Efficient Object Embedding for Spliced Image Retrieval

Bor-Chun Chen[1,2] Zuxuan Wu[3*] Larry S. Davis[1] Ser-Nam Lim[2]

[1]University of Maryland, College Park [2]Facebook AI [3]Fudan University

{sirius, lsd}@cs.umd.edu, zxwu@fudan.edu.cn, sernamlim@fb.com

Abstract

Detecting spliced images is one of the emerging challenges in computer vision. Unlike prior methods that focus on detecting low-level artifacts generated during the manipulation process, we use an image retrieval approach to tackle this problem. When given a spliced query image, our goal is to retrieve the original image from a database of authentic images. To achieve this goal, we propose representing an image by its constituent objects based on the intuition that the finest granularity of manipulations is oftentimes at the object-level. We introduce a framework, object embeddings for spliced image retrieval (OE-SIR), that utilizes modern object detectors to localize object regions. Each region is then embedded and collectively used to represent the image. Further, we propose a student-teacher training paradigm for learning discriminative embeddings within object regions to avoid expensive multiple forward passes. Detailed analysis of the efficacy of different feature embedding models is also provided in this study. Extensive experimental results show that the OE-SIR achieves state-of-the-art performance in spliced image retrieval.

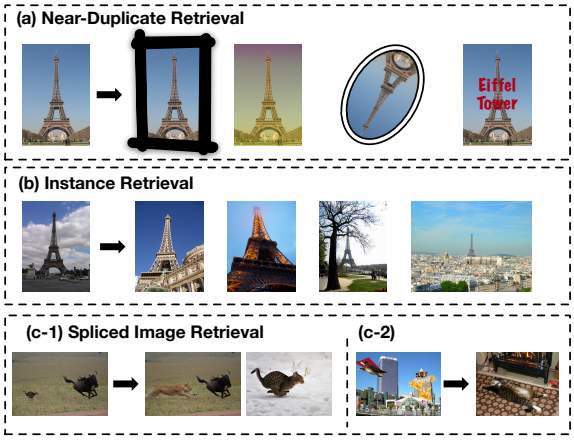

Figure 1. Three different types of image retrieval tasks. (a) The traditional image retrieval algorithm tries to retrieve near-duplicate images using low-level image statistics. (b) Instance retrieval tries to retrieve the same instance (e.g. building) under different viewpoint, illumination, and occlusion. (c) Spliced image retrieval (SIR) tries to retrieve authentic images used to create the spliced query image. Results from SIR contain images with large variation, so it is difficult to learn a single embedding that is suitable for the SIR task.

1. Introduction

With the proliferation of social media platforms and the availability of user-friendly image editing software, adversaries can now easily share spliced images on the Internet and reach millions of people with malicious intent to spread misinformation, disrupt democratic processes, and commit fraud. The ability to detect such spliced images is thus an increasingly important research area. Most existing work learns a mapping function between a spliced image and its corresponding label map, where each pixel in the map denotes whether the pixel has been modified or not [6, 28, 59, 76]. However, such training strategies require dense pixel-level annotations, which are expensive to obtain and thus prevent their abilities to scale. In this paper, we formulate splicing detection as an image retrieval task: given a spliced query image and a large-scale image database, our goal is to retrieve images in the database that are authentic versions of the query image. We describe this as **Spliced Image Retrieval (SIR)** problem. Once the original images are retrieved, we can then localize the spliced regions in these images by comparing the query and the retrieved images.

In contrast to traditional image retrieval which usually focuses on retrieving near-duplicate images or images containing specific instances, SIR focuses on retrieving authentic images that were used to create the query. Figure 1 shows examples of near-duplicate retrieval, instance retrieval, and SIR. As shown in the figure, SIR faces a different set of challenges compared to near duplicate or instance retrieval task. First, the query image may con-

*Work done during author was in Facebook AI.

tain both manipulated as well as non-manipulated regions (cf. Figure 1 (c-1)). When comparing with images in the database, the query image should be region-specific rather than using the entire image as in many current image retrieval systems [5, 21, 49, 22]. Secondly, retrieved images may not overlap and query expansion [12], a common practice in this research area, does not apply. Using Figure 1 (c-1) as an example, if we were to use the first image to do query expansion, we might retrieve more images of similar lions and horses but we will not be able the retrieve the image of the cat running in the snow since two images do not share any overlapping content. Third, the query image and the authentic image might have extreme diverse backgrounds (cf. Figure 1 (c-2)), which may cause traditional image retrieval algorithms to fail.

To mitigate these issues, we take advantage of recent advances in object detection [14, 53, 23] and propose Object Embeddings for Spliced Image Retrieval (OE-SIR). Instead of using a single global representation, OE-SIR generates object-level representations per object region that is then collectively used to represent the image. By representing an image at object-level granularity, and assuming that image manipulations are frequently done by manipulating objects (e.g., faces, logos, etc.), we can extract similar object embeddings from both spliced and authentic images, achieving the purpose of SIR.

Given detected objects, the next challenge lies in deriving robust feature representations for the task of spliced image retrieval. It is appealing to directly use features from detection networks as such embeddings are trained with additional location information and it is computationally efficient with a single forward pass. However, while extensive studies have been conducted on image retrieval, most of them were focused only on embeddings provided by classification networks [5, 1, 64, 4, 21, 72]. In light of this, we provide a detailed analysis of embeddings derived from different pre-trained object detectors, and compare them with image classification models. Our analysis shows that even though the object detection networks are trained with additional annotations, the resulting embeddings are significantly worse than those from classification models for image retrieval. This suggests a computationally expensive two-step process for SIR—detecting objects with object detectors and then encode them with pretrained classification models.

We propose a student-teacher training regime to explore the best of both worlds for computational efficiency, *i.e.*, reliable bounding boxes produced by detectors and discriminative features computed with classification models. This is achieved by training a lightweight student network on top of the detection model that projects feature maps of the detection model into a more discriminative feature space guided by the teacher model. The student network decouples feature learning from localization, preserving the discriminative power of the features for classification.

The contributions of this work include: (1) We introduce the task of spliced image retrieval and propose OE-SIR that derives object-level embeddings. (2) We provide a detailed analysis of embeddings extracted from different pre-trained models and show that embeddings extracted from object detection models are less discriminative than those from image classification models. (3) We show that OE-SIR can outperform traditional image retrieval baseline and achieve significantly better results with two SIR datasets. (4) OE-SIR demonstrates state-of-the-art performance in detecting spliced regions by utilizing the original image.

2. Background and Related Work

Image forensics. Finding manipulated images is an important topic in media forensics research. Traditional approaches [70, 40, 18] usually focus on finding low-level artifacts in the manipulated images. Recently, with the success of deep learning in computer vision, many people also turn to deep learning algorithms [76, 28, 6, 59, 69] to detect manipulated images. Specifically, Zhou *et al*. [76] utilize object detection framework [53] to detect manipulated region in images. In contrast to their approach, we use an object detection framework for learning object embeddings and use the embeddings to retrieve spliced images. Most previous approaches try to detect manipulation from a single image. There are a few recent studies focusing on provenance analysis [43], constructing a graphical relationship of all manipulated images. However, these approaches usually treat detecting manipulated content as a segmentation task, which requires dense annotations as supervision. In this paper, we formulate the problem as a retrieval task without the need to use pixel-level annotations.

Content-based image retrieval. Image retrieval aims to identify relevant images from an image database given a query image based on the image content. Early work [57] 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 features [39] and indexing methods [62, 30, 31] achieved robustness against illumination and geometric variations. With the recent broad adoption of convolutional neural networks (CNN), different techniques have been proposed for global feature extraction [5, 4, 64, 1, 21, 49, 22], local feature extraction [45, 42, 73], embedding learning [44, 68, 65, 19], as well as geometric alignment [54, 55, 41] using deep networks. Zheng *et al*. [74] 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 and it

is aiming for SIR.

Representation learning from large-scale datasets. Previous works mainly studied the transferability of embeddings extracted from classification models that have been trained on datasets such as ImageNet to other tasks [15, 60, 71, 27, 2, 33]. For instance, [60] reports comprehensive results of applying embeddings from the ImageNet-trained classification model to object detection, scene recognition, and image retrieval. In contrast, the efficacy of embeddings obtained from object detection models trained on large-scale datasets such as COCO [37] and OpenImages [35] 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 the retrieval task.

Object detection aims to detect different objects in an input image. Girshick *et al*. [20] proposed one of the first deep learning based object detection models, R-CNN, which improved the accuracy significantly compared to traditional methods [13, 17, 14]. Since then many enhancements [53, 52, 36, 61] have been made to improve the accuracy as well as the training/inference time. Teichmann *et al*. [63] utilized a specialized landmark detection model to aggregate deep local features [45] for landmark retrieval. A comprehensive survey of recent deep learning based object detection methods can be found in [38]. By taking advantage of recent success in object detection, our model can learn discriminative object-level embeddings for image retrieval. Joint detection and feature extraction has recently been used for person search tasks [9, 16]. However, these approaches requires annotations of bounding boxes as well as fine-grained person identities in the boxes. Therefore, these approaches can not directly apply to our task.

Knowledge distillation [7, 3, 26, 56, 10] compress a complex model into a simpler one while maintaining the accuracy of the model. Bucilua *et al*. [7] first proposed to train a single model to mimic the outputs of an ensemble of models. Ba *et al*. [3] adopted a similar idea to compress deep neural networks. Hinton *et al*. [26] 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 to provide additional guidance during training, which we show is effective for training the student network.

3. Method

Given an image, our goal is to learn a feature embedding which models the image at the object-level such that it can be used to detect whether an object is spliced. Figure 2 shows the overview of our proposed OE-SIR framework. First, an object index is built with all available authentic images using the object embedding network described in Section 3.1. When a query image arrives, an object embedding is computed with the same network, and then used to perform an object-level nearest neighbor search to retrieve authentic images. Finally, by comparing the query image with retrieved authentic images, we can localize the spliced region as described in Section 3.2.

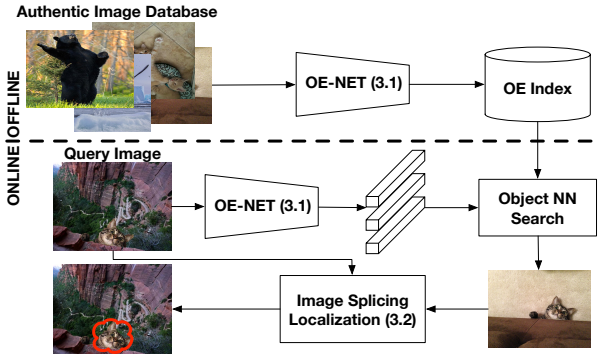

Figure 2. Overview of the proposed OE-SIR framework. A set of authentic images go through the proposed object embedding network (OE-NET) to build an object index offline. When a query image comes in, it first goes through the same network to extract object embeddings, and these embeddings are used to retrieve the authentic image with the object-level nearest neighbor search. Once the original image is retrieved, we can compare it with the query images to localize the spliced region.

3.1. Object Embedding Network (OE-NET)

Object Detection and Feature Extraction Model. The first step of OE-SIR is to train an object detection model M_o and a feature extraction model M_f:

$$B, S = M_o(I), \qquad C = M_f(I), \tag{1}$$

where $B \in R^{n\times 4}$ denotes the bounding box coordinates for n predicted objects in an image I, $S \in N^n$ is the object index, and $C \in R^{w\times h\times d}$ is a convolution feature map. In addition, w, h, d is the width, height and the number of channels of the feature map. We adopt the Faster-RCNN [53] object detection framework by minimizing the following multi-task loss during training for M_o:

$$L(\{p_i\},\{ti\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \tag{2}$$

$$\lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*), \tag{3}$$

where L_{reg} is the bounding box regression loss and L_{cls} is the classification loss, p_i, p_i^* are the predict class label and ground truth label; t_i, t_i^* are the predict box label and ground truth. The loss is minimized with SGD on standard detection datasets.

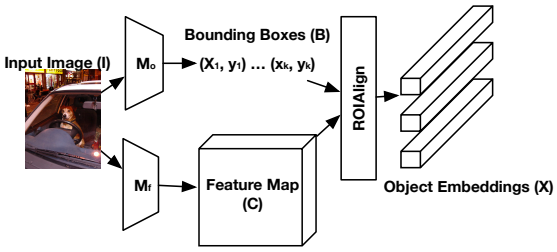

Figure 3. Configuration of the proposed object embedding network. The image first goes through object detector (M_o) to extract object bounding boxes, and a separate feature network (M_f) to extract discriminative feature map. We then use the detected bounding boxes to extract object embeddings from the feature map with ROIAlign layer.

For M_f, since we do not have additional training data available, we utilize a pre-trained image classification model (*i.e.*, ResNet [24]) as our feature extraction network. We provide a detailed analysis of how we select our feature extraction model in Section 4.1. After both detection and feature extraction model are trained, we can extract object-level embeddings using the ROIAlign layer [23] with B as hard attention over the feature map C:

$$X = ROIAlign(C, B), \tag{4}$$

where $X \in R^{(n \times d)}$ are object embeddings of the image, which contain n predicted objects.

While it is possible to use a shared model for both object detection and feature extraction, we find that training two separate models provides many benefits. First, as we show in Section 4.1, jointly learning classification and localization reduces the discriminative power of the embeddings. Therefore, separate models ensure that we have better embeddings for retrieval. Second, since the detection model and the feature extraction model are independent, we can change the feature extraction model for a different task without retraining the object detection model. However, despite these advantages, such a two-step process is computationally expensive, requiring two forward passes for the same image. This limits the deployment of such models in resource-constrained environments such as mobile devices, robots, *etc*.

Efficient Object Embeddings Extraction. We now introduce how to use a single model which explores the best of both worlds—robust bounding box detection with object detectors and discriminative feature computation with a classification model—such that given an image, object-embeddings can be efficiently computed with a single forward pass. Towards this goal, we use knowledge distillation [26] to save computation. One straightforward way is to train a student network completely from scratch to mimic the outputs of the classification model. However, this defeats the purpose of using a single model and neglects useful information from the backbone of the detector. Instead, we introduce a guided framework by using features from the detection backbone to train a student model.

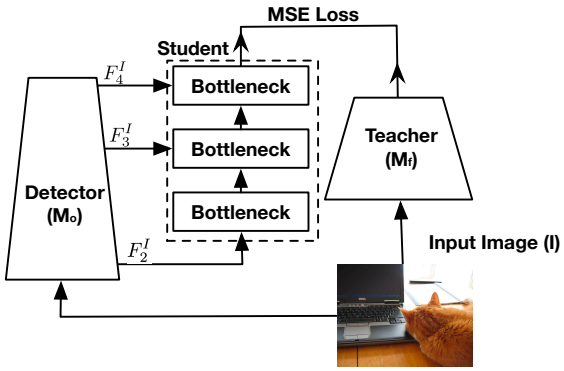

Figure 4. Knowledge distillation with the student network. Lightweight student network (f) utilize the information from detector backbone to learn discriminative feature map from the teacher network with mean square loss. Once the student model is trained, we can use single forward pass to extract discriminative object embeddings.

More specifically, we consider the classification model as a teacher and we attach a lightweight branch to the detector as a student model to approximate outputs from the teacher model. The reasons for using a parallel branch as the student network are two-fold: (1) we wish to reuse information from detector backbones to guide the training of the student model; (2) the student network can produce discriminative features for retrieval task while preserving the ability of the main detector branch to generate accurate bounding boxes.

Formally, given an image I, the student network $f(F_2^I, F_3^I, F_4^I; \theta_s)$ is a lightweight model parameterized by the weight θ_s. It takes as inputs feature maps after the second residual stage of the detector model F_i^I to approximate the outputs of the classification model $M_f(I)$ with three bottleneck convolutional layers. We denote the outputs of the i-th ($i \in \{1, 2, 3\}$) bottleneck-layer as y_i. To effectively leverage information from the multi-scale feature maps of the detector model, before feeding y_i to the next layer, we modify it in an addition manner:

$$y_i = y_i + F_{i+2}^I, \quad \text{if } 1 \leq i \leq 2, \tag{5}$$

where F_{i+2}^I denotes the outputs of the feature maps of the 3-rd ($i = 1$) and 4-th ($i = 2$) residual stage of the detector. 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, the outputs of the student network are used to approximate the classification model by mini-

mizing the following mean-squared loss:

$$\min_{\theta_s} \sum_{I} ||f(F_2^I, F_3^I, F_4^I; \theta_s) - M_f(I)||_2. \quad (6)$$

Since the student model is optimized with multi-scale guidance from the detector, it can utilize both high-level and low-level features to learn discriminative features efficiently.

3.2. Spliced Image Retrieval and Localization

Given a query image represented by n object embeddings $I_q = \{x_1^q, x_2^q, ..., x_n^q\}$, and a database image $I_d = \{x_1^d, x_2^d, ..., x_m^d\}$, the distance between the image pair is calculated as the minimum distance between pairwise object embeddings:

$$D(I_q, I_d) = \min_{i,j} ||x_i^q - x_j^d||_2^2. \quad (7)$$

Note that while we are using multiple embeddings per image, m and n are usually really small and we can use quantization and indexing [58] to speed up the retrieval process. For application with 100 million images and 32 bytes per embeddings with average eight embeddings per image, the total storage requirement is only around 24G, which can be easily stored in the memory of a single server.

Localization. We use Eq. 7 to retrieve the top-1 database image, I_d^*, which gives us a matching object pair:

$$(i, j) = \arg\min_{i,j} ||x_i^q - x_j^d||_2^2. \quad (8)$$

We can then use the bounding box (b_i, b_j) information to estimate the geometric transformation T between the image pair [47]. The final localization map Q can then be generated by comparing convolutional feature map:

$$Q = ||M_f(T(I_q)) - M_f(I_d^*)||_2^2. \quad (9)$$

After comparison, we apply a threshold to Q to generate a binary mask for localization.

4. Experimental Results

4.1. Feature Extraction Model

Harvesting data and annotations for splicing detection is expensive, and thus OE-SIR is built upon *pre-trained* models that are widely available. Unlike specific tasks such as landmark retrieval [48], where there are usually additional sets of landmark images to train a good model, it is imperative that we select a good feature extraction model that provides discriminative embeddings. Here, we first provide a detailed analysis of embeddings extracted from different pre-trained models trained on image classification, object detection, and instance segmentation models using four common image retrieval benchmarks.

Model	Train Set (# of Img. / Cls.)	$\mathcal{R}$Oxf	$\mathcal{R}$Par	CUB200	Cars196
Faster-RCNN [53]	COCO (330K / 80)	18.7	28.3	4.1	2.4
Faster-RCNN-FPN [36]		20.4	31.0	3.3	3.1
Mask-RCNN [23]		20.7	33.0	3.0	2.4
Mask-RCNN-FPN [36]		34.2	48.1	2.9	3.6
ResNet50 [24]	ImageNet (1.2M / 1K)	**40.1**	**57.3**	**21.2**	**11.1**
Faster-RCNN [53]	OpenImagesV4 (1.7M / 601)	19.5	32.3	4.7	2.2
ResNet50 [24]		**41.2**	**61.2**	**19.3**	**11.0**

Table 1. Image retrieval performance (mAP) with embeddings extracted from different pre-trained models for four different retrieval benchmarks. Even though all detection and instance 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.

Retrieval benchmark. We consider four datasets for benchmarking, including USCB bird dataset [66] (**CUB200**), Stanford car dataset [34] (**Cars196**), and two landmark datasets, $\mathcal{R}$Oxford5K [48] (**$\mathcal{R}$Oxf**) and $\mathcal{R}$Pairs6K [48] (**$\mathcal{R}$Par**). For CUB200 and Cars196, we follow the same protocol in [46] and use leave-one-out partitions to evaluate every image in the test set. For $\mathcal{R}$Oxford5K and $\mathcal{R}$Paris6K, we follow the medium protocol described in [48], 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 seven different pre-trained models including (1) **Faster-RCNN** [53] and (2) Faster-RCNN with feature pyramid networks [36] (**Faster-RCNN-FPN**) trained on COCO [37], (3) **Mask-RCNN** [23] and (4) Mask-RCNN with feature pyramid networks (**Mask-RCNN-FPN**) trained on COCO with bounding box and mask annotations, and (5) **ResNet50** [24] trained on ImageNet. To control the effect of different training data, we also compare with (6) **Faster-RCNN** and (7) **ResNet50** trained with the same dataset (OpenImagesV4 [35]). We adopt open source implementation[1] of Faster-RCNN and Mask-RCNN with ResNet50 as a backbone feature extractor for all our detection and segmentation models. For all Faster-RCNN and Mask-RCNN models, we use weights from the ImageNet classification model to initialize the backbone network and use the default $3\times$ learning rate schedule to train the models. We use images from OpenImagesV4 to learn a PCA whitening matrix for post processing [29].

During test time, we first resized the image to a maximum size of 1024×1024, followed by extracting features from conv5_3 layer [24] and using max-pooling to produce image embeddings from different pre-trained models. We

[1] https://github.com/facebookresearch/detectron2

	FLOPs	# Params.	$\mathcal{R}$Oxf	$\mathcal{R}$Par
Faster-RCNN	-	-	25.4	34.4
S_1	1.49×10^9	8.02×10^6	32.1	55.2
S_2	1.13×10^9	7.93×10^6	43.3	56.3
S_3	1.13×10^9	7.93×10^6	**50.2**	**65.2**
Teacher (ResNet50)	3.33×10^9	8.54×10^6	53.4	69.7

Table 2. FLOPs, number of parameters and mAP for different student models. The performance of the proposed S_3 achieves better performance while using fewer FLOPs and model parameters comparing to two other baseline student model.

then use cosine similarity between embeddings for retrieval ranking. We do not apply any post-processing tricks such as multi-scale ensemble and query expansion except PCA whitening.

Embeddings comparison. 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 decrease the performance of the embeddings on some of the dataset, suggesting that localization constraints could 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 but degrades on other datasets. 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 comparing Faster-RCNN (OpenImages) with ResNet50 (OpenImages) which are trained with the same training data, but with Faster-RCNN utilizing more human annotations (*i.e.*, bounding boxes), embeddings learned from the classification model still significantly outperform embeddings learned from the detection model. 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 as we mentioned in Section 3.1. Based on the analysis, we select the ResNet50 classification model as our feature extraction model for SIR.

Note that different spatial pooling techniques [48] and post-processing steps such as dimensionality reduction [29] have been shown to greatly affect retrieval performance. Furthermore, embeddings from different layers of the network also perform differently. We provide detailed analysis in the supplementary material for selecting these parameters.

4.2. Student Networks

We compare the student network proposed in Section 3.1 with two baseline versions: (1) S_1: Lightweight network with five bottleneck layers without any guidance information from the detector backbone. (2) S_2: Lightweight network with three bottleneck layers with taken F_2^I from the detector backbone as an input feature map. (3) S_3: proposed network with multi-scale inputs from the detector backbone. See supplementary material for an illustration of these 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 [32] optimizer with a learning rate of 1e-3 and batch size of 64 to train all the student models for 20,000 iterations. During inference, we use the minimum distance between pairwise object embedding derived from the student networks to retrieve database images. Table 2 shows the performance of different student models in terms of mAP as well as the computational cost and model parameters evaluating on the landmark dataset $\mathcal{R}$Oxf and $\mathcal{R}$Par. S_1 achieves the worst performance and it struggles to learn discriminative embeddings. S_2 achieves slightly better performance than S_1 by reusing the low-level feature maps from the detector. Utilizing the guidance from multi-scale feature maps of the detection model, our best student model is S_3, which achieves up to 93.5% of the original performance, but only requires one-third of the FLOPs used by the teacher networks. Note that mAP of the teacher model is higher than the image-level retrieval results in Section 4.1 which demonstrates the importance of utilizing object embeddings. Additional results on landmark retrieval can be found in the supplementary material.

4.3. Spliced Image Retrieval

To demonstrate the effectiveness of our approach for SIR, we conduct experiments on two different benchmarks. (1) **COCO-Fake**. COCO-Fake consists of 58 query images with spliced objects generated by the method described in [8], and 10,000 authentic images from the COCO dataset, including images that are used to create the queries. (2) Photoshop Image Retrieval dataset (**PIR**). The images are collected from the publicly available PS-Battles Dataset [25]. We use 70,389 spliced images as queries and 10,592 authentic images as the database. Since we mostly care about whether we can retrieve the correct match in the top rank, we use recall at K (R@K) as our evaluation metric, which shows the percentage of queries that have the correct match in the top K rank. Note that since only one image is expected to be retrieved per query, recall at K metric is identical to accuracy at K.

Table 3 shows retrieval results compared to different image retrieval methods using the same ImageNet feature

Method	COCO-Fake		PIR	
	R@1	R@10	R@1	R@10
SPoC [4]	29.3	34.3	43.2	46.6
MAC [51]	29.3	34.8	52.6	59.9
R-MAC [64]	37.9	42.5	51.6	58.5
GeM [50]	37.9	43.7	48.2	54.2
OE-HoG [13]	43.1	48.3	49.8	53.6
OE-FasterRCNN [53]	39.7	55.1	48.7	54.8
OE-SIR (Ours)	**70.7**	**84.5**	**58.6**	**67.7**

Table 3. Performance on COCO-Fake and PIR dataset. Our approach outperforms other baseline approaches for retrieving authentic images with spliced objects.

extraction model including (1) SPoC descriptors [4], (2) maximum activations of convolutions (MAC) [51], (3) regional maximum activation of convolutions (R-MAC), and (4) generalized mean pooling (GeM) [50]. On COCO-Fake, our model performs significantly better because all query images are with small spliced objects and the traditional image retrieval approach fails in this case. On PIR, where it contains in-the-wild spliced images, our approach still achieves better performance. Figure 5 shows some examples of the retrieval result. Figure 5 (a) are the query images; Figure 5 (b) show rank-1 retrieved results by MAC. MAC retrieves images with similar scenes but fails to retrieve original images that contain the spliced objects from the query image. Figure 5 (c) shows the rank-1 results retrieved by OE-SIR. More qualitative results including some failure cases can be found in the supplementary material. Note that while most of the examples in the dataset are spliced for entertainment purposes, spliced images can easily be used for malicious intent to spread misinformation.

Object-level retrieval. We also compare with two additional baseline methods that utilize the same object-level retrieval framework as the proposed method: (1) OE-HoG: After object detection, we extract histogram of oriented gradients [13] from each object to build object index and use object-level HoG for retrieval. (2) OE-FasterRNN: Directly using object features from the Faster-RCNN network for object-level retrieval. By utilizing the object-level search, simple handcraft features with object embeddings (OE-HoG) can achieve competitive performance compare to deep learning based image retrieval approach, which demonstrate the importance of the object-level search. On the other hand, OE-FasterRCNN performs worse than the proposed method, which also confirms the finding in Section 4.1 that jointly learning classification and localization degrades the discriminative power of the embeddings.

Number of object embeddings. Table 4 shows the performance of OE-SIR when using different numbers of object embeddings. We select up to k objects in each image based on the confidence score of the detection model. Using more embeddings results in a higher recall, however, it also requires more memory. We found that using up to 8 object embeddings per image is a good trade-off since it requires a reasonable amount of memory and increasing the number of embeddings provides little performance gain.

# of objects	PIR		
	R@1	R@10	R@100
1	54.7	62.7	69.5
2	56.1	64.6	71.1
4	57.6	66.3	72.9
8	**58.6**	67.7	74.1
16	58.4	**67.8**	**74.7**

Table 4. Performance of OE-SIR with different numbers of object embeddings by varying the detection threshold. The model with more embeddings achieves higher performance while requiring more memory storage.

Method	MCC	F1
NOI [40]	0.172	0.269
CFA [18]	0.050	0.190
RGB-N [76]	0.334	0.379
Self-Consistency [28]	0.102	0.276
GSR-Net [75]	0.439	0.489
OE-SIR (Ours)	**0.721**	**0.732**

Table 5. Image splicing localization performance. The proposed method significantly outperforms other state-of-the-art methods because our model can utilize the retrieved original image to localize the spliced regions. Note that our goal is to show the ability of image splicing localization using the retrieval approach, and our number is not directly comparable to previous methods since we utilize the original image.

4.4. Image Splicing Localization

Dataset. We show the performance of image splicing localization in a widely used image forensics dataset, COVERAGE [67] dataset. COVERAGE contains 100 spliced images with its original version generate by copy-move manipulation. The spliced objects are used to superimpose similar objects in the original authentic images and thus are challenging for humans to recognize visually without close inspection. Figure 6 (a) shows some example of images in the COVERAGE dataset.

Evaluation metrics. We use the pixel-level F1 score and MCC as the evaluation metrics when comparing to other approaches and we follow the same measurement as [75], by varying the prediction threshold to get a binary prediction mask and report the optimal score over the whole dataset.

Comparisons to state-of-the-art methods. We compare several state-of-the-art image manipulation detection al-

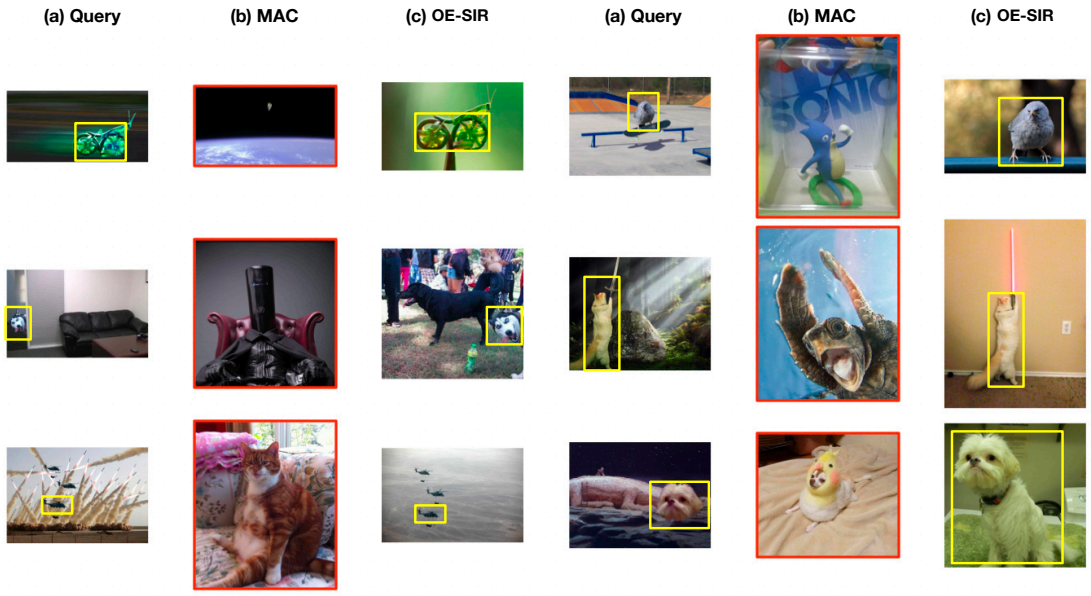

Figure 5. Example images and rank one results in the PIR dataset. (a) spliced query images. (b) Rank-1 result from MAC (c) Rank-1 result from OE-SIR. The red border indicates incorrect matches and the yellow bounding box shows the matching objects. OE-SIR can retrieve the original images while global image embeddings fail to retrieve the correct match due to background clutter.

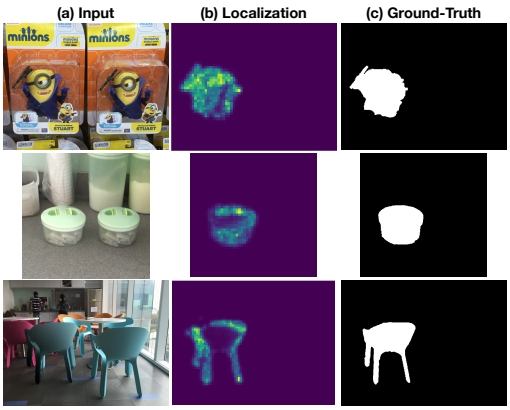

Figure 6. Qualitative results of image splicing localization on the COVERAGE dataset. The proposed method is able to correctly localize the spliced region by comparing the spliced image with the retrieved original image.

gorithms including (1) NoI [40]: A noise inconsistency method which predicts regions as manipulated where the local noise is inconsistent with authentic regions. (2) CFA [18]: A CFA based method that estimates the internal CFA pattern of the camera for every patch in the image and segments out the regions with anomalous CFA features as manipulated regions. (3) RBG-N [76]: A two-stream Faster R-CNN based approach which combines features from the RGB and noise channel to make the final prediction. (4) Self-Consistency [28]: A self-consistency approach that utilizes metadata to learn features useful for manipulation localization. The prediction is made patch by patch and post-processing like mean-shift [11] is used to obtain the pixel-level manipulation prediction. (5) GSR-NET [75]: a segmentation-based approach with the generative adversarial network. (6) OE-SIR: Our spliced image retrieval approach. We first retrieve from the database (cf. Eq. 7) the top-1 database image. Following Eq. 8 and 9, we then obtain the binary localization map corresponding to the top-1 image, after which we compute the MCC and F1 score. Table 5 shows the performance of different methods. Our approach outperforms other state-of-the-art methods by a significant margin because we can effectively utilize a database of original images. As we can see in Figure 6 (a) many of these spliced images are quite challenging even for a human. However, it becomes easier if we can compare it with the original images. Figure 6 (b) shows some qualitative results of the proposed method.

5. Conclusion

We describe an alternative approach to spliced image detection by casting it as an image retrieval task. Extensive experiments show the effectiveness of the proposed approach compares to the baseline algorithm. In addition, this paper provides a comprehensive analysis of features from pretrain models, allowing for a better selection of embedding models, which could be useful even for other tasks. Future research direction includes analysis of the effect of different object detectors as well as utilizing unsupervised or self-supervised methods to learn better features for the OE-SIR framework.

References

[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef 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. 2

[2] Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. Factors of transferability for a generic convnet representation. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1790–1802, 2016. 3

[3] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in neural information processing systems*, pages 2654–2662, 2014. 3

[4] Artem Babenko and Victor Lempitsky. Aggregating local deep features for image retrieval. In *Proceedings of the IEEE international conference on computer vision*, pages 1269–1277, 2015. 2, 7

[5] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural codes for image retrieval. In *European conference on computer vision*, pages 584–599. Springer, 2014. 2

[6] Jawadul H Bappy, Amit K Roy-Chowdhury, Jason Bunk, Lakshmanan Nataraj, and BS Manjunath. Exploiting spatial structure for localizing manipulated image regions. In *Proceedings of the IEEE international conference on computer vision*, pages 4970–4979, 2017. 1, 2

[7] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541. ACM, 2006. 3

[8] Bor-Chun Chen and Andrew Kae. Toward realistic image compositing with adversarial learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8415–8424, 2019. 6

[9] Di Chen, Shanshan Zhang, Jian Yang, and Bernt Schiele. Norm-aware embedding for efficient person search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12615–12624, 2020. 3

[10] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In *Advances in Neural Information Processing Systems*, pages 742–751, 2017. 3

[11] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE transactions on pattern analysis and machine intelligence*, 17(8):790–799, 1995. 8

[12] Ondrej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007. 2

[13] Navneet Dalal and Bill 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. 3, 7

[14] Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona. Fast feature pyramids for object detection. *IEEE transactions on pattern analysis and machine intelligence*, 36(8):1532–1545, 2014. 2, 3

[15] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655, 2014. 3

[16] Wenkai Dong, Zhaoxiang Zhang, Chunfeng Song, and Tieniu Tan. Bi-directional interaction network for person search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2839–2848, 2020. 3

[17] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009. 3

[18] Pasquale Ferrara, Tiziano Bianchi, Alessia De Rosa, and Alessandro Piva. Image forgery localization via fine-grained analysis of cfa artifacts. *IEEE Transactions on Information Forensics and Security*, 7(5):1566–1577, 2012. 2, 7, 8

[19] Weifeng Ge. Deep metric learning with hierarchical triplet loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 269–285, 2018. 2

[20] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra 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. 3

[21] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. Deep image retrieval: Learning global representations for image search. In *European conference on computer vision*, pages 241–257. Springer, 2016. 2

[22] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus. End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision*, 124(2):237–254, 2017. 2

[23] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2, 4, 5

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4, 5

[25] Silvan Heller, Luca Rossetto, and Heiko Schuldt. The PS-Battles Dataset – an Image Collection for Image Manipulation Detection. *CoRR*, abs/1804.04866, 2018. 6

[26] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3, 4

[27] Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016. 3

[28] Minyoung Huh, Andrew Liu, Andrew Owens, and Alexei A Efros. Fighting fake news: Image splice detection via

learned self-consistency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 101–117, 2018. 1, 2, 7, 8

[29] Hervé Jégou and Ondřej Chum. Negative evidences and cooccurences in image retrieval: The benefit of pca and whitening. In *European conference on computer vision*, pages 774–787. Springer, 2012. 5, 6

[30] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Improving bag-of-features for large scale image search. *International journal of computer vision*, 87(3):316–336, 2010. 2

[31] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128, 2011. 2

[32] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[33] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? *arXiv preprint arXiv:1805.08974*, 2018. 3

[34] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013. 5

[35] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv:1811.00982*, 2018. 3, 5

[36] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017. 3, 5

[37] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 3, 5

[38] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *arXiv preprint arXiv:1809.02165*, 2018. 3

[39] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 2

[40] Babak Mahdian and Stanislav Saic. Using noise inconsistencies for blind image forensics. *Image and Vision Computing*, 27(10):1497–1503, 2009. 2, 7, 8

[41] Iaroslav Melekhov, Aleksei Tiulpin, Torsten Sattler, Marc Pollefeys, Esa Rahtu, and Juho Kannala. Dgc-net: Dense geometric correspondence network. *arXiv preprint arXiv:1810.08393*, 2018. 2

[42] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *Advances in Neural Information Processing Systems*, pages 4826–4837, 2017. 2

[43] Daniel Moreira, Aparna Bharati, Joel Brogan, Allan Pinto, Michael Parowski, Kevin W Bowyer, Patrick J Flynn, Anderson Rocha, and Walter J Scheirer. Image provenance analysis at scale. *IEEE Transactions on Image Processing*, 27(12):6109–6123, 2018. 2

[44] Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 360–368, 2017. 2

[45] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3456–3465, 2017. 2, 3

[46] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio 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. 5

[47] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 5

[48] Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej 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. 5, 6

[49] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *European conference on computer vision*, pages 3–20. Springer, 2016. 2

[50] Filip Radenović, Giorgos Tolias, and Ondrej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 7

[51] Ali S Razavian, Josephine Sullivan, Stefan Carlsson, and Atsuto Maki. Visual instance retrieval with deep convolutional networks. *ITE Transactions on Media Technology and Applications*, 4(3):251–258, 2016. 7

[52] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali 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. 3

[53] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 2, 3, 5, 7

[54] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6148–6157, 2017. 2

[55] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. End-to-end weakly-supervised semantic alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6917–6925, 2018. 2

[56] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014. 3

[57] Yong Rui, Thomas S Huang, and Shih-Fu Chang. Image retrieval: Current techniques, promising directions, and open issues. *Journal of visual communication and image representation*, 10(1):39–62, 1999. 2

[58] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Spreading vectors for similarity search. *arXiv preprint arXiv:1806.03198*, 2018. 5

[59] Ronald Salloum, Yuzhuo Ren, and C-C Jay Kuo. Image splicing localization using a multi-task fully convolutional network (mfcn). *Journal of Visual Communication and Image Representation*, 51:201–209, 2018. 1, 2

[60] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan 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. 3

[61] Bharat Singh, Mahyar Najibi, and Larry S Davis. Sniper: Efficient multi-scale training. In *Advances in Neural Information Processing Systems*, pages 9333–9343, 2018. 3

[62] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *null*, page 1470. IEEE, 2003. 2

[63] Marvin Teichmann, Andre Araujo, Menglong Zhu, and Jack Sim. Detect-to-retrieve: Efficient regional aggregation for image search. *arXiv preprint arXiv:1812.01584*, 2018. 3

[64] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. *arXiv preprint arXiv:1511.05879*, 2015. 2, 7

[65] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. Deep metric learning with angular loss. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2593–2601, 2017. 2

[66] 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. 5

[67] Bihan Wen, Ye Zhu, Ramanathan Subramanian, Tian-Tsong Ng, Xuanjing Shen, and Stefan Winkler. Coverage — a novel database for copy-move forgery detection. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 161–165. IEEE, 2016. 7

[68] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2840–2848, 2017. 2

[69] Yue Wu, Wael Abd-Almageed, and Prem Natarajan. Busternet: Detecting copy-move image forgery with source/target localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 168–184, 2018. 2

[70] Shuiming Ye, Qibin Sun, and Ee-Chien Chang. Detecting digital image forgeries by measuring inconsistencies of blocking artifact. In *2007 IEEE International Conference on Multimedia and Expo*, pages 12–15. IEEE, 2007. 2

[71] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014. 3

[72] Joe Yue-Hei Ng, Fan Yang, and Larry S Davis. Exploiting local features from deep networks for image retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 53–61, 2015. 2

[73] Xu Zhang, Felix X Yu, Sanjiv Kumar, and Shih-Fu Chang. Learning spread-out local feature descriptors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4595–4603, 2017. 2

[74] Liang Zheng, Yi Yang, and Qi Tian. Sift meets cnn: A decade survey of instance retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 40(5):1224–1244, 2018. 2

[75] Peng Zhou, Bor-Chun Chen, Xintong Han, Mahyar Najibi, and Larry S Davis. Generate, segment and replace: Towards generic manipulation segmentation. In *Proceedings of the AAAI conference on artificial intelligence*, 2020. 7, 8

[76] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry 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. 1, 2, 7, 8