J. Vis. Commun. Image R. 79 (2021) 103253

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Full length article

Instance search via instance level segmentation and feature representation ☆

Yu Zhan, Wan-Lei Zhao *

The School of Information Science and Technology, Xiamen University, Xiamen, 361005, PR China

ARTICLE INFO

Keywords:
Instance search
Instance segmentation
CNN

ABSTRACT

Instance search is an interesting task as well as a challenging issue due to the lack of effective feature representation. In this paper, an instance level feature representation built upon fully convolutional instance-aware segmentation is proposed. The feature is ROI-pooled from the segmented instance region. So that instances in various sizes and layouts are represented by deep features in uniform length. This representation is further enhanced by the use of deformable ResNeXt blocks. Superior performance is observed in terms of its distinctiveness and scalability on a challenging evaluation dataset built by ourselves. In addition, the proposed enhancement on the network structure also shows superior performance on the instance segmentation task.

1. Introduction

With the proliferation of massive multimedia contents in our daily life, it is desired that users are allowed to browse over relevant images/videos in which the specified visual instance (e.g., an object or a landmark or a person) appears. This is known as instance search [1], which arises from several application scenarios such as online product search in the shopping website, video editing, and person re-identification, etc.

Instance search is essentially different from conventional content-based image retrieval (CBIR) [2,3] in several perspectives. First of all, in instance search, the query is a visual object that is outlined (usually by a bounding box) in an image. While in CBIR, the whole image is treated as the query. Secondly, instance search requires the intended visual objects to come from the same instance (while possibly under different transformations) as the query [1]. In contrast, CBIR only requires the returned contents to be visually similar as the query image no matter whether they share the same origin. Moreover, instance search should localize the target instance in the returned images.

There are basically two stages in visual content search system, namely feature representation [4–16] and fast retrieval [17–20]. In the whole process, feature representation plays the key role to the success of the system. On one hand, features are required to be robust to various image transformations, such as scaling, rotation and occlusions, motion blur, etc. On the other hand, they should be distinctive enough so that the retrieval quality does not suffer severe degradation as the scale of the reference set grows.

In the existing solutions, instance search has been mainly addressed by conventional approaches that are originally designed for image search [1,2], such as bag-of-visual words (BoVW) [4], RoI-BoVW [10], VLAD [5] and FV [9]. All these approaches are built upon image local features such as SIFT [21], RootSIFT [22], SURF [23]. Although local features are much more distinctive than global features, they are still unsuitable for instance search task. First of all, local features are not robust to out-of-plane rotation and deformation, both of which are widely observed in the real world. Moreover, it is not rare that very few local features are extracted from transparent objects (e.g., bottles) or objects with flat surface (e.g., balls). Additionally, it is not guaranteed that the regions covered by local features are exactly from one instance. As a result, the local features used to describe a target instance are more or less contaminated by the contents from the background. For this reason, similar as global features, isolated feature representation for individual instances is not desirable.

Recently, pre-trained CNNs are gradually introduced to image retrieval tasks [13–16,24–26] due to their great success in visual object classification tasks [27]. In the existing practices, image features are typically extracted from the whole image or a series of local regions with convolution or fully connected layers. Encouraging results are observed on the landmark retrieval tasks in [14,15]. However, they are unfeasible for instance representation since it is essentially a type of global feature. The feature vector is comprised by a mixture of activations from a variety of latent instances in the image. Although recent research [28,29] attempts to localize the representation to regional level, exhaustive sliding search or feature aggregation is still inevitable. Moreover, since such region level representation is given by a coarsely restricted region, their improvement is still limited.

In this paper, an instance level feature representation is proposed, which is based on an effective instance segmentation approach, namely

☆ This paper has been recommended for acceptance by Zicheng Liu.

* Corresponding author.
E-mail addresses: yeezytaughtme@stu.xmu.edu.cn (Y. Zhan), wlzhao@xmu.edu.cn (W.-L. Zhao).

https://doi.org/10.1016/j.jvcir.2021.103253
Received 18 August 2018; Received in revised form 7 June 2021; Accepted 25 July 2021
Available online 30 July 2021

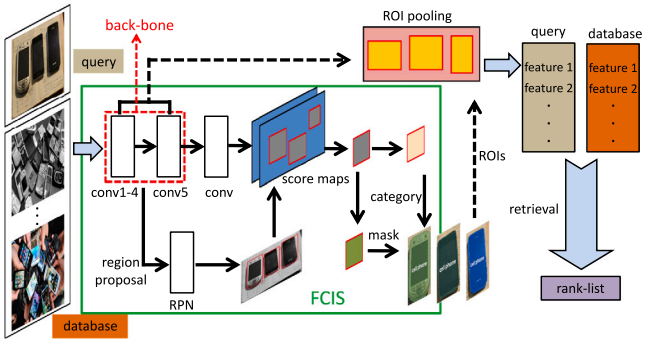


Fig. 1. Framework of instance-level feature representation from convolutional activations of FCIS [30]. Processing flow with black arrows and dashed lines denote the proposed modification and enhancement over FCIS.

fully convolutional instance-aware semantic segmentation (FCIS) [30]. Individual instances present in the image are detected and segmented on pixel level by FCIS. This is essentially different from the approach presented in [31], in which the segmentation only reaches to the semantic category level. With the instance level segmentation, feature representation of each instance is derived from the feature maps of convolution layers using ROI pooling. So that instances in different sizes and layouts are represented with the feature vectors of the same size. In order to enhance the performance, two modifications have been made on the FCIS network.

- The back-bone network of FCIS is replaced with a more powerful ResNeXt-101 [32] without increasing extra FLOPs complexity or the number of parameters;
- To enable the receptive field to be adaptive to the various shape of potential objects, the plain layer in ResNeXt-101's final stage is replaced with deformable convolution [33].

To the best of our knowledge, this is the first piece of work that visual instances are represented by features derived exactly from the instance region. Moreover, due to the lack of publicly available testing benchmark for instance search, a new dataset called *Instance-160* is constructed by harvesting test videos that are originally used for visual object tracking evaluation.

2. Framework for instance search

2.1. Instance level feature representation

Fully convolutional instance-aware semantic segmentation (FCIS) [30] is designed primarily for instance segmentation and detection. The framework of FCIS is given as a sub-figure in Fig. 1, which is inside the bounding box in green. In the network, the idea of "position-sensitive score map" is adopted to perform segmentation and detection simultaneously. These two sub-tasks share the same set of score maps by assembling operation according to the region of interest (ROI). ROIs are generated by region proposal network (RPN), which is added on top of "conv4". The score maps output "inside" and "outside" scores for the mask prediction and classification jointly. For details, readers are referred to [30].

As seen from Fig. 1, there are three outputs from FCIS for one image, namely the segmented instances (given as instance masks) and the corresponding category label, along with the bounding box of each instance. In order to extract the feature for each segmented instance, another pipeline is introduced into FCIS framework. Namely, with the generated bounding box, ROI pooling is performed on the feature maps that are generated in the convolution stages. This feature extraction pipeline is shown on the up-right of Fig. 1. Since the size of feature map is different from the input image and varies from layer to layer, bounding box of each instance is scaled accordingly to fit the size of the feature map when we perform ROI pooling. The maximum activation is extracted from the scaled ROI region as one dimension of the feature representation. This ROI pooling is applied on all feature maps in the same layer. As a consequence, the size of the output feature equals to the number of the feature maps. Instances in different sizes and layouts are represented with the same size of feature vectors. Since the segmentation is precise and clean, this feature representation is on instance level of real sense. All per-ROI computation is simple and fast with a negligible cost, compared with forward pass. In recent research on image captioning, the instance level feature is produced by Mask R-CNN [34], which shares similar spirit as ours. However, different from our framework, the features are fed to GCN-LSTM for caption generation, which is less demanding on either the accuracy of instance localization and distinctiveness of the feature representation.

Intuitively, convolution layers keep more abstract visual information as network goes deeper. It is therefore widely believed that shallower convolution layers are more suitable for low level feature representation. In our framework, the ROI pooling could be possibly applied on "conv2" to "conv5" and "conv" in Fig. 1. Namely, given a segmented instance region, the segmented region is first projected to the corresponding region on the feature maps of certain layer, *e.g.* "conv5". On every channel of feature map, max-pooling is performed on the segmented region, and therefore, is reduced to a float number. By concatenating all float numbers with respect to the order of n channels, a feature vector of n dimensions is produced. As a result, instances of various size and layouts are represented by a feature vector with uniform size. The size of feature n is the same as the number of channels in the layer. In the experiment, a comparative study is made to show the distinctiveness of the feature extracted from these layers. In addition, we also test the possibility of concatenating features ROI-pooled from different stages. The concatenation is performed with two, three and four stages of features. As will be revealed in the experiment, concatenating features from "conv3" and "conv4" leads to a good trade-off between feature distinctiveness and computation costs. Features are l_2-normalized before and after the concatenation.

2.2. Performance enhancement

In order to boost the performance of the proposed feature representation, the FCIS is modified in two aspects. Namely, the ResNet-101 [35], upon which FCIS is built, is replaced by more powerful ResNeXt-101 [32]. In addition, to enable the network to be more robust to severe shape variations, deformable convolution [33] is adopted in the last three bottle-neck blocks of ResNeXt-101.

As pointed out in [30], the performance of ResNet [35] gets saturated when its depth reaches to *152*. To further improve the accuracy of this back-bone network, ResNet-101 is replaced by ResNeXt-101 [32] which corresponds to "conv1-4" and "conv-5" in Fig. 1. Compared to ResNet, ResNeXt increases the *cardinality* of the building blocks. Fig. 2 show the difference between blocks of ResNet and ResNeXt. *Cardinality* refers to the size of same-topology transformation aggregated in the building block. The cardinality of building blocks in our case is set to *32*. This is to control the FLOPs complexity on the same level as ResNet. Similar as ResNet-101, the weights of the model are initialized from ImageNet [27] classification task. The layers (i.e., deformable convolution layer and RPN) absent from the pre-trained model are randomly initialized.

Visual instances usually undergo various irregular geometric transformations in real scenario, which causes heavy deformations in their appearances. Plain convolution modules in CNNs are inherently vulnerable to such kind of transformations. Inspired from [33], deformable convolutions are introduced to replace the plain convolution in the last three bottle-neck blocks of ResNeXt-101 to alleviate this problem (illustrated in Fig. 2). Fig. 2(d) shows the sampling structure

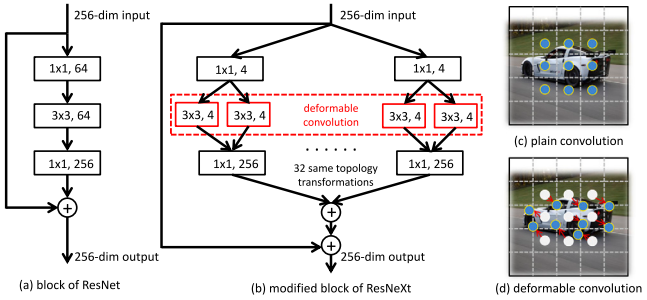


Fig. 2. Comparison between ResNet and ResNeXt blocks. In figure (b), ResNeXt's block [32] is embedded with deformable convolution [33] with cardinality of *32*. The size of filter and the number of filters are shown on each convolution layer. In the enhanced instance feature design, structure in (b) is adopted. The last *3* bottle-neck blocks of ResNeXt-101 are replaced by deformable convolution given in figure (d).

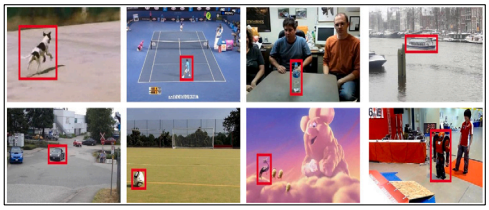
of deformable convolution in contrast to plain one (Fig. 2(c)). The deformable convolution calculates a set of offsets for the ultimate sampling locations to better adapt to the deformations of the instance. The offsets are easily learned by applying a convolutional layer over the same input feature maps. As is revealed later in the experiments, both modifications proposed in this section boost the performance of instance segmentation and instance search.

3. Evaluation dataset construction

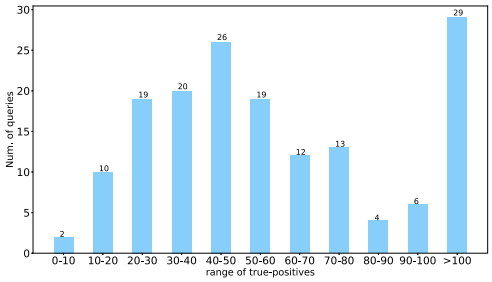
Since the initiatives of instance search task in TRECVID [1], several instance search approaches have been proposed one after another over the past few years. However, the publicly available evaluation benchmark is slow to occur. Approaches [28,29,31] aiming for instance search are only evaluated on landmark datasets, typically Oxford5k [36], Paris6k [37] and Holidays [37]. The evaluation does not reflect the real challenges, such as motion blur, partial occlusion, deformation and mutual object embedding, that instance search faces in the general cases. Dataset maintained by TRECVID [1] avoids such kind of disadvantages, whereas it is only open to TRECVID participants. In this paper, a new dataset, namely *Instance-160* is introduced. As visual object tracking and instance search are two similar tasks, *Instance-160* is built based on the video sequences used for visual object tracking evaluation. On one hand, this avoids the painstaking efforts to annotate the instances from new video sequences. On the other hand, videos that are used for visual object tracking have been widely accepted benchmarks. The variety of variations and transformations that could happen on visual instances are incorporated.

In the object tracking, the tracking algorithm is required to track the target object (selected on the first frame) in the rest of video frames. In order to verify the robustness of the tracking algorithm, the test videos are collected from different scenarios and cover a wide range of objects. Most popular evaluation benchmarks are OTB2015 [38] and ALOV++ [39]. They are collected from diverse circumstance including illuminations, transparency, specularity, confusion with similar objects, clutter, occlusion, severe deformation, motion blur and low contrast. Since instance search arises from similar application scenarios as object tracking, the same challenges are seen in instance search. Nevertheless, it is worth noting that instance search is different from object tracking. The latter assumes the visual object varies following the temporal order. For this reason, the temporal information is more or less capitalized in various object tracking algorithms. While this is not the case for instance search. Moreover, the tracking algorithm is allowed to update the feature representation from time to time as the tracking continues. In contrast, feature representation, once has been designed, is fixed all the way in instance search.

When we construct *Instance-160*, *58* and *102* sequences are selected from *100* and *300* video sequences from OTB2015 and ALOV++ respectively. The videos inside which the target instances are not covered



(a) Eight sample queries in *Instance-160*



(b) Distribution of true-positive

Fig. 3. Sample queries from *Instance-160* and the number of true-positive distribution in *Instance-160*.

by Microsoft COCO's *80* categories are omitted. For each video, the first frame in which the query instance is given by a bounding box is extracted as the query side. For the rest, one frame is extracted for every other *4* frames as the reference dataset. This results in *11,885* reference images in total. Sample queries are seen in Fig. 3(a). The distribution about the number of true-positives for all queries are shown in Fig. 3(b). As shown in the figure, more than *90%* of the queries have more than *20* true-positives for each.

4. Experiments

In this section, the proposed approach for instance search is evaluated on the dataset introduced in Section 3. Additionally, in order to verify the scalability of the presented approach, another *1* million images randomly crawled from Flickr are incorporated as distractors. The performance evaluation is studied in comparison to several representative approaches. They are BoVW [4], BoVW+HE [37], R-MAC [28], Deepvision [31] and CroW [29]. The last three are based on deep features. For each CNN-based method, the network is initialized with the default pre-trained model and configuration described in the corresponding paper. For BoVW and BoVW+HE, the same visual vocabulary sized of *65,536* are used. The binary signature in HE is set to *64* bits. The performance is measured by mAP at top-*k*, where *k* varies from *10* to *100*. This is due to the fact that more than *95%* the queries have more than *10* corresponding true-positives as shown in Fig. 3(b).

Under the same training protocol introduced in [30], the feasibility of the proposed enhancement strategies is validated on PASCAL VOC 2012 [40]. Thereby, FCIS and FCIS in-planted with the proposed enhancement strategies are trained on Microsoft COCO 2014 [41]. All the experiments are conducted on a workstation with four Nvidia Titan X GPUs and one *3.20* GHz Intel CPU setup.

4.1. Configuration test on FCIS

Theoretically speaking, feature ROI-pooled from any layer could be used to represent the detected instance. The distinctiveness of these features varies from layer to layer. In the first experiment, the distinctiveness of instance-wise representation that are extracted from different layers is studied. The feature representation with the best distinctiveness (reflected by the highest mAP) is selected as the final

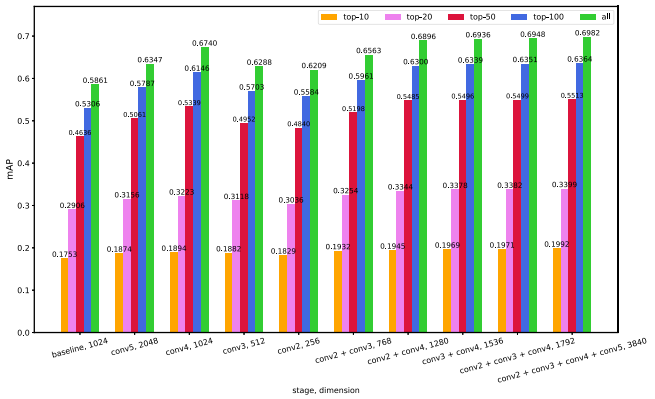


Fig. 4. Performance of deep features extracted from different stages' convolution layer, including experiments with feature concatenation.

feature representation. Additionally, we also investigate the possibility of concatenating features from different layers.

According to our observation, the category label for the segmented instance from FCIS is in high accuracy. It is therefore could be adopted for early pruning. Namely, the instance query only needs to compare with the candidate instances which share the same category label. Such pruning strategy speeds up the retrieval by two times without notable drop in mAP. In the following experiments, pruning scheme is adopted as default configuration for our approach.

In the first experiment, the distinctiveness of features from different layers of FCIS network is studied. We also investigate the performance of hybrid features that combining features from two layers. Feature derived from the "conv" (see Fig. 1) layer is given as comparison baseline.

Fig. 4 summarizes the performance with features extracted from different stages. In the figure, mAP@*10* and mAP@*20* for all the configurations are low since not all potential true-positives are considered due to the fact that *90%* queries have more than *20* true-positives (see Fig. 3(b)). As expected, features derived from intermediate layers perform better over feature from baseline ("conv"). Performance drops when features are derived from the shallow layers, such as "conv2" and "conv3". This basically indicates that it is sub-optimal to employ representations only kept with local visual patterns. As seen in the figure, all three different combinations between features from different layers lead to better results. Concatenating features from all four layers gives the best results, whereas with the highest dimensionality. As seen from the figure, concatenating features from "conv3" and "conv4" achieves similar performance but with only about a half of feature dimension. This observation is consistent even when we change the back-bone network from ResNet to ResNeXt as will be shown in the later experiment. It indicates that appearance of the instance is mostly encoded in these intermediate stages. As a result, we choose to concatenate only two stages of features in the later experiment for computation efficiency.

4.2. FCIS+XD versus FCIS

In this section, we are going to investigate the performance achieved by two enhancement strategies proposed in Section 2.2. Since FCIS is primarily designed for instance segmentation, the effectiveness of the enhanced FCIS network is studied first on instance segmentation task. In the experiment, the performance of FCIS with ResNeXt back-bone network and deformable convolution layer are studied both as separate runs and as a combination. FCIS supported with deformable convolution is denoted as FCIS+D. FCIS supported with ResNeXt-101 is denoted as FCIS+X. FCIS+XD denotes that FCIS powered by both enhancement strategies.

Table 1
Performance comparison (measured by mAP^r) of FCIS with its variants on PASCAL VOC 2012 [40].

Approach	mAP^r@0.5	mAP^r@0.7
FCIS	0.657	0.521
FCIS+D	0.667	0.528
FCIS+X	0.658	0.526
FCIS+XD	**0.675**	**0.539**

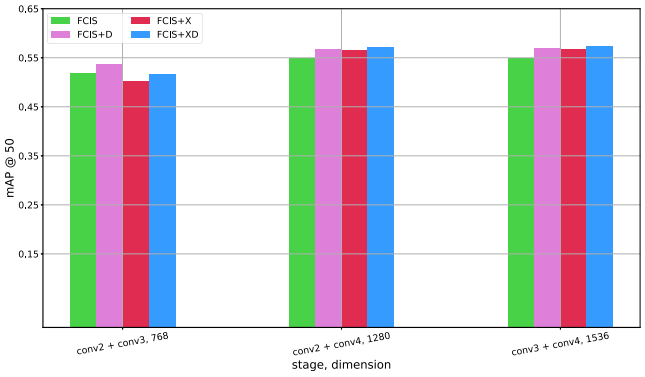


Fig. 5. Performance of FCIS, FCIS+D, FCIS+D and FCIS+XD with the hybrid features from different layers. The performance is measured by mAP at top-*50* on *Instance-160*.

The performance evaluation is conducted on PASCAL VOC 2012 [40] and Microsoft COCO 2014 test-dev [41]. $mAP^r@r$ is adopted for the evaluation. It basically calculates the mean of Average Precision (AP) measured for a method for which the corresponding recall exceeds *r*. Notice that it is essentially different from mAP that we use to evaluate the instance search performance.

The performance of instance segmentation using FCIS and its variants is summarized in Tables 1 and 2. On the two datasets PASCAL VOC 2012 and Microsoft COCO 2014, both networks individually supported by deformable convolution layers and ResNeXt bottle-neck blocks (denoted as FCIS+D and FCIS+X respectively) are able to achieve better results in comparison to original FCIS architecture. When both of these enhancement strategies are adopted (given as FCIS+XD), the best segmentation accuracy is attained. As we verified on PASCAL VOC 2012 and Microsoft COCO 2014, the segmentation accuracy of FCIS is relatively improved by *2.7%* and *5.5%* measured with mAP^r@0.5 respectively by FCIS+XD. Such results indicate that the enhancement strategies proposed in Section 2.2 are all effective in boosting the performance of instance segmentation task.

In addition, we further study the performance of the features derived from FCIS+D, FCIS+X and FCIS+XD when they are adopted for instance search task. Similar as FCIS, the other three networks are trained on Microsoft COCO 2014 [41]. Fig. 5 presents the performance of FCIS and its variants on *Instance-160*. mAPs at top-*50* are presented. Since hybrid features from different layers are always better than the ones from single layer, the results of features derived from single layer are omitted. As seen from the figure, hybrid features from "conv3 + conv4" achieve the best result. This is consistent with the observation on the results shown in Fig. 4. In the following experiments, hybrid feature from "conv3" and "conv4" is selected as the feature representation for each detected instance.

Table 3 further shows the performance of FCIS and its enhancements on *Instance-160*. For all different networks, the features are extracted from "conv3" and "conv4". Due to the high accuracy on instance level segmentation, superior performance is observed with FCIS+XD across all the rankings. It outperforms FCIS by a constant *2–3%* margin. In the rest of our experiments, FCIS with ResNeXt back-bone network and deformable convolution, namely FCIS+XD is selected as the standard configuration for our approach.

Table 2
Performance comparison (measured by mAP^r) of FCIS with its variants on Microsoft COCO 2014 test-dev [41].

Approach	mAP^r@[0.5:0.95]	mAP^r@0.5	mAP^r@[0.5:0.95] (small)	mAP^r@[0.5:0.95] (mid)	mAP^r@[0.5:0.95] (large)
FCIS	0.292	0.495	0.071	0.313	0.500
FCIS+D	0.288	0.498	0.070	0.309	0.514
FCIS+X	0.296	0.513	0.081	0.319	0.515
FCIS+XD	**0.303**	**0.522**	**0.082**	**0.326**	**0.528**

Table 3
Performance (mAP) of FCIS, FCIS+D, FCIS+X and FCIS+XD with the hybrid features of "conv3+conv4" on *Instance-160*.

Approach	top-10	top-20	top-50	top-100	all
FCIS	0.1969	0.3378	0.5496	0.6339	0.6936
FCIS+D	0.2089	0.3517	0.5688	0.6535	0.7127
FCIS+X	0.2078	0.3522	0.5682	0.6537	0.7125
FCIS+XD	**0.2109**	**0.3558**	**0.5747**	**0.6585**	**0.7237**

Table 4
Performance (mAP) of FCIS+XD compared to five representative approaches in the literature.

Approach	top-10	top-20	top-50	top-100	all
BoVW [4]	0.1061	0.1651	0.2483	0.2806	0.3141
BoVW+HE [37]	0.1483	0.2359	0.3553	0.4033	0.4375
R-MAC [28]	0.1014	0.1685	0.2680	0.3071	0.3577
CroW [29]	0.0733	0.1296	0.2391	0.2840	0.3375
DV-Res [31]	0.1763	0.2908	0.4609	0.5239	0.5790
DV-Vgg [31]	0.1939	0.3282	0.5413	0.6660	0.7306
FCIS+XD	**0.2109**	**0.3558**	**0.5747**	0.6585	0.7237

4.3. Comparison to state-of-the-art approaches

In this section, the performance of proposed FCIS+XD is studied in comparison to five representative approaches in the literature. They are two local feature based approaches BoVW [4] and BoVW+HE [37] and three deep feature based approaches R-MAC [28], Deepvision [31] (denoted as DV-Vgg) and CroW [29]. For Deepvision, the search is carried out in two steps. In the first step, the top-ranked candidates are produced by image level comparison. In the second step, instance level search is carried out on the top-*100* candidates. In order to make a more fair comparison between Deepvision and our approach, another run is also conducted for Deepvision. In this new run, back-bone network of Deepvision is replaced by ResNet-101, which becomes the same as FCIS. The filtering scheme in the first step is disabled. This run is denoted as DV-Res.

Table 4 shows the performance from all approaches. As seen from the table, DV-Vgg and FCIS+XD show considerably better performance than the rest. BoVW+HE still shows competitive performance in comparison to deep feature approaches such as R-MAC and CroW. Although the results from Deepvision are very close to FCIS+XD, they do not reflect real behavior of Deepvision. In *Instance-160*, the videos are primarily collected from visual tracking evaluation. In many cases, the query instance shares similar background scene as the reference images. So that true instances are retrieved by Deepvision due to their similar background. For this reason, the image-wise feature representation in Deepvision still works seemingly well. However, the performance of Deepvision drops considerably when the target instances are cluttered in different backgrounds. This will be confirmed by another experiment afterwards. Another disadvantage for Deepvision lies in its low accuracy of generated instance bounding box. As shown in the table, the mAP of DV-Res is even lower than original FCIS (see Table 3) although it is already powered by ResNet. This is mainly caused by its imprecise feature representation of each instance. In contrast, FCIS+XD is able to generate precise instance-level bounding-boxes owing to its precise object category-level classification and pixel-level mask prediction.

Table 5
Performance comparison (measured by mAP) of our method to Deepvision on *40* queries in which heavy background variations are observed.

Approach	top-10	top-20	top-50	top-100	all
DV-Vgg [31]	0.1925	0.2979	0.4642	0.5585	0.5894
FCIS	0.2521	0.4075	0.6094	0.6582	0.6975
FCIS+XD	**0.2624**	**0.4301**	**0.6467**	**0.6975**	**0.7366**

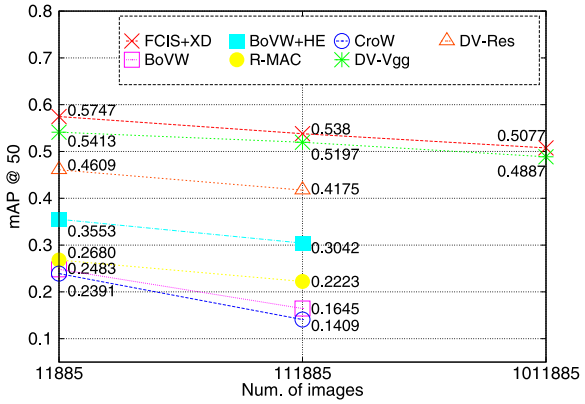


Fig. 6. Scalability test in comparison to five state-of-the-art approaches. The performance is measured by mAP at top-*50*.

In order to further confirm our observation about Deepvision, *40* queries from *Instance-160*, in which severe background variations are observed, are selected to verify its real behavior. Table 5 shows the performance of FCIS, FCIS+XD and Deepvision on *40* queries. As observed from the table, the performance of Deepvision drops considerably compared to that of Table 4. As the background scenes from the instance query and the reference images are dissimilar, the first round search in Deepvision becomes ineffective since it is based on image-wise feature. As the consequence, decent results are not expected from the re-ranking stage since many true-positives are already missed in the first stage. Another disadvantage of this approach is that one has to keep two types of features. One is on image level, another is on region level, which induce heavy computational overhead.

4.4. Scalability test

In this section, the scalability of the proposed feature representation is studied. In the experiment, *1* million distractor images are added in the reference set. The same processing pipeline is undertaken on this *1* million images. In the experiment, five representative approaches are considered. For FCIS+XD, *1,648,654* instances are extracted from the distractor images, each of which is represented as an *1,536*-dimensional feature vector.

As seen from Fig. 6, FCIS+XD shows the best scalability. It outperforms Deepvision by a constant margin. As the computation cost is high and the results from BoVW, BoVW+HE, R-MAC, CroW and DV-Res are already much poorer than FCIS+XD and Deepvision (DV-Vgg) with 100 K distractors, further verification on the whole *1* million distractors is not carried out for these approaches. Fig. 7 shows six instance search results produced by FCIS+XD. As shown in the figure,

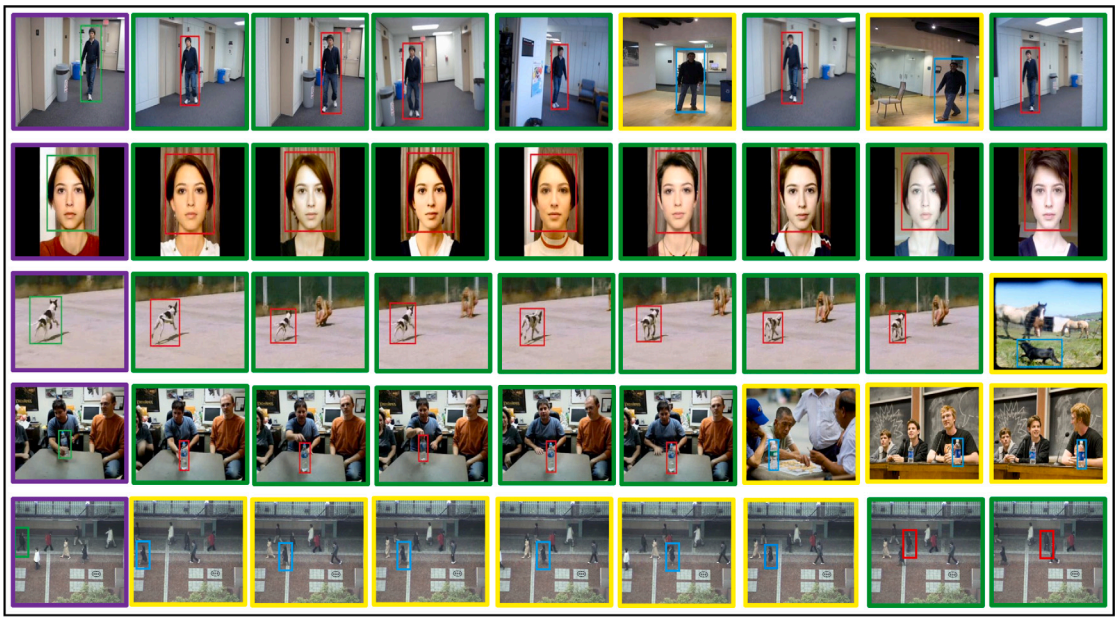


Fig. 7. Top-*8* search results of six sample queries produced by FCIS+XD with *1* million distractor images (best viewed in color). The instances highlighted by a green bounding-box in the first column are the query instances. The top-*8* search results for each query are listed in the following columns of each row. The false-positives are outlined with bounding-box in blue, while the true-positives are outlined by bounding-box in red.

all the top-*8* results for each individual query are meaningful. The last row in Fig. 7 shows one typical example where our approach fails. False positive results are returned when the false instances demonstrating similar appearance and scale as the query instance. Although a few false-positive instances are returned, they indeed exhibit very close appearance as the query.

5. Conclusion

We have presented a promising way of instance level feature representation for instance search. This representation is built upon a fully convolutional network that is originally used for instance segmentation. With the precise instance segmentation, the feature is derived by ROI pooling on the feature maps. To further boost its performance, two enhancement strategies are proposed. The distinctiveness and scalability of this feature have been comprehensively studied. As shown in the experiment, it outperforms most of the representative approaches in the literature. Considering the lack of publicly available evaluation benchmark, a medium-scale dataset for instance search is introduced by harvesting videos from object tracking benchmarks. Currently, the types of instances that our approach could handle with are restricted to Microsoft COCO-*80* categories. Although it already covers variety of instances that we encounter in the daily life, exploring more generic instance segmentation model that works beyond *80* categories will be our future research focus.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by National Natural Science Foundation of China under grants 61572408 and 61972326, and the grants of Xiamen University, PR China 20720180074.

References

[1] G. Awad, W. Kraaij, P. Over, S. Satoh, Instance search retrospective with focus on TRECVID, IJMIR 6 (1) (2017) 1–29.
[2] Y. Rui, T.S. Huang, S. Chang, Image retrieval: Current techniques, promising directions, and open issues, JVCIR 10 (1) (1999) 39–62.
[3] A. AlZu'bi, A. Amira, N. Ramzan, Semantic content-based image retrieval: A comprehensive study, JVCIR 32 (2015) 20–54.
[4] J. Sivic, A. Zisserman, Video google: A text retrieval approach to object matching in videos, in: ICCV, 2003, pp. 1470–1477.
[5] H. Jégou, M. Douze, C. Schmid, P. Pérez, Aggregating local descriptors into a compact image representation, in: CVPR, 2010, pp. 3304–3311.
[6] H. Jégou, M. Douze, C. Schmid, Improving bag-of-features for large scale image search, IJCV 87 (3) (2010) 316–336.
[7] C. Reta, J.A. Cantoral-Ceballos, I.S. Moreno, J.A. Gonzalez, R. Alvarez-Vargas, N. Delgadillo-Checa, Color uniformity descriptor: An efficient contextual color representation for image indexing and retrieval, JVCIR 54 (2018) 39–50.
[8] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: CVPR, 2005, pp. 886–893.
[9] F. Perronnin, J. Sánchez, T. Mensink, Improving the fisher kernel for large-scale image classification, in: ECCV, 2010, pp. 143–156.
[10] J. Zhang, D. Li, Y. Zhao, Z. Chen, Y. Yuan, Representation of image content based on roi-bow, JVCIR 26 (2015) 37–49.
[11] R. Raveaux, J. Burie, J. Ogier, Structured representations in a content based image retrieval context, JVCIR 24 (8) (2013) 1252–1268.
[12] X. Wang, Z. Wang, A novel method for image retrieval based on structure elements' descriptor, JVCIR 24 (1) (2013) 63–74.
[13] A. Babenko, A. Slesarev, A. Chigorin, V.S. Lempitsky, Neural codes for image retrieval, in: ECCV, 2014, pp. 584–599.
[14] A.S. Razavian, H. Azizpour, J. Sullivan, S. Carlsson, CNN features off-the-shelf: An astounding baseline for recognition, in: CVPR Workshops, 2014, pp. 512–519.
[15] A. Babenko, V.S. Lempitsky, Aggregating deep convolutional features for image retrieval, ICCV (2015).
[16] J.Y. Ng, F. Yang, L.S. Davis, Exploiting local features from deep networks for image retrieval, 2015, CoRR abs/1504.05133.
[17] H. Jégou, M. Douze, C. Schmid, Product quantization for nearest neighbor search, IEEE TPAMI 33 (1) (2011) 117–128.
[18] M. Datar, N. Immorlica, P. Indyk, V.S. Mirrokni, Locality-sensitive hashing scheme based on p-stable distributions, in: ACM Symposium on Computational Geometry, 2004, pp. 253–262.
[19] M. Muja, D.G. Lowe, Fast approximate nearest neighbors with automatic algorithm configuration, in: VISAPP, 1, 2009, pp. 331–340.
[20] M. Muja, D.G. Lowe, Scalable nearest neighbor algorithms for high dimensional data, IEEE TPAMI 36 (11) (2014) 2227–2240.
[21] D.G. Lowe, Distinctive image features from scale-invariant keypoints, IJCV 60 (2) (2004) 91–110.
[22] R. Arandjelovic, A. Zisserman, Three things everyone should know to improve object retrieval, in: CVPR, 2012, pp. 2911–2918.

[23] H. Bay, T. Tuytelaars, L.J.V. Gool, SURF: speeded up robust features, in: ECCV, 2006, pp. 404–417.
[24] L. Zheng, Y. Zhao, S. Wang, J. Wang, Q. Tian, Good practice in CNN feature transfer, 2016, CoRR abs/1604.00133.
[25] R. Arandjelovic, P. Gronát, A. Torii, T. Pajdla, J. Sivic, Netvlad: CNN architecture for weakly supervised place recognition, in: CVPR, 2016, pp. 5297–5307.
[26] L. Xie, L. Zheng, J. Wang, A.L. Yuille, Q. Tian, Interactive: Inter-layer activeness propagation, in: CVPR, 2016, pp. 270–279.
[27] J. Deng, W. Dong, R. Socher, L. Li, K. Li, F. Li, Imagenet: A large-scale hierarchical image database, in: CVPR, 2009, pp. 248–255.
[28] G. Tolias, R. Sicre, H. Jégou, Particular object retrieval with integral max-pooling of CNN activations, ICLR (2016).
[29] Y. Kalantidis, C. Mellina, S. Osindero, Cross-dimensional weighting for aggregated deep convolutional features, in: ECCV Workshops, 2016, pp. 685–701.
[30] Y. Li, H. Qi, J. Dai, X. Ji, Y. Wei, Fully convolutional instance-aware semantic segmentation, in: CVPR, 2017, pp. 4438–4446.
[31] A. Salvador, X. Girói Nieto, F. Marqués, S. Satoh, Faster R-CNN features for instance search, in: CVPR Workshops, 2016, pp. 394–401.
[32] S. Xie, R.B. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: CVPR, 2017, pp. 5987–5995.
[33] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, Deformable convolutional networks, in: ICCV, 2017, pp. 764–773.
[34] T. Yao, Y. Pan, Y. Li, T. Mei, Hierarchy parsing for image captioning, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019.
[35] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: CVPR, 2016, pp. 770–778.
[36] J. Philbin, O. Chum, M. Isard, Sivic, A. Zisserman, Object retrieval with large vocabularies and fast spatial matching, in: CVPR, 2007.
[37] H. Jégou, M. Douze, C. Schmid, Hamming embedding and weak geometric consistency for large scale image search, in: ECCV, 2008, pp. 304–317.
[38] Y. Wu, J. Lim, M. Yang, Object tracking benchmark, IEEE TPAMI 37 (9) (2015) 1834–1848.
[39] A.W.M. Smeulders, D.M. Chu, R.C. et al., Visual tracking: An experimental survey, IEEE TPAMI 36 (7) (2014) 1442–1468.
[40] M. Everingham, L.J.V. Gool, C.K.I. Williams, J.M. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, IJCV 88 (2) (2010) 303–338.
[41] T. Lin, M. Maire, S.J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft COCO: common objects in context, in: ECCV, 2014, pp. 740–755.

Yu Zhan received his Bachelor degree of Computer Science from Xiamen University, China in 2016. He is currently a graduate student at Department of Computer Science, Xiamen University. His research interest is content-based image retrieval and instance search.

Wan-Lei Zhao received his Ph.D. degree from City University of Hong Kong in 2010. He received M.Eng. and B.Eng. degrees in Department of Computer Science and Engineering from Yunnan University in 2006 and 2002 respectively. He currently works with Xiamen University as an associate professor, China. Before joining Xiamen University, he was a Postdoctoral Scholar in INRIA, France. His research interests include multimedia information retrieval and video processing.