The Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI-22)

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

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 amount of 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 OVIS1400 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 27.8% on OVIS40 and achieves a recall@30 of 21.3% on OVIS1400 dataset under the most challenging settings.

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 (Google 2021), in a single day. Taking a textual search query, *e.g.*, a word "`ovis`" 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.

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)[1], 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 (Liu et al. 2016; Cao et al. 2018; Yu, Wu, and Yuan 2017; Yu et al. 2018, 2020; Johnson et al. 2015; Faghri et al. 2017; Zhang et al. 2020). 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 or 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 sim-

[1]We use the two terms, "visual instance" and "image patch", interchangeably in this manuscript.

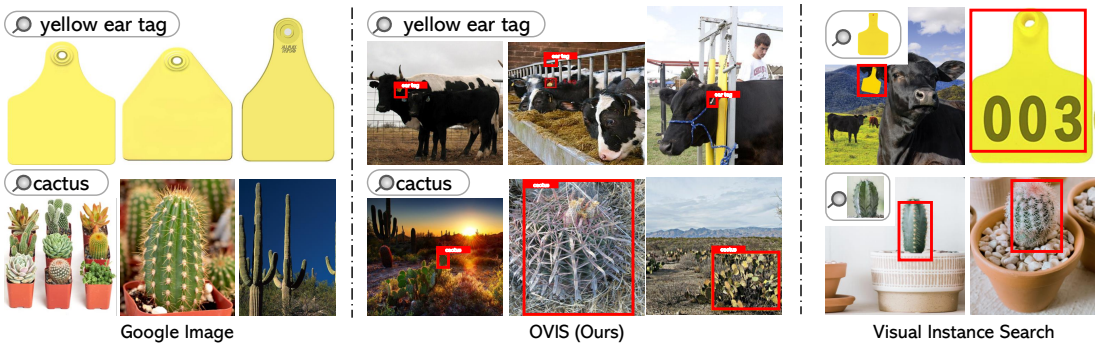

Figure 1: A comparison between Google Image, ViSA, *i.e.*, our model for OVIS and visual instance search (VIS) (Yu et al. 2017). Google Image and ViSA take as input a textual search query, *e.g.*, `cactus`, while VIS takes an image as input. Google Image returns images, while ViSA and VIS return visual instances (shown with red boxes), including small ones, in context. Google Image relies on textual metadata of images, while ViSA and VIS rely on visual contents only.

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

Table 1: Comparison of different tasks related to OVIS. IR: image retrieval; VIS: visual instance search; WSOD: weakly supervised object detection; OV-CLS: open-vocabulary image classification (Frome et al. 2013) (* indicates that (Frome et al. 2013) is not able to classify images whose labels do not have word vectors).

ilar 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 OVIS1400 datasets with ∼6K and ∼5K visual instances, which corresponds to 40 and 1,400 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.

Backgrounds

We compare OVIS with four related tasks: image retrieval (IR), weakly supervised object detection (WSOD), visual instance search (VIS) and open-vocabulary image classification (OV-CLS). Key features of these tasks are shown in Tab. 1. In addition to these tasks, OVIS is also closely related to image-text retrieval and large-vocabulary instance segmentation (Gupta, Dollar, and Girshick 2019; Wu et al. 2020).

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 (Liu et al. 2016; Cao et al. 2018; Yu, Wu, and Yuan 2017; Yu et al. 2018) 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.

Visual Instance Search (VIS): Given an image representing a visual instance as input, the goal of VIS is to retrieve images in which the instance to be searched exist and localize the instances in the retrieved images (Yu et al. 2017, 2020). Different from VIS, OVIS takes as input a textual search query.

Weakly Supervised Object Detection (WSOD): WSOD aims to train an object detector without using bounding-box annotations. Prior studies (Ren et al. 2020; Huang et al. 2020; Zeng et al. 2019; Yang, Li, and Dou 2019; Wen et al. 2016; Hong, Yuan, and Das Bhattacharjee 2017; Hong et al. 2019) use image tags as supervision, and Ye *et al.* (Ye et al. 2019) use image-caption pairs as supervision. A recent work (Zareian et al. 2021) propose to use captions as supervision to train an "open-vocabulary" object detector. In addition, Weakly Supervised Object Localization (WSOL) (Zhou et al. 2016; Bilen and Vedaldi 2016; Choe et al. 2020; Oquab et al. 2015) is a related topic to WSOD. WSOL aims to localize a single class-specific region in an image. Contrary to OVIS, most existing work on WSOD and WSOL only focuses on a fixed set of object classes.

Open-Vocabulary Image Classification (OV-CLS): Given an image as input, the goal of OV-CLS (Frome et al. 2013) 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 models (Li et al. 2020c; Chen et al. 2020; Zhou et al. 2020; Su et al. 2020; Tan and Bansal 2019; Li et al. 2020a; Cao et al. 2020; Li et al. 2020b) are successful in learning cross-modal representations for various tasks, *e.g.*, visual question answering (Antol et al. 2015), visual captioning (Liu, Ren, and Yuan 2020), using captions as supervision. While our model, *i.e.*, ViSA, is trained using a large corpus of image-caption pairs similar to existing VLP models, our model is directly applied to OVIS after being trained (VLP models have to be *finetuned* for *downstream* tasks). In

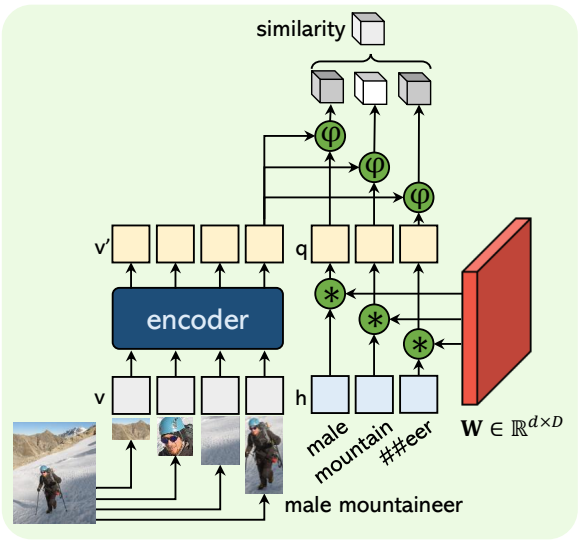

Figure 2: The way our model computes the similarity between a textual query "male mountaineer" and the 4-th visual instance in an image at test time. Our model consists of a visual-semantic encoder and a base-token embedding matrix $\mathbf{W}$. ⊛: matrix multiplication operation; ⊙: similarity measure,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

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.

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[2] in the set of all *147K* words in current use, and an *arbitrary* visual instance.

As shown in Fig. 2, our model consists of a visual-semantic encoder, *i.e.*, a Transformer encoder (Vaswani et al. 2017), 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.

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 147*K* 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 the experiment section. 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 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 or 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 Fig. 3) and then predict the masked token from the other tokens in the caption

[2]We focus on short queries as more than 80% of web search queries have less than 4 words (Spink et al. 2001).

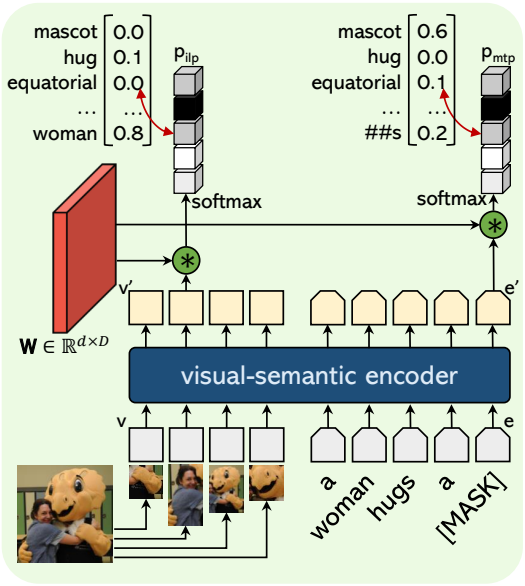

Figure 3: The way our model is trained. Each training sample is an image-caption pair, represented as visual instance features □ and token embeddings ⌓. We train our model using two tasks: masked token prediction (MTP), that aims to predict the masked token, and visual instance label prediction (ILP)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

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 Fig. 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 $\mathrm{softmax}$ function ($\mathbf{p}_{\text{mtp}}$ is shown on the top right part of Fig. 3). During training, we adopt a negative log-likelilood (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 Fig. 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 Fig. 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 Fig. 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 Fig. 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 precompute and store the similarities between all visual instances in the image database to be searched and all columns in $\mathbf{W}$ (as shown in Fig. 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 (Bentley 1975), can be used to further accelerate the search process. Fast nearest neighbor methods (Berchtold et al. 1998; Hwang, Han, and Ahn 2012; Li et al. 2016; Yu, Wu, and Yuan 2017; Yu et al. 2018, 2020; Hong et al. 2019) 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.

Datasets

We create two datasets, *i.e.*, OVIS40 and OVIS1400, to benchmark OVIS methods. Both datasets contain ∼101K 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. 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 3,378 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

	OVIS40-small				OVIS40-medium				OVIS40-large			
Model	mAP_{30}	mAP_{50}	mAP_{70}	mAP_{all}	mAP_{30}	mAP_{50}	mAP_{70}	mAP_{all}	mAP_{30}	mAP_{50}	mAP_{70}	mAP_{all}
Oscar	45.2	34.3	24.3	34.6	42.1	31.9	22.5	32.2	25.7	19.2	12.9	19.2
ViSA	**56.5**	**44.5**	**31.8**	**44.0**	**52.1**	**40.9**	**28.3**	**40.5**	**36.2**	**28.0**	**19.3**	**27.8**

Table 2: Comparison of ViSA and Oscar (Li et al. 2020c) on three subsets of OVIS40.

40 categories of visual instances to be searched. The three subsets contain ∼12K, ∼24K and ∼96K 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 $4\times$, $8\times$ and $30\times$ more than those in OVIS-medium, OVIS-small and the number of images with annotated visual instances.

OVIS1400: OVIS1400 contains 1,400 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 OVIS1400 appears in the set of textual visual instance labels (seen labels) used for ILP during training. $\sim$85% of the 1,400 queries from OVIS1400 dataset are neither synonyms nor hypernyms (super-class) of any seen labels; $\sim$27% of the queries are hyponyms (sub-class) of a seen label; 58% have no relation to any seen labels. More importantly, ∼170 queries ($\sim$12%) are adjectives, *e.g.*, "`equestrian`", "`misty`" and ∼80 queries ($\sim$6%) are verbs, while all seen labels are nouns.

Experiments

Setup

Training Corpus. We use three image captioning datasets, *i.e.*, Conceptual Captions (CC) (Sharma et al. 2018), SBU Captions (Ordonez, Kulkarni, and Berg 2011) COCO Captions (Lin et al. 2014) 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 (Krishna et al. 2017) to train our model (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 (Devlin et al. 2018). 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 ∼30 epochs with a batch size of 512 using AdamW optimizer (Loshchilov and Hutter 2019). 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 or queries in OVIS40 and OVIS1400 and the visual instance label annotations are discussed in the dataset section), to provide the positions of visual instances in images, and extract visual instance features with its ResNet101 (He et al. 2016) 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 (as shown in Fig. 4). Note that traditional methods, *e.g.*, EdgeBox (Zitnick and Dollár 2014) or Selective Search (Uijlings et al. 2013) 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 methods 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. Top-k recall (recall@k) is adopted as well. We compute mAP and precision at three IoU thresholds: 30%, 50% and 70% and denote the results as $\text{mAP@k}_{30/50/70}$ and $\text{prec@k}_{30/50/70}$[3].

Experiments on OVIS40

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

Comparison with Oscar: We compare the performances of our model, *i.e.*, ViSA and Oscar (Li et al. 2020c) on all three subsets of OVIS40. Tab. 2 shows the performance of the two methods.

We see that ViSA outperforms Oscar across all eight metrics. On OVIS40-small, mAP_{all} of Oscar is 34.6%, while that of ViSA is 44.0% (9.4% more than the mAP_{all} of Oscar). On OVIS40-medium, the mAP_{all} of ViSA is 40.5%, which is 8.7% higher that of Oscar. ViSA maintains its superiority over Oscar on OVIS40-large (27.8% mAP_{all}, 8.6% more than that of Oscar). We also see that the mAP_{all} of ViSA drops by 3.5% and 12.7% as the number of distractors increases drastically from ∼12K to ∼24K and then to ∼96K. This shows that there is a $\sim$3% performance degradation of ViSA as the number of distractors *doubles*, thus 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

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

	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		46.3	34.3	23.6	34.7	43.4	32.4	22.7	32.9
MTP & ILP		**52.1**	**40.9**	**28.4**	**40.5**	**48.9**	**38.8**	**27.4**	**38.4**
ILP	OVIS40-large	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MTP		27.8	20.3	13.7	20.6	25.8	19.1	13.0	19.3
MTP & ILP		**36.2**	**28.0**	**19.3**	**27.8**	**32.0**	**25.5**	**18.1**	**25.2**

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	56.4	44.0	30.2	43.5	**52.1**	40.5	27.7	40.1	36.1	27.8	18.8	27.6
DP	55.0	42.0	29.1	42.0	50.8	38.7	26.7	38.7	35.8	27.2	18.6	27.2
NDP	**56.5**	**44.5**	**31.8**	**44.0**	**52.1**	**40.9**	**28.3**	**40.5**	**36.2**	**28.0**	**19.3**	**27.8**

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.

by training our model using different components of ViSA. Tab. 3 shows the performance of our model trained using different training schemes.

We can see from the 1^{st} row and the 4^{th} row that training with visual instance label prediction (ILP) loss results in a model that is not able to perform OVIS. The reason is that the model trained using ILP loss only learns to predict textual labels for visual instances of a *closed* set of categories. Therefore, the trained model can not be used to search for other visual instances. If we train our model using masked token prediction (MTP) loss only (the 2^{nd} row and the 5^{th} row), the learned model achieves mAP_{all} of 34.7% and 20.6% on OVIS40-medium and OVIS40-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 the MTP loss. The performance of our model becomes even better, if it is trained using both the MTP loss and the ILP loss, *i.e.*, our proposed training scheme. The increases in mAP_{all} and prec_{all} are 5.8% and 5.5% on OVIS40-medium and 6.8% and 5.9% on OVIS40-large. This shows that the two loss are *complementary* to each other and are *essential* for aligning the representations of visual instance and textual queries.

Comparison of Different Similarity Measures ψ: Tab. 4 compares three different instantiations of the similarity measure ϕ, *i.e.*, cosine similarity, dot product similarity (DP) and normalized dot product similarity (NDP). NDP utilizes softmax function to normalize dot product similarity across all the 31,069 tokens in the dictionary D. Interestingly, they perform similarly. mAP_{all} of cosine, DP and NDP differ by less than 2.0%, 1.8% and 0.6% on OVIS40-small, OVIS40-medium and OVIS40-large, respectively. The small gaps show that the performance of ViSA is not sensitive to the choice of the similarity measure ψ.

Model	mAP_{30}	mAP_{50}	mAP_{70}	mAP_{all}
Oscar	11.2	9.6	8.5	9.8
ViSA	**23.5**	**19.9**	**16.9**	**20.1**

Table 5: Comparison of ViSA and Oscar (Li et al. 2020c) on OVIS1400 dataset. We adopt mAP as the evaluation metric.

Model	recall_{30}	recall_{50}	recall_{70}	$\text{recall}_{\text{all}}$
Oscar	9.9	8.4	6.9	8.4
ViSA	**24.8**	**21.4**	**17.8**	**21.3**

Table 6: Comparison of ViSA and Oscar (Li et al. 2020c) on OVIS1400 dataset. We adopt recall as the evaluation metric.

Experiments on OVIS1400

As OVIS1400 contains a total number of 1,400 queries, the cost for annotating all the visual instances corresponding to the 1,400 queries in a large number of images is rather large. We first evaluate the performance of ViSA and Oscar on a set of 3,266 images using mAP@6 as the metric (all the visual instances in this set of images that correspond to the 1,400 queries are annotated). We then add ∼96K distractors to this set and compare the performance of ViSA and Oscar using recall@30 as the metric (recall is meaningful, even if not all the visual instances are annotated).

Tab. 5 shows a comparison of ViSA and Oscar on OVIS1400 dataset (using mAP as the evaluation metric). Specifically, the mAP_{70}, which is computed at a rather high IoU threshold of 0.7, of ViSA is 16.9%. This shows the ability of ViSA to search for a wide range of categories of visual instances. Comparing to Oscar, ViSA improves mAP across all IoU thresholds by more than 8.4%, thus validating the advantages of ViSA. Tab. 6 compares the performance of ViSA and Oscar using recall as the evaluation metric. We see that ViSA maintains its advantages over Oscar. ViSA

Figure 4: Visualization of top ranked visual instances returned by ViSA for three textual queries. *ADJ, N, NP* stand for adjective, noun and noun phrase, respectively.

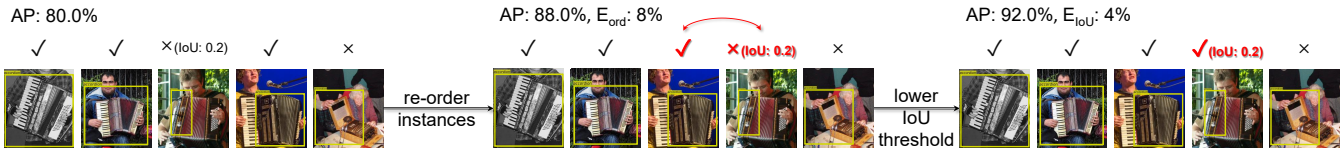

Figure 5: An illustration of the proposed error analysis pipeline. Given a search query "`accordion`", the original AP@5 is 80%. We first eliminate the order error by re-ordering the visual instances, such that TPs are ranked at higher orders than FPs. The change of AP before and after re-ordering is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

outperforms Oscar by 12.9% in terms of $\mathrm{recall_{all}}$. Tab. 6 also demonstrates ViSA's ability to handle more than 96K distractors.

Qualitative Results

Fig. 4 shows the top-ranked visual instances returned by ViSA for three challenging queries, *i.e.* "`equestrian`" (adjective), "`afro`" (noun) and "`water reflection`" (noun phrase). We see that ViSA not only returns the images that contain the visual instances, but also accurately localize the visual instances.

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 $\mathrm{E_{ord}}$ are caused by ranking false positives (FPs) at higher order than true positives (TPs). (2) IoU errors $\mathrm{E_{IoU}}$ 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.

Fig. 5 shows our proposed pipeline which quantitatively analyzes the influence of the three types of errors. The left most part of Fig. 5 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 $\mathrm{E_{ord}}$, 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 $\mathrm{E_{IoU}}$, which is 4% in this example. The gap between the AP after lowering the IoU threshold and 100% is defined as the background error $\mathrm{E_{bg}}$, which is 8% in this example.

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 OVIS1400, to benchmark OVIS methods, including ViSA. Experiments on both datasets verify the effectiveness of ViSA in performing OVIS.

References

Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Zitnick, C. L.; and Parikh, D. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, 2425–2433.

Bentley, J. L. 1975. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9): 509–517.

Berchtold, S.; Ertl, B.; Keim, D. A.; Kriegel, H.-P.; and Seidl, T. 1998. Fast nearest neighbor search in high-dimensional space. In *Proceedings 14th International Conference on Data Engineering*, 209–218. IEEE.

Bilen, H.; and Vedaldi, A. 2016. Weakly supervised deep detection networks. In *CVPR*, 2846–2854.

Cao, J.; Gan, Z.; Cheng, Y.; Yu, L.; Chen, Y.-C.; and Liu, J. 2020. Behind the Scene: Revealing the Secrets of Pre-trained Vision-and-Language Models. *arXiv preprint arXiv:2005.07310*.

Cao, Y.; Long, M.; Liu, B.; and Wang, J. 2018. Deep cauchy hashing for hamming space retrieval. In *CVPR*, 1229–1237.

Chen, Y.-C.; Li, L.; Yu, L.; El Kholy, A.; Ahmed, F.; Gan, Z.; Cheng, Y.; and Liu, J. 2020. UNITER: UNiversal Image-TExt Representation Learning. In *ECCV*, 104–120. Springer.

Choe, J.; Oh, S. J.; Lee, S.; Chun, S.; Akata, Z.; and Shim, H. 2020. Evaluating weakly supervised object localization methods right. In *CVPR*, 3133–3142.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Faghri, F.; Fleet, D. J.; Kiros, J. R.; and Fidler, S. 2017. Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612*.

Frome, A.; Corrado, G. S.; Shlens, J.; Bengio, S.; Dean, J.; Ranzato, M.; and Mikolov, T. 2013. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, 2121–2129.

Google. 2021.

Gupta, A.; Dollar, P.; and Girshick, R. 2019. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5356–5364.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Hong, W.; Tang, X.; Meng, J.; and Yuan, J. 2019. Asymmetric Mapping Quantization for Nearest Neighbor Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Hong, W.; Yuan, J.; and Das Bhattacharjee, S. 2017. Fried binary embedding for high-dimensional visual features. In *CVPR*, 2749–2757.

Huang, Z.; Zou, Y.; Bhagavatula, V.; and Huang, D. 2020. Comprehensive Attention Self-Distillation for Weakly-Supervised Object Detection. *Advances in neural information processing systems*.

Hwang, Y.; Han, B.; and Ahn, H.-K. 2012. A fast nearest neighbor search algorithm by nonlinear embedding. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 3053–3060. IEEE.

Johnson, J.; Krishna, R.; Stark, M.; Li, L.-J.; Shamma, D.; Bernstein, M.; and Fei-Fei, L. 2015. Image retrieval using scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3668–3678.

Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.-J.; Shamma, D. A.; et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1): 32–73.

Li, L.; Chen, Y.-C.; Cheng, Y.; Gan, Z.; Yu, L.; and Liu, J. 2020a. HERO: Hierarchical Encoder for Video+ Language Omni-representation Pre-training. In *EMNLP*.

Li, L. H.; You, H.; Wang, Z.; Zareian, A.; Chang, S.-F.; and Chang, K.-W. 2020b. Weakly-supervised VisualBERT: Pre-training without Parallel Images and Captions. *arXiv preprint arXiv:2010.12831*.

Li, X.; Yin, X.; Li, C.; Zhang, P.; Hu, X.; Zhang, L.; Wang, L.; Hu, H.; Dong, L.; Wei, F.; et al. 2020c. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*, 121–137. Springer.

Li, Z.; Liu, X.; Wu, J.; and Su, H. 2016. Adaptive Binary Quantization for Fast Nearest Neighbor Search. In *ECAI*, 64–72.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*, 740–755. Springer.

Liu, H.; Wang, R.; Shan, S.; and Chen, X. 2016. Deep supervised hashing for fast image retrieval. In *CVPR*, 2064–2072.

Liu, S.; Ren, Z.; and Yuan, J. 2020. Sibnet: Sibling convolutional encoder for video captioning. *IEEE transactions on pattern analysis and machine intelligence*, 43(9): 3259–3272.

Loshchilov, I.; and Hutter, F. 2019. Decoupled weight decay regularization. *ICLR*.

Oquab, M.; Bottou, L.; Laptev, I.; and Sivic, J. 2015. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In *CVPR*, 685–694.

Ordonez, V.; Kulkarni, G.; and Berg, T. 2011. Im2text: Describing images using 1 million captioned photographs. *Advances in neural information processing systems*, 24: 1143–1151.

Ren, Z.; Yu, Z.; Yang, X.; Liu, M.-Y.; Lee, Y. J.; Schwing, A. G.; and Kautz, J. 2020. Instance-Aware, Context-Focused, and Memory-Efficient Weakly Supervised Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10598–10607.

Sharma, P.; Ding, N.; Goodman, S.; and Soricut, R. 2018. 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*, 2556–2565.

Spink, A.; Wolfram, D.; Jansen, M. B.; and Saracevic, T. 2001. Searching the web: The public and their queries. *Journal of the American society for information science and technology*, 52(3): 226–234.

Su, W.; Zhu, X.; Cao, Y.; Li, B.; Lu, L.; Wei, F.; and Dai, J. 2020. Vl-bert: Pre-training of generic visual-linguistic representations. *ICLR*.

Tan, H.; and Bansal, M. 2019. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*.

Uijlings, J. R.; Van De Sande, K. E.; Gevers, T.; and Smeulders, A. W. 2013. Selective search for object recognition. *International journal of computer vision*, 104(2): 154–171.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wen, Y.; Zhang, K.; Li, Z.; and Qiao, Y. 2016. A discriminative feature learning approach for deep face recognition. In *ECCV*, 499–515. Springer.

Wu, J.; Song, L.; Wang, T.; Zhang, Q.; and Yuan, J. 2020. Forest R-CNN: Large-vocabulary long-tailed object detection and instance segmentation. In *Proceedings of the 28th ACM International Conference on Multimedia*, 1570–1578.

Yang, K.; Li, D.; and Dou, Y. 2019. Towards precise end-to-end weakly supervised object detection network. In *ICCV*, 8372–8381.

Ye, K.; Zhang, M.; Kovashka, A.; Li, W.; Qin, D.; and Berent, J. 2019. Cap2Det: Learning to amplify weak caption supervision for object detection. In *ICCV*, 9686–9695.

Yu, T.; Meng, J.; Fang, C.; Jin, H.; and Yuan, J. 2020. Product Quantization Network for Fast Visual Search. *International Journal of Computer Vision*, 1–19.

Yu, T.; Wu, Y.; Bhattacharjee, S.; and Yuan, J. 2017. Efficient object instance search using fuzzy objects matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Yu, T.; Wu, Y.; and Yuan, J. 2017. Hope: Hierarchical object prototype encoding for efficient object instance search in videos. In *CVPR*, 2424–2433.

Yu, T.; Yuan, J.; Fang, C.; and Jin, H. 2018. Product quantization network for fast image retrieval. In *ECCV*, 186–201.

Zareian, A.; Rosa, K. D.; Hu, D. H.; and Chang, S.-F. 2021. Open-vocabulary object detection using captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 14393–14402.

Zeng, Z.; Liu, B.; Fu, J.; Chao, H.; and Zhang, L. 2019. Wsod2: Learning bottom-up and top-down objectness distillation for weakly-supervised object detection. In *ICCV*, 8292–8300.

Zhang, Q.; Lei, Z.; Zhang, Z.; and Li, S. Z. 2020. Context-aware attention network for image-text retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3536–3545.

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning deep features for discriminative localization. In *CVPR*, 2921–2929.

Zhou, L.; Palangi, H.; Zhang, L.; Hu, H.; Corso, J. J.; and Gao, J. 2020. Unified Vision-Language Pre-Training for Image Captioning and VQA. In *AAAI*, 13041–13049.

Zitnick, C. L.; and Dollár, P. 2014. Edge boxes: Locating object proposals from edges. In *European conference on computer vision*, 391–405. Springer.