach, we can still get a vector $\mathbf{o} \in \mathbb{R}^{S+1}$ from the classification layer of Faster-RCNN network that is used in inference. The details are given in Sec. 4.3.
- Ours (L'_{mm}): It uses our proposed architecture with word vectors as mentioned in Fig. 2. We train the network with the loss L'_{mm} discussed in Sec. 4.3. This approach does not use unseen word vectors and meta-class annotation.
- Ours (L_{cls}): This approach is same as Ours (L'_{mm}) but the training uses the loss L_{cls} discussed in Sec. 4.2. It takes advantage of unseen word vectors and meta-class annotation.

5.2 ZSD Performance on ILSVRC-2017 detection

We use two different architectures i.e., VGG-16 (V) [51] and ResNet-50 (R) [16] as the backbone of the Faster-

RCNN during the first training step. In second step, we experiment with both Word2vec and GloVe as the semantic embedding vectors used to define $\mathbf{W}_2$. Fig. 7 illustrates some qualitative ZSD examples.

Overall results: Table 2 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. Other than V+w2v case, L_{cls} achieves a higher mAP than L'_{mm} because L_{cls} considers both unseen semantics and meta-class information during training. Only for V+w2v case, the performance goes down from L'_{mm} to L_{cls}. This trend is likely due to the relatively higher noise in the w2v compared to GloVe, since even for R+w2v, the performance gain from L'_{mm} to L_{cls} is not huge. *(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. Moreover, the small performance improvement from ZSD to ZSMD in comparison to ZST to ZSMT shows that the correct localization of unseen classes is a more challenging problem as compared to their recognition (that is targetted in a multi-class labeling problem, i.e., ZST and ZSMT). *(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 [64] 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 3, 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-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 similar classes are not present (mAP 6.5 vs 4.4) because meta-class annotation could not provide sufficient supervision due to visual dissimilarity within the same superclass. The performances for some of the classes in the "Similar classes present" category are like the classes in the "Similar classes NOT present category" because those similar classes may not have sufficient instances in the training dataset to correctly relate seen to unseen. For example, unseen 'horizontal bar' has a small number of similar instances like seen 'balance beam' in the training dataset compared to unseen 'train' and seen 'bus'.

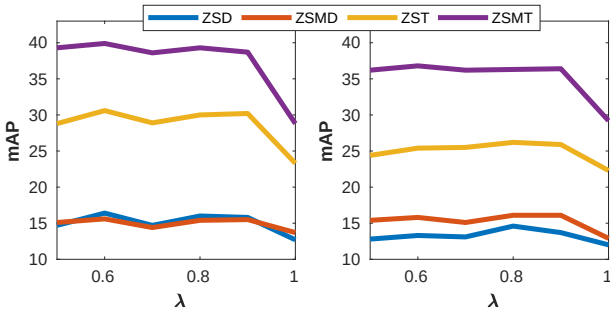

Fig. 6: Effect of varying λ in different zero-shot tasks for ResNet+w2v (left) and ResNet+glo (right).

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. Such ambiguities can happen because of object size, orientation, image clutter which make an object different from the description within the word vectors. As an example, we refer to Figure 8, where bounding boxes are incorrectly detected, although the class labels are present in the image.

Varying λ: The hyperparameter λ controls the weight between L_{mm} and L_{mc} in L_{cls}. In Fig. 6, 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. Low values of λ (i.e., $\lambda < .5$) are not reported as they lead to low emphasis on max-margin loss, resulting in somewhat lower performance.

More ablation studies: In Table 4, we compare our methods with different experimental settings. *(1)* No pre-trained model: This experiment does not use any pre-trained model. For this case, we use the training set of the proposed ImageNet-ZSD dataset and train with the max-margin loss L'_{mm}. We obtain mAP = 5.4. Note that the training was done for the same number of iterations as before, i.e., with 10 million mini-batches having one image per mini-batch (equivalent to 7 days training with a single GPU). One possible reason of low performance is that the network is not fully converged within these iterations. However, given a single GPU available to us, training a network on ImageNet DEC from scratch would take much longer which is not a feasible solution. Therefore, we opted for backbone initialization with a pre-trained network, which significantly accelerates the network convergence rate, making it feasible within the available computational budget. Furthermore, the ILSVRC detection dataset has less number of images than ILSVRC recognition and the exclusion of unseen images further reduces the data available for training. These two factors contribute towards a relatively lower performance mark for the model trained without pre-trained backbone. *(2)* Excluding overlapped unseen classes from the pre-trained model: In a recent study, [56] showed that such overlap of unseen classes introduces significant bias in the recognition performance. We empirically evaluate this bias in the detection case for the first time. We get an mAP of 12.7 (compared to previous mAP of 15.0) after excluding all overlapping unseen classes from the backbone pre-training. This shows that the existence of an overlap in the backbone can lead to higher results in the detection setting, similar to the case observed in recognition. This is despite the fact that pre-trained weights are subsequently updated based on *only* seen instances and later based on word vectors with our proposed loss. However, note that this choice of ImageNet pretrained backbone was made to be consistent with the competitive approaches [5,8] and the exact same setting is used in our baseline for fairness. *(3)* Softmax: As the standard choice for classification is to train the network with a softmax cross-entropy loss, in this experiment, we replace our max-margin loss with softmax loss. We get an mAP = 13.8 whereas with max-margin loss the mAP = 15.0. In both cases, unseen classes are not pre-defined during training. It tells that our proposed max-margin loss is better suited in ZSD settings because it can align features and semantics in a better way. In contrast, softmax loss tries to align feature and its true semantics but does not maximize the separation of the true class from rest of the classes. *(4)* ℓ_2 normalization: The word vector model can generate a vector of millions of words. But, we only use a small subset of it, which are the names of detection classes. We apply ℓ_2 normalization on this small subset to make zero mean and unit standard deviation. Using this step, we achieve a better performance than without performing this step (10.9 vs. 15.0).

Method	No pre-training	No overlap	No ℓ_2 norm	Softmax loss	Ours
mAP	5.4	12.7	10.9	13.8	15.0

Table 4: (*left* to *right*) Performance comparison when no pretrained model is used, no overlap of unseen with pretrained classes exists, without using ℓ_2 normalization on word vectors, applying softmax cross-entropy loss and our method.

Class	Seen (177)	Unseen (23)	All (200)
accuracy	47.1	49.5	47.3

Table 5: The unseen object proposal quality and its comparison with seen classes.

Unseen proposal quality: The RPN within our model generates object proposals that are later used for classification. Although the RPN is trained with only seen instances, it can localize both seen and unseen objects. This is because the RPN is trained in a class agnostic manner. RPN predicts an objectness score and bounding box regression parameters for each anchor box. During this learning process, the RPN goal is to maximize the overlap between ground-truth bounding boxes and pre-defined anchor boxes. Since no class information is used, RPN learns what qualifies an object in general. Thus, irrespective of an object class, RPN can provide object proposals. In this experiment, we attempt to assess the quality of object proposals found by RPN. Given an image as input, RPN is set to provide a maximum of 100 proposals. Then, we apply NMS on those proposals to remove highly overlapping proposals. For each ground-truth bounding box in an input image, we calculate IoU with all proposals. If any of the ground-truth boxes get a suitable match with IoU $\geq .5$, we consider that the box is correctly localized. In this way, we calculate the percentage of correctly localized ground truth box for each class. In Table 5, we report this percentage for 177 seen and 23 unseen and all 200 classes. It shows that RPN is successful in covering a significant amount of seen and unseen bounding boxes. Here, because of the different ratio and frequency of seen and unseen objects, the unseen class percentage becomes higher than seen.

Method	ZSD	GZSD		
	mAP/RE	Seen mAP/RE	Unseen mAP/RE	HM mAP/RE
LAB [5]	0.27/20.52	-	-	-
SB [5]	0.70/24.39	-	-	-
DSES [5]	0.54/**27.19**	-/15.02	-/**15.32**	-/15.17
Ours	**5.05**/12.27	**13.93**/**20.42**	**2.55**/12.42	**4.31**/**15.45**

Table 6: Performance on ZSD and GZSD tasks on MSCOCO dataset.

Method	pullover	dress	ankle-boot	mean
Demirel *et al.* [8]	49.0	49.0	95.0	64.9
Ours	**70.4**	**58.6**	**99.6**	**76.2**

Table 7: Performance on ZSD tasks on Fashion-ZSD Dataset

5.3 ZSD on MS-COCO

Recently, Bansal *et al.* [5] proposed a seen/unseen split on MS-COCO (2014) dataset for evaluating zero-shot object detection. Out of total 80 classes they used 48 and 17 classes for seen and unseen respectively. This setting considers 73,774 images containing seen objects and 6,608 images for testing unseen objects. In this paper, we adopt this setting to compare our method with [5]. In Table 6, we report both ZSD and GZSD performances on mAP and Recall@100 based evaluation. For fair comparison, our results are based on only L_{mm}, i.e., using $\lambda = 1$ so that the training do not have access about the unseen knowledge. For ZSD task, with mAP our method beats LAB, DSES and SB [5] with a large margin (5.05 vs. 0.27, 0.54 and 0.70). However, with Recall@100, we notice an opposite trend. Although [5] proposed Recall@100 to evaluate ZSD, we argue that this metric is sub-optimal because it does not penalize for wrong bounding box detections by a model[3]. For GZSD, our method successfully outperforms DSES [5] in Recall measure. However, we support ZSD evaluations based on mAP measure similar to the traditional object detection problem since it is a more comprehensive evaluation measure.

5.4 ZSD on Fashion-MNIST

Demirel *et al.* [8] generate a toy dataset for ZSD based on Fashion-MNIST [60]. This dataset includes three objects per image to make it suitable for multi-object detection. Moreover, to increase the task complexity, some generated images contain randomly cropped objects as clutter. They use 7 seen and 3 unseen classes and 8.3k, 8k and 8k images for training, validation and testing, respectively. In Table 7, we adopt their settings to compare our ZSD method with [8]. The results show that our proposed approach performs favorably well against [8]. Note that, we do not use pre-defined unseen in this experiment.

[3] Although, we acknowledge that Recall@100 stays an appropriate measure for large-scale datasets that are not fully labeled (such as Visual Genome-see Sec. 5.5).

Method	SB [5]	DSES [5]	LAB [5]	Ours
Recall	4.09	4.75	**5.40**	2.02

Table 8: Performance on ZSD task on Visual Genome dataset.

5.5 ZSD on Visual Genome (VG)

In Table 8, we perform experiments with VG dataset [22] using Bansal *et al.* settings [5]. Our performance with max-margin loss in terms of recall@100 is 2.02, whereas [5] reports 5.4 in the same setting. We believe one possible reason of this performance gap is that our approach considers relatively less number of bounding box proposals as compared to [5]. Since the average number of instances per image is very high for VG dataset 21.24 (MSCOCO has 7.7), considering a large number of proposals per image is useful during training on VG. However, our Faster-RCNN based model runs in an end-to-end manner on a single GPU, putting a restriction on the number of proposals (only 32 bounding boxes per image are considred in our case). In contrast, [5] used an off-line bounding box predictor (based on Edge-box proposals), which allows them to consider a significantly large number of proposals per image. Additionally, [5] is a background-aware approach. Instead of one general background class, their LAB variant considers an extensive number of 1673 classes (those are neither seen nor unseen) as the background, which may have been a contributing factor for their approach on the VG dataset. Therefore, as a future work, one can consider combining such a background-aware approach, together with the proposed Faster-RCNN model to further improve ZSD performance on VG dataset.

5.6 ZSD on CUB

We evaluate the ZSD performance of the baseline and our proposed method based on a single bounding box per image provided in CUB dataset [53]. Table 9 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

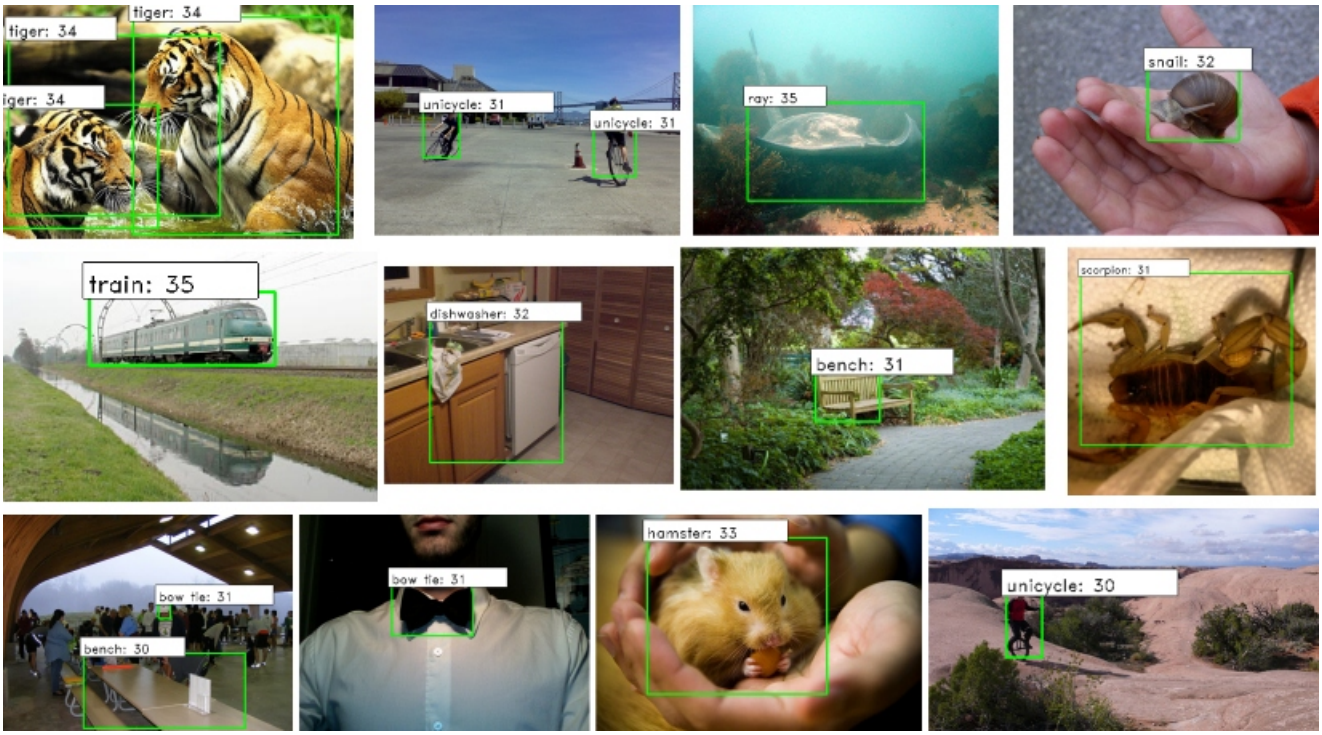

Fig. 7: Selected examples of ZSD of our cluster ($\lambda = .6$) method with R+w2v, using the prediction score threshold = 0.3. The numbers represents the prediction scores in percent. Images are from ILSVRC-2017 detection dataset

any meta-class for the CUB classes. Therefore, we use $\lambda = 1$ for CUB related experiments.

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

5.7 Zero Shot Recognition on CUB

Being a detection model, the proposed network can also perform traditional Zero Shot Recognition (ZSR). We evaluate ZSR performance on popular Caltech-UCSD Birds-200-2011 (CUB) dataset [53]. This dataset contains 11,788 images from 200 classes and provides a single bounding box per image. Following standard train/test split [56], we use 150 seen and 50 unseen classes for experiments. For semantics embedding, we use 400-d word2vec (w2v) and GloVe (glo) vector [55]. Note that, we do not use per image part annotation (like [1]) and descriptions (like [64]) 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 10. We find our proposed solution achieves a significant improvement in performance compared to state-of-the-art methods.

Top1 Accuracy	Network	w2v	glo
Akata'16 [1]	V	33.90	-
DMaP-I'17[27]	G+V	26.38	30.34
SCoRe'17[35]	G	31.51	-
Akata'15 [3]	G	28.40	24.20
LATEM'16 [55]	G	31.80	32.50
DMaP-I'17 [27]	G	26.28	23.69
Ours	R	**36.77**	**36.82**

Table 10: 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.

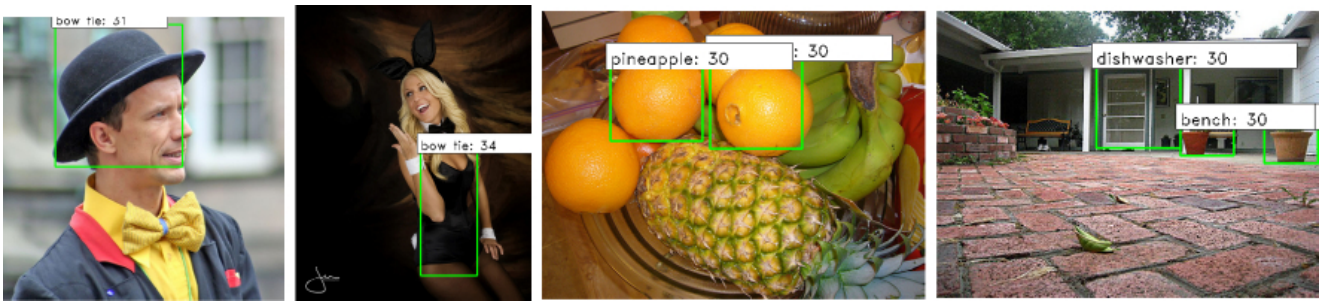

Fig. 8: Examples of incorrect detection but correct classification. The unseen class 'bow-tie', 'pineapple' and 'bench' are incorrectly localized in these images. . Images are from ILSVRC-2017 detection dataset

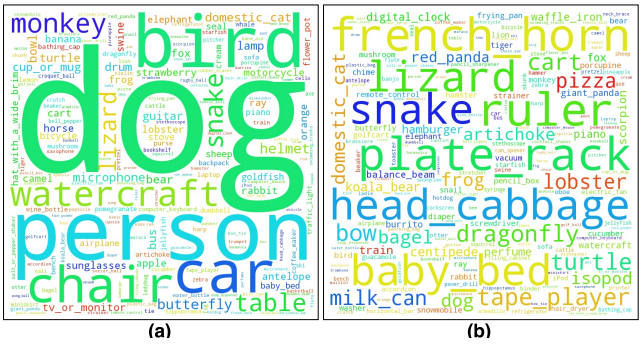

Fig. 9: Word cloud based on (a) number of object instance (b) Mean object size in pixel

5.8 Qualitative results

We provide examples of ZSD in Fig. 7 using ILSVRC-2017 detection dataset. One can find that the prediction score threshold is lower (0.3 used in the examples) than the value (greater than 0.5) used in traditional object detection like faster-RCNN [46]. 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 generally noisy. This degrades the overall confidence for ZSD.

In the last layer of the box regression branch, our method does not have specified bounding boxes for unseen 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 does not always provide/obtain very accurate localizations, as illustrated in Fig. 8.

5.9 Discussion

ZSD Challenges: In general, detection is a more difficult task than recognition/tagging because the bounding box must be located at the same time. The strict requirement of not using any unseen class images during the zero-shot training is a tough enough condition for recognition/tagging tasks, which is significantly intensified in detection tasks. We have used the ILSVRC-2017 detection dataset to evaluate some baseline performances of the proposed problem. This dataset has 200 classes, including a total of 478,807 object instances of different shapes/sizes and distributions (see Figure 9). Within these, we define M = 14 meta classes, which contain one or more specific classes. Figure 4 describes the normalized number of instances per class within each meta class. Considering this challenging dataset, here we describe some other difficulties of the zero shot detection task:

- *Rarity:* The ILSVRC dataset contains a long-tail distribution issue, i.e., many rare classes have a lower 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 lie in the rarest 50% in the distribution. This affects the zero-shot version of the problem as well.
- *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 has a different number of classes and there exists a high visual diversity between meta-class images. Since being in a same meta-class does not guarantee visual similarity, it is difficult to learn relationships for the unseen categories that are quite different from the seen categories in the same super-class. As an example, 'tiger' has more similar classes than 'ray'. The scarcity of similar classes produces an inadequate description

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 11: 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.

of the unseen class, which eventually affects 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 as they are generated automatically from unannotated text mining. This also affects the zero-shot detection performance significantly.

Seen vs. Unseen Class Performance: The overall performance of ZSD is dependent on the learning of seen classes. Therefore, the performance of seen-class detection can be an indication of how ZSD works. To this end, we also study the detection performance for seen classes of the ILSVRC validation dataset after the first step of faster-RCNN training (Table 11). It indicates the baseline performance of seen classes that leads to our final ZSD performance on the unseen. The baseline method result is better than our proposed approaches. This is justifiable since our proposed methods generate predictions for both seen and unseen class together, which somewhat sacrifices the seen performance to achieve distinction among all seen and unseen classes. Table 11 also compares the seen result with the unseen performance. It is found that the performance of selected unseen classes is similar to that of seen classes for our (L_{cls}) method. This indicates the 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 cases, one can define a meta-class in an unsupervised manner by applying a clustering mechanism on the original semantic embedding.

ZSL vs ZSD loss: Many traditional non-end-to-end trainable ZSR methods consider different aspects of regularization [35], transductive setting [27], metric learning [32], domain adaptation [20] and class attribute association [4] etc. Similarly, the end-to-end trainable ZSR methods [64, 25] employ different non-linearities in feature and semantic pipeline. However, those traditional loss formulations must be redesigned for ZSD to be compatible with both classification and box detection losses.

Future challenges: The ZSD problem warrants further investigation. *(1)* Instead of mapping image features to the semantic space, the reverse mapping can help ZSD as it does for the case of ZSR [21, 64]. *(2)* ZSD might benefit from fusing different word vectors (word2vec and GloVe). *(3)* Like generalized ZSL [61, 56, 27], a generalized ZSD setting can be explored, which represents a more realistic set-up. *(4)* Moreover, weakly/semi-supervised version of ZSD/GZSD is also an interesting direction for further research.

6 Conclusion

While traditional ZSL research focuses only on object recognition, we propose to extend the problem to object detection (ZSD). To this end, we offer a new experimental protocol for the ILSVRC-2017 dataset, specifying the seen-unseen, train-test split. We also develop an end-to-end trainable CNN model to solve this problem. Our proposed approach employs a novel loss function to relate semantic and visual features of seen object classes with the unseen objects. We show that our solution is better than a strong baseline and recently reported zero-shot detection approaches.

Overall, this research throws several new challenges to the ZSL community. To make long-standing progress in ZSL, the community needs to move forward in the detection setting rather than merely recognition. Furthermore, the interesting extensions of ZSD setting, such as the any-shot detection [42], can lead to more practical scenarios close to the real-world.

References

1. Akata, Z., Malinowski, M., Fritz, M., Schiele, B.: Multi-cue zero-shot learning with strong supervision. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
2. Akata, Z., Perronnin, F., Harchaoui, Z., Schmid, C.: Label-Embedding for Image Classification. IEEE Transactions on Pattern Analysis and Machine Intelligence **38**(7), 1425–1438 (2016). DOI 10.1109/TPAMI.2015.2487986
3. Akata, Z., Reed, S., Walter, D., Lee, H., Schiele, B.: Evaluation of output embeddings for fine-grained image classification. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 07-12-June-2015, pp. 2927–2936 (2015). DOI 10.1109/CVPR.2015.7298911
4. Al-Halah, Z., Tapaswi, M., Stiefelhagen, R.: Recovering the missing link: Predicting class-attribute associations for unsupervised zero-shot learning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)

5. Bansal, A., Sikka, K., Sharma, G., Chellappa, R., Divakaran, A.: Zero-shot object detection. In: The European Conference on Computer Vision (ECCV) (2018)
6. Changpinyo, S., Chao, W.L., Gong, B., Sha, F.: Synthesized classifiers for zero-shot learning. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2016-January, pp. 5327–5336 (2016)
7. Dai, J., Li, Y., He, K., Sun, J.: R-FCN: Object detection via region-based fully convolutional networks. arXiv preprint arXiv:1605.06409 (2016)
8. Demirel, B., Cinbis, R.G., Ikizler-Cinbis, N.: Zero-shot object detection by hybrid region embedding. In: British Machine Vision Conference (BMVC) (2018)
9. Demirel, B., Gokberk Cinbis, R., Ikizler-Cinbis, N.: Attributes2classname: A discriminative model for attribute-based unsupervised zero-shot learning. In: The IEEE International Conference on Computer Vision (ICCV) (2017)
10. Deng, J., Ding, N., Jia, Y., Frome, A., Murphy, K., Bengio, S., Li, Y., Neven, H., Adam, H.: Large-scale object classification using label relation graphs. In: ECCV, pp. 48–64. Springer (2014)
11. Deutsch, S., Kolouri, S., Kim, K., Owechko, Y., Soatto, S.: Zero shot learning via multi-scale manifold regularization. In: CVPR (2017)
12. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pp. 1778–1785. IEEE (2009)
13. Frome, A., Corrado, G.S., Shlens, J., Bengio, S., Dean, J., Ranzato, M.A., Mikolov, T.: Devise: A deep visual-semantic embedding model. In: C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K.Q. Weinberger (eds.) Advances in Neural Information Processing Systems 26, pp. 2121–2129. Curran Associates, Inc. (2013)
14. Fu, Y., Yang, Y., Hospedales, T., Xiang, T., Gong, S.: Transductive multi-label zero-shot learning. arXiv preprint arXiv:1503.07790 (2015)
15. Fu, Z., Xiang, T., Kodirov, E., Gong, S.: Zero-shot learning on semantic class prototype graph. IEEE Transactions on Pattern Analysis and Machine Intelligence **PP**(99), 1–1 (2017). DOI 10.1109/TPAMI.2017.2737007
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
17. Hu, R., Xu, H., Rohrbach, M., Feng, J., Saenko, K., Darrell, T.: Natural language object retrieval. In: CVPR, pp. 4555–4564 (2016)
18. Jayaraman, D., Grauman, K.: Zero-shot recognition with unreliable attributes. In: Advances in neural information processing systems, pp. 3464–3472 (2014)
19. Jetley, S., Sapienza, M., Golodetz, S., Torr, P.H.: Straight to shapes: Real-time detection of encoded shapes. arXiv preprint arXiv:1611.07932 (2016)
20. Kodirov, E., Xiang, T., Fu, Z., Gong, S.: Unsupervised domain adaptation for zero-shot learning. In: The IEEE International Conference on Computer Vision (ICCV) (2015)
21. Kodirov, E., Xiang, T., Gong, S.: Semantic autoencoder for zero-shot learning. In: CVPR (2017)
22. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., Bernstein, M.S., Fei-Fei, L.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. International Journal of Computer Vision **123**(1), 32–73 (2017)
23. Lampert, C., Nickisch, H., Harmeling, S.: 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, pp. 951–958 (2009). DOI 10.1109/CVPRW.2009.5206594
24. Lampert, C.H., Nickisch, H., Harmeling, S.: Attribute-based classification for zero-shot visual object categorization. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(3), 453–465 (2014). DOI 10.1109/TPAMI.2013.140
25. Lei Ba, J., Swersky, K., Fidler, S., et al.: Predicting deep zero-shot convolutional neural networks using textual descriptions. In: CVPR, pp. 4247–4255 (2015)
26. Li, X., Liao, S., Lan, W., Du, X., Yang, G.: Zero-shot image tagging by hierarchical semantic embedding. In: RDIR, pp. 879–882. ACM (2015)
27. Li, Y., Wang, D., Hu, H., Lin, Y., Zhuang, Y.: Zero-shot recognition using dual visual-semantic mapping paths. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
28. Li, Z., Gavves, E., Mensink, T., Snoek, C.G.: Attributes make sense on segmented objects. In: European Conference on Computer Vision, pp. 350–365. Springer (2014)
29. Li, Z., Tao, R., Gavves, E., Snoek, C., Smeulders, A.: Tracking by natural language specification. In: CVPR, pp. 6495–6503 (2017)
30. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV, pp. 740–755. Springer (2014)
31. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: Single Shot MultiBox Detector, pp. 21–37. Springer International Publishing, Cham (2016). DOI 10.1007/978-3-319-46448-0_2. URL `https://doi.org/10.1007/978-3-319-46448-0_2`
32. Maxime Bucher, S.H., Jurie, F.: Improving semantic embedding consistency by metric learning for zero-shot classification. In: Proceedings of The 14th European Conference on Computer Vision (2016)
33. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K.Q. Weinberger (eds.) Advances in Neural Information Processing Systems 26, pp. 3111–3119. Curran Associates, Inc. (2013)
34. Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM **38**(11), 39–41 (1995)
35. Morgado, P., Vasconcelos, N.: Semantically consistent regularization for zero-shot recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
36. Norouzi, M., Mikolov, T., Bengio, S., Singer, Y., Shlens, J., Frome, A., Corrado, G.S., Dean, J.: Zero-shot learning by convex combination of semantic embeddings. arXiv preprint arXiv:1312.5650 (2013)
37. Palatucci, M., Pomerleau, D., Hinton, G.E., Mitchell, T.M.: Zero-shot learning with semantic output codes. In: Y. Bengio, D. Schuurmans, J.D. Lafferty, C.K.I. Williams, A. Culotta (eds.) Advances in Neural Information Processing Systems 22, pp. 1410–1418. Curran Associates, Inc. (2009)
38. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)

39. Rahman, S., Khan, S., Barnes, N.: Polarity loss for zero-shot object detection. arXiv preprint arXiv:1811.08982 (2018)
40. Rahman, S., Khan, S., Barnes, N.: Transductive learning for zero-shot object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 6082–6091 (2019)
41. Rahman, S., Khan, S., Barnes, N.: Improved visual-semantic alignment for zero-shot object detection. In: AAAI, pp. 11,932–11,939 (2020)
42. Rahman, S., Khan, S., Barnes, N., Khan, F.S.: Any-shot object detection. arXiv preprint arXiv:2003.07003 (2020)
43. Rahman, S., Khan, S., Porikli, F.: A unified approach for conventional zero-shot, generalized zero-shot, and few-shot learning. IEEE Transactions on Image Processing **27**(11), 5652–5667 (2018). DOI 10.1109/TIP.2018.2861573
44. Rahman, S., Khan, S., Porikli, F.: Zero-shot object detection: Learning to simultaneously recognize and localize novel concepts. In: C.V. Jawahar, H. Li, G. Mori, K. Schindler (eds.) Computer Vision – ACCV 2018, pp. 547–563. Springer International Publishing, Cham (2019)
45. Redmon, J., Farhadi, A.: Yolo9000: Better, faster, stronger. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
46. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(6), 1137–1149 (2017). DOI 10.1109/TPAMI.2016.2577031
47. Romera-Paredes, B., Torr, P.: An embarrassingly simple approach to zero-shot learning. In: Proceedings of The 32nd International Conference on Machine Learning, pp. 2152–2161 (2015)
48. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) **115**(3), 211–252 (2015). DOI 10.1007/s11263-015-0816-y
49. Schonfeld, E., Ebrahimi, S., Sinha, S., Darrell, T., Akata, Z.: Generalized zero- and few-shot learning via aligned variational autoencoders. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
50. Shigeto, Y., Suzuki, I., Hara, K., Shimbo, M., Matsumoto, Y.: Ridge regression, hubness, and zero-shot learning. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 135–151. Springer (2015)
51. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
52. Socher, R., Ganjoo, M., Manning, C.D., Ng, A.: Zero-shot learning through cross-modal transfer. In: C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K.Q. Weinberger (eds.) Advances in Neural Information Processing Systems 26, pp. 935–943. Curran Associates, Inc. (2013)
53. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset. Tech. Rep. CNS-TR-2011-001, California Institute of Technology (2011)
54. Wang, X., Ji, Q.: A unified probabilistic approach modeling relationships between attributes and objects. Proceedings of the IEEE International Conference on Computer Vision pp. 2120–2127 (2013). DOI 10.1109/ICCV.2013.264
55. Xian, Y., Akata, Z., Sharma, G., Nguyen, Q., Hein, M., Schiele, B.: Latent embeddings for zero-shot classification. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
56. Xian, Y., Lampert, C.H., Schiele, B., Akata, Z.: Zero-shot learning - a comprehensive evaluation of the good, the bad and the ugly. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–1 (2018). DOI 10.1109/TPAMI.2018.2857768
57. Xian, Y., Lampert, C.H., Schiele, B., Akata, Z.: Zero-shot learning - a comprehensive evaluation of the good, the bad and the ugly. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–1 (2018). DOI 10.1109/TPAMI.2018.2857768
58. Xian, Y., Lorenz, T., Schiele, B., Akata, Z.: Feature generating networks for zero-shot learning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
59. Xian, Y., Sharma, S., Schiele, B., Akata, Z.: F-vaegan-d2: A feature generating framework for any-shot learning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
60. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms (2017)
61. Xu, X., Shen, F., Yang, Y., Zhang, D., Shen, H.T., Song, J.: Matrix tri-factorization with manifold regularizations for zero-shot learning. In: Proc. of CVPR (2017)
62. Ye, M., Guo, Y.: Zero-shot classification with discriminative semantic representation learning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
63. Yu, F.X., Cao, L., Feris, R.S., Smith, J.R., Chang, S.F.: Designing category-level attributes for discriminative visual recognition. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pp. 771–778 (2013). DOI 10.1109/CVPR.2013.105
64. Zhang, L., Xiang, T., Gong, S.: Learning a deep embedding model for zero-shot learning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
65. Zhang, Y., Gong, B., Shah, M.: Fast zero-shot image tagging. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
66. Zhang, Z., Saligrama, V.: Zero-shot learning via semantic similarity embedding. In: The IEEE International Conference on Computer Vision (ICCV) (2015)
67. Zhang, Z., Saligrama, V.: Zero-shot learning via joint latent similarity embedding. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
68. Zhu, P., Wang, H., Bolukbasi, T., Saligrama, V.: Zero-shot detection. arXiv preprint arXiv:1803.07113 (2018)