OVIS: Open-Vocabulary Visual Instance Search via Visual-Semantic Aligned Representation Learning

Sheng Liu[1] Kevin Lin[2] Lijuan Wang[2] Junsong Yuan[1] Zicheng Liu[2]
[1]University at Buffalo [2]Microsoft
{sliu66, jsyuan}@buffalo.edu {keli, lijuanw, zliu}@microsoft.com

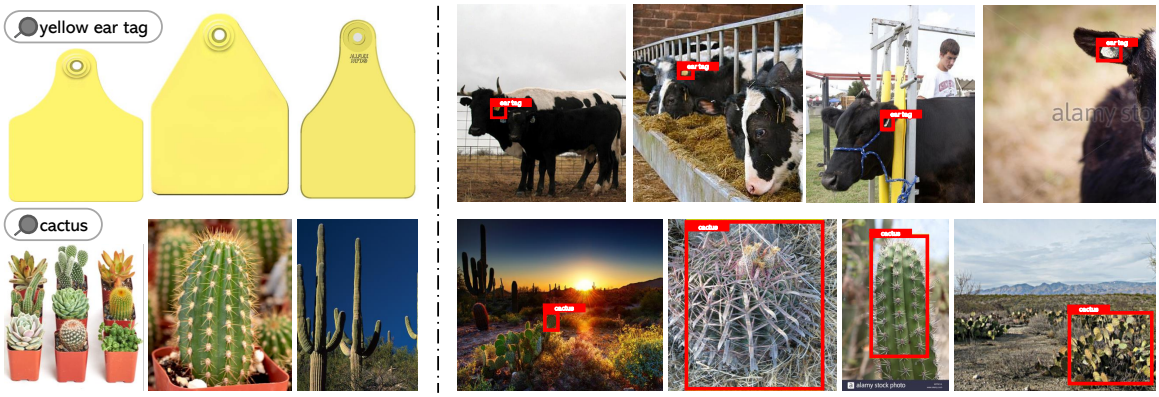

Figure 1: Comparison between Google Image and ViSA, *i.e.*, our model for OVIS. Similarities: both take as input a textual search query, *e.g.*, `yellow ear tag`, `cactus`, and return a ranked list of visual contents. Differences: (1) Google Image returns images, while ViSA returns visual instances, *i.e.*, localized regions (shown with red bounding boxes). ViSA is able to find *small* visual instances and visual instances in *context*. (2) Google Image heavily relies on textual metadata of images, while ViSA relies on visual contents only (more details are provided in the supplementary materials).

Abstract

We introduce the task of open-vocabulary visual instance search (OVIS). Given an arbitrary textual search query, Open-vocabulary Visual Instance Search (OVIS) aims to return a ranked list of visual instances, i.e., image patches, that satisfies the search intent from an image database. The term "open vocabulary" means that there are neither restrictions to the visual instance to be searched nor restrictions to the word that can be used to compose the textual search query. We propose to address such a search challenge via visual-semantic aligned representation learning (ViSA). ViSA leverages massive image-caption pairs as weak image-level (not instance-level) supervision to learn a rich cross-modal semantic space where the representations of visual instances (not images) and those of textual queries are aligned, thus allowing us to measure the similarities between any visual instance and an arbitrary textual query. To evaluate the performance of ViSA, we build two datasets named OVIS40 and OVIS1600 and also introduce a pipeline for error analysis. Through extensive experiments on the two datasets, we demonstrate ViSA's ability to search for visual instances in images not available during training given a wide range of textual queries including those composed of uncommon words. Experimental results show that ViSA achieves an mAP@50 of 21.9% on OVIS40 under the most challenging setting and achieves an mAP@6 of 14.9% on OVIS1600 dataset.

1. Introduction

The sheer number of image searches perfectly reflects its importance. Tens of millions of image searches are carried out in a single day by image search engines, *e.g.*, Google [1], in a single day. Taking a textual search query, *e.g.*, a word "`ovis`"[1] as input, an image search engine returns a list of images relevant to the query. In this sense, an image search engine can be viewed as mapping textual search queries to visual search results. Despite promising text-to-image search results, image search engines like Google often rely on textual descriptions of images, *e.g.*, alt-texts and titles, and not on visual contents of images. In addition, existing image search engine typically returns a whole image rather than locating the textual query in the image.

[1] `ovis`: a genus of mammals, whose species are known as sheep [2].

In this work, we introduce the task of open-vocabulary visual instance search (OVIS). Given a textual search query, *e.g.*, "`ovis`", "`marble column`", OVIS aims to return visual instances, *i.e.*, image patches (instead of images)[2], which are relevant to the query, solely relying on the visual contents of images. We use the term "open" as we do not limit the visual instances that can be searched, it can be instances of any objects, movements and attributes. In contrast, works on image retrieval mainly focus on retrieving *whole images* of a *closed* set of classes [27, 8, 43, 44, 42, 20, 12, 46]. Furthermore, we do not restrict the words that can be used in the textual queries. Words from any part of speech can be used, *e.g.*, nouns, verbs and adjectives.

The vast number of the visual instances to be searched and the textual search queries makes OVIS a challenge. While state-of-the-art computer vision models have achieved great success in many areas, they often have a closed vocabulary limited by the annotated categories. The vocabulary of an object detector is limited, for example, by the number of object classes with bounding box annotations. They cannot detect classes of objects with no bounding box annotations. However, it is infeasible for us to create a sufficiently large dataset that, covers all the possibilities of the visual instances as well as the textual search queries, due to their large numbers.

To address this challenge, we propose to use a large number of image captions that can be collected by a web crawler to train our model. However, captions describe images rather than visual instances. Therefore, captions can only serve as weak supervision, as we have to associate words / phrases of the captions with visual instances in images without explicit supervision. This is achieved with the help of masked token prediction, which is a task that attempts to predict the masked token in the caption based on visual instances in the image and the other tokens. In order to correctly predict the masked token, our model must attend to visual instances relevant to the masked token. In this way, an implicit association is achieved. As a result, our model is able to encode visual instances and textual search queries into representations that are aligned in a common semantic space. In other words, visual instances and textual search queries with similar semantics have similar representations. We also use a small number of textual visual instance labels so that our model can explicitly associate visual instances to tokens in the labels during training. While we only use a *small closed* set of textual labels, they serve as anchors that ease the learning of the alignment between the representations of visual instances and textual queries.

We collect OVIS40 and OVIS1600 datasets with $\sim$ 6K and $\sim$ 5K visual instances, which corresponds to 40 and $1,600$ sophisticated queries with different characteristics in order to evaluate our model. These two datasets can serve as benchmarks for future research in this direction. In addition, we propose an error analysis pipeline with which the sources of error in OVIS models can be analyzed.

[2]The terms, "visual instance" and "image patch", are used interchangeably in this manuscript.

	IR	WSOD	OV-CLS	OVIS
Incomplete supervision?	✗	✓	✓	✓
Instance-level?	✗	✓	✗	✓
Open-vocabulary?	✗	✗	✓*	✓

Table 1: Comparison of different tasks related to OVIS. IR: image retrieval; WSOD: weakly supervised object detection; OV-CLS: open-vocabulary image classification [13] (* indicates that [13] is not able to classify images whose labels do not have word vectors).

2. Backgrounds

We compare OVIS with three related tasks: image retrieval (IR), weakly supervised object detection (WSOD) and open-vocabulary image classification (OV-CLS). Key features of these tasks are shown in Table 1. In addition to these tasks, OVIS is also closely related to image-text retrieval and large-vocabulary instance segmentation [14, 39].

Image Retrieval (IR): Given an image as input, the goal of IR is to retrieve images that are similar or have similar semantics to the given image in an image database. Supervised hashing [27, 8, 43, 44, 42] has become a paradigm for IR due to its low computational cost. In contrast to OVIS, IR models are trained to retrieve images of a *closed* set of classes in a supervised manner. Moreover, IR models retrieve images, while OVIS retrieves visual instances.

Weakly Supervised Object Detection (WSOD): WSOD aims to train an object detector without using bounding-box annotations. Prior studies [31, 18, 45, 40, 38, 17, 16] use image tags as supervision, and Ye *et al.* [41] use image-caption pairs as supervision. In addition, Weakly Supervised Object Localization (WSOL) [47, 6, 10, 29] is a related topic to WSOD. WSOL aims to localize a single class-specific region in an image. Contrary to OVIS, WSOD and WSOL only focus on a fixed set of object classes.

Open-Vocabulary Image Classification (OV-CLS): Given an image as input, the goal of OV-CLS [13] is to assign a class label to the image. The main difference between OVIS and OV-CLS is that OV-CLS assigns image-level labels, while OVIS returns a list of visual instances, *i.e.*, image patches.

Vision-Language Pre-Training: Vision-language pre-trained (VLP) models [24, 9, 48, 34, 35, 22, 7, 23] are highly successful in learning cross-modal representations for various vision-language tasks using image captions as supervision. Our model is not a VLP model, as our model

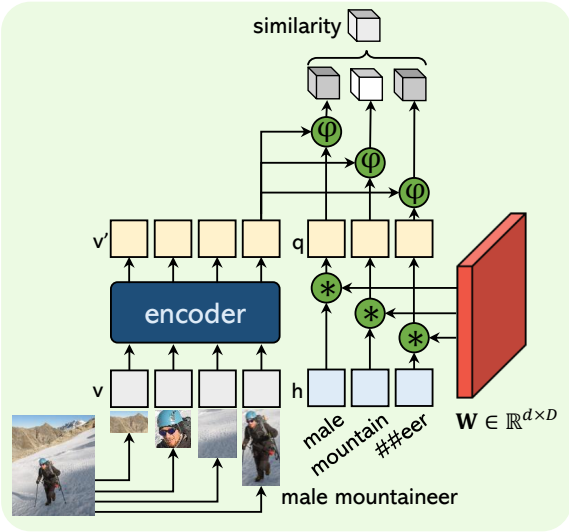

Figure 2: The way our model computes the similarity between a textual query "male mountaineer" and the 4-th visual instance in an image at test time. Our model consists of a visual-semantic encoder and a base-token embedding matrix $\mathbf{W}$. ⊛: matrix multiplication operation; ⓟ: similarity measure, *e.g.*, cosine similarity.

is directly applied to OVIS after being trained, while VLP models have to be *finetuned* for *downstream* tasks. In addition, our model is mainly trained in a weakly-supervised manner as captions only provide image-level (not instance-level) annotations, while VLP models have to be finetuned in a supervised manner.

3. Method

In this section, we first introduce how our model can be used at test time (assuming it has been trained). We then introduce how we train our model via visual-semantic aligned representation learning. To this end, we discuss a preprocessing scheme that could be used to speed up the search process.

Inference: Essentially, a search problem, *e.g.*, OVIS, can be solved once we are able to measure the similarity between a search query and the items to be searched in a database, as items can be ranked and selected according to their similarity with the given query. In our case, we aim to compute the similarity between a textual search query, *i.e.*, an *arbitrary* word or phrase consisting of less than 4 words[3] in the set of all *147K* words in current use, and an *arbitrary* visual instance.

As shown in Figure 2, our model consists of a visual-semantic encoder, *i.e.*, a Transformer encoder [37], and a base-token embedding matrix $\mathbf{W} \in \mathbb{R}^{d \times D}$, where D is the size of the dictionary of our model.

[3]We focus on short queries as more than 80% of web search queries have less than 4 words [33].

Given a textual query, we tokenize the query into a set of tokens in the dictionary of our model. For example, "male mountaineer" is tokenized into "male", "mountain" and "##eer". Thanks to tokenization, our model can handle any word in the set of *147K* words in current use, even if it does not appear in our model's dictionary, *e.g.*, "mountaineer". We then encode the tokens into vector representations in a semantic space $\mathcal{S}$ via $\mathbf{q}_i = \mathbf{W} \cdot \mathbf{h}_i$, where $\mathbf{q}_i \in \mathbb{R}^d$ and $\mathbf{h}_i \in \{0,1\}^D$ denote the vector representation and one-hot vector of the i-th token. In other words, we encode each token using a column of the base-token embedding matrix $\mathbf{W}$.

Given an image $\mathbf{I}$, we use a pretrained visual backbone to identify n visual instances in it and extract their features, $[\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n], \mathbf{v}_j \in \mathbf{R}^d$. The sequence of features are encoded *jointly* by our visual-semantic encoder into a sequence of *contextualized* representations, $[\mathbf{v}'_1, \mathbf{v}_2, \ldots, \mathbf{v}_n], \mathbf{v}_j \in \mathbf{R}^d$, which are in the same semantic space $\mathcal{S}$ as token representations.

We then compute the similarity between the representation of a visual instance $\mathbf{v}'_j$ and the representation of each token $\mathbf{q}_i$ with a similarity measure $\psi : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$, *e.g.*, cosine similarity. We will compare different instantiation of ψ in Section 5.2. The similarity between a visual instance and a textual query is the average of the similarity between the visual instance's representation and each token's representation computed with ψ. Visual instances are ranked according to their similarities with the textual query.

To ensure that the computed similarities are meaningful, it is essential that the representations of both the tokens, *i.e.*, columns of $\mathbf{W}$, and those of visual instances are aligned in the same semantic space $\mathcal{S}$. Similarity between representations in different semantic space is not meaningful, for example, similarity between the feature of a visual instance $\mathbf{v}_j$ and the one-hot vector of a token $\mathbf{h}_i$ is meaningless. Therefore, our goal is to train the visual-semantic encoder and the base-token embedding matrix $\mathbf{W}$ so that they can align representations of visual instances and tokens in a common semantic space $\mathcal{S}$. In other words, our goal is to ensure representations of visual instances and tokens with similar semantics have great similarities, while those with different semantics have little similarities, for example, visual instances of a mountaineer are very similar to token "mountain" and token "##eer" and have little similarities with token "dolphin".

Visual-Semantic Aligned Representation Learning: Should we were able to build a dataset containing all possible visual instances and all possible textual search queries with which to search them, we would be able to learn such an alignment in a supervised manner by directly maximizing the similarity between a visual instance and search query whose semantics are alike. However, it is infeasible to build such an enormous dataset. Therefore, we propose

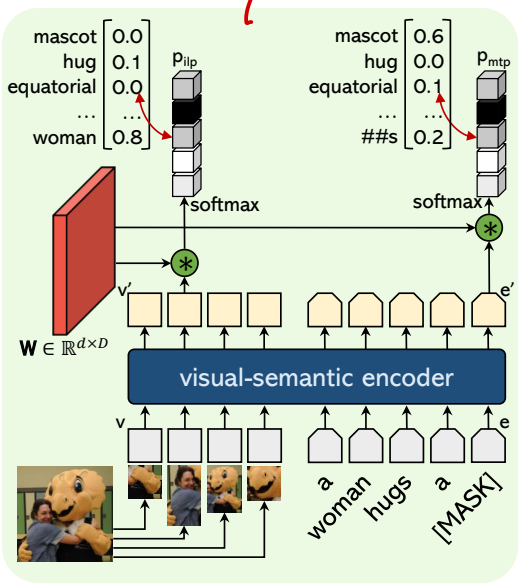

Figure 3: The way our model is trained. Each training sample is an image-caption pair, represented as visual instance features □ and token embeddings ⌒. We train our model using two tasks: masked token prediction (MTP), that aims to predict the masked token, and visual instance label prediction (ILP) whose goal is to predict the textual labels for visual instances that belong to a *small closed* set of classes. ⊛: matrix multiplication operation.

to learn the alignment via visual-semantic aligned representation learning, which mainly leverages image captions collected by a web crawler, as image-level supervision. As captions describe images instead of visual instances, it is therefore important that, during training, we can make associations between words / phrases in the captions and visual instances in images.

To achieve such a goal, we simply mask a percentage of tokens (replace with a special "[MASK]" token) in a caption at random (*e.g.*, the 5-th token in Figure 3) and then predict the masked token from the other tokens in the caption and the visual instances in the image described by the caption. Such a process is often referred to as masked token prediction (MTP). As shown in Figure 3, the visual-semantic encoder takes a concatenation of m caption token embeddings $[\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m], \mathbf{e}_j \in \mathbb{R}^d$ and n visual instance features $[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n], \mathbf{x}_i \in \mathbb{R}^d$ as input. It encodes both of them together into $[\mathbf{e}_1^{'}, \mathbf{e}_2^{'}, \ldots, \mathbf{e}_m^{'}]$ and $[\mathbf{v}_1^{'}, \mathbf{v}_2^{'}, \ldots, \mathbf{v}_n^{'}]$. $\mathbf{e}_i^{'}$ and $\mathbf{v}_j^{'}$ are contextualized representations of $\mathbf{e}_i$ and $\mathbf{v}_j$, respectively. Suppose the i-th token is masked (replaced by "[MASK]"). We then predict the i-th token via $\mathbf{l} = \mathbf{e}_i^{'} \cdot \mathbf{W}, \mathbf{l} \in \mathbb{R}^D$. $\mathbf{l}$ contains logits, which are then normalized into probabilities $\mathbf{p}_{\text{mtp}}$ using softmax function ($\mathbf{p}_{\text{mtp}}$ is shown on the top right part of Figure 3). During training, we adopt a negative log-likelihood (NLL) loss, which allows our model to maximize the probability of the ground-truth masked token, *e.g.*, the probability of "mascot" shown as the first element of $\mathbf{p}$ in Figure 3.

While MTP enables our model to learn the alignment implicitly with an "intermediary", we propose to learn the alignment explicitly without the "intermediary". To do this, we let our model predict the textual class labels of visual instances, which belong to a *small closed* set of classes. We refer to this process as visual instance label prediction (ILP). As shown in Figure 3, ILP is done similar to MTP. We use NLL loss as the loss function for ILP so that our model can predict the correct textual label for the visual instances, *e.g.*, "woman" for the second visual instance in Figure 3. Since we only predict labels of visual instances of a closed set of classes, such a loss is only applied to labelled visual instances, for example, it is only applied to the 2nd one of the class "woman" in Figure 3 as other visual instances are not labelled. If an image does not have any labelled visual instance, we do not apply such a loss at all. While ILP is applied to visual instances of a closed set of classes, the representations of these visual instances □ are aligned directly with columns of $\mathbf{W}$ without the "intermediary", *i.e.*, tokens' representations ⌒. Hence, the representations of these visual instances can serve as anchors that facilitate learning of representations of other "open-vocabulary" visual instances via MTP. In this sense, MTP and ILP complement to each other. We train our model by minimizing the sum of the loss of MTP and that of ILP.

Preprocessing Scheme: As mentioned when we introduce our inference scheme, the similarity between a visual instance and a textual query is indeed the average of the similarities between the visual instance's representation and columns in the base-token embedding matrix $\mathbf{W}$ which represents tokens in the textual query. Therefore, we can pre-compute and store the similarities between all visual instances in the image database to be searched and all columns in $\mathbf{W}$ (as shown in Figure 2). Such a process speeds up the search process at test time, because there is no need to compute the similarities at test time (computing similarities between d dimensional vector representations of visual instances and tokens is relatively time-consuming). In practice, indexing methods, *e.g.*, KD-tree [4], can be used to further accelerate the search process. Fast nearest neighbor methods [5, 19, 25, 43, 44, 42, 16] can also be used if for some reason there is need to compute similarities at test time. However, that is not the focus of this paper.

4. Datasets

We create two datasets, *i.e.*, OVIS40 and OVIS1600, to benchmark OVIS methods. Both datasets contain ∼117K images, that differ considerably in contents, resolutions, and so on. In order to better simulate a real image database, the two datasets contain not only natural color images, but also man-made images, *e.g.*, cartoons, and grayscale images.

Model	OVIS40-small				OVIS40-medium				OVIS40-large			
	mAP_{30}	mAP_{50}	mAP_{70}	mAP_{all}	mAP_{30}	mAP_{50}	mAP_{70}	mAP_{all}	mAP_{30}	mAP_{50}	mAP_{70}	mAP_{all}
Det+DeViSE [13]	8.6	5.1	2.4	5.4	8.0	4.2	1.9	4.7	5.0	2.8	1.1	3.0
ViSA	50.8	35.0	18.5	34.8	46.9	30.6	14.4	30.6	33.6	21.9	10.3	21.9

Table 2: Comparison of ViSA and a combination of a detector and DeViSE, *i.e.*, Det+DeViSE, on three subsets of OVIS40.

There is no overlap between images in the two datasets and those in the training corpus.

OVIS40: OVIS40 is composed of visual instances of 40 categories of objects whose names are *uncommon* nouns, *e.g.*, "`afro`", "`fresco`", "`pagoda`" and are used as textual queries. In total, human labelers annotate 5,959 visual instances in 3,535 images for the 40 queries. On average, 149.0 visual instances are annotated for each query. None of the visual instances' name appears in the set of textual visual instance labels (seen labels) used for ILP during training. 88% of the visual instances' names are not synonyms or hypernyms (super-classes) of any seen labels; 38% of the names are hyponyms (sub-classes) of a seen label, 50% of the names have no relation to any seen labels.

OVIS40 has three different subsets, *i.e.*, OVIS40-small, OVIS40-medium, OVIS40-large. They differ in the numbers of distractors, *i.e.*, images that do not contain any of the 40 categories of visual instances to be searched. The three subsets contain ∼10K, ∼20K and ∼114K distractors, respectively. The varied number of distractors ensures that the three subsets have different degree of difficulty. OVIS40-large is particularly challenging as the number of distractors in it is 4×, 10× and 32× more than those in OVIS-medium, OVIS-small and the number of images with annotated visual instances.

OVIS1600: OVIS1600 contains 1,600 different categories of visual instances, including visual instances of objects, motions and visual instances with certain attributes, which are to be searched using queries composed of nouns, verbs (*e.g.*, "`running`", "`standing`") and adjectives (*e.g.*, "`equestrian`", "`misty`"). A total of 4,832 visual instances from 3,266 images are annotated. None of the queries in OVIS1600 appears in the set of textual visual instance labels (seen labels) used for ILP during training. 86% of the 1,600 queries from OVIS1600 dataset are neither synonyms nor hypernyms (super-class) of any seen labels; 27% of the queries are hyponyms (sub-class) of a seen label; 59% have no relation to any seen labels. More importantly, 192 queries (12%) are adjectives, *e.g.*, "`equestrian`", "`misty`" and 92 queries (6%) are verbs, while all seen labels are nouns.

There are a total of ∼117K images in OVIS1600, including ∼114K distractors (more than 95% of images are distractors). The large number of distractors not only simulate real application scenarios, but also make it quite challenging to find the visual instances to be searched.

We refer our reader to the supplementary materials for more statistics about the two datasets.

5. Experiments

5.1. Setup

Training Corpus. We use three image captioning datasets, *i.e.*, Conceptual Captions (CC) [32], SBU Captions [30] COCO Captions [26] to train our model (for MTP). CC is composed of 3.3M image-caption pairs collected by a web crawler. SBU Captions and COCO Captions contain 870K and 580K image-caption pairs, respectively. We also use 98K images with a set of 1,600 categories of visual instance label annotations from VisualGenome [21] (for ILP).

Implementation Details. Our visual-semantic encoder is implemented as a 12-layer Transformer encoder, with a hidden size of 768. Its parameters are initialized with those of BERT-Base [11]. The dictionary D of our model contains 31,069 tokens. Hence, the base-token embedding matrix $\mathbf{W}$ is of size $768 \times 31,069$. We train our ViSA model for 50 epochs with a batch size of 512 using AdamW optimizer [28]. The learning rate is set to 0.00001.

We adopt a Faster R-CNN, which is trained on VisualGenome using the *same* visual instance label annotations we use for ILP (relationship between visual instances / queries in OVIS40 and OVIS1600 and the visual instance label annotations are discussed in Section 4), to provide the positions of visual instances in images, and extract visual instance features with its ResNet101 [15] backbone. While the Faster R-CNN is trained to detect 1,600 categories objects, it performs surprisingly well at providing the positions of visual instances, even if the visual instances have no relation to any of the 1,600 categories according to WordNet hierarchy [3] (as shown in Figure 4). Note that traditional methods, *e.g.*, EdgeBox [49] or Selective Search [36] can be used to *directly* replace the Faster R-CNN to provide the positions of visual instances.

Evaluation Metrics. We evaluate the performance of OVIS method using mean average precision@k (mAP@k), which considers k top-ranked visual instances. We also adopt top-k precision (prec@k) as an auxiliary metric to show the percentage of true positives in the returned visual instances. We compute mAP and precision at three IoU thresholds: 30%, 50% and 70% and denote the results as $mAP@k_{30/50/70}$ and

	Subset	mAP				Precision			
		mAP_{30}	mAP_{50}	mAP_{70}	mAP_{all}	$prec_{30}$	$prec_{50}$	$prec_{70}$	$prec_{all}$
ILP	OVIS40-medium	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MTP		34.8	22.9	14.4	24.0	28.1	18.4	11.4	19.3
MTP & ILP		46.9	30.6	14.4	30.6	47.0	31.5	16.6	31.6
ILP	OVIS40-large	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MTP		28.5	17.8	11.4	19.2	17.7	11.0	6.7	11.8
MTP & ILP		33.6	21.9	10.3	21.9	32.4	21.4	10.3	21.4

Table 3: Comparison of models trained using different training scheme. ILP: using visual instance label prediction only; MTP: using masked token prediction only; MTP & ILP: using both MTP and ILP (our proposed training scheme).

	OVIS40-small				OVIS40-medium				OVIS40-large			
ψ	mAP_{30}	mAP_{50}	mAP_{70}	mAP_{all}	mAP_{30}	mAP_{50}	mAP_{70}	mAP_{all}	mAP_{30}	mAP_{50}	mAP_{70}	mAP_{all}
cosine	48.1	33.7	16.5	32.8	44.0	30.3	16.0	30.0	29.3	19.8	9.2	19.5
DP	50.8	35.0	18.5	34.8	46.9	30.6	14.4	30.6	33.6	21.9	10.3	21.9
NDP	49.1	32.8	17.7	33.2	46.9	30.2	14.6	30.5	31.7	19.1	8.3	19.7

Table 4: Comparison of the three choices for the similarity measure ψ on OVIS40. cosine: cosine similarity; DP: dot product similarity; NDP: normalized dot product similarity.

prec@$k_{30/50/70}$[4].

5.2. Experiments on OVIS40

We adopt mAP@50 as the evaluation metric for all the experiments on OVIS40. The best performance is shown in orange red in all the tables.

Comparison with DeViSE: We compare the performances of our model, *i.e.*, ViSA, and DeViSE[13] on all three subsets of OVIS40. As DeViSE is not able to perform OVIS, we modify it by combining it with a Faster R-CNN, which is the one used by our model, and adding all the queries in OVIS40 to DeViSE's dictionary so that it can learn their embeddings (this cannot be done in practice as queries are not known in advance). The modified model is trained using the same data as our model and is denoted as Det+DeViSE.

Table 2 shows the performance of ViSA and Det+DeViSE. We see that ViSA outperforms Det+DeViSE across all metrics. On OVIS-small, ViSA achieves 34.8% on mAP_{all}, which is 5.4× more than that of Det+DeViSE. mAP_{all} of ViSA is 5.5× more than that of Det+DeViSE on OVIS-medium and is 6.3× more on OVIS-large. ViSA maintains it superiority over DeViSE as the number of distractors in the database rapidly increases. We also see that mAP_{all} of ViSA decreases from 34.8% to 30.4% and from 30.4% to 21.9%, as the number of distractors increases by 10K (2×) and by 94K (4.7×). Despite that the number of distractors grows by *9.4 times*, mAP_{all} of ViSA only decreases by 12.9%, demonstrating ViSA's ability to handle tens of thousands of distractors and its potential to handle even larger number of distractors.

Comparison of Different Training Schemes: To analyze our proposed training scheme, *i.e.*, visual-semantic aligned (ViSA) representation learning, we conduct ablation studies by training our model using different components of ViSA. Table 3 shows the performance of our model trained using different training schemes.

We can see from the 1st row and the 4th row that training with visual instance label prediction (ILP) results in a model that is not able to perform OVIS. The reason is that ILP only trains the model to predict textual labels of visual instances of a *closed* set of categories. Thus, the trained model can not be used to search for other visual instances. If we train our model with masked token prediction (MTP) only (the 2nd row and the 5th row), the learned model achieves mAP_{all} of 24.0% and 19.2% on OVIS-medium and OVIS-large, respectively. This shows that our model implicitly learns to align representations of visual instances and textual search queries in a common semantic space with the help of MTP. The performance of our model becomes even better, if it is train with both MTP and ILP (our proposed training scheme). The increases in mAP_{all} and $prec_{all}$ are 10.6% and 12.3% on OVIS40-medium and 2.7% and 9.6% on OVIS40-large. This MTP and ILP are *complementary* to each other and are *essential* for aligning the representations of visual instance and textual queries.

Comparison of Similarity Measures ψ: Table 4 compares three different instantiations of the similarity measure ϕ, *i.e.*, cosine similarity, dot product similarity (DP) and nor-

[4] "@k" may be abbreviated if there is no confusion.

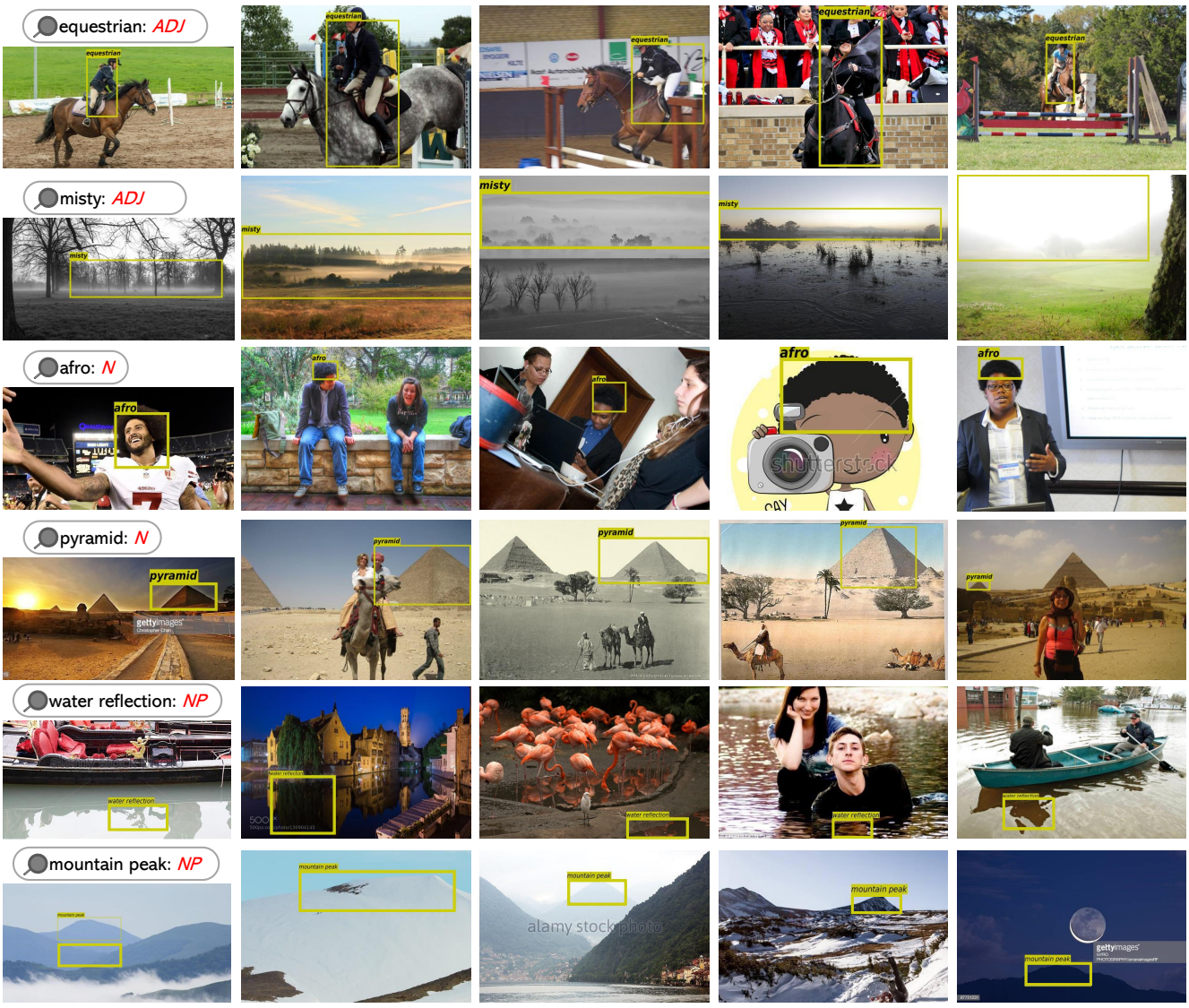

Figure 4: Visualization of top ranked visual instances returned by ViSA for six textual queries. Five queries have no relation to any visual instance labels used during training. "afro" is a hyponym of label "hair". *ADJ, N, NP* stand for adjective, noun and noun phrase, respectively.

malized dot product similarity (NDP). Interestingly, they perform similarly. mAP_{all} of cosine, DP and NDP differ by less than 2.0%, 0.6% and 2.4% on OVIS40-small, OVIS40-medium and OVIS40-large, respectively. The small gaps show that the performance of our model is not sensitive to the choice of the similarity measure ψ.

5.3. Experiments on OVIS1600

We adopt mAP@6 as the evaluation metric. Table 5 shows a comparison of ViSA and DeViSE on OVIS1600 dataset. To enable DeViSE to perform OVIS, we make the same modifications to DeViSE as introduced in Section 5.2. Comparing to Det+DeViSE, ViSA improves mAP by more than 10% across all IoU thresholds on OVIS1600. Specifically, ViSA achieves 12.4% on mAP_{70}, which is computed at a rather high IoU threshold of 0.7, and also achieves 14.9% on mAP_{all}. ViSA demonstrates its ability in searching for more than 1,600 queries in an image database composed of more than 117K images.

Model	mAP_{30}	mAP_{50}	mAP_{70}	mAP_{all}
Det+DeViSE [13]	2.6	2.0	1.9	2.2
ViSA	17.6	14.6	12.4	14.9

Table 5: Comparison of ViSA and a combination of a detector and DeViSE, *i.e.*, Det+DeViSE, on OVIS1600.

Figure 5: Visualization of the visual instances returned by ViSA given a query "bass". ViSA returns visual instances that are relevant to *both* of the two dramatically different meanings of "bass".
bass: (1) a name shared by many species of fish; (2) a member of the guitar family

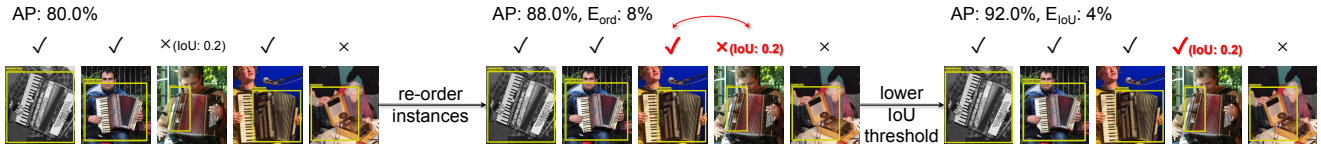

Figure 6: An illustration of the proposed error analysis pipeline. Given a search query "accordion", the original AP@5 is 80%. We first eliminate the order error by re-ordering the visual instances, such that TPs are ranked at higher orders than FPs. The change of AP before and after re-ordering is defined as the order error E_{ord} (8%). We then eliminate the IoU error by lowering the IoU threshold to 0.01. The increase in AP is defined as the IoU error E_{IoU} (4%). The gap between the AP after lowering the IoU threshold and 100% is defined as the background error E_{bg} (8%).

5.4. Qualitative Results

Figure 4 shows the top-ranked visual instances returned by ViSA for six challenging queries from three different part of speeches, *i.e.* adjective ("equestrian", "misty"), noun ("afro", "pyramid") and noun phrase ("water reflection", "mountain peak"). We see that ViSA not only returns the images that contain the visual instances, but also accurately localize the visual instances.

Figure 5 visualizes the top-ranked visual instances returned for a query "bass", which has two different meanings (1) a name shared by many species of fish, (2) a member of the guitar family. Interestingly, ViSA returns visual instances that are relevant to *both* of the two dramatically different meanings of "bass", indicating that ViSA is capable of capturing *multiple* aspects of the semantic meanings of a textual search query.

5.5. Error Analysis

We introduce a pipeline for analyzing errors made by OVIS methods, including but not limited to ViSA. There are three types of errors that prevent an OVIS method from achieving an mAP of 100%. (1) Order errors E_{ord} are caused by ranking false positives (FPs) at higher order than true positives (TPs). (2) IoU errors E_{IoU} are caused by low IoU between the returned visual instances and the annotated visual instances. (3) Background errors are caused by returning visual instances from distractors, *i.e.*, images that do not contain any visual instances relevant to the query.

Figure 6 shows our proposed pipeline which quantitatively analyzes the influence of the three types of errors. The left most part of Figure 6 shows five top-ranked visual instances for query "accordion". We first eliminate the order error by re-ordering the list of returned visual instances, such that TPs are ranked at higher orders than FPs. The change of AP before and after re-ordering is defined as the order error E_{ord}, which is 8% in this example. We then eliminate the IoU error by lowering the IoU threshold to 0.01. The increase of AP brought by lowering the IoU threshold is defined as the IoU error E_{IoU}, which is 4% in this example. The gap between the AP after lowering the IoU threshold and 100% is defined as the background error E_{bg}, which is 8% in this example. In the supplementary materials, we present an analysis of our model using such a pipeline.

6. Conclusion

In this work, we introduce the task of open-vocabulary visual instance search (OVIS), whose goal is to search for visual instances in a large-scale image database that are relevant to textual search queries. We propose a visual-semantic aligned representation learning (ViSA) method for OVIS. With the two complementary tasks of masked token prediction and visual instance label prediction, ViSA aligns representations of the visual instances and those of textual queries in a common semantic space, in which their similarities can be measured. We create two datasets, *i.e.*, OVIS40 and OVIS1600, to benchmark OVIS methods, including ViSA. Experiments on both datasets verify the effectiveness of ViSA in performing OVIS.

References

[1] https://www.google.com/. 1

[2] https://en.wikipedia.org/wiki/Ovis. 1

[3] https://wordnet.princeton.edu/. 5

[4] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975. 4

[5] Stefan Berchtold, Bernhard Ertl, Daniel A Keim, H-P Kriegel, and Thomas Seidl. Fast nearest neighbor search in high-dimensional space. In *Proceedings 14th International Conference on Data Engineering*, pages 209–218. IEEE, 1998. 4

[6] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *CVPR*, pages 2846–2854, 2016. 2

[7] Jize Cao, Zhe Gan, Yu Cheng, Licheng Yu, Yen-Chun Chen, and Jingjing Liu. Behind the scene: Revealing the secrets of pre-trained vision-and-language models. *arXiv preprint arXiv:2005.07310*, 2020. 2

[8] Yue Cao, Mingsheng Long, Bin Liu, and Jianmin Wang. Deep cauchy hashing for hamming space retrieval. In *CVPR*, pages 1229–1237, 2018. 2

[9] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *ECCV*, pages 104–120. Springer, 2020. 2

[10] Junsuk Choe, Seong Joon Oh, Seungho Lee, Sanghyuk Chun, Zeynep Akata, and Hyunjung Shim. Evaluating weakly supervised object localization methods right. In *CVPR*, pages 3133–3142, 2020. 2

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 5

[12] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612*, 2017. 2

[13] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129, 2013. 2, 5, 6, 7

[14] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5356–5364, 2019. 2

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 5

[16] Weixiang Hong, Xueyan Tang, Jingjing Meng, and Junsong Yuan. Asymmetric mapping quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 2, 4

[17] Weixiang Hong, Junsong Yuan, and Sreyasee Das Bhattacharjee. Fried binary embedding for high-dimensional visual features. In *CVPR*, pages 2749–2757, 2017. 2

[18] Zeyi Huang, Yang Zou, Vijayakumar Bhagavatula, and Dong Huang. Comprehensive attention self-distillation for weakly-supervised object detection. *Advances in neural information processing systems*, 2020. 2

[19] Yoonho Hwang, Bohyung Han, and Hee-Kap Ahn. A fast nearest neighbor search algorithm by nonlinear embedding. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3053–3060. IEEE, 2012. 4

[20] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3668–3678, 2015. 2

[21] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017. 5

[22] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+ language omni-representation pre-training. In *EMNLP*, 2020. 2

[23] Liunian Harold Li, Haoxuan You, Zhecan Wang, Alireza Zareian, Shih-Fu Chang, and Kai-Wei Chang. Weakly-supervised visualbert: Pre-training without parallel images and captions. *arXiv preprint arXiv:2010.12831*, 2020. 2

[24] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*, pages 121–137. Springer, 2020. 2

[25] Zhujin Li, Xianglong Liu, Junjie Wu, and Hao Su. Adaptive binary quantization for fast nearest neighbor search. In *ECAI*, pages 64–72, 2016. 4

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 5

[27] Haomiao Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. Deep supervised hashing for fast image retrieval. In *CVPR*, pages 2064–2072, 2016. 2

[28] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *ICLR*, 2019. 5

[29] Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In *CVPR*, pages 685–694, 2015. 2

[30] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. *Advances in neural information processing systems*, 24:1143–1151, 2011. 5

[31] Zhongzheng Ren, Zhiding Yu, Xiaodong Yang, Ming-Yu Liu, Yong Jae Lee, Alexander G Schwing, and Jan Kautz. Instance-aware, context-focused, and memory-efficient weakly supervised object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10598–10607, 2020. 2

[32] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 2556–2565, 2018. 5

[33] Amanda Spink, Dietmar Wolfram, Major BJ Jansen, and Tefko Saracevic. Searching the web: The public and their queries. *Journal of the American society for information science and technology*, 52(3):226–234, 2001. 3

[34] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *ICLR*, 2020. 2

[35] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019. 2

[36] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013. 5

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 3

[38] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, pages 499–515. Springer, 2016. 2

[39] Jialian Wu, Liangchen Song, Tiancai Wang, Qian Zhang, and Junsong Yuan. Forest r-cnn: Large-vocabulary long-tailed object detection and instance segmentation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1570–1578, 2020. 2

[40] Ke Yang, Dongsheng Li, and Yong Dou. Towards precise end-to-end weakly supervised object detection network. In *ICCV*, pages 8372–8381, 2019. 2

[41] Keren Ye, Mingda Zhang, Adriana Kovashka, Wei Li, Danfeng Qin, and Jesse Berent. Cap2det: Learning to amplify weak caption supervision for object detection. In *ICCV*, pages 9686–9695, 2019. 2

[42] Tan Yu, Jingjing Meng, Chen Fang, Hailin Jin, and Junsong Yuan. Product quantization network for fast visual search. *International Journal of Computer Vision*, pages 1–19, 2020. 2, 4

[43] Tan Yu, Yuwei Wu, and Junsong Yuan. Hope: Hierarchical object prototype encoding for efficient object instance search in videos. In *CVPR*, pages 2424–2433, 2017. 2, 4

[44] Tan Yu, Junsong Yuan, Chen Fang, and Hailin Jin. Product quantization network for fast image retrieval. In *ECCV*, pages 186–201, 2018. 2, 4

[45] Zhaoyang Zeng, Bei Liu, Jianlong Fu, Hongyang Chao, and Lei Zhang. Wsod2: Learning bottom-up and top-down objectness distillation for weakly-supervised object detection. In *ICCV*, pages 8292–8300, 2019. 2

[46] Qi Zhang, Zhen Lei, Zhaoxiang Zhang, and Stan Z Li. Context-aware attention network for image-text retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3536–3545, 2020. 2

[47] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, pages 2921–2929, 2016. 2

[48] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and vqa. In *AAAI*, pages 13041–13049, 2020. 2

[49] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European conference on computer vision*, pages 391–405. Springer, 2014. 5