Abstract

We present an analysis of embeddings extracted from different pre-trained models for content-based image retrieval. Specifically, we study embeddings from image classification and object detection models. We discover that even with additional human annotations such as bounding boxes and segmentation masks, the discriminative power of the embeddings based on modern object detection models is significantly worse than their classification counterparts for the retrieval task. At the same time, our analysis also unearths that object detection model can help retrieval task by acting as a hard attention module for extracting object embeddings that focus on salient region from the convolutional feature map. In order to efficiently extract object embeddings, we introduce a simple guided student-teacher training paradigm for learning discriminative embeddings within the object detection framework. We support our findings with strong experimental results.

1 Introduction

Convolutional neural networks trained on large-scale image classification datasets such as ImageNet [10] have been shown to be an effective generic feature extractor that can be applied to different vision tasks. These include modern object detection frameworks such as Faster-RCNN [44], which utilizes the same network architecture pre-trained on image classification datasets for feature extraction. With the availability of large-scale object detection and segmentation datasets such as COCO [31] and OpenImagesV4 [29] that come with additional bounding boxes and mask annotations, we explore whether features extracted from models trained on them would display similar effectiveness as a generic feature extractor. While ImageNet classification embeddings have been extensively studied [3, 12, 21, 27, 49, 58], little work has focused on analyzing embeddings extracted from object detection models. In this paper, we investigate the performance of such embeddings for image retrieval.

Our analysis shows that even though object detection or semantic segmentation model utilizes additional annotations, the embedding learned from these models is significantly less discriminative than embeddings learn from classification models when conducting image retrieval. This suggests that the joint learning of classification and localization leads to degradation of the discriminative power of the resulting embeddings. However, we also discover that by retrieving similar objects as opposed to images, we can significantly improve image retrieval performance. For best of both worlds, we show that by utilizing object detection as a hard attention module to extract embeddings from classification model pertaining to the object regions, it allows the model to focus on salient regions and at the same time ignore background clutter.
For applications with an efficiency requirement, we propose a guided student-teacher training regime. We first train a teacher classification network with image-level labels as a discriminative feature extractor. This is followed by training a light-weight student network on top of the detection model that projects the feature map of the detection model into a more discriminative feature space guided by the teacher model. During image retrieval, we use the object detector as a hard attention module and extract object-level embeddings from the output of the student network with a single forward pass. This is as opposed to maintaining a separate feature extractor and an object detector, which would require two forward passes. Such a student network would still decouple feature learning from localization, which helps to preserve the discriminative power of the features. It is also possible to learn different student transformations without re-training the object detection model.

Our contributions include: (1) We empirically show that embeddings extracted from object detection models are less discriminative than embeddings extracted from image classification models when the task of image retrieval is considered. (2) We demonstrate that an object detector can help image retrieval performance by acting as a hard attention module. (3) For efficiency, we propose a student-teacher training paradigm, which allows us to extract discriminative object embeddings in a single forward pass. (4) Finally, extensive experimental results show the advantage of the proposed approach. Further, we also demonstrate the efficacy of our approach for near duplicate object retrieval, which allows for an important application in detecting image splicing.

2 Background and Related Work

Representation learning from large-scale datasets. Previous works mainly studied the transfer-ability of embeddings extracted from classification models that have been trained on datasets such as ImageNet to other tasks \cite{3, 12, 21, 27, 49, 58}. For instance, \cite{49} reports comprehensive results of applying embeddings from ImageNet-trained classification model to object detection, scene recognition, as well as image retrieval. In contrast, the efficacy of embeddings obtained from object detection models trained on large-scale object detection datasets such as COCO \cite{31} and OpenImages \cite{29} has not been widely studied. In this work, we provide an analysis of embeddings extracted from different models pre-trained on large-scale datasets for retrieval task.

Content-based image retrieval aims to retrieve relevant images from an image database given a query image based on the image content. Early work \cite{48} used global color and texture statistics such as color histogram and Gabor wavelet transform to represent the image. Later advances on instance retrieval using local feature \cite{33} and indexing methods \cite{24, 25, 51} achieved robustness against illumination and geometric variations. With the recent broad adoption of convolutional neural networks (CNN), different techniques has been proposed for global feature extraction \cite{2, 5, 6, 16, 40, 54}, local feature extraction \cite{35, 37, 59}, embedding learning \cite{14, 36, 55, 57}, as well as geometric alignment \cite{34, 45, 46} using deep networks. Zheng \textit{et al.} \cite{60} provide a comprehensive review of recent approaches towards image retrieval. Different from traditional image retrieval using either global features or local features, our approach generates a few discriminative object embeddings utilizing object detection models for image retrieval.

Object detection aims to detect different objects in an input image. Girshick \textit{et al.} \cite{15} proposed one of the first deep learning based object detection models, R-CNN, which improved the accuracy significantly compared to traditional methods \cite{9, 11, 13}. Since then many enhancements \cite{30, 43, 44, 50} have been made to improve accuracy as well as the training/inference time. A comprehensive survey of recent deep learning based object detection methods can be found in \cite{32}. By taking advantage of recent success in object detection, our model can learn discriminative object-level embeddings for image retrieval. Most recently, Teichmann \textit{et al.} \cite{53} utilized a specialized landmark detection model to aggregate deep local features \cite{37} for landmark retrieval. Object detection has also been used to improve the performance of other vision tasks such as visual question answering \cite{11}.

Knowledge distillation \cite{4, 7, 8, 20, 47} compress a complex model into a simpler one while maintaining the accuracy of the model. Bucilua \textit{et al.} \cite{7} first proposed to train a single model to mimic the outputs of an ensemble of models. Ba \textit{et al.} \cite{4} adopted a similar idea to compress deep neural networks. Hinton \textit{et al.} \cite{20} further generalized the idea with temperature cross entropy loss. Our student-teacher approach is related to knowledge distillation, which learns a simple student model to mimic the output of a complex one. What is different is that we leverage a detection network
Table 1: Image retrieval performance (mAP) with embeddings extracted from different pre-trained models for four different retrieval benchmarks. Even though all detection and semantic segmentation models are initialized with weights trained on ImageNet classification dataset, the embeddings learned from these models perform significantly worse than embeddings learned from the classification model.

| Model (Training Set)          | # of Img. / Cls. | ROxf | RPar | CUB200 | Cars196 |
|-------------------------------|------------------|------|------|--------|---------|
| Faster-RCNN (COCO)**[44]**    | 330K / 80        | 28.03| 41.66| 5.63   | 3.70    |
| Faster-RCNN (OpenImagesV4)**[44]** | 1.7M / 601   | 28.26| 40.07| 8.77   | 3.17    |
| Mask-RCNN (COCO)**[17]**      | 330K / 80        | 27.15| 42.60| 4.94   | 3.32    |
| ResNet50 (ImageNet)**[18]**   | 1.2M / 1K        | 35.52| 47.30| 20.31  | 6.36    |

Figure 1: Detailed analysis of embeddings learned from different models. Embeddings learned from ImageNet classification model consistently achieve the best performance for different pooling techniques and PCA dimension.

to provide additional guidance during training, which we show is effective for training the student network.

3 Analyzing Embeddings for Image Retrieval

3.1 Embeddings from Pre-trained Models

We first provide a detailed analysis of embeddings extracted from different pre-trained models, including image classification, object detection and semantic segmentation models using four different retrieval benchmarks.

Retrieval benchmark. We consider four datasets for benchmarking, including USCB bird dataset [56] (CUB200), Stanford car dataset [28] (Cars196), and two landmark datasets, ROxford5K [39] (R Oxf) and RPairs6K [39] (R Par). For CUB200 and Cars196 we follow the same protocol in [38] and use leave-one-out partitions to evaluate on every images in the test set. For ROxford5K and RPairs6K we follow the medium protocol described in [39], using 70 and 55 images as queries, 4,993 and 6,322 images as database. We use mean average precision (mAP) to measure the performance of different embeddings.

Pre-trained models. We consider four different pre-trained models including (1) Faster-RCNN [44] trained on COCO (Faster-RCNN (COCO)), (2) Faster-RCNN, trained on OpenImagesV4 [29] (Faster-RCNN (OpenImagesV4)), (3) Mask-RCNN [17] trained on COCO with bounding box and mask annotations, and (4) ResNet50 [18] trained on ImageNet. We adopt open source implementation of Faster-RCNN and Mask-RCNN with ResNet50 as a backbone feature extractor for our detection and segmentation models and the same backbone as our classification model. For all Faster-RCNN and Mask-RCNN models, we use weights from the ImageNet classification model to initialize the backbone network and use default 1x learning rate schedule to train the models, except for OpenImagesV4 for which we use 8x schedule due to the larger dataset size.

1https://github.com/facebookresearch/maskrcnn-benchmark
Table 2: Performance of image retrieval by utilizing object detection model. We use object detection as a hard attention module for extracting object-level regional embeddings from convolutional feature maps for image retrieval. Retrieval performance in terms of mean average precision (mAP) and precision at ten (P@10) both shows significant improvement compared to using a single embedding from the whole image.

| Dataset Embeddings | ROxf mAP | P@10 | RPar mAP | P@10 | CUB200 mAP | P@10 | Cars196 mAP | P@10 |
|--------------------|----------|------|----------|------|------------|------|-------------|------|
| CNN                | 35.52    | 56.14| 47.30    | 93.57| 20.31      | 41.87| 6.36        | 25.19|
| CNN-OE             | 40.84    | 65.71| 57.58    | 96.00| 23.57      | 46.95| 9.41        | 34.69|

We extract features from conv5_x layer and use max pooling to produce image embeddings from different pre-trained models and use cosine similarity between embeddings for retrieval ranking. Table 1 shows the mean average precision of different models when used as feature extractors on the four retrieval benchmarks. Comparing Faster-RCNN (COCO) and Mask-RCNN (COCO), we note that additional mask annotations actually decrease the performance of the embeddings on some of the dataset, suggesting that additional localization constraints might even hurt the retrieval performance further. Also, by increasing the size of the training set from COCO to OpenImagesV4, the Faster-RCNN performance improves on some datasets while reducing on some other dataset. Most importantly, although all the models are initialized with weights trained on ImageNet classification, embeddings extracted from detection and segmentation models perform significantly worse than the embeddings from the ImageNet classification model. Even when training on the OpenImagesV4 dataset, which has a comparable training size as ImageNet, we still have a large gap compared to the classification embeddings. This suggests that enforcing both classification and localization during training compromises the discriminative ability of the embedding. Consequently, decoupling localization and classification might be crucial for learning embeddings that are effective for image retrieval.

Note that different spatial pooling techniques and post-processing steps such as dimensionality reduction and embedding whitening have been shown to greatly affect retrieval performance. Given a convolutional feature map from conv5_x layer $F \in \mathbb{R}^{W \times H \times C}$, we consider the following pooling functions $P: \mathbb{R}^{W \times H \times C} \rightarrow \mathbb{R}^C$: (1) sum pooling (SPoC), (2) max pooling (Max), (3) regional max pooling (R-MAC), and (4) generalized mean pooling (GeM). We also perform experiments while varying the number of dimensions in PCA from 256 to 2,048 with whitening. Figure 1 shows a detailed analysis of the effect of different pooling techniques and post-processing steps. Figure 1 (a) shows retrieval performance of four benchmarks with different PCA dimensions. By selecting the appropriate PCA dimension for the dataset, we can greatly improve the retrieval performance. For instance, in Cars196, by projecting the embeddings to 256-dimensional space with PCA whitening, we can improve mAP from 6.36% to 9.63%. However, embeddings from classification model (ResNet50) consistently perform the best for all dimensions, which confirms our previous observation. Figure 1 (b) shows the mAP for different pooling techniques. While pooling methods also greatly affect the performance, ResNet50 embeddings again consistently achieve the best performance among embeddings from different pre-training models.

3.2 Can Object Detection Help Image Retrieval?

Even though the embeddings extracted from object detection models are less discriminative, here we show how localization can be beneficial when conducting image retrieval. Using the same benchmarks, we show that by explicitly utilizing object bounding boxes predicted by the detection model as a hard attention mechanism, thereby ignoring background clutter, image retrieval performance can be improved. Specifically, for each image, we first deploy the object detection model trained on the OpenImagesV4 dataset to detect up to eight bounding boxes per image. For each bounding box, object-level embedding is extracted from conv5_x layer of ResNet50 model pre-trained on ImageNet using an ROI align layer. To compute similarity, we use the maximum similarity between pairwise objects embeddings. Table 2 shows mAP and precision at ten (P@10) of image retrieval when using the image embeddings (CNN) and the object-level embeddings (CNN-OE). CNN-OE achieves better performance on all benchmarks, which suggests the detection model can help retrieval by acting as a hard attention mechanism.
Figure 2: Overview of the student-teacher training paradigm. We first train a teacher classification network to learn discriminative features, and a separate object detection model for bounding box prediction. Finally, we train a compact student network to transform the feature map from the detection model to the discriminative feature space, guided by the teacher model.

Figure 3: Three different types of student networks. (a) Compact student network \(S_{full}\) that directly takes input images and tries to mimic the output of the teacher network. This can be considered as a simple model compression approach. (b) Student network that utilizes the low-level features from the detection model \(S_{top}\), it is more compact compared to \(S_{full}\), since it reuses the lower layers from the detection model. (c) Student network with multi-scale guidance \(S_{guided}\). It takes both high-level and low-level feature maps from the detection model as guidance to learn the discriminative features from teacher network.

4 Efficient Image Retrieval using Object Embeddings

Section 3.2 provides a simple approach toward utilizing object detection for improving retrieval performance. However, CNN-OE uses two separate models: a classification model used for generating discriminative feature maps, and a detection model responsible for the hard attention, resulting in two forward passes during inference. To be more efficient, we propose to use knowledge distillation [20] to combine the two models. Figure 2 shows the overview of our approach for image retrieval. During training, we first train a classification teacher model which learns to generate discriminative features as well as a separate object detection model. We then train a student network that transforms the feature map from the object detection model to the teacher model. During test time, the combined model outputs both the bounding box predictions as well as the discriminative feature maps. ROI align layer with spatial pooling is used to extract object embeddings from the feature maps to perform retrieval.

Training student networks. Figure 3 illustrates three different types of student networks. We first consider a simple model compression approach by training a compact student model to directly mimic the output of the teacher network (cf. Figure 3(a)). Given an input image \(I\), a pre-trained teacher network \(T: R^{W \times H \times 3} \rightarrow R^{W/32 \times H/32 \times C}\), we construct a student network \(S_{full}: R^{W \times H \times 3} \rightarrow R^{W/32 \times H/32 \times C}\) with one convolutional layers and four bottleneck layers with skip connections [18] and parameters \(\theta_s\). We directly minimize the mean squared error between the output feature maps using gradient decent:

\[
\min_{\theta_s} \sum_I ||S_{full}(I; \theta_s) - T(I)||_2. \quad (1)
\]
We use images from the OpenImageV4 dataset to train different student models. Note that the training where $L$ also requires the least amount of computation, with only one-fourth of the FLOPs used by the teacher $S$ model while obtaining up to 93.8% of the performance.

Table 3: FLOPs, number of parameters and mAP on four different benchmarks for different student models. The performance of the proposed $S_{\text{guided}}$ achieves better performance while using fewer FLOPs and model parameters comparing to two other baseline student model.

| Embeddings         | FLOPs   | # Params. | ROxO | RPar | CUB200 | Cars196 |
|--------------------|---------|-----------|------|------|--------|---------|
| Faster-RCNN        | -       | -         | 31.66 | 49.18 | 8.99   | 3.68    |
| Student - Full ($S_{\text{full}}$) | $1.49 \times 10^9$ | $8.02 \times 10^6$ | 29.85 | 45.38 | 9.89 | 5.34 |
| Student - Top ($S_{\text{top}}$)     | $1.13 \times 10^9$ | $7.93 \times 10^6$ | 32.10 | 47.58 | 11.36 | 5.81 |
| Student - Guided ($S_{\text{guided}}$) | $0.82 \times 10^9$ | $5.17 \times 10^6$ | **38.21** | **54.01** | **16.74** | **6.32** |
| Teacher (CNN-OE)   | $3.33 \times 10^9$ | $8.54 \times 10^6$ | 40.84 | 57.58 | 23.57 | 9.41 |

It is commonly believed that the shallower layers in convolutional neural networks learn common low-level features such as edges which can be useful for all visual tasks. Since we already compute these low-level features in the detection model, we can reuse them for training the student model.

The detection model’s backbone network is represented as $(D_{l4} \circ D_{l3} \circ D_{\text{lower}})(\cdot)$, where $D_{\text{lower}}: R^{W \times H \times 3} \rightarrow R^{W \times \frac{H}{4} \times \frac{C}{8}}$, denotes the lower layers in the network; $D_{l3}: R^{W \times \frac{H}{4} \times \frac{C}{8}} \rightarrow R^{W \times \frac{H}{8} \times \frac{C}{4}}$, and $D_{l4}: R^{W \times \frac{H}{8} \times \frac{C}{4}} \rightarrow R^{W \times \frac{H}{16} \times \frac{C}{2}}$ are the higher layers. We consider a student model $S_{\text{top}}: R^{W \times \frac{H}{4} \times \frac{C}{8}} \rightarrow R^{W \times \frac{H}{8} \times \frac{C}{4}}$ that only contains the top layers (cf. Figure 3(b)). Reusing the lower layers from the detection network $D_{\text{lower}}: R^{W \times H \times 3} \rightarrow R^{W \times \frac{H}{4} \times \frac{C}{8}}$, the mean squared error between the output feature maps is minimized:

$$\text{minimize}_{\theta_s} \sum_{i} ||S_{\text{full}}(D_{\text{lower}}(I); \theta_s) - T(I)||_2.$$  

Lastly, we propose a guided student model $S_{\text{guided}}: (R^{W \times \frac{H}{4} \times \frac{C}{8}}, R^{W \times \frac{H}{8} \times \frac{C}{4}}, R^{W \times \frac{H}{16} \times \frac{C}{2}}) \rightarrow R^{W \times \frac{H}{8} \times \frac{C}{4}}$ that uses multi-scale feature maps from the detection backbone network as guidance to learn discriminative embeddings (cf. Figure 3(c)), with each layer $L_i$ of $S_{\text{guided}}$ defined as:

$$y_i = L_i(y_{i-1} + g_{i-1}),$$

where $L_i$ is a bottleneck layer, $y_i$ is the output of layer $i$, and $g_1, g_2, g_3$ are the guidance inputs from the detection backbone network with $y_0 = g_1$ and $g_0 = 0$. Here, we assume the guidance has the same dimension as the layer output of the student model. For different dimensions, a linear transformation is applied to map them into the same space. Finally, we minimize the mean squared error between the output of the student model $S_{\text{guided}}$ and the teacher model $T$:

$$\text{minimize}_{\theta_s} \sum_{i} ||S_{\text{guided}}(D_{\text{lower}}(I), (D_{l3} \circ D_{\text{lower}})(I), (D_{l4} \circ D_{l3} \circ D_{\text{lower}})(I); \theta_s) - T(I)||_2.$$  

Student model with multi-scale guidance can utilize both high-level and low-level features learned in the detection model. As shown in Section 5.1, this is essential for learning discriminative features.

5 Experimental Details

5.1 Experiment with different Student Networks

We use images from the OpenImageV4 dataset to train different student models. Note that the training of the student model is unsupervised and does not require any manual annotations. We use Adam optimizer with a learning rate of 1-e3 and batch size of 64 to train all the student models for 20,000 iterations. Table 3 shows the performance of different student models in term of mAP. $S_{\text{full}}$ achieves the worst performance and it struggles to learn discriminative embeddings. $S_{\text{top}}$ achieves slightly better performance than $S_{\text{top}}$ by reusing the low-level feature maps from the detection model. Utilizing the guidance from multi-scale feature maps of the detection model, our guided student model $S_{\text{guided}}$ obtains the best performance. Note that the proposed guided student model actually also requires the least amount of computation, with only one-fourth of the FLOPs used by the teacher model while obtaining up to 93.8% of the performance.
Table 4: Comparison with state-of-the-art approaches on CUB200 and Cars196. Object embeddings from ImageNet pre-trained model (CNN-OE) obtain competitive results on CUB200. By fine-tuning on the training set corresponding to each benchmark (CNN-FT-OE), we can achieve state-of-the-art retrieval performance. With PCA dimensionality reduction to embedding size of 512 (CNN-FT-OE + PCA), our approach still maintains competitive results among SOTAs.

| Method    | Network       | Dimension | CUB200 mAP | CUB200 P@1 | Cars196 mAP | Cars196 P@1 |
|-----------|---------------|-----------|------------|------------|-------------|-------------|
| ProxyNCA  | Inception BN  | 64        | -          | 49.2       | -           | 73.2        |
| Angular Loss | GoogLeNet     | 512       | -          | 54.7       | -           | 71.4        |
| Margin Loss | ResNet50     | 128       | -          | 63.6       | -           | 79.6        |
| HTL       | Inception BN  | 512       | -          | 57.1       | -           | 81.4        |
| CNN-OE    | ResNet50     | 2048      | 20.3       | 56.5       | 6.4         | 49.2        |
| CNN-FT-OE | ResNet50     | 2048      | 23.6       | 61.6       | 9.4         | 59.9        |
| CNN-FT-OE + PCA |           | 512       | 23.2       | 67.6       | 18.8        | 81.9        |

5.2 Comparison with State-of-the-Art (SOTA)

Table [4] compares the proposed method with state-of-the-art embedding learning approaches on CUB200 and Cars196 image retrieval datasets. We compare several embedding learning approaches including ProxyNCA [36], Angular Loss [55], Margin Loss [57], and Hierarchical Triplet Loss (HTL) [14]. Note that it is hard to fairly compare different methods as they use different network architecture or embedding dimension. Nevertheless, we show the precision at one (P@1) of the proposed method to provide insights into how it compares with state of the art. Further note that while the student models achieve better efficiency, the performance does understandably degrade as shown in Table [3] when compared to the teacher model. For the strongest comparison, we therefore use the teacher model here.

For clarity, we list all the network architectures used by different approaches and use PCA dimensionality reduction to reduce the dimension of the embeddings of our approach to 512. By using object embeddings from ImageNet pretrained model (CNN-OE), we can achieve 61.6% precision on CUB200, which is already quite competitive against state-of-the-art embedding learning approach. For a fair comparison, we note that the SOTA methods have all been trained on the training sets of CUB200 and Cars196, while CNN-OE is simply using the weights of ImageNet classification model. For this reason, we also fine-tune the classification model with the training set corresponding to each benchmark (CNN-FT-OE), after which we are able to achieve the state-of-the-art performance of 68.2% on CUB200. For Cars196, CNN-OE improves performance of CNN from 49.2% to 59.9%, but it is still lower than the state-of-the-art performance. With fine-tuning and dimensionality reduction, we also achieve state-of-the-art performance on this benchmark. It is interesting to note that PCA dimensionality reduction actually improves the performance of our approach on Cars196.

Table [5] compares different landmark retrieval approaches including sum pooling of convolutions (CNN-SPoC) [5], maximum activation of convolutions (CNN-MAC) [42], regional maximum activation (CNN-R-MAC) [54], and generalized mean pooling (CNN-GeM) [41] on R_Oxford5K and R_Paris6K dataset. It is hard to directly compare with the SOTA results for these two datasets in [39], which employ ResNet101, and an array of post-processing steps such as query expansion that is conducted at system-level, discriminative whitening, multi-scale inference, as well as fine-tuning with landmark datasets. To ensure a fair comparison, the evaluations are done as follow. We employ ResNet50 trained on ImageNet for all the methods. We then add on top the different approaches (SPoC, MAC, R-MAC, GeM). We also do not conduct any post-processing steps. Our approach (CNN-OE) achieves the best performance among other approaches using the same pre-trained network; more importantly, our approach still maintains competitive results using the compact student network (CNN-OE-S_{guided}) described in Section [4].
Table 5: (Left) Comparison of different approaches on ROxford5k and RParis6k datasets. Our approach achieves the best performance among other baselines even when a compact student model is deployed. (Right) performance on PhotoShop Image Retrieval (PIR) dataset. Our approach is especially suitable for retrieving tampered images with spliced objects.

| Method                  | ROxford5K | RParis6K | PIR     |
|-------------------------|-----------|----------|---------|
|                         | mAP @10   | mAP @10  | mAP @10 |
| CNN-SPoC [5]            | 23.33     | 40.71    | 40.84   |
| CNN-MAC [42]            | 35.52     | 56.14    | 46.89   |
| CNN-R-MAC [54]          | 36.99     | 57.34    | 44.21   |
| CNN-GeM [41]            | 33.49     | 53.55    | 44.40   |
| CNN-OE-S guided (Ours)  | 38.21     | 59.57    | 52.16   |
| CNN-OE (Ours)           | 40.84     | 65.71    | 54.06   |

Figure 4: Example images and rank one results in the PIR dataset. PIR contains 3,278 query images and 60,550 tampered images derived from the query images. First and the fourth column shows query images, second and fifth column show rank-1 result from CNN-MAC; third and sixth shows rank-1 result from CNN-OE. Red border indicates incorrect matches and yellow bounding box shows the matching objects.

5.3 Near-Duplicate Object Retrieval

One interesting capability of our proposed approach is in retrieving near-duplicate objects in images. Having demonstrated that our approach works well in image retrieval, the same rationale that object regions help avoid the influence of background clutter should also apply. This capability has an important application in detecting tampered images that contain spliced objects [61], where a given image can be queried against a repository of images to detect near-duplicate object associations. Due to the proliferation of social media platforms, such application is becoming increasingly important, where it has been shown that there is a strong correlation between tampered images and the spread of misinformation.

To demonstrate the effectiveness of our approach for near-duplicate object retrieval, we construct a benchmark that we refer to as Photoshop Image Retrieval dataset (PIR). The images are collected from the publicly available PS-Battles dataset [19] by selecting 3,278 original images as queries and 60,550 tampered images as the database. Each query has at least ten tampered versions in the database. Table 5 shows retrieval results compared to different image retrieval methods. Our approach achieves better performance because it can retrieve small spliced objects as a result of the hard attention provided by the detection model. Figure 4 shows some examples of the retrieval result. The first and the fourth columns are the query images, second and fifth columns show rank-1 retrieved results by CNN-MAC. CNN-MAC retrieves images with similar scenes but fails to retrieve tampered images that contain the spliced objects from the query image. The third and sixth columns show the rank-1 results retrieved by CNN-OE.
6 Conclusion

We provided analysis of embeddings learned from different models and demonstrated that embeddings learned from detection models are less discriminative than their classification counterpart. Based on our analysis, we proposed an approach that uses detection as a source of hard attention to improve retrieval performance. Our experimental results showed that the proposed approach achieves state-of-the-art performance on different retrieval benchmarks. For applications with efficiency requirement, we have also introduced a student-teacher training regime that only needs a single forward pass during inference. Lastly, we also show how our approach can be applied towards near-duplicate object retrieval.

References

[1] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6077–6086, 2018.
[2] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5297–5307, 2016.
[3] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson. Factors of transferability for a generic convnet representation. IEEE transactions on pattern analysis and machine intelligence, 38(9):1790–1802, 2016.
[4] J. Ba and R. Caruana. Do deep nets really need to be deep? In Advances in neural information processing systems, pages 2654–2662, 2014.
[5] A. Babenko and V. Lempitsky. Aggregating local deep features for image retrieval. In Proceedings of the IEEE international conference on computer vision, pages 1269–1277, 2015.
[6] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. In European conference on computer vision, pages 584–599. Springer, 2014.
[7] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil. Model compression. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 535–541. ACM, 2006.
[8] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker. Learning efficient object detection models with knowledge distillation. In Advances in Neural Information Processing Systems, pages 742–751, 2017.
[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In international Conference on computer vision & Pattern Recognition (CVPR’05), volume 1, pages 886–893. IEEE Computer Society, 2005.
[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.
[11] P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. IEEE transactions on pattern analysis and machine intelligence, 36(8):1532–1545, 2014.
[12] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In International conference on machine learning, pages 647–655, 2014.
[13] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. IEEE transactions on pattern analysis and machine intelligence, 32(9):1627–1645, 2009.
[14] W. Ge. Deep metric learning with hierarchical triplet loss. In Proceedings of the European Conference on Computer Vision (ECCV), pages 269–285, 2018.
[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 580–587, 2014.
[16] A. Gordo, J. Almazán, J. Revaud, and D. Larlus. Deep image retrieval: Learning global representations for image search. In European conference on computer vision, pages 241–257. Springer, 2016.
[17] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 2961–2969, 2017.
[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–
778, 2016.
[19] S. Heller, L. Rossetto, and H. Schuldt. The PS-Battles Dataset – an Image Collection for Image Manipulation Detection. CoRR, abs/1804.04866, 2018.
[20] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015.
[21] M. Huh, P. Agrawal, and A. A. Efros. What makes imagenet good for transfer learning? arXiv preprint arXiv:1608.08614, 2016.
[22] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.
[23] H. Jégou and O. Chum. Negative evidences and co-occurrences in image retrieval: The benefit of pca and whitening. In European conference on computer vision, pages 774–787. Springer, 2012.
[24] H. Jégou, M. Douze, and C. Schmid. Improving bag-of-features for large scale image search. International journal of computer vision, 87(3):316–336, 2010.
[25] H. Jegou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. IEEE transactions on pattern analysis and machine intelligence, 33(1):117–128, 2011.
[26] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.
[27] S. Kornblith, J. Shlens, and Q. V. Le. Do better imagenet models transfer better? arXiv preprint arXiv:1805.08974, 2018.
[28] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In 4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13), Sydney, Australia, 2013.
[29] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Mallocci, T. Duerig, and V. Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. arXiv:1811.00982, 2018.
[30] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2117–2125, 2017.
[31] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In European conference on computer vision, pages 740–755. Springer, 2014.
[32] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen. Deep learning for generic object detection: A survey. arXiv preprint arXiv:1809.02165, 2018.
[33] D. G. Lowe. Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60(2):91–110, 2004.
[34] I. Melekhov, A. Tiulpin, T. Sattler, M. Pollefeys, E. Rahtu, and J. Kannala. Dgc-net: Dense geometric correspondence network. arXiv preprint arXiv:1810.08393, 2018.
[35] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas. Working hard to know your neighbor’s margins: Local descriptor learning loss. In Advances in Neural Information Processing Systems, pages 4826–4837, 2017.
[36] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh. No fuss distance metric learning using proxies. In Proceedings of the IEEE International Conference on Computer Vision, pages 360–368, 2017.
[37] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han. Large-scale image retrieval with attentive deep local features. In Proceedings of the IEEE International Conference on Computer Vision, pages 3456–3465, 2017.
[38] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4004–4012, 2016.
[39] F. Radenović, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5706–5715, 2018.
[40] F. Radenović, G. Tolias, and O. Chum. Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In European conference on computer vision, pages 3–20. Springer, 2016.
[41] A. S. Razavian, J. Sullivan, S. Carlsson, and A. Maki. Visual instance retrieval with deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence, 2018.
[42] A. S. Razavian, J. Sullivan, and A. Carlsson. Visual instance retrieval with deep convolutional networks. Ite Transactions on Media Technology and Applications, 4(3):251–258, 2016.
[43] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016.
[44] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015.
[45] I. Rocco, R. Arandjelovic, and J. Sivic. Convolutional neural network architecture for geometric matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6148–6157, 2017.
[46] I. Rocco, R. Arandjelović, and J. Sivic. End-to-end weakly-supervised semantic alignment. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6917–6925, 2018.
[47] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fittnets: Hints for thin deep nets. arXiv preprint arXiv:1412.6550, 2014.
[48] Y. Rui, T. S. Huang, and S.-F. Chang. Image retrieval: Current techniques, promising directions, and open issues. Journal of visual communication and image representation, 10(1):39–62, 1999.
[49] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pages 806–813, 2014.
[50] B. Singh, M. Najibi, and L. S. Davis. Sniper: Efficient multi-scale training. In Advances in Neural Information Processing Systems, pages 9333–9343, 2018.
[51] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In null, page 1470. IEEE, 2003.
[52] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1–9, 2015.
[53] M. Teichmann, A. Araujo, M. Zhu, and J. Sim. Detect-to-retrieve: Efficient regional aggregation for image search. arXiv preprint arXiv:1812.01584, 2018.
[54] G. Tolias, R. Sicre, and H. Jégou. Particular object retrieval with integral max-pooling of cnn activations. arXiv preprint arXiv:1511.05879, 2015.
[55] J. Wang, F. Zhou, S. Wen, X. Liu, and Y. Lin. Deep metric learning with angular loss. In Proceedings of the IEEE International Conference on Computer Vision, pages 2593–2601, 2017.
[56] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.
[57] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krahenbuhl. Sampling matters in deep embedding learning. In Proceedings of the IEEE International Conference on Computer Vision, pages 2840–2848, 2017.
[58] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In Advances in neural information processing systems, pages 3320–3328, 2014.
[59] X. Zhang, F. X. Yu, S. Kumar, and S.-F. Chang. Learning spread-out local feature descriptors. In Proceedings of the IEEE International Conference on Computer Vision, pages 4595–4603, 2017.
[60] L. Zheng, Y. Yang, and Q. Tian. Sift meets cnn: A decade survey of instance retrieval. IEEE transactions on pattern analysis and machine intelligence, 40(5):1224–1244, 2018.
[61] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis. Learning rich features for image manipulation detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1053–1061, 2018.