ALADDIN: A LOCALITY ALIGNED DEEP MODEL FOR INSTANCE SEARCH

Wenhui Jiang [1], *Zhicheng Zhao* [1,2], *Fei Su* [1,2], *Anni Cai* [1,2]

[1] School of Information and Communication Engineering
[2] Beijing Key Laboratory of Network System and Network Culture
Beijing University of Posts and Telecommunications, Beijing, China

ABSTRACT

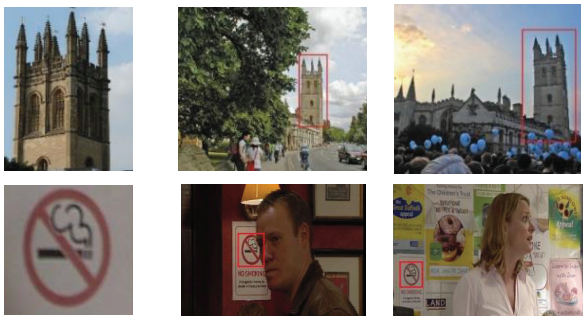
Most instance search systems are based on modeling local features. It remains a challenge to apply deep learning techniques into this task because of the asymmetrical similarity between the query region and dataset images. In this paper, we propose ALADDIN, A Locality Aligned Deep moDel for INstance search. This model deals with the asymmetrical similarity by searching query instances at the scale of aligned target regions instead of the whole image. Towards discriminative region representations, we utilize a deep convolutional network which captures both intra-class and inter-class distinctions of the regions. In addition, we propose a semi-supervised method to collect appropriate data to train the network. Extensive experiments confirm that our method is more suitable for generic instance search than most conventional methods, and outperforms the best CNNs-based method in both accuracy and efficiency.

Index Terms— Deep learning, asymmetrical similarity, object proposal, intra-class distinction, instance search

1. INTRODUCTION

Instance search (aka object search) aims at retrieving the images including object or scene similar to the given query region. It remains a challenging task mainly due to two problems: 1) Robust image representation: The same object may appear quite different because of viewpoint, illumination, occlusion and so on; 2) Asymmetrical search [1]: The query object may cover only a small part of a dataset image, therefore the real signal on the relevant region will drown in the noise from the background. For example, consider the problem of retrieving all the scenes that contain "Oxford Magdalen Tower" in Figure 1. The Tower is surrounded by a significant amount of clutter and the visual object varies largely due to different viewpoints and illumination. Therefore, learning a robust visual representation for objects and eliminating the impact of cluttered background are keys to accurate instance search.

This work is supported by Chinese National Natural Science Foundation (61471049, 61372169, 61532018) and China Scholarship Council.



(a) Query (b) Query objects in dataset images

Fig. 1. Illustrations of asymmetrical search. The query region, which is delimited by a bounding box, may cover only a small part in relevant dataset images.

Most recent instance search systems [2][3][4] rely on local features such as SIFT to deal with appearance variations, and then design image similarity functions to discount for the clutter. For example, Zhu [1] proposed an asymmetrical dissimilarity to down-weight the contribution of local features from background. Tao [5] considered many boxes per database image as candidate targets to search locally in the picture. Although these local features-based methods are successful in several applications, their performance is quite limited due to the representation ability of the hand-crafted features. Besides, local features are incapable of describing small or smooth objects [6], which largely exist among natural images.

Deep learning based methods have improved the state-of-art of many recognition tasks such as image classification [7] and object detection [8]. Moreover, Wang [9] and Wan [10] showed that convolutional neural networks (CNNs) are effective in image representation for content-based image retrieval (CBIR). However, instance retrieval is more difficult – discounting for background variations would require training on a very large specific dataset which is presently not available. To address this issue, Razavian [11] proposed to divide the query and dataset images into fixed sub-regions and ranked the dataset images based on the best-matched sub-regions. However, it lacks the invariance to object translation and scale change.

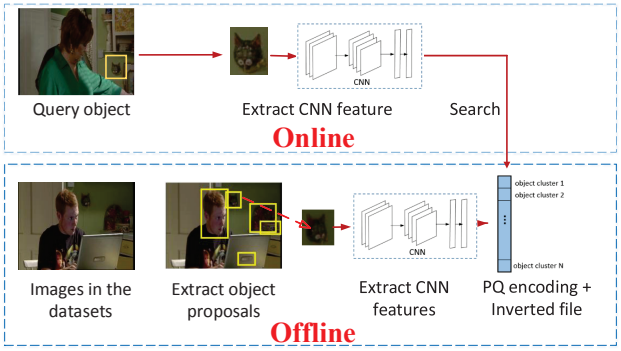


Fig. 2. A schematic view of our proposed method. Our system consists of three modules. The first generates object proposals for dataset images. The second module is a large CNN that extracts a feature vector for each proposal. The third module is an indexing system that efficiently stores and organizes the feature vectors.

In this paper, we attempt to address two major questions. First, how to deal with the asymmetrical search problem when representing an image using CNNs. Second, how to collect relevant training data. Training data is critical for learning CNNs, but for instance search, there are few labelled examples.

Towards these goals, we propose ALADDIN, **A** **L**ocality **A**ligned **D**eep mo**D**el for **IN**stance search. A brief illustration of ALADDIN is shown in Figure 2. In this model, we propose to use object proposal [12] to target query region in a set of candidate regions. In this way, for each relevant image in the dataset, at least one candidate region is approximately aligned to the query object. Then the asymmetrical object retrieval is converted into symmetrical object matching. To capture instance-level discriminative information among these regions, a deep convolutional network is designed. In addition, we propose a semi-supervised way to collect training data to train this network. We show that our method significantly outperforms the best CNN-based method in both accuracy and efficiency.

Major contributions of this work are two-folds:

1) A framework based on deep convolutional network for instance search. To the best of our knowledge, deep learning has not been successfully applied to instance search before;

2) A semi-supervised method to collect training data. Recent works on CBIR [10] collected training data in a supervised way: the query object is known as a priori, then relevant images are carefully collected to form the training set – this is infeasible for real applications. In our method, no prior information and very little supervision is required for collecting labelled training data.

2. ARCHITECTURE OVERVIEW

Current deep learning models are insufficient for capturing the asymmetrical similarity relationship [1] between query objects and dataset images. To address this problem, we utilize EdgeBox [12] to decompose a dataset image D into a small set of regions such that each object appearing in D is approximately aligned to one of the regions:

$$D = \{R_1, R_2, ..., R_k\} \tag{1}$$

EdgeBox generates around 2000 candidate regions per image. We notice that some of these regions show very little intensity variations, and they could hardly represent any object. As an improvement, we discard these regions by setting a small threshold on the standard variance of the region's pixel intensity. Such a simple procedure reduces the number of proposed regions by about 10% without hurting the recall rate of object detection.

Search then proceeds at the scale of candidate object regions instead of the entire image. For accurate region matching, we leverage the power of CNNs to learn discriminative features for the aligned object regions. Specifically, the deep network is learned to project aligned object regions into a new feature subspace, under which patches depicting the same object are closer to each other than patches of other objects. The implementation of the network is presented in Section 3.

With the discriminative features extracted from CNNs, the similarity between the query image Q and a dataset image D is determined by the maximum scored object region R_m:

$$sim(Q, R_i) = \frac{< \boldsymbol{q}, \boldsymbol{r} >}{\|\boldsymbol{q}\|_2 \|\boldsymbol{r}\|_2} \tag{2}$$

$$Sim(Q, D) = \max_{R_i \in D} sim(Q, R_i) \tag{3}$$

where $\boldsymbol{q}$ and $\boldsymbol{r}$ represent the feature vectors of image Q and R_i extracted from CNNs.

Exhaustive region search requires 2000-fold increase in search time and memory storage. To solve this problem, we propose to encode each feature vector using product quantization (PQ) [13]. The similarity between query feature and one

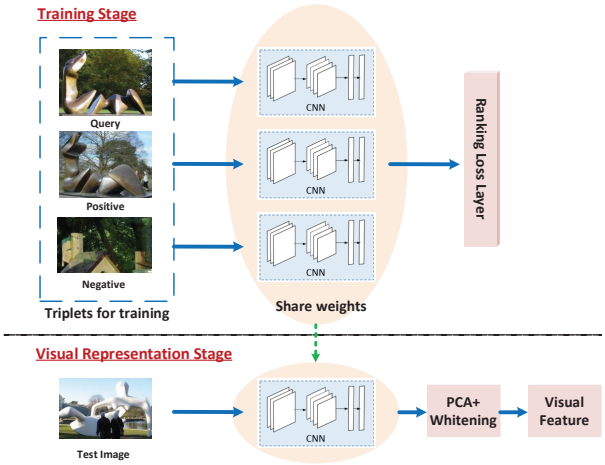


Fig. 3. The architecture of the deep network for feature representation.

encoded feature could be calculated very fast using a look-up table. We further incorporate these encoded features into an inverted file system. This makes our retrieval system very efficient.

3. NETWORK FOR FEATURE REPRESENTATION

3.1. Network Architecture

It is shown that CNNs trained on large datasets (DeCAF) are generic and can help in other computer vision problems [14][15]. However, this feature mainly focuses on *category-level* image similarity [9]. In this paper, we aim at training an end-to-end deep network for better *instance-level* feature representation. The learned model keeps patches from the same object to be closer than those of different objects by a large margin. Towards this goal, we construct a set of triplets $\mathscr{T}$ where $T_i = \{Q_i, P_i, N_i\}$. In a triplet, (Q_i, P_i) represents a pair of image regions depicting the same object, and (Q_i, N_i) denotes two patches of different objects. Following [10][16], we can define a ranking-based loss function:

$$\ell(\mathscr{T}) = \sum_i \max\{0, \gamma - sim(Q_i, P_i) + sim(Q_i, N_i)\} \quad (4)$$

where γ represents the margin, and the similarity function is determined by Equation 2.

In our implementation, except the loss layer, we retain a similar architecture as AlexNet [7]. As is shown in Figure 3, in the training step, the network takes a set of triplets as inputs. The three examples in a triplet are fed into individual CNNs that share weights, and the ranking loss layer defined in Equation 4 is placed on top of the outputs of the three CNNs. While in the testing step, the test image is fed into the trained CNN directly for feature extraction.

Following [8, 17], we use ImageNet-trained model as initializing model, and then fine-tune the parameters with the triplet training data $\mathscr{T}$. The optimization can be done by applying Stochastic Gradient Descent (SGD) algorithm with Back-propagation (BP).

3.2. A semi-supervised way to collect relevant training data

As with most machine learning problems, it is challenging to collect large datasets for training deep networks. Since we expect to extract features from segmented object regions through CNNs, it would be better to collect a set of labelled triplets from aligned object patches instead of entire images [18]. Towards this goal, we propose a semi-supervised method to mine "hard negative" triplets as training data. We define "hard negatives" as the triplets in which DeCAF [14] fails to tell the positive element from the negative one. The method proceeds in four steps as shown in Figure 4:

1) For each dataset, select 2000 most likely object regions according to the objectness score returned from [12].

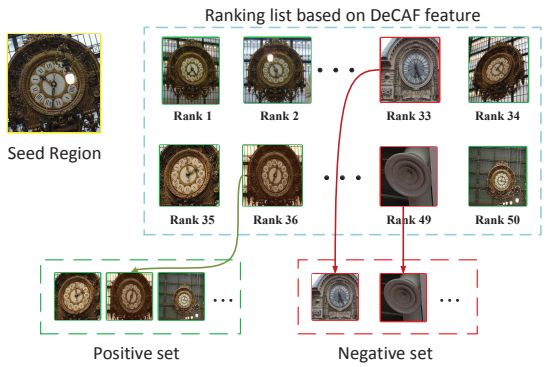


Fig. 4. Illustration of our semi-supervised way to collect training data. The correctly verified patches are labelled by green bounding boxes, while the incorrect patches are labelled by red boxes.

Among these regions, the most obvious texture-less regions could be easily recognized and filtered out. The remaining regions serve as seed regions.

2) For each seed region S, search for the best matching region in each dataset based on DeCAF feature, and rank the returned regions according to the similarity. The top 50 returned regions are retained as candidates — they are regarded as positive regions determined by DeCAF.

3) Verify the correctness of the candidates. We combine conventional local features (BoW [19] and BoB [6]) with RANSAC [19] to verify if the candidate region is a correct match to the seed region. Specifically, for a candidate region, if more than ten local feature matches are verified by RANSAC, it is considered as a correct match. We denote the set of verified correct matches as well as the seed region by $\mathcal{C}_p$, whilst the set of incorrect matches by $\mathcal{C}_n$.

4) Divide $\mathcal{C}_p$ into two halves — one as queries and the other one as positives. Take regions in $\mathcal{C}_n$ as negatives. Then a set of triplets is generated as training data.

4. EXPERIMENTAL RESULTS

4.1. Datasets and baselines

We test the results on Oxford5k [19], Paris6k [20], and a more challenging dataset – Sculpture6k [6] which is featured for smooth and texture-less objects.

We evaluate our method with two groups of baselines. The first group includes four bag-of-feature (BoF) based models, *i.e.*, bag-of-words (BoW) [21], bag-of-boundaries (BoB) [6], Asymmetrical similarity (AS) [1] and Localized search (LS) [5]. The other group includes state-of-the-art methods based on deep learning, *i.e.*, DeCAF [14], ReDSL [10], CNN-ss [11] and Deep Local Features (DLF) [22]. We denote the **L**ocality **A**ligned scheme as "LA". The performance is evaluated by mean average precision (mAP) [19].

Table 1. The performance of instance search on three datasets (%). The average mAP is denoted as Generic.

Model	Oxford	Paris	Sculpture	Generic
BoW [21]	64.78	53.29	8.0	42.03
BoB [6]	–	–	45.4 [6]	–
AS [1]	77.80	–	–	–
LS [5]	73.40 [5]	–	–	–
DeCAF [14]	30.4	57.6	38.4	42.13
ReDSL [10]	65.41	76.50	53.48	56.48
CNN-ss [11] [1]	67.19	75.59	41.33	61.37
DLF[22]	64.9	69.4	–	–
LA+DeCAF	67.79	75.80	51.68	65.09
ALADDIN	**77.97**	**78.28**	**59.26**	**71.83**

Table 2. Efficiency of our method compared with other works

Model	Dim	runtime (s)	
		Oxford 5k	Paris 6k
BoW [21]	1M	0.031	0.042
DeCAF [14]	4096	0.241	0.301
CNNss [11]	35k	1.783	2.009
ALADDIN	$\sim 8k$	**0.009**	**0.011**

4.2. Implementations

Our deep model is implemented by *Caffe* [23]. We take the activations of the first fully-connected layer as features for the aligned object patches. The features are compressed by PCA to 512 dimensions and followed by whitening [24]. To encode the features using PQ, the 512-D features are grouped into 16 parts, each part is encoded by 8 bits (256 clusters).

4.3. Performance evaluation

Deep Feature VS BoF: As is shown in Table 1, BoW works well on Oxford5k and Paris6k, but shows limited performance on Sculpture6k. That is because SIFT-like features are incapable of describing smooth and texture-less objects. BoB is not tested on Oxford5k and Paris6k since it is designed specifically for smooth objects. In contrast, deep convolutional models are capable of describing more generic objects.

Ours VS Other Deep Models: 1) *How important are CNN features at aligned object region level?* From Table 1, it is clear that although DeCAF [14] is helpful features in many computer vision problems, they do not deal with the asymmetrical similarity associated in the task of instance search. In contrast, by decomposing the entire image into a set of candidate object regions (LA + DeCAF), the performance boosts significantly on three datasets. Also, LA + DeCAF outperforms CNNss, because compared with locality aligned scheme, CNNss lacks invariance to object translation and scale change. 2) *How important it is to capture instance-level image distinction?* We further compare the result of searching with DeCAF with the one that features are learned to capture instance-level image distinction. It is clear that by further leveraging instance-level distinction (ALADDIN), the retrieval performance boosts by 6.74% compared with LA+DeCAF on three datasets in average. A clear example is shown in Figure 5. We also compare the results of learning a ranking model on top of entire images (ReDSL) and on aligned object patches (ALADDIN). It is obvious that learning a ranking model on top of aligned object patches is better, that is because learning CNNs from scratch is insufficient to capture the asymmetrical similarity, but by decomposing the image into aligned object regions, we make the subsequent training of convolutional networks drastically easier.

[1]For a fair comparison, we re-implement the CNNs with *Caffe* [23]. Note that [11] applied *Overfeat* in his implementation, which achieved a better baseline result.

Fig. 5. Comparison of ranking example of LA+DeCAF (upper) and ALADDIN (bottom). False results are marked with red boxes.

4.4. Efficiency analysis

We compare the speed efficiency with BoW[21], DeCAF [14] and CNNss [11] in Table 2 [2]. BoW is the simplest baseline of local features-based method. DeCAF stands for holistic features. CNNss gives the state-of-art performance among deep learning based methods. The speed of the algorithms is evaluated by the average processing time per query. It is clear that CNNss is the slowest due to exhaustive search among split sub-regions. BoW model is much faster because of its sparse distribution of feature. Compared with BoW model in which the query image is represented by hundreds of local features, query in our method is represented by only one feature vector, *i.e.*, the holistic CNN feature. Therefore, our method reduces the search time compared with BoW model.

5. CONCLUSIONS

In this paper, we proposed a locality aligned deep model for instance search. Our method addresses the problem of asymmetrical similarity by decomposing the dataset images into aligned object regions, and leverages the strength of CNN for image representation. The CNN is learned using a training set collected in a semi-supervised method. Extensive experiments on three benchmark datasets confirm that our method is more suitable for generic instance search than most conventional methods, and significantly outperforms the best CNNs-based method in both accuracy and efficiency.

[2]Hardware information: Intel Xeon E5-2609 2.40GHz CPU (8 Cores) and 48 GB RAM

6. REFERENCES

[1] Caizhi Zhu, Herve Jegou, and Shin'ichi Satoh, "Query-adaptive asymmetrical dissimilarities for visual object retrieval," in *ICCV*, 2013.

[2] Paul Over, George Awad, Martial Michel, Jonathan Fiscus, Wessel Kraaij, Alan F. Smeaton, Georges Quenot, and Roeland Ordelman, "Trecvid 2015 – an overview of the goals, tasks, data, evaluation mechanisms and metrics," in *Proceedings of TRECVID 2015*, 2015.

[3] Liang Zheng, Shengjin Wang, Wengang Zhou, and Qi Tian, "Bayes merging of multiple vocabularies for scalable image retrieval," in *CVPR*, 2014.

[4] Liang Zheng, Shengjin Wang, and Qi Tian, "Coupled binary embedding for large-scale image retrieval," *TIP*, vol. 23, no. 8, pp. 3368–3380, 2014.

[5] Ran Tao, Efstratios Gavves, Cees G M Snoek, and Arnold W M Smeulders, "Locality in Generic Instance Search from One Example," in *CVPR*, 2014.

[6] R Arandjelovic and Andrew Zisserman, "Smooth Object Retrieval using a Bag of Boundaries," in *ICCV*, 2011.

[7] Alex Krizhevsky and Geoffrey E Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *NIPS*, 2012.

[8] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.

[9] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu, "Learning Fine-grained Image Similarity with Deep Ranking," in *CVPR*, 2014.

[10] Ji Wan, Dayong Wang, Steven C H Hoi, and Pengcheng Wu, "Deep Learning for Content-Based Image Retrieval : A Comprehensive Study," in *MM*, 2014.

[11] Ali Sharif, Razavian Hossein, Azizpour Josephine, and Sullivan Stefan, "CNN Features off-the-shelf : an Astounding Baseline for Recognition," in *CVPR Workshop*, 2014.

[12] C Lawrence Zitnick and Piotr Dollár, "Edge boxes: Locating object proposals from edges," in *ECCV*, 2014, pp. 391–405.

[13] Hervé Jégou, Matthijs Douze, and Cordelia Schmid, "Product Quantization for Nearest Neighbor Search," *TPAMI*, vol. 33, no. 1, pp. 117–128, 2011.

[14] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, Trevor Darrell, Trevor Eecs, and Berkeley Edu, "DeCAF : A Deep Convolutional Activation Feature for Generic Visual Recognition," in *ICML*, 2014.

[15] Liang Zheng, Shengjin Wang, Lu Tian, Fei He, Ziqiong Liu, and Qi Tian, "Query-adaptive late fusion for image search and person re-identification," in *CVPR*, 2015.

[16] Pengcheng Wu, Steven C.H. Hoi, Hao Xia, Peilin Zhao, Dayong Wang, and Chunyan Miao, "Online multimodal deep similarity learning with application to image retrieval," in *MM*, 2013.

[17] Pulkit Agrawal, Ross Girshick, and Jitendra Malik, "Analyzing the Performance of Multilayer Neural Networks for Object Recognition," in *ECCV*, 2014.

[18] Wanli Ouyang, Xiaogang Wang, Xingyu Zeng, Shi Qiu, and Xiaoou Tang, "DeepID-Net: Deformable Deep Convolutional Neural Networks for Object Detection," in *CVPR*, 2015.

[19] James Philbin, O Chum, Michael Isard, Josef Sivic, and Andrew Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *CVPR*, 2007.

[20] James Philbin, O Chun, Michael Isard, Josef Sivic, and Andrew Zisserman, "Lost in quantization:improving particular object retrieval in large scale image databases," in *CVPR*, 2008.

[21] Relja Arandjelovi and Andrew Zisserman, "Three things everyone should know to improve object retrieval," in *CVPR*, 2012.

[22] Joe Yue-Hei Ng, Fan Yang, and Larry S. Davis, "Exploiting Local Features from Deep Networks for Image Retrieval," in *CVPR Workshop*, 2015.

[23] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[24] Hervé Jégou and Ondřej Chum, "Negative evidences and co-occurences in image retrieval: The benefit of pca and whitening," in *ECCV*, 2012, pp. 774–787.