Detect-to-Retrieve: Efficient Regional Aggregation for Image Search

Marvin Teichmann*
University of Cambridge, UK
mttt2@eng.cam.ac.uk

André Araujo* Menglong Zhu Jack Sim
Google AI, USA
{andrearaujo, menglong, jacksim}@google.com

Abstract

Retrieving object instances among cluttered scenes efficiently requires compact yet comprehensive regional image representations. Intuitively, object semantics can help build the index that focuses on the most relevant regions. However, due to the lack of bounding-box datasets for objects of interest among retrieval benchmarks, most recent work on regional representations has focused on either uniform or class-agnostic region selection. In this paper, we first fill the void by providing a new dataset of landmark bounding boxes, based on the Google Landmarks dataset, that includes 86k images with manually curated boxes from 15k unique landmarks. Then, we demonstrate how a trained landmark detector, using our new dataset, can be leveraged to index image regions and improve retrieval accuracy while being much more efficient than existing regional methods. In addition, we introduce a novel regional aggregated selective match kernel (R-ASMK) to effectively combine information from detected regions into an improved holistic image representation. R-ASMK boosts image retrieval accuracy substantially with no dimensionality increase, while even outperforming systems that index image regions independently. Our complete image retrieval system improves upon the previous state-of-the-art by significant margins on the Revisited Oxford and Paris datasets. Code and data available at the project webpage: https://github.com/tensorflow/models/tree/master/research/delf.

1. Introduction

In this paper, we address the image retrieval problem: given a query image, a system should efficiently retrieve similar images from a database. Image retrieval systems are usually composed of two main stages: (1) *filtering*, where an efficient technique ranks database images according to their similarity with respect to the query; (2) *re-ranking*, where a small number of the most similar database images from the first stage are inspected in more detail and re-ranked.

Traditionally, hand-crafted local features [21, 6] were coupled to Bag-of-Words-inspired techniques [36, 26, 27, 14, 15, 16, 38] to construct high-dimensional representations used in the filtering step. Local feature matching and geometric verification [26, 27, 3] (commonly using RANSAC [8]) have been used as effective re-ranking strategies. Recently, several deep learning techniques have been proposed for these two stages. Global image representations based on convolutional neural networks (CNN) can produce compact embeddings to enable fast similarity computation in the filtering step [5, 4, 40, 1, 9, 30]. Local image representations can also be extracted using CNNs, suitable to re-ranking via

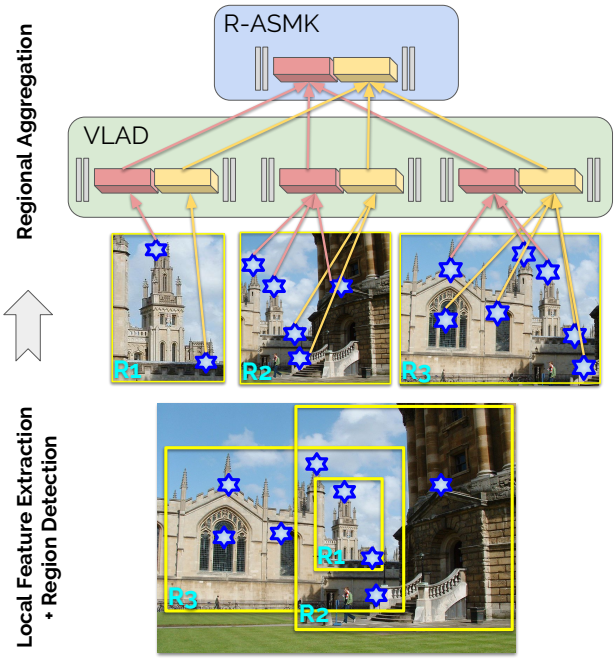


Figure 1: Overview of our proposed regional aggregation method. Deep local features (stars) and object regions (boxes) are extracted from an image. Regional aggregation proceeds in two steps, using a large codebook of visual words (red and yellow visual words are depicted): first, per-region VLAD description; second, sum pooling and per-visual word normalization. Our final regionally aggregated image representation can be combined to selective match kernels and provide improved image similarity estimation: we refer to this technique as regional aggregated selective match kernels (R-ASMK). It leverages detected regions to improve image retrieval at no dimensionality increase when compared to the original ASMK method [38].

*Both authors contributed equally to this work.

spatial matching and geometric verification [25, 24, 23].

Today's image retrieval systems tend to fail when relevant objects do not occupy a large enough fraction of database images, typically in cluttered scenes. Often, these objects produce local features that can be used to find local matches against the query in the re-ranking stage. However, such cluttered images usually fail to reach the re-ranking stage, since their initial representation does not lead to high similarity when compared to the query during the filtering stage. The most common solution to estimate an improved similarity with respect to the query image is to extract and separately store image representations for regions-of-interest in the database, using a fixed regional grid [2, 31] or a class-agnostic detector [37, 17]. However, the existing region selection techniques produce a large number of irrelevant regions. In a recent large-scale experimental image retrieval evaluation, Radenovic *et al*. [28] concluded that such regional search approaches impose too high of a cost in terms of memory and latency, with only small accuracy gains.

Contributions. (1) Our first contribution is aimed at improving region selection: we introduce a dataset of manually boxed landmark images, with $86k$ images from $15k$ unique classes, and we show that detectors can be trained for robust landmark localization. (2) Our second contribution is to leverage the trained detector and produce more efficient regional search systems, which improve accuracy for small objects with only a modest increase to the database size – much more efficiently than previously proposed techniques. (3) In our third contribution, we propose regional aggregated match kernels to leverage selected image regions and produce a discriminative image representation, illustrated in Fig. 1. This new representation outperforms regional search systems significantly, while at the same time being more efficient: only one descriptor needs to be stored per image. Our image retrieval system outperforms previously published results by 9.3% absolute mean average precision on the Revisited Oxford-Hard dataset, and 1.9% on the Revisited Paris-Hard dataset [28].

2. Related Work

Datasets. To the best of our knowledge, no manually curated datasets of landmark bounding boxes exist. Gordo *et al*. [9] use SIFT [21] matching to estimate boxes in landmark images. Such boxes are biased towards the feature extraction and matching technique, and may contain localization errors. Their dataset contains $49k$ boxed images, from 586 landmarks. In comparison, we use human raters to annotate the regions of interest, and produce $86k$ boxed images from $15k$ landmarks. The OpenImages dataset [19] contains $9M$ images, annotated with generic object bounding boxes. Some of them may be considered landmarks, for example: buildings, towers, skyscrapers, billboards. However, these classes make for a small fraction of the entire dataset.

Regional search and aggregation. Region selection has been explored in image retrieval systems. They have been used with two different purposes: (i) regional search: selected regions are encoded independently in the database, allowing for retrieval of subimages; (ii) regional aggregation: selected regions are used to improve image representations. In the following, we review these two types of approaches.

Regional search. Many papers propose to describe regions using VLAD [15] or Fisher Vectors [16]: Arandjelovic and Zisserman [2] use a multi-scale grid to extract 14 regions per image; Tao *et al*. [37] use Selective Search [41] with thousands of regions per image; Kim *et al*. [17] use maximally stable extremal regions (MSER) [22]. Razavian *et al*. [31] use a multi-scale grid with 30 regions per image, and compute the similarity of two images by taking into account the distances between all region pairs. Iscen *et al*. [13, 12] leverage multi-scale grids in conjunction with CNN features [29], to enable query expansion via diffusion. More recently, Radenovic *et al*. [28] performed a comprehensive evaluation of retrieval techniques and concluded that existing regional search methods may improve recognition accuracy, however at significantly larger memory and complexity costs. In contrast, our Detect-to-Retrieve framework aims at efficient regional search via the use of a custom trained detector.

Regional aggregation. Tolias *et al*. [40] leverage the grid structure from [31] to pool pretrained CNN features [18, 35] into compact representations; approximately 20 regions are selected per image. Radenovic *et al*. [29] build upon [40] by re-training features on a dataset collected in an unsupervised manner. Gordo *et al*. [9] train a region proposal network [32] from semi-automatic bounding box annotations, to replace the grid from [40]. Hundreds of regions per image are considered in this case. Our work departs from these papers by using a small set of regions (fewer than 5 per image), and by formulating regional aggregation as a new match kernel (instead of regional sum-pooling as in [40, 9]).

3. Google Landmark Boxes Dataset

In this section, we introduce our newly collected Google Landmark Boxes dataset, describing the manual annotation process. Our work builds upon the recent Google Landmarks dataset (GLD) [25], whose training set contains $1.2M$ images of $15k$ unique landmarks, with a wide variety of objects including buildings, monuments, bridges, statues as well as natural landmarks such as mountains, lakes and waterfalls.

Each image in this dataset is considered to only depict one landmark. In some cases, a landmark may consist of a set of buildings: for example, skylines, which are common in this dataset, are considered as a single landmark. Since GLD is collected in a semi-automatic manner considering popular touristic locations, it is sometimes ambiguous what the landmark of interest may be. When collecting bounding box annotations, our goal is to capture the most prominent

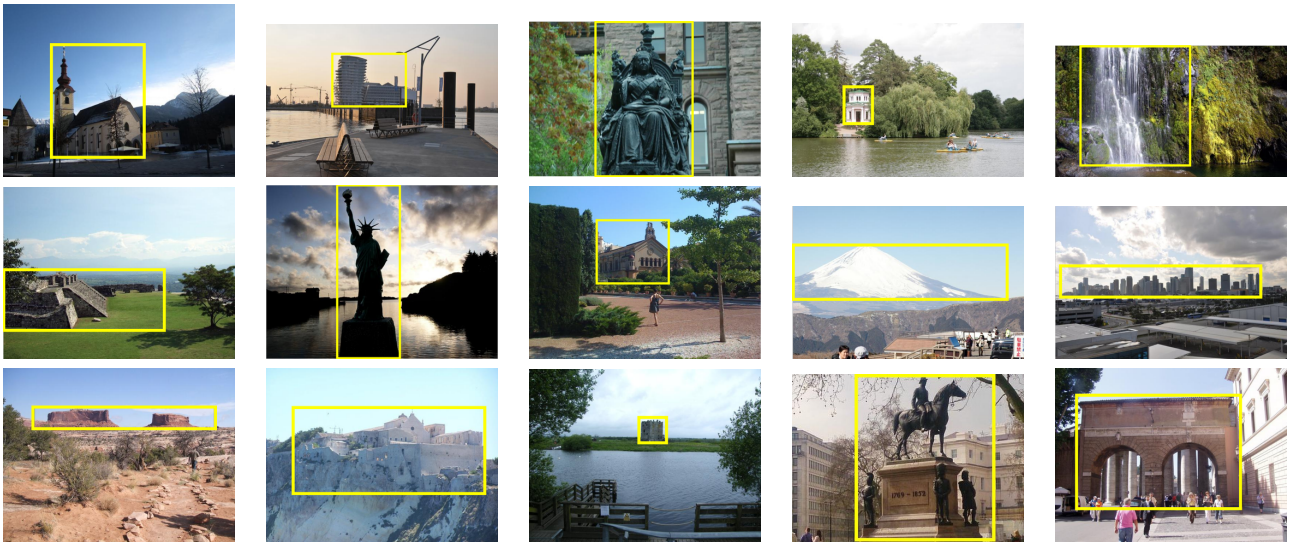


Figure 2: Examples of annotated images from our Google Landmark Boxes dataset. A box is drawn around the most prominent landmark depicted in the image. The dataset contains a wide variety of objects, ranging from man-made to natural landmarks.

landmark in the image, according to the fact that each image is only assigned one landmark label. Each box should reflect the main object (or set of objects) which is showcased in each dataset image. For this reason, we instructed human operators to draw at most one box per image.

One of the main challenges in such a fine-grained dataset is the inherent long tail of number of image samples per class. In GLD, some landmarks are associated to several thousands of images, while for about half of the classes only 10 or fewer images are provided. Our goal is to represent landmarks in a balanced manner in our new dataset, such that trained detectors are able to localize a wide variety of objects. For this reason, we first separate part of the $1.2M$ training set into a validation set. We randomly select four training and four validation images per landmark. In total, this yields $58k$ and $36k$ boxed images for training and validation, respectively. Note that this means that for about 40% of landmarks, all available images are annotated.

Examples of annotated images are shown in Fig. 2. In some cases, it is not possible to identify a prominent landmark (see Fig. 3): the landmark of interest may be occluded, or the image may actually show the surroundings of a landmark. We remove such corner cases from our dataset (this applied to about 8% of images which were initially selected), leading to a final dataset with $54k$ and $32k$ boxed images for training and validation, respectively.

4. Regional Search and Aggregation

We present techniques that enhance image retrieval performance by utilizing bounding boxes predicted by a trained landmark detector. In particular, our approach builds on top of deep local features (DELF) [25] and aggregated selective match kernels (ASMK) [38], which were recently shown to achieve state-of-the-art performance on a large-scale image retrieval benchmark [28].

4.1. Background

We briefly review the aggregated match kernel framework by Tolias *et al.* [38]. An image X is described by a set $\mathcal{X} = \{x_1, x_2, \ldots, x_M\}$ containing M local descriptors, each of dimension D. A codebook $\mathcal{C}$ comprising C visual words, learned using k-means, is used to quantize the descriptors. Denote $\mathcal{X}_c = \{x \in \mathcal{X} : q(x) = c\}$ as the subset of descriptors from X which are assigned to visual word c by the nearest neighbor quantizer $q(x)$.

According to this framework, the similarity between two images X and Y, represented by local descriptor sets $\mathcal{X}$ and $\mathcal{Y}$, can be computed as:

$$K(X,Y) = \gamma(\mathcal{X})\gamma(\mathcal{Y}) \sum_{c \in \mathcal{C}} \sigma(\Phi(\mathcal{X}_c)^T \Phi(\mathcal{Y}_c)) \qquad (1)$$

where $\Phi(\mathcal{X})$ is an aggregated vector representation, $\sigma(.)$ denotes a scalar selectivity function and $\gamma(\mathcal{X}) = \left(\sum_c \sigma(\Phi(\mathcal{X}_c)^T \Phi(\mathcal{X}_c))\right)^{-1/2}$ is a normalization factor. This formulation encompasses popular local feature aggregation techniques, such as Bag-of-Words [36], VLAD [15] and ASMK [38].

In particular, for VLAD, $\sigma(u) = u$ and $\Phi(\mathcal{X}_c)$ corresponds to an aggregated residual $V(\mathcal{X}_c) = \sum_{x \in \mathcal{X}_c} x - q(x)$. For ASMK, $\sigma(u)$ corresponds to a thresholded polynomial selectivity function

$$\sigma(u) = \begin{cases} \text{sign}(u)|u|^\alpha, & \text{if } u > \tau \\ 0, & \text{otherwise} \end{cases} \qquad (2)$$

Figure 3: Examples of Google Landmarks dataset images which do not depict a prominent landmark. In such cases (about 8% of images), no boxes were drawn, and the images were not included in the Google Landmark Boxes dataset.

where usually $\alpha = 3$ and $\tau = 0$; and $\Phi(\mathcal{X}_c)$ corresponds to a normalized aggregated residual $\hat{V}(\mathcal{X}_c) = V(\mathcal{X}_c)/\|V(\mathcal{X}_c)\|$.

4.2. Regional Search

In this section, we consider image retrieval systems where regional descriptors are stored independently in the database. Denote the query image as X, and the database of N images as $\{Y^{(n)}\}$, $n = 1, 2, \ldots, N$. We are mainly interested in the experimental configuration where a query contains a well-localized region-of-interest (*i.e.*, the query in practice contains only one region), which is a common setting in image retrieval. For the n-th database image, regions $r_n = 1, \ldots, R_n$ are predicted by a landmark detector, defining the subimages $\{Y^{(n,r_n)}\}$. We denote $Y^{(n,1)} = Y^{(n)}$ as the subimage corresponding to the original image, and always consider it as a valid region. To leverage uncluttered representations, we store aggregated descriptors independently for each subimage, which leads to a total of $\sum_{n=1}^{N} R_n$ items in the database.

To compute the similarity between the query X and a database image $Y^{(n)}$, we consider max-pooling or average-pooling individual regional similarities, respectively:

$$\text{sim}_{MAX}(X, Y^{(n)}) = max_{r=1,\ldots,R_n} K\left(\mathcal{X}, \mathcal{Y}^{(n,r)}\right) \quad (3)$$

$$\text{sim}_{AVG}(X, Y^{(n)}) = \frac{1}{R_n} \sum_{r=1}^{R_n} K\left(\mathcal{X}, \mathcal{Y}^{(n,r)}\right) \quad (4)$$

Max-pooling corresponds to assigning a database image's score considering only its highest-scoring subimage. Average pooling aggregates contributions from all subimages. These two variants are compared in Sec. 5.

4.3. Regional Aggregated Match Kernels

Storing descriptors of each region independently in the database incurs additional cost for both memory and search computation. In this section, we consider utilizing the detected bounding boxes to instead improve the aggregated representations of database images – producing discriminative descriptors at no additional cost. We extend the aggregated match kernel framework of Tolias *et al.* [38] to regional aggregated match kernels, as follows.

We start by noting that the average pooling similarity Eq. (4) can be rewritten as:

$$\text{sim}_{AVG}(X, Y^{(n)}) = \gamma(\mathcal{X}) \sum_{c} \sum_{r} \frac{\gamma(\mathcal{Y}^{(n,r)})}{R_n} \sigma\left(\Phi(\mathcal{X}_c)^T \Phi(\mathcal{Y}_c^{(n,r)})\right) \quad (5)$$

Simple regional aggregation. For VLAD, this can be further expanded as:

$$\begin{aligned}
&\text{sim}^{(\text{R-VLAD})}(X, Y^{(n)}) \\
&= \gamma(\mathcal{X}) \sum_{c} \sum_{r} \frac{\gamma(\mathcal{Y}^{(n,r)})}{R_n} V(\mathcal{X}_c)^T V(\mathcal{Y}_c^{(n,r)}) \\
&= \sum_{c} \gamma(\mathcal{X}) V(\mathcal{X}_c)^T \sum_{r} \frac{\gamma(\mathcal{Y}^{(n,r)})}{R_n} V(\mathcal{Y}_c^{(n,r)}) \quad (6) \\
&= \sum_{c} V_R(\mathcal{X}_c)^T V_R(\{\mathcal{Y}_c^{(n,r)}\}_r) \quad (7)
\end{aligned}$$

where we define

$$V_R(\{\mathcal{Y}_c^{(n,r)}\}_r) = \frac{1}{R_n} \sum_{r} \gamma(\mathcal{Y}^{(n,r)}) V(\mathcal{Y}_c^{(n,r)}) \quad (8)$$

Using this definition, note that $V_R(\mathcal{X}_c) = \gamma(\mathcal{X}) V(\mathcal{X}_c)$. This derivation indicates that average pooling of regional VLAD similarities can be performed using aggregated regional descriptors and does not require storage of each region's representation separately[1]. We refer to this simple regional aggregated kernel as R-VLAD.

A similar derivation can be obtained for ASMK in the case where $\sigma(.)$ is the identity function (*i.e.*, no selectivity is applied), by replacing $V(\mathcal{X}_c)$ by $\hat{V}(\mathcal{X}_c)$ in Eq. (6). A straightforward matching kernel using this idea would apply the selectivity function when comparing the query ASMK representation against this aggregated representation. We refer to this aggregation variant as Naive-R-ASMK.

Both the R-VLAD and Naive-R-ASMK kernels present an important problem when using many detected regions per image and large codebooks. For a given image region,

[1] Another way to see that this applies to VLAD kernels is to note that VLAD similarity is computed via a simple inner product, and that the average inner product with a set of vectors equals the inner product with the set average; *i.e.*, for vector x and set $\{y_n\}$, $\frac{1}{N}\sum_n x^T y_n = x^T \left(\frac{1}{N}\sum_n y_n\right)$.

most visual words will not be associated to any local feature, leading to many all-zero residuals for the region. For visual words that correspond to visual patterns observed in only a small number of regions, this will lead to substantially downweighted residuals. We propose to fix this weakness by developing the R-ASMK kernel as follows, inspired by the changes introduced by the original ASMK with respect to VLAD.

R-ASMK. We define the R-ASMK similarity between a query and a database image as:

$$\text{sim}^{(\text{R-ASMK})}(X, Y^{(n)}) = \sum_c \sigma\left(\hat{V}_R(\mathcal{X}_c)^T \hat{V}_R(\{\mathcal{Y}_c^{(n,r)}\}_r)\right) \quad (9)$$

where $\hat{V}_R(\{\mathcal{Y}_c^{(n,r)}\}_r) = \frac{V_R(\{\mathcal{Y}_c^{(n,r)}\}_r)}{\|V_R(\{\mathcal{Y}_c^{(n,r)}\}_r)\|}$ is the normalized regionally aggregated residual corresponding to visual word c.

R-AMK. The kernels we presented in this section can be regarded as different instantiations of a general regional aggregated match kernel (R-AMK), defined as follows:

$$K_R(X, Y) = \sum_{c \in \mathcal{C}} \sigma\left(\Phi_R(\{\mathcal{X}_c^{(r)}\}_r)^T \Phi_R(\{\mathcal{Y}_c^{(r)}\}_r)\right) \quad (10)$$

where $\{\mathcal{X}_c^{(r)}\}_r$ denotes the sets of local descriptors quantized to visual word c, from each region of X. Φ_R specializes to V_R for R-VLAD, and to $\hat{V}_R$ for R-ASMK. Note that this definition involves regional aggregation for both images, while in this work we focus on the asymmetric case where regional aggregation is applied to the database image only. The asymmetric case is more relevant when the query image is itself a well-localized region-of-interest, which is a common setup in image retrieval benchmarks.

Binarization. For codebooks with a large number of visual words, the storage cost for such aggregated representations may be prohibitive. Binarization is an effective strategy to allow scalable retrieval in these cases. We adopt a similar binarization strategy as [38], where a binarized version of Φ_R can be obtained by the elementwise function $b(x) = +1$ if $x > 0, -1$ otherwise. We denote the binarized version by a $\star$ superscript (*e.g.*, R-ASMK* is the binarized version of R-ASMK).

5. Experiments

We present two types of experiments: first, landmark detection, to assess the quality of object detector models trained on the new dataset. Second, we utilize the detected landmarks to enhance image retrieval systems.

5.1. Landmark Detection

We train two types of detection models on the bounding box data we have collected and described in Sec. 3: a single shot Mobilenet-V2 [33] based SSD detector [20] and a two stage Resnet-50 [10] based Faster-RCNN [32]. Standard object detection evaluation metric Average Precision (AP) measured at 50% Intersection-over-Union ratio is used during evaluation. Both models reach about 85% AP on the validation set within $500k$ steps (85.61%, 84.37% respectively). The models are trained with publicly available Tensorflow Object Detection API [11]. The results indicate that accurate landmark localization can be trained using our dataset. The Mobilenet-V2-SSD variant runs at 27ms per image, while the Resnet-50-Faster-RCNN runs at 89ms, both numbers on a TitanX GPU.

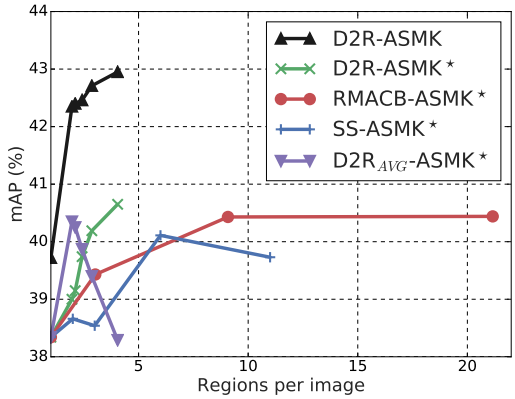
5.2. Image Retrieval

We perform regional search and regional aggregation experiments. The following describes the experimental setup.

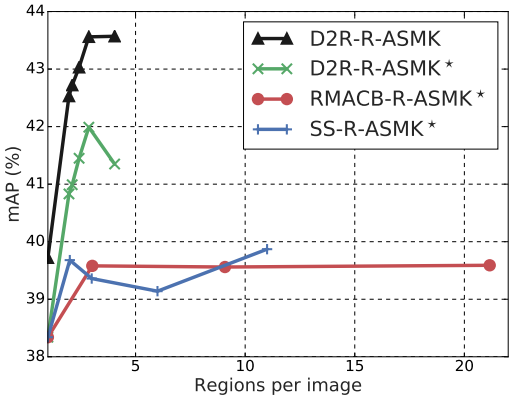
Datasets. We use the Oxford [26] and Paris [27] datasets, which have recently been revisited to correct annotation mistakes, add new query images and introduce new evaluation protocols [28]; the datasets are referred to as $\mathcal{R}$Oxf and $\mathcal{R}$Par, respectively. There are 70 query images for each dataset, with 4993 (6322) database images in the $\mathcal{R}$Oxf ($\mathcal{R}$Par) dataset. We report results on the Medium and Hard setups; for ablations, we focus more specifically on the Hard setup. Performance is measured using mean average precision (mAP) and mean precision at rank 10 (mP@10). We also perform large-scale experiments using the $\mathcal{R}$1M distractor set [28], which contains 1,001,001 images.

Image representation. We use the following setup in our experiments, except where indicated otherwise. The released DELF model [25] (pre-trained on the dataset from [9]) is used, with the default configuration (maximum of 1000 features per region are extracted, with a required minimum attention score of 100), except that the feature dimensionality is set to 128 as in previous work [28]. A 1024-sized codebook is used when computing aggregated kernels; as common practice, codebooks are trained on $\mathcal{R}$Oxf for retrieval experiments on $\mathcal{R}$Par, and vice-versa. We focus on improving the core image representations for retrieval, and do not consider query expansion (QE) [7] techniques such as Hamming QE [39], α QE [30] or diffusion [13, 12]; these methods could be incorporated to our system to obtain even stronger retrieval performance.

Region selection techniques. For our Detect-to-Retrieve (D2R) framework, we adopt the trained Faster R-CNN detector described in Sec. 5.1. We compare against previously proposed region selection techniques for image retrieval: the uniform grid from [31, 40] (denoted RMACB, for "RMAC boxes") and Selective Search (SS) [41, 37]. To vary the number of regions per image, we do as follows: (i) for D2R,



(a) Regional search evaluation.



(b) Regional aggregation evaluation.

Figure 4: Regional search and aggregation evaluations of different image representations, on $\mathcal{R}$Oxf-Hard. (a) Regional search: each regional representation is stored independently in the database, leading to increased memory requirements. Our D2R-ASMK variants achieve significant improvements over the single-image baseline while requiring substantially fewer boxes compared to other region selection approaches. (b) Regional aggregation: each region contributes to the aggregated representation for the entire image. The aggregated descriptor dimensionality is identical to single-image baseline that does not use regions. Our D2R-R-ASMK variants leverage the different landmark regions to compose a strong image representation, which is even more effective than storing each regional representation separately.

Method	Det. Thresh.	$\mathcal{R}$Oxf-Hard mAP	$\mathcal{R}$Oxf-Hard Size	$\mathcal{R}$Par-Hard mAP	$\mathcal{R}$Par-Hard Size
ASMK*	—	38.3	1	54.2	1
D2R-ASMK*	0.7	39.2	2.1	56.0	2.2
	0.5	39.7	2.4	56.2	2.4
	0.3	40.2	2.9	56.3	2.9
	0.1	40.7	4.1	56.7	3.9
D2R-R-ASMK*	0.7	41.0	1	56.2	1
	0.5	41.5	1	56.2	1
	0.3	**42.0**	1	56.3	1
	0.1	41.4	1	**56.8**	1

Table 1: Retrieval mAP and relative database size for the different region-based techniques introduced in this work, on the $\mathcal{R}$Oxf-Hard and $\mathcal{R}$Par-Hard datasets, as a function of the landmark detector threshold used for region selection. D2R-ASMK* uses max-pooling similarity from Eq. (3). The performances of both D2R-ASMK* and D2R-R-ASMK* tend to improve as the detection threshold decreases (more regions are selected). D2R-R-ASMK* outperforms D2R-ASMK* consistently, with a smaller memory footprint.

we vary the landmark detector threshold; (ii) for RMACB, we sweep the number of levels from 1 to 3; (iii) for SS, we select the top $\{1, 2, 5, 10\}$ boxes per image (as in this case there are no confidence scores associated to regions). For all region selection techniques, we add the original image as one of the selected regions.

Implementation details. We implemented the aggregated kernel framework from scratch in Python/Tensorflow. As a comparison against the reference MATLAB implementation [38], our ASMK* with a 1024-sized codebook and DELF features obtains 37.91% mAP in the $\mathcal{R}$Oxf-Hard dataset, while the reference implementation obtains 37.08%. Note that the reference implementation uses a similar configuration as Hamming Embedding (HE) [14], with a projection matrix before binarizarion, residuals computed with respect to the median, and IDF. We did not find consistent improvements using these, so we use the simpler version as described in Sec. 4. Similarly, the reference implementation uses multiple visual word assignments, but our preliminary experiments show improved results using single assignment, making retrieval faster and simpler – therefore we adopt single assignment in our experiments. We extended this implementation to support our regional search and aggregation techniques.

5.2.1 Regional Search

We compare aggregated match kernels, region selection techniques and similarity computation methods on the $\mathcal{R}$Oxf-Hard dataset. When performing regional search, multiple regions are selected per image and stored independently in the database, leading to increased memory cost. Fig. 4a presents results for ASMK variants, where all techniques use max-pooling similarity from Eq. (3), except for D2R$_{AVG}$-ASMK*, which uses average-pooling similarity from Eq. (4). Combining our proposed D2R regions with ASMK enhances mAP by 3.23% when using an average of 4.05 regions per image.

We compare the different region selection approaches using ASMK*. Our D2R-ASMK* achieves 40.65% mAP when using 4.05 regions per image, improvement of 2.31% over the single-image ASMK* baseline. Other region selection approaches improve retrieval accuracy, but with significantly larger memory requirements. RMACB-ASMK* requires 9.08 regions/image to achieve 40.43% mAP (this is

Method	Medium								Hard							
	$\mathcal{R}$Oxf		$\mathcal{R}$Oxf+$\mathcal{R}$1M		$\mathcal{R}$Par		$\mathcal{R}$Par+$\mathcal{R}$1M		$\mathcal{R}$Oxf		$\mathcal{R}$Oxf+$\mathcal{R}$1M		$\mathcal{R}$Par		$\mathcal{R}$Par+$\mathcal{R}$1M	
	mAP	mP@10	mAP	mP@10	mAP	mP@10	mAP	mP@10	mAP	mP@10	mAP	mP@10	mAP	mP@10	mAP	mP@10
AlexNet-GeM [30]	43.3	62.1	24.2	42.8	58.0	91.6	29.9	84.6	17.1	26.2	9.4	11.9	29.7	67.6	8.4	39.6
VGG16-GeM [30]	61.9	82.7	42.6	68.1	69.3	97.9	45.4	94.1	33.7	51.0	19.0	29.4	44.3	83.7	19.1	64.9
ResNet101-R-MAC [9]	60.9	78.1	39.3	62.1	78.9	96.9	54.8	93.9	32.4	50.0	12.5	24.9	59.4	86.1	28.0	70.0
ResNet101-GeM [30]	64.7	84.7	45.2	71.7	77.2	98.1	52.3	95.3	38.5	53.0	19.9	34.9	56.3	89.1	24.7	73.3
ResNet101-GeM↑+DSM [34]	65.3	87.1	47.6	76.4	77.4	99.1	52.8	96.7	39.2	55.3	23.2	37.9	56.2	89.9	25.0	74.6
HesAff-rSIFT-ASMK⋆ [38]	60.4	85.6	45.0	76.0	61.2	97.9	42.0	95.3	36.4	56.7	25.7	42.1	34.5	80.6	16.5	63.4
HesAff-rSIFT-ASMK⋆+SP [38]	60.6	86.1	46.8	79.6	61.4	97.9	42.3	95.3	36.7	57.0	26.9	45.3	35.0	81.7	16.8	65.3
HesAff-HardNet-ASMK⋆+SP [24]	65.6	90.2	–	–	65.2	98.9	–	–	41.1	59.7	–	–	38.5	87.9	–	–
DELF-ASMK⋆+SP [25, 28]	67.8	87.9	53.8	81.1	76.9	99.3	57.3	98.3	43.1	62.4	31.2	50.7	55.4	**93.4**	26.4	75.7
DELF-ASMK⋆ (reimpl.)	65.7	87.9	–	–	77.1	98.7	–	–	41.0	57.9	–	–	54.6	90.9	–	–
DELF-D2R-R-ASMK⋆ (ours)	69.9	89.0	–	–	78.7	99.0	–	–	45.6	61.9	–	–	57.7	93.0	–	–
— DELF-GLD (ours)	73.3	90.0	61.0	84.6	**80.7**	99.1	**60.2**	97.9	47.6	64.3	33.6	53.7	**61.3**	**93.4**	**29.9**	82.4
DELF-ASMK⋆+SP (reimpl.)	68.9	90.9	–	–	76.6	98.7	–	–	46.6	66.7	–	–	52.2	87.6	–	–
DELF-D2R-R-ASMK⋆+SP (ours)	71.9	91.3	–	–	78.0	**99.4**	–	–	48.5	66.7	–	–	54.0	87.6	–	–
— DELF-GLD (ours)	**76.0**	**93.4**	**64.0**	**87.7**	80.2	99.1	59.7	**99.0**	**52.4**	**70.9**	**38.1**	**61.3**	58.6	91.0	29.4	**83.9**

Table 2: Comparison of proposed techniques against state-of-the-art methods, on the $\mathcal{R}$Oxford ($\mathcal{R}$Oxf) and $\mathcal{R}$Paris ($\mathcal{R}$Par) datasets (and their large-scale extensions $\mathcal{R}$Oxf+$\mathcal{R}$1M and $\mathcal{R}$Par+$\mathcal{R}$1M), with Medium and Hard evaluation protocols. Previously published results are presented in the first block of rows. The second and third block of rows present our experimental results, considering systems without and with spatial verification (SP), respectively. In this experiment, we use codebooks with $65k$ visual words, to make our results comparable to previous work [28]. DELF-GLD indicates a version of DELF which we re-trained on the Google Landmarks dataset. Our methods achieve equal or improved performance for all evaluation protocols, datasets and metrics.

0.22% mAP below the previously mentioned D2R-ASMK⋆ operating point, despite requiring 2.24× more memory). SS-ASMK⋆ benefits from some regions, while performance decreases when a large number of regions are selected, since many of those regions are irrelevant.

Average pooling of individual regional similarities improves upon the single-image baseline significantly, at low overhead memory requirements: D2R$_{AVG}$-ASMK⋆ achieves 40.35% mAP with only 1.96× storage cost. Note that in this case performance drops significantly as more regions are added, since irrelevant regional similarities are added to the final image similarity. We also experimented with a D2R-VLAD representation: mAP improves from 30.17% (single-image) to 33.87% (2.87 regions/image).

Tab. 1 further presents D2R-ASMK⋆ results on the $\mathcal{R}$Par-Hard dataset. Regional search enables 2.5% mAP improvement at 3.9 regions/image. Note that our D2R approach is effective even if the landmarks in the Google Landmark Boxes dataset present much larger variability than the landmarks encountered in the $\mathcal{R}$Oxf/$\mathcal{R}$Par datasets.

5.2.2 Regional Aggregated Match Kernels

In this section, we evaluate the proposed regional aggregated match kernels. In this experiment, region selection is used to produce an improved image representation, with no increase in the aggregated descriptor dimensionality. Fig. 4b compares different aggregation methods and region selection approaches, on the $\mathcal{R}$Oxf-Hard dataset. Both our proposed D2R-R-ASMK and D2R-R-ASMK⋆ variants achieve substantial improvements compared to their baselines which do not use boxes for aggregation: 3.85% and 3.65% absolute mAP improvements, respectively. We also compare our D2R approach against other region selection methods. RMACB and SS improve upon the baseline, however with limited gain of at most 1.5% mAP.

More interestingly, our proposed kernels outperform even the regional search configuration where each region is indexed separately in the database. Tab. 1 compiles experimental results on $\mathcal{R}$Oxf-Hard and $\mathcal{R}$Par-Hard. Our D2R-R-ASMK⋆ method outperforms the best regional search variant on both datasets, respectively by 1.3% and 0.1% absolute mAP, with relative storage savings of 4.1× and 3.9×.

In another ablation experiment, we assess the performance of simpler regional aggregation methods: R-VLAD and Naive-R-ASMK. We use the trained detector to select regions. For R-VLAD, mAP on $\mathcal{R}$Oxf improves from 30.17% (single-image) to 30.91% when using 2.4 regions per image, but degrades quickly as more regions are considered. In particular, when setting a very low detection threshold (0.01) to obtain 10.2 regions per image, performance degenerates to 16.46% mAP – this agrees with the intuition that a large number of regions is detrimental to R-VLAD. For Naive-R-ASMK, no improvement is obtained when detected regions are used: mAP drops from 39.72% to 31.42% when 1.96 regions per image are used, and similarly degenerates to 9.2% when using 10.2 regions per image. In comparison, using the same detection threshold of 0.01, R-ASMK⋆ obtains 41.6% mAP, *i.e.*, performance is high even if using a large number of regions, due to the improved aggregation technique.

5.2.3 Comparison Against State-of-the-Art

We compare our D2R-R-ASMK⋆ technique against state-of-the-art image retrieval systems. To make our system comparable with previously published results [28], for this experiment we use a codebook with $65k$ visual words. We also further experiment with re-training the DELF local feature on the Google Landmarks dataset (denoted as DELF-

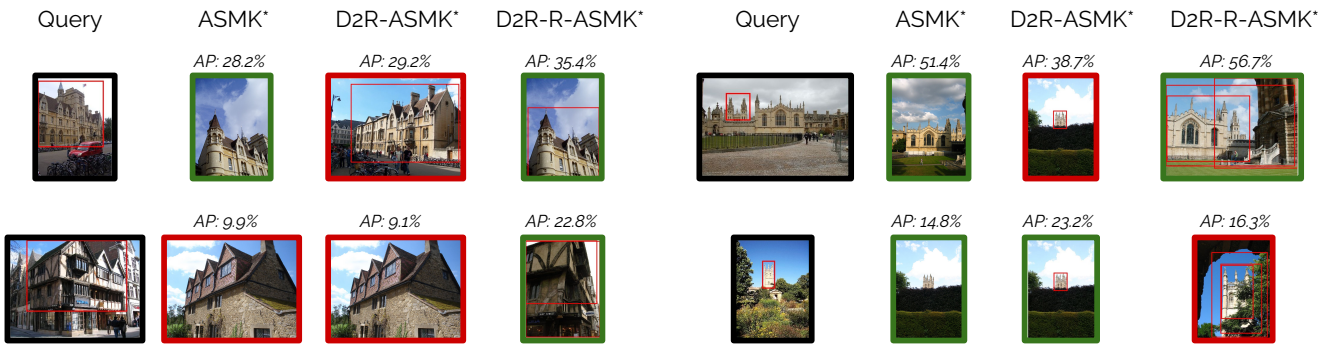


Figure 5: Qualitative results for ASMK⋆ (baseline single-image method), D2R-ASMK⋆ (regional search) and D2R-R-ASMK⋆ (regional aggregation) on $\mathcal{R}$Oxf-Hard. Four queries are presented, with their regions-of-interest highlighted. For each method, we show the first ranked image where the methods disagree. Red borders indicate incorrect results, and green borders indicate correct results. For D2R-ASMK⋆, we box the region used for the result (or leave unboxed if the region corresponds to the entire image). For D2R-R-ASMK⋆, we box all regions used for aggregation. We also present average precision (AP) for each method and query.

GLD). Spatial verification (SP) is used to re-rank the top 100 database images (we use RANSAC with an Affine model).

Table 2 presents experimental results on $\mathcal{R}$Oxf and $\mathcal{R}$Par, using the Medium and Hard protocols, also including the large-scale setup with $\mathcal{R}$1M. Our proposed D2R-R-ASMK⋆ representation by itself, without spatial verification, already improves mAP when comparing against all previously published results. SP further boosts performance by about 3% mAP on $\mathcal{R}$Oxf; surprisingly, it slightly degrades performance on the $\mathcal{R}$Par dataset. Re-training DELF on GLD improves performance by around 4%. Our best results improve upon the previous state-of-the-art by 8.2% mAP on $\mathcal{R}$Oxf-Medium, 1.8% mAP on $\mathcal{R}$Par-Medium, 9.3% mAP on $\mathcal{R}$Oxf-Hard and 1.9% in $\mathcal{R}$Par-Hard (with similar gains in the large-scale setup).

Memory. Our DELF-D2R-R-ASMK⋆ descriptors have the exact same dimensionality as DELF-ASMK⋆. However, DELF-ASMK⋆ is sparser and consumes less memory in practice: 10.3GB, compared to 27.6GB for DELF-D2R-R-ASMK⋆, in the large-scale $\mathcal{R}$Oxf+$\mathcal{R}$1M dataset. This is still much less than other local feature based approaches; *e.g.* HesAff-rSIFT-ASMK⋆ requires 62GB [28], and HesAffNet-HardNet++-ASMK⋆ [24] requires approximately 86.8GB.

5.2.4 Discussion

Our experiments demonstrate that selecting relevant image regions can help boost image retrieval performance significantly. In our regional aggregation method, the detected regions allow for effective re-weighting of local feature contributions, emphasizing relevant visual patterns in the final image representation. Note, however, that it is crucial to perform **both** region selection **and** regional aggregation in a suitable manner. If the selected regions are not relevant to the objects of interest, regional aggregation cannot be very effective, as shown in Fig. 4b. Also, our experiments with naive versions of regional aggregation indicate that the aggregation needs to be performed in the right way: this is related to the poor R-VLAD and Naive-R-ASMK results.

It may initially seem unintuitive that the regional search method underperforms when compared to our regional aggregation technique. However, this can be understood by observing some retrieval result patterns, which are presented in Fig. 5. The addition of separate regional representations to the database may help retrieval of relevant small objects in cluttered scenes, as illustrated with the successful bottom-right D2R-ASMK⋆ retrieved image. However, it also increases the chances of finding localized regions which are similar but do not correspond to the same landmark, as illustrated with the top two cases.

Regional aggregation, on the other hand, can help retrieval by re-balancing the visual information presented in an image. The top-right D2R-R-ASMK⋆ result shows a database image where the detected boxes do not precisely cover the query object; instead, several selected regions cover it, and consequently its features are boosted. A similar case is illustrated in the bottom-left example, where the main detected region in the database image does not cover the object of interest entirely. The features inside the main box are boosted but those outside are also used, generating a more suitable representation for image retrieval.

6. Conclusions

In this paper, we present an efficient regional aggregation method for image retrieval. We first introduce a dataset of landmark bounding boxes, and show that landmark detectors can be trained and leveraged for extracting regional representations. Regional search using our detectors not only provides superior retrieval performance but also much better efficiency than existing regional methods. In addition, we propose a novel regional aggregated match kernel framework that further boosts the retrieval accuracy. Our full system achieves state-of-the-art performance by a large margin on two image retrieval datasets.

Appendix A. Discussion: Detection Helps Finding Relevant Features

In this section, we analyze the detector's ability to focus on relevant landmarks by empirically estimating the proportion of relevant local features located within or without predicted bounding boxes. We extract and match local features for image pairs that are known to depict the same landmark. A local feature is declared to be relevant if it is an inlier to a high-confidence estimated geometric transformation.

More specifically, we use DELF local features [25] and a Faster-RCNN [32] landmark detector trained on our new dataset. $10k$ image pairs are collected from the Google Landmarks dataset [25]. Local feature matching is performed via nearest neighbor search followed by geometric verification (RANSAC [8] with an affine model). Fig. 6 plots the relevance probabilities as a function of the DELF local feature attention scores (these attention scores can be interpreted as a measure of a local feature's "landmarkness"). The blue curve denotes features that are located within bounding boxes, while the red curve represents features located outside bounding boxes.

The curves show that local features located within bounding boxes are much more likely to be relevant: for two features with the same attention score, the relevance probability for a feature located within a predicted box is approximately 3 to $4\times$ larger than that for a local feature located outside the box. Note how feature relevance increases with attention scores, as expected, but the predicted boxes can provide important extra information to effectively select the best features. This can be interpreted as the merging of two information streams: *bottom-up* (DELF attention scores estimate per-local feature relevance) and *top-down* (landmark detector estimates relevance of large regions).

Our proposed R-ASMK⋆ can be seen as a local feature re-weighting mechanism, which favors features located within detected regions. The experimental results obtained on the $\mathcal{R}$Oxford and $\mathcal{R}$Paris datasets confirm that re-weighting features within detected regions boosts image retrieval performance substantially.

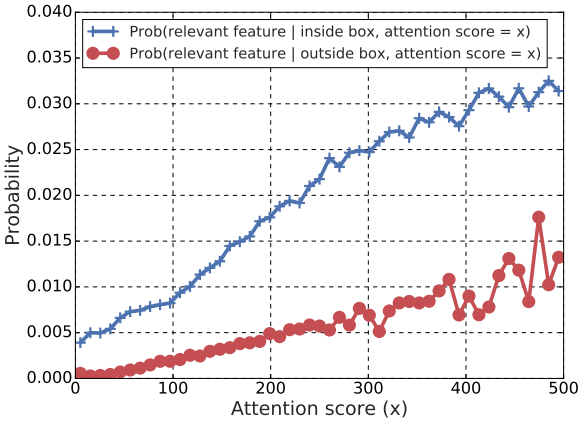


Figure 6: Relevance probability of a DELF local feature, as a function of its attention score. The blue curve denotes features inside predicted bounding boxes, while the red curve denotes features outside them. The detected boxes provide valuable information that can be used to improve image representations for retrieval tasks.

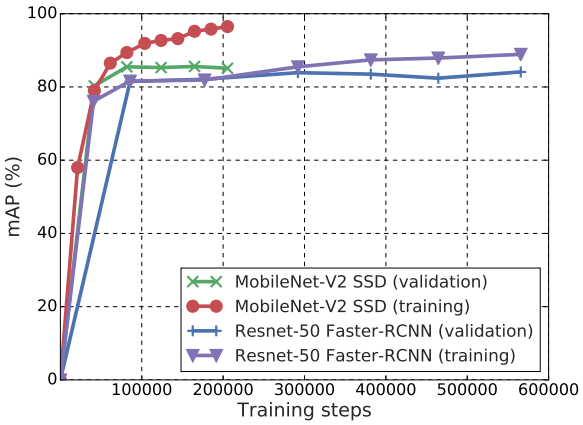


Figure 7: Mean average precision @ IOU=0.5 for the two trained landmark detectors, as a function of the number of training steps.

Appendix B. Detection Experiments

We present learning curves for the trained detectors, and examples of detected regions compared to ground-truth.

Learning Curves. We train both Faster-RCNN and SSD based object detection models on our dataset. Fig. 7 shows the comparison of learning progression of the two models. Both models converge to around 85% mAP within 600k training steps. The MobilenetV2 SSD model trains much faster than the Resnet-50 Faster-RCNN, due to much smaller model size and larger batch size (32 vs. 1, respectively). We also observe that SSD-based model slightly outperforms the Faster-RCNN base model despite having a smaller/weaker feature extractor. We conjecture that the advantage is due to the multi-scale feature map of SSD capturing the landmarks at different scales better than Faster-RCNN, which operates on a single feature map.

Detection Examples. To illustrate the effectiveness of our trained detectors, we present examples of detection using the SSD model. Fig. 8 shows examples for a variety of landmarks with different scales, occlusion and lighting conditions. In addition, we also show some failure cases in Fig. 9 where the object of interest has ambiguous semantic boundary (resulting in double-detection) or is very hard to distinguish from the scene (resulting in missed detection). For both figures, only detections with confidence probability more than 0.2 are shown.

Appendix C. Region Selection Comparison

In this section, we present landmark detection results on the $\mathcal{R}$Oxford and $\mathcal{R}$Paris datasets (Fig. 10 and Fig. 11, respectively), comparing with the selected regions by competitive approaches (RMAC boxes and Selective Search). The three methods use a configuration that produces a roughly similar number of regions per image: D2R with detection threshold 0.1 (about 4 regions per image), RMAC boxes with 2 levels (9 regions per image), and Selective Search with 6 selected regions per image. Note that our image retrieval experiments always use the whole image as a valid region, but in these visualizations we do not box the whole image, for a more concise presentation.

The figures show that our trained landmark detector tends to focus on the most prominent landmark regions in the image. RMAC boxes correspond to a fixed multi-scale grid, where the selected regions only depend on the input image size, not on its contents. This leads to regularly spaced boxes which do not usually overlap well with landmarks. Selective search produces boxes corresponding to prominent objects in the scene, which may or may not correspond to landmarks.

References

[1] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. In *Proc. CVPR*, 2016. 1

[2] R. Arandjelovic and A. Zisserman. All About VLAD. In *Proc. CVPR*, 2013. 2

[3] Y. Avrithis and G. Tolias. Hough Pyramid Matching: Speeded-up Geometry Re-ranking for Large Scale Image Retrieval. *IJCV*, 2014. 1

[4] A. Babenko and V. Lempitsky. Aggregating Local Deep Features for Image Retrieval. In *Proc. ICCV*, 2015. 1

[5] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural Codes for Image Retrieval. In *Proc. ECCV*, 2014. 1

[6] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-Up Robust Features (SURF). *CVIU*, 2008. 1

[7] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval. In *Proc. ICCV*, 2007. 5

[8] M. Fischler and R. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 1981. 1, 9

[9] A. Gordo, J. Almazan, J. Revaud, and D. Larlus. Deep Image Retrieval: Learning Global Representations for Image Search. In *Proc. ECCV*, 2016. 1, 2, 5, 7

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Proc. CVPR*, 2016. 5

[11] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al. Speed/Accuracy Trade-offs for Modern Convolutional Object Detectors. In *Proc. CVPR*, 2017. 5

[12] A. Iscen, Y. Avrithis, G. Tolias, T. Furon, and O. Chum. Fast Spectral Ranking for Similarity Search. In *Proc. CVPR*, 2018. 2, 5

[13] A. Iscen, G. Tolias, Y. Avrithis, T. Furon, and O. Chum. Efficient Diffusion on Region Manifolds: Recovering Small Objects with Compact CNN Representations. In *Proc. CVPR*, 2017. 2, 5

[14] H. Jégou, M. Douze, and C. Schmid. Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search. In *Proc. ECCV*, 2008. 1, 6

[15] H. Jégou, M. Douze, C. Schmidt, and P. Perez. Aggregating Local Descriptors into a Compact Image Representation. In *Proc. CVPR*, 2010. 1, 2, 3

[16] H. Jégou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid. Aggregating Local Image Descriptors into Compact Codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012. 1, 2

[17] H. J. Kim, E. Dunn, and J.-M. Frahm. Predicting Good Features for Image Geo-Localization Using Per-Bundle VLAD. In *Proc. ICCV*, 2015. 2

[18] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet Classification with Deep Convolutional Neural Networks. In *Proc. NIPS*, 2012. 2

[19] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, 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. 2

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single Shot Multibox Detector. In *Proc. ECCV*, 2016. 5

[21] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 2004. 1, 2

[22] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust Wide-Baseline Stereo from Maximally Stable Extremal Regions. *Image and Vision Computing*, 2004. 2

[23] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas. Working Hard to Know your Neighbor's Margins: Local Descriptor Learning Loss. In *Proc. NIPS*, 2017. 2

[24] D. Mishkin, F. Radenovic, and J. Matas. Repeatability Is Not Enough: Learning Affine Regions via Discriminability. In *Proc. ECCV*, 2018. 2, 7, 8

[25] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han. Large-Scale Image Retrieval with Attentive Deep Local Features. In *Proc. ICCV*, 2017. 2, 3, 5, 7, 9

[26] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object Retrieval with Large Vocabularies and Fast Spatial Matching. In *Proc. CVPR*, 2007. 1, 5

[27] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in Quantization: Improving Particular Object Retrieval in Large Scale Image Databases. In *Proc. CVPR*, 2008. 1, 5

[28] F. Radenović, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum. Revisiting Oxford and Paris: Large-Scale Image Retrieval Benchmarking. In *Proc. CVPR*, 2018. 2, 3, 5, 7, 8

[29] F. Radenović, G. Tolias, and O. Chum. CNN Image Retrieval Learns from BoW: Unsupervised Fine-Tuning with Hard Examples. In *Proc. ECCV*, 2016. 2

[30] F. Radenović, G. Tolias, and O. Chum. Fine-tuning CNN Image Retrieval with No Human Annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 1, 5, 7

[31] A. S. Razavian, J. Sullivan, S. Carlsson, and A. Maki. Visual Instance Retrieval with Deep Convolutional Networks. *ITE Transactions on Media Technology and Applications*, 2016. 2, 5

[32] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Proc. NIPS*, 2015. 2, 5, 9

[33] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation. In *Proc. CVPR*, 2018. 5

[34] O. Simeoni, Y. Avrithis, and O. Chum. Local Features and Visual Words Emerge in Activations. In *Proc. CVPR*, 2019. 7

[35] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *Proc. ICLR*, 2015. 2

[36] J. Sivic and A. Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *Proc. ICCV*, 2003. 1, 3

[37] R. Tao, E. Gavves, C. G. M. Snoek, and A. W. M. Smeulders. Locality in Generic Instance Search from One Example. In *Proc. CVPR*, 2014. 2, 5

[38] G. Tolias, Y. Avrithis, and H. Jegou. Image Search with Selective Match Kernels: Aggregation Across Single and Multiple Images. *IJCV*, 2015. 1, 3, 4, 5, 6, 7

[39] G. Tolias and H. Jegou. Visual Query Expansion with or without Geometry: Refining Local Descriptors by Feature Aggregation. *Pattern Recognition*, 2014. 5

[40] G. Tolias, R. Sicre, and H. Jégou. Particular Object Retrieval with Integral Max-Pooling of CNN Activations. In *Proc. ICLR*, 2015. 1, 2, 5

[41] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective Search for Object Recognition. *IJCV*, 2013. 2, 5

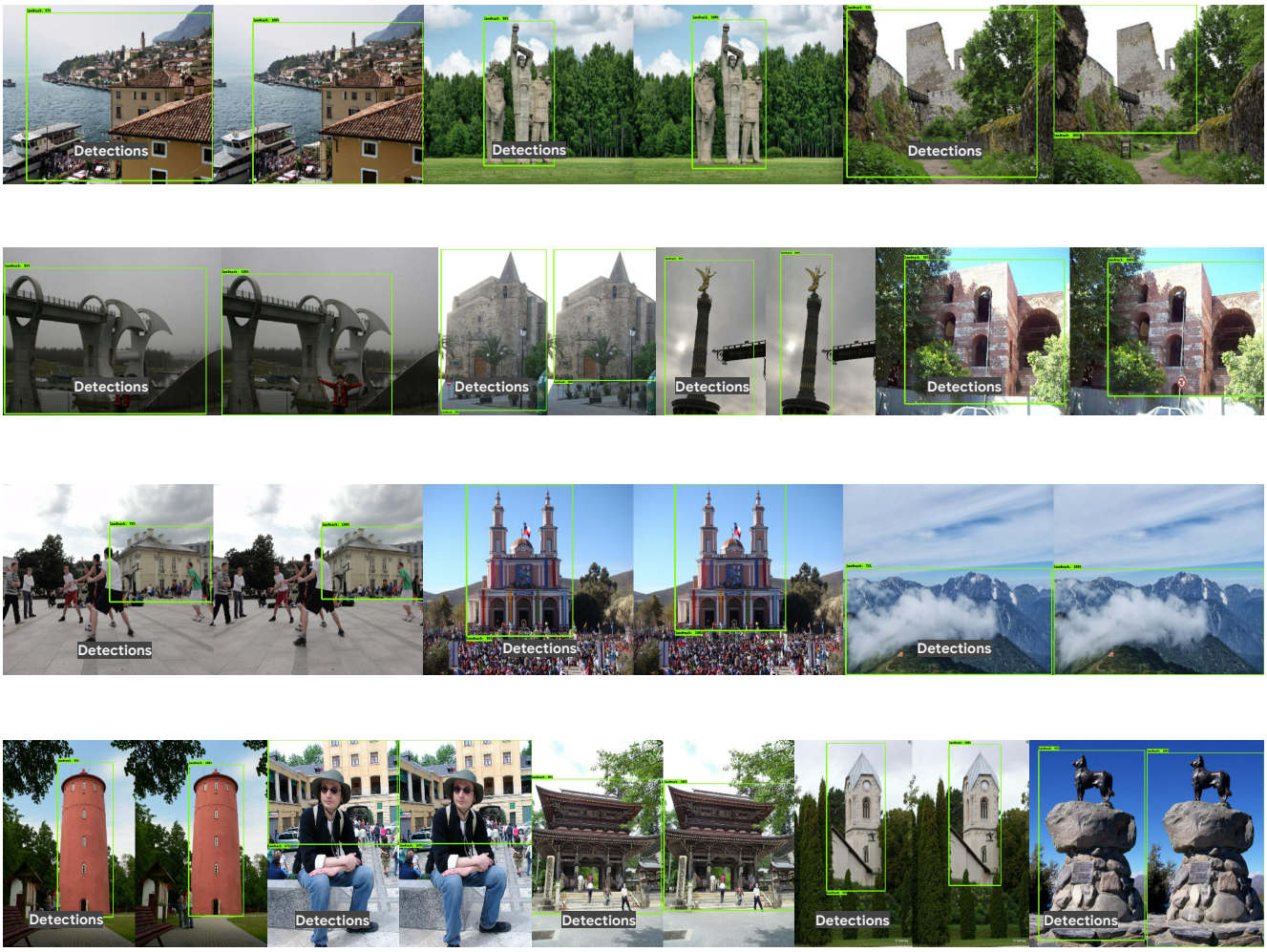


Figure 8: Detection (on the left) versus ground-truth (on the right) on the Google Landmarks dataset.

Figure 9: Two failure detection cases. On the right are the ground-truth images, and on the left are the outputs of the detector (if any).

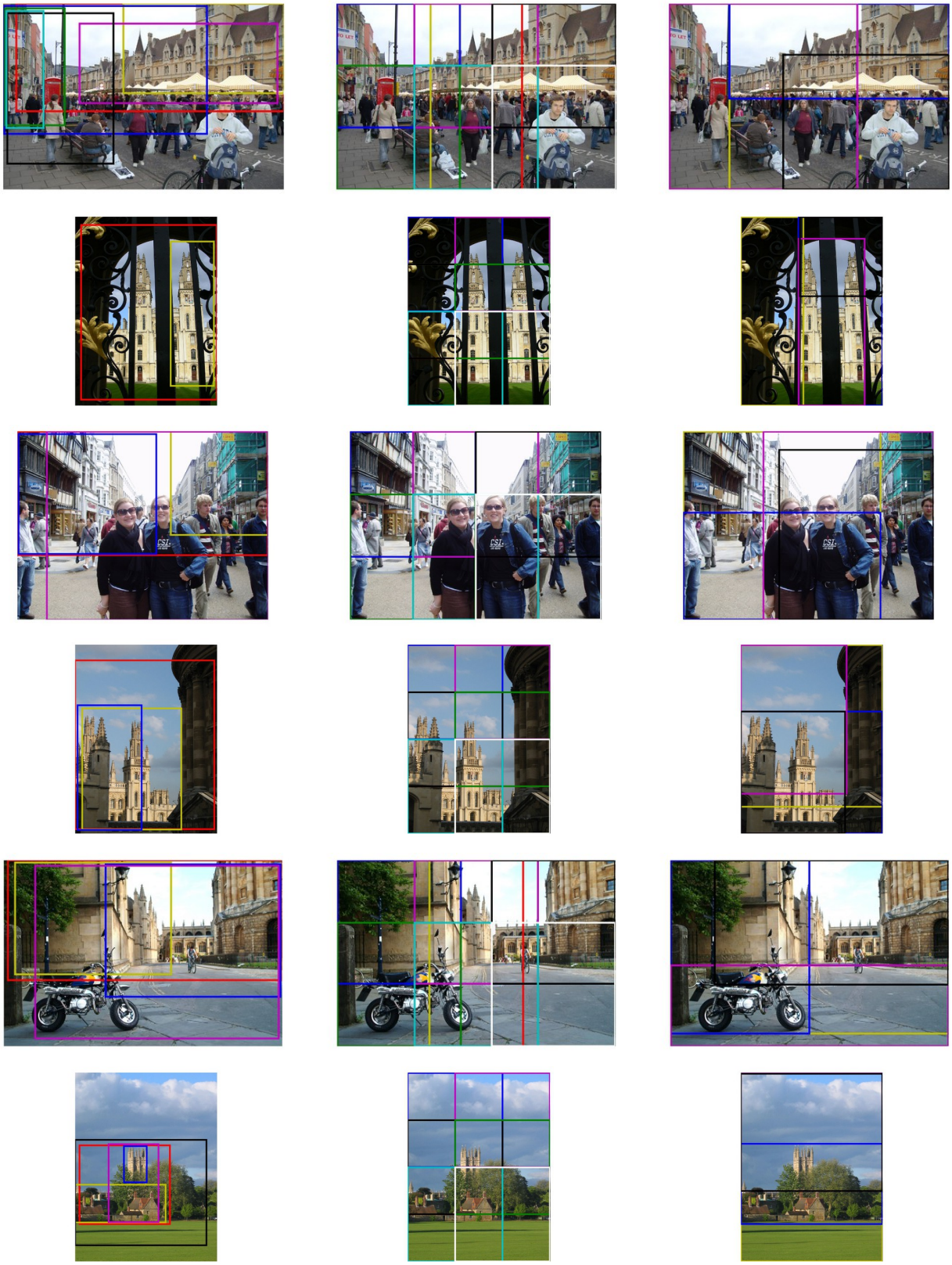


Figure 10: Examples of selected regions for the three methods compared in the paper, on the $\mathcal{R}$Oxford dataset. Left: our D2R approach, with detection threshold of 0.1 (4.1 regions per image). Center: RMAC boxes (fixed multi-scale grid), with 2 levels (9 regions per image). Right: Selective search, with 6 regions per image. Note that edges for some regions overlap in some cases, so not all regions may be clearly visible.

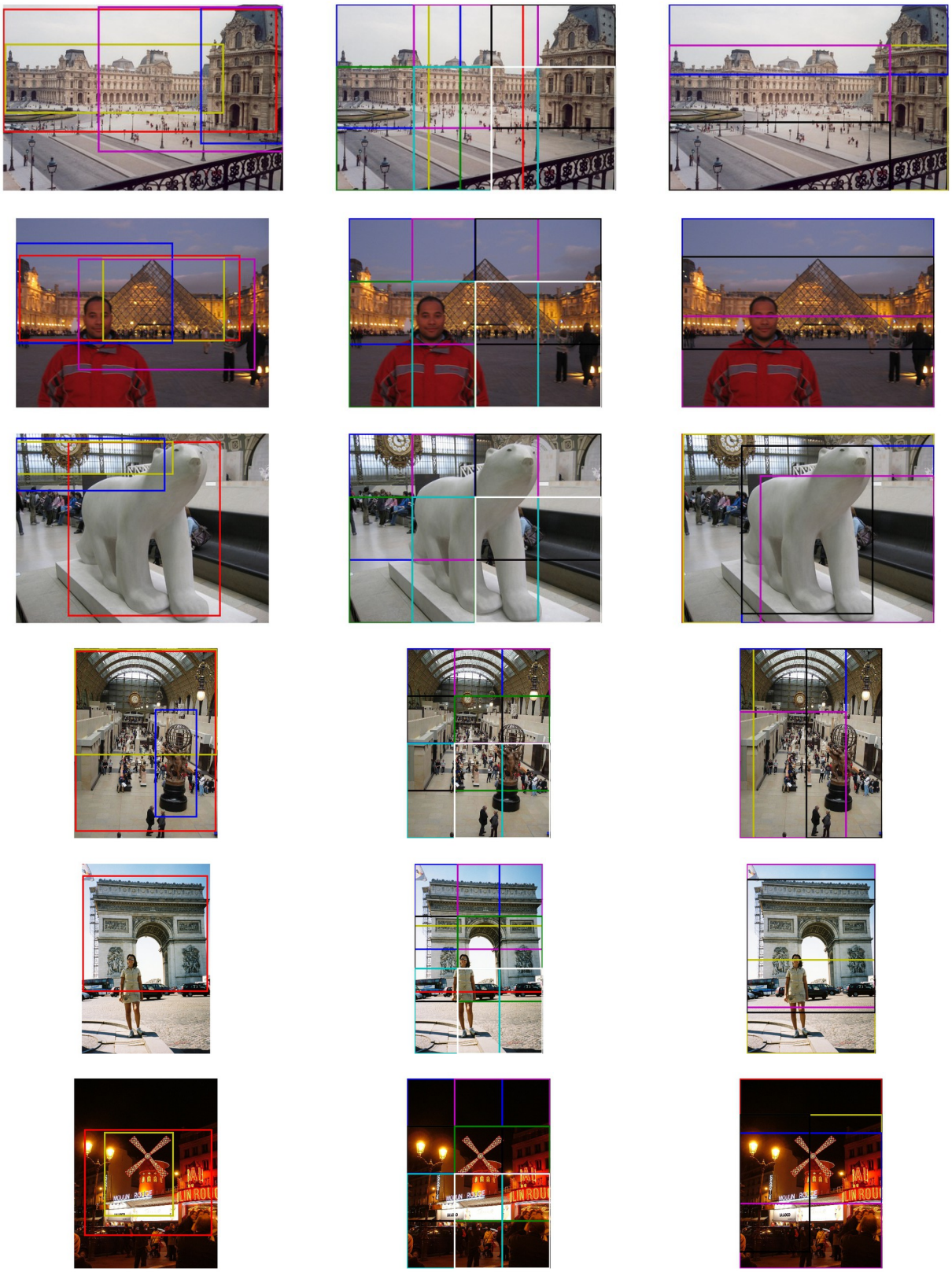


Figure 11: Examples of selected regions for the three methods compared in the paper, on the $\mathcal{R}$Paris dataset. Left: our D2R approach, with detection threshold of 0.1 (3.9 regions per image). Center: RMAC boxes (fixed multi-scale grid), with 2 levels (9 regions per image). Right: Selective search, with 6 regions per image. Note that edges for some regions overlap in some cases, so not all regions may be clearly visible.