Citation: Mtope, Franck Romuald Fotso and Wei, Bo (2020) Region-DH: Region-based Deep Hashing for Multi-Instance Aware Image Retrieval. In: 2020 International Joint Conference on Neural Networks (IJCNN 2020). Neural Networks (IJCNN) . IEEE, Piscataway, pp. 2007-2013. ISBN 9781728169279, 9781728169262

Published by: IEEE

URL: https://doi.org/10.1109/IJCNN48605.2020.9207485 <https://doi.org/10.1109/IJCNN48605.2020.9207485>

Region-DH: Region-based Deep Hashing for Multi-Instance Aware Image Retrieval

Franck Romuald Fotso Mtope[†], Bo Wei[‡]
[†]Research and Innovation, Cognitive Data System SARL, Yaoundé, Cameroon
[‡]Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne, United Kingdom
Email: [†]franck@conids.cm, [‡]bo.wei@northumbria.ac.uk

***Abstract*—This paper introduces an instance-aware hashing approach Region-DH for large-scale multi-label image retrieval. The accurate object bounds can significantly increase the hashing performance of instance features. We design a unified deep neural network that simultaneously localizes and recognizes objects while learning the hash functions for binary codes. Region-DH focuses on recognizing objects and building compact binary codes that represent more foreground patterns. Region-DH can flexibly be used with existing deep neural networks or more complex object detectors for image hashing. Extensive experiments are performed on benchmark datasets and show the efficacy and robustness of the proposed Region-DH model.**

***Index Terms*—Imaging hashing, deep learning**

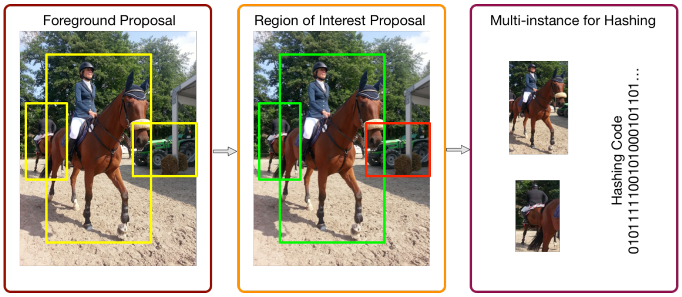

Fig. 1. Multi-instance for Hashing

I. Introduction

The Content-Based Image Retrieval (CBIR) technique is broadly used in many applications, such as fingerprint identification [2], crime prevention [21], etc. CBIR searches similar images in a large scale database from a query image based on a pairwise comparison of features extracted from images. The hashing mechanism is a key technique commonly used by CBIR to reduce feature size, improve search speed, and minimize storage. The hashing process mainly aims to extract a low-dimensional features representation. Many research works have been conducted in hashing mechanisms [11], [15], [17], [19].

The existing hashing techniques can be categorized according to two criteria: the nature of data use and the training process. Regarding the nature of data use, unsupervised and supervised methods are two main groups of image hashing approaches. The unsupervised methods can learn hashing functions from unlabeled training data [8], [11], [16], [20], [22], while the supervised methods instead leverage semantic information in training data for learning hash functions [10], [13], [15], [17]. Motivated by the successful application of deep learning in the imaging processing and computer vision domain, many research works have introduced deep learning into image hashing and taken advantage of deep Convolutional Neural Networks (referred to as deep hashing networks) to analyze labeled images, extract usable patterns and further create an improved feature representation for hash functions [12], [14], [23], [25], [26]. The existing hashing methods also have different training strategies. Pairwise and point-wise are two main approaches to learn hashing functions. The pairwise approaches [12], [23], [26] use a pair of images containing similar objects to learn the similarity between the derived patterns and features. In contrast, the point-wise approaches [14], [25] use one image to learn features automatically during each training iteration when using deep learning techniques.

These supervised methods have achieved remarkable performance on single or multiple label datasets. However, according to our best knowledge, the existing methods mainly focus on semantic information for the similarity ranking. In other words, these methods have a strong assumption that images have a good quality of image labels, and the objects of interest occupy the majority of the labeled images. However, they cannot be directly applied for real application scenarios without further optimization because captured images usually contain multiple objects and Region of Interest (ROI) is only a part of the whole image. Different from most of the existing works, one instance-aware similarity technique [12] uses one set of region proposals during the training process. Instances here are the region proposals with label probabilities above a threshold. This technique has the huge potential to be widely deployed because of its ability to recognize objects and analyze their similarity from ROI instead of the whole image. However, this introduction of object proposals involves a new challenge that inaccurate bounding boxes will result in extra bias. To solve this problem and improve the qualities of learned bounding boxes, instance features are considered and leveraged for multi-label image retrieval and region proposals.

In this research, we propose Region-DH, a novel region-based deep hashing method. The proposed method simultaneously learns hash functions and object bounds by taking advantage of the deep learning techniques (as shown in Figure 1). We also propose a multi-task loss function, which involves components required to map instance features into efficient binary codes for object detection. To summarize, the main contributions in this paper are listed as follows:

- We propose an innovative unified deep neural network for simultaneous object detection and image hashing.
- An efficient and effective instance-aware hashing is proposed, which takes advantage of more accurate object bounds with the consideration of object recognition. The proposed method reduces the bias caused by incorrect object proposals and improves hashing performance.
- A hash code for each image is derived using the proposed method. The hash code also combines and encodes instance information which contains the object proposals.
- We extensively evaluate Region-DH on two public multi-label datasets. The experiments have shown that the proposed method can successfully incorporate instance information into compact binary codes, which outperforms existing state-of-the-art hashing techniques.

The following of this paper is organized as follows: Section II shows the related works pertaining to the image retrieval systems. Section III describes our proposed deep neural network. Section IV shows the configuration and process of our experiments, then the quantitative analysis of our results. Section V concludes this research.

II. Related Works

A. Hashing Methods

In computer vision, hashing methods are commonly used to compress image representation into a small vector while preserving the semantic information. In this section, we introduce the hashing methods. First, we will describe the unsupervised methods. Unsupervised methods use a set of unlabeled data to learn hash functions. They aim to catch common patterns hidden in feature representation and generalize them into compact hash codes while preserving similarity between images. Representative techniques are Kernelized LSH (KLSH) [11], Semantic Hashing [19], Anchor Graph Hashing [16], Spectral Hashing [22] and Iterative Quantization (ITQ) [8]. Supervised methods instead learn hash functions from labeled data to derive similar binary codes from similar images [10], [13], [15], [17]. Data are grouped by semantic to learn hash functions. Significant techniques are Binary Reconstruction Embedding (BRE) [10], Supervised Hashing with kernels (KSH) [15], Minimal Loss Hashing (MLH) [17], and Column Generation Hashing (CGHash) [13]. Moreover, deep neural networks have also been successfully applied for hashing methods, called deep hashing [12], [14], [23], [25], [26]. They take advantage of deep features learned by deep neural networks to approximate hash codes. Instance-Aware Hashing (IAH) [12] learns instance-aware image representations for multi-label data. Deep Multilevel Semantic Similarity Preserving Hashing (DMSSPH) [23] takes advantage of labeled data for supervised methods and learns compact binary codes for multi-label data and takes the best use of supervised data in the form of labels to maximize the distance of dissimilar pairs. Lin et al [14] propose a deep learning framework to learn hash codes by using a hidden layer to represent the latent concepts that dominate each category. Instance Similarity Deep Hashing (ISDH) [26] defines a new pair-wise similarity metric preserving hashing into an instance similarity hashing. The instance similarity is simultaneously used for feature learning and hash coding. SSDH [25] learns hash functions as a latent layer by minimizing an objective function defined over classification error and other hash code properties. Regarding the training process, whether data are labeled or unlabeled, there are two main approaches: the pair-wise approach and point-wise approach. The pair-wise (or triplet-wise) approach consists of using two (or three) input images to learn over the similar or dissimilar samples during the training phase. Although this approach has shown significant performance on previous works (e.g., [12], [23], [26]), they require a high cost for computation and storage. The point-wise approaches [14], [25] efficiently learn binary codes over deep features generated via transfer learning while satisfying a classification objective.

B. Object Detection

Recently, computer vision research has observed significant improvements with the success of deep convolutional networks. Well-known object detection techniques or frameworks are Regional CNN (R-CNN) [7], SPPnet [9], Fast R-CNN [6], Faster R-CNN [18]. R-CNN [7] aims to combine region proposals with convolutional neural networks to localize and segment objects. SPPnet [9] allows the sharing of convolutional layers between object proposals and generates a fixed-length representation regardless of image size. Fast R-CNN [6] proposes a new training algorithm that fixes the downside of R-CNN and SPPnet. Faster R-CNN [18] introduces a Region Proposal Network (RPN) and shares full-image convolutional features with a detection network without a huge amount of extra cost in object proposal generation. Our paper instead aims to increase the performance of instance-aware hashing for a fast and efficient generation of hash codes. IAH [12] addressed the problem of instance-aware hashing by using object proposals, while SSDH [25] relies on strong hashing constraints with a classification objective. Differently from these two works, we propose a unified deep neural network that focuses on objects detected and learns similarity beyond the image labels.

III. Region-based Deep Hashing

A. The Overview of the Proposed Architecture

Figure 2 shows the overview of the proposed Region-Based Deep Hashing (Region-DH) for multi-instance image retrieval. This proposed architecture includes 4 main modules: object location, bounding box regression, instance-level hashing, and image-level hashing. In the following part, we will overview these main modules.

Object Location (labeled as (1) in Figure 2): this module involves three sequential components, i.e. a backbone deep neural network (e.g. VGG16), a region proposal network (RPN), and the loss functions (penalizing the regions proposed by RPN). The backbone network in this module first learns

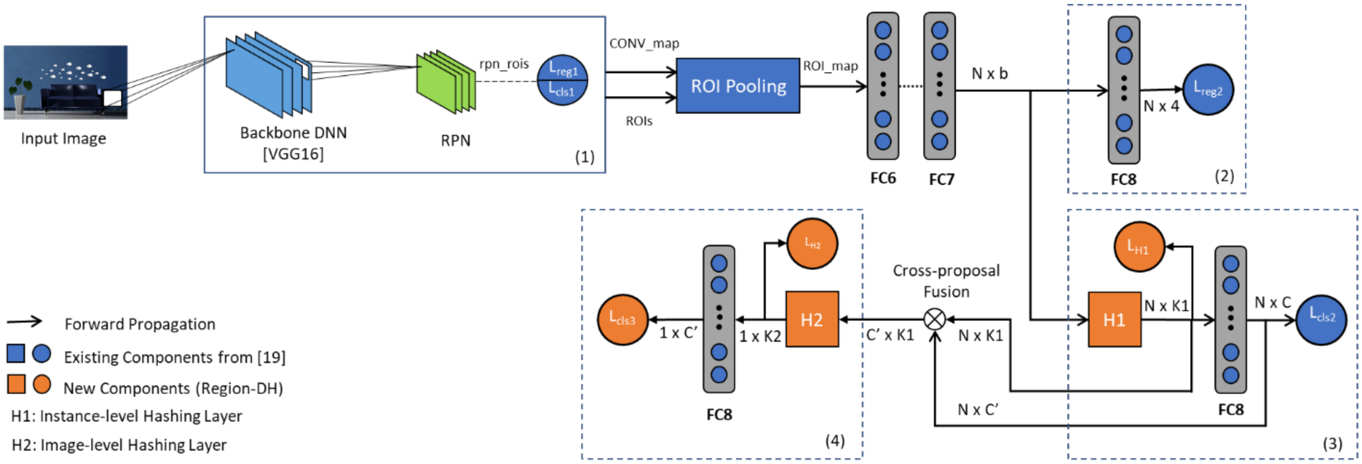

Fig. 2. The overview of the proposed Region-Based Deep Hashing (Region-DH) for Multi-instance image retrieval. This architecture relies on a backbone deep neural network (DNN), e.g. VGG16 or ResNet [24], coupled with a Region Proposal Network (RPN). Region-DH takes an image as the input. The backbone network and RPN are used to learn the features and location of objects. It also determines hashing functions via both instance-level hashing (H1) and image-level hashing (H2).

features and extract patterns from input images. These features are further used in an RPN. RPN will make out a set of multiple regions of interest (denoted as rpn_roi). Moreover, two loss functions [18] L_{reg1} and L_{cls1} are employed here. L_{reg1} and L_{cls1} measure the regression loss on proposal boxes and the cross-entropy loss on prediction, respectively. The main goal of this module is to predict multiple foreground proposals for ROIs, and a proportion N of those will be selected for the ROI pooling and fully connected layers (FC6 and FC7 in Figure 2).

Bounding Box Regression (labeled as (2) in Figure 2): this module focuses on predicting bounding box coordinates through features generated by FC7. These features are fed into a fully connected layer FC8, which carries out a $N \times 4$ matrix for the previously selected N ROIs and their predicted coordinates. The loss function [18] L_{reg2} is used here to penalize coordinates predicted.

Instance-level Hashing and Classification: this module (labeled as (3) in Figure 2) handles the hashing process over the regions resulted from the previous modules. One hashing code will be gauged for each instance. A set of hashing functions are learned via the latent layer H1 from the fully connected layer FC7. The latent layer H1 aims to encode FC7 activations, which have a significant impact on the instance classification objective. An $N \times K1$ matrix is obtained from the latent layer H1 for representing the $K1$-bits binary codes for the N ROIs. The fully connected layer FC8, along with a softmax activation function, produces a $N \times C$ matrix that is imposed for the classification purpose. Specifically, FC8 conducts prediction for the N ROIs. In this module, two loss functions L_{H1} and L_{cls2} are used to penalize the instance binary codes and prediction, respectively.

Cross-proposal Fusion: In this component, we aim to embed the ROI prediction probabilities into instances features and finalize instance-level hashing. One similar method is suggested in [12]. We consider the ROI probabilities from the instance-level hashing along with object recognition probabilities from the classification module. In other words, this cross-proposal fusion penalizes the embeddings with low recognition confidence and the background proportion in one image.

Image-level Hashing: this module (labeled as (4) in Figure 2) is after the cross-proposal fusion to generate hashing code for images. The cross-proposal fusion finalize instance-aware embedding with a $C' \times K1$ matrix, which are used to learn hashing functions in the latent layer H2 of this module. The latent layer H2 encodes the relevant embedding features and produces the final $K2$-bits binary code representing both the semantic information and features for images, which are employed for further the image classification. This module also uses two loss functions L_{H2} and L_{cls3} to penalize the image binary code and the image multi-label classification, respectively. We finally use a softmax function for the classification purpose.

In the following parts of this section, we demonstrate the details of the key components of our proposed method.

B. Binary Code Learning

In this section, we will show the details of binary code learning that is used in layers H1 and H2, as shown in Figure 2 (described in Sec. 3.2). Relying on a learning process, this component generates binary codes for both instances and images. To achieve this, the latent layers H1 and H2 encode the activated weights of previous layers into binary values {0,1}. We use a binary hashing method from [25] in H1 and H2, as shown in the following equation:

$$q_n = \frac{1}{2}\left[\operatorname{sign}\left(A_n^{L-1} - 0.5\right) + 1\right] \tag{1}$$

$$sign(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{2}$$

where L is the index of the hashing layer, A_n^{L-1} is the activation matrix of the layer FC7, and q_n is the approximate values for the n-th training sample. Moreover, the binary approximation of A_n values needs to follows two strong constraints, i.e., the balance property and evenly distribution. More details about the constraints can be found in [25].

C. Overall Learning Objective

The proposed architecture aims to realize the accurate, efficient, and simultaneous ROI localization, object classification, and hashing. To achieve these, we use the following overall multi-task loss functions:

$$L = \alpha L_{det} + \beta L_{cls} + \gamma L_H \tag{3}$$

where the weighting hyperparameters α, β, γ control the impact of the ROI detection loss L_{det}, the classification loss L_{cls}, and the hashing loss L_H, respectively. We show the definitions for these three loss functions in the following.

Equation 4 demonstrates the loss function for ROI detection.

$$L_{det} = \alpha_1 L_{reg_1} + \alpha_2 L_{reg_2} \tag{4}$$

where L_{reg_1} and L_{reg_2} refer to the regression losses for ROI detection in module (1) and (2) with their corresponding weighting hyperparameters α_1 and α_2.

Equation 5 demonstrates the loss function for the components related to classification.

$$L_{cls} = \beta_1 L_{cs1} + \beta_2 L_{cs2} + \beta_3 L_{cls3} \tag{5}$$

where L_{cs1}, L_{cs2} and L_{cls3} are multi-class cross-entropy losses for classification purposes (described in section 3.2) with their corresponding weighting hyperparameters β_1, β_2, β_3. L_{cs1}, L_{cs2} and L_{cls3} are used in modules (1) (3) (4) as shown in Figure 2.

Equation 6 demonstrates the loss function for hashing functions.

$$L_H = \gamma^{H_1} L_{H_1} + \gamma^{H_2} L_{H_2} \tag{6}$$

where L_{H_1} and L_{H_2} are hashing losses with the weighting hyperparameters γ^{H_1} and γ^{H_2} The loss function for hashing $L_{H_i}, i \subset 1, 2$ is defined in the following equation:

$$L_{H_i} = \gamma_1^{H_i} E_1^{H_i} + \gamma_2^{H_i} E_2^{H_i} \tag{7}$$

We use $E_1^{H_i}$ and $E_2^{H_i}$ for the hashing constraints as shown in Equation 8 and Equation 9. $i = 1$ is for instance-level hashing, while $i = 2$ for image-level hashing.

$$E_1^{H_i} = -\frac{1}{k} \sum_{n=1}^{N} \sum_{r=1}^{R} \| q_{nr}^{H_i} - 0.5e \| \tag{8}$$

$$E_2^{H_i} = \sum_{n=1}^{N} \sum_{r=1}^{R} \| \operatorname{mean}(q_{nr}^{H_i}) - 0.5 \| \tag{9}$$

where N is the number of training samples in each batch, R is the number of ROIs found for each image, e is a k-dimensional identity vector. $q_{nr}^{H_i}$ is the binary hashing code vector from the latent layer H1 for the n-th training sample and r-th ROI.

IV. Evaluation

To show the performance of our proposed approach Region-DH, we use two public datasets to demonstrate the effectiveness and robustness of our method. We carry out extensive experiments on different benchmarks. In this section, we will show the details of the evaluations.

A. Datasets

In this paper, we will use the following two datasets to evaluate the proposed methods.

VOC 2007 [4]: It is a dataset widely used for object detection, image segmentation, and image retrieval. It consists of 9,963 images annotated for 20 categories of objects, and the images are multi-labeled. These images are organized as test set (4,952 images), training set (2,501 images), training and validation set (5,011 images), and validation set (2,510 images) to facilitate image processing and machine learning relevant research.

VOC 2012 [5]: Similar as VOC 2007, VOC 2012 has 20 categories and 11,540 multi-labeled images. This dataset has a training set (5,717 images), and a validation set (5,823 images). Different from VOC 2007, VOC 2012 does not contain a test set.

B. Metrics

We use the following two commonly used evaluation metrics adopted: Average Cumulative Gains (ACG) and Mean Average Precision (MAP), which are also adopted in previous works [12], [25]. ACG is mainly used for multi-label image retrieval as defined in Equation 10.

$$ACG = \frac{1}{k} \sum_{i=1}^{k} s(i) \tag{10}$$

It is the mean of the similarities $s(i)$ between the query image and each of the top-k matches or retrieved images. The similarity $s(i)$ is the number of shared labels between the i retrieved image and the query image.

MAP is also commonly used by any image retrieval system to assess the precision of the retrieval system and find k similar images which share at least one label with the query one. MAP is defined in Equation 11.

$$MAP = \frac{1}{N_r} \sum_{i=1}^{k} \frac{N_r(i)}{i} r(i) \tag{11}$$

where N_r is the number of relevant image corresponding to the query image. $r(i)$ is binary to show the relevance for the i-th image in the query image, i.e. $r(i) = 1$ if there is at least one share label. $N_r(j)$ is the number of relevant images in the top i images.

C. Experiment Details

To conduct our experiments and evaluate the proposed method, we adopt the following settings for the proposed method Region-DH and baselines used for evaluations. We mainly compare the proposed method with DLBHC [14], SSDH [25], and IAH [12][1]. Since we implement Region-DH with TensorFlow [1] instead of Caffe from their official code, we re-implement DLBHC and SSDH using the same TensorFlow framework for a fair comparison of performance. The results regarding IAH are from [12] with the use of GoogleNet.

Concerning the implementation of our method Region-DH, we use VGG16 [20] as the backbone network to compute convolutional features. The same network is used for the baselines DLBHC and SSDH. The used backbone network weights are initialized with a VGG16 model pretrained on ImageNet [3], and the initial learning rate is set to 0.001. The latent layers in the proposed model have the same configuration as [25]. Specifically, we use a fully connected layer initialized by a normal distribution with a zero-mean and a standard deviation of 0.005. The number of outputs of the latent layer is the length of the binary codes. The learning algorithm is the stochastic gradient with the momentum set to 0.9 and a weight decay of 0.0001.

For the multi-task loss function, we set all the weights to 1 for the equal consideration of every component. For the hashing, parameters are set as $\gamma^{H_1} = 0.05$ and $\gamma^{H_2} = 10$. When re-implementing baselines, we use the default settings and achieve the equivalent performance in the official DLBHC [14] and SSDH [25] implementation. These settings ensure the convergence of the network and show the equivalent performances in the existing studies on the test set. In our evaluation, we train the baselines and Region-DH on three lengths of hash code, i.e., 32, 48, and 64 bits.

D. Results

1) Ranking on VOC 2007: In the dataset VOC 2007, we train the baselines selected and our method on the training and validation set (5,011 images). To perform the retrieval evaluation, we use the test set (4,952 images) as query images and the training and validation set as the target images.

For each training process, we first use the latest checkpoint model to extract hash codes and store them into an HDF5 file. Then, we use each query hash code to search for the top-k similar images in the target images. We fix k as 1,000 in our experiment, perform queries for each image, and compare the proposed method with baselines using the metrics ACG and MAP introduced in Section IV-B.

Table I and Table II show the comparison between the previous works (DLBHC, SSDH, IAH) and the proposed method Region-DH, under evaluation metrics: ACG and MAP. As shown in Table I for the performances in ACG, the proposed method has an improvement of 3.18% - 4.79% compared with IAH ,5.08% - 6.89% compared with SSDH, and 7.37% - 12.16% compared with DLBHC. Table II shows their performances in MAP, where it can be seen that the proposed method has a considerable improvement of 3.3% - 4.59% compared with IAH and 5.46% - 7.5% compared with SSDH, and 7.06% - 30.67% compared with DLBHC.

TABLE I

Comparison of Region-DH against baselines w.r.t. ACG

Methods	VOC 2007 ACG		
	32 bits	48 bits	64 bits
DLBHC	0.5646	0.6044	0.6311
SSDH	0.6354	0.6271	0.6359
IAH	0.6436	0.6514	0.6569
Region-DH	**0.6862**	**0.6832**	**0.7048**

TABLE II

Comparison of Region-DH against baselines w.r.t. MAP

Methods	VOC 2007 MAP		
	32 bits	48 bits	64 bits
DLBHC	0.5965	0.7946	0.8582
SSDH	0.8282	0.8595	0.8715
IAH	0.8702	0.8765	0.8829
Region-DH	**0.9032**	**0.9141**	**0.9288**

2) Ranking on VOC 2012: Because there is no test set in VOC 2012, we train our models on the train set with 5,717 images and use them as the target images for the retrieval process. The validation set is for the queries (5,823 images)[2]. Then, we apply queries images to find the best matches.

Table III and Table IV show the performance of our method Region-DH against the baselines on the VOC 2012 dataset. Concerning the ACG, the proposed Region-DH outperforms the baselines with an improvement of 0.82% - 4.56% compared with IAH, 3.35% - 6.41% compared with SSDH, and 7.37% - 9.04% compared with DLBHC. Concerning the MAP, our method demonstrates an improvement of 2.92% - 5.60% compared with IAH, 4.13% - 4.49% compared with SSDH, and 4.84% - 24.06% compared with DLBHC.

TABLE III

Comparison of Region-DH against baselines w.r.t. ACG

Methods	VOC 2012 ACG		
	32 bits	48 bits	64 bits
DLBHC	0.5741	0.5879	0.6151
SSDH	0.6143	0.6200	0.6248
IAH	0.6396	0.6465	0.6433
Region-DH	**0.6478**	**0.6783**	**0.6889**

TABLE IV

Comparison of Region-DH against baselines w.r.t. MAP

Methods	VOC 2012 MAP		
	32 bits	48 bits	64 bits
DLBHC	0.6282	0.8273	0.8625
SSDH	0.8239	0.8616	0.8696
IAH	0.8396	0.8579	0.8549
Region-DH	**0.8688**	**0.9041**	**0.9109**

[1] The official code for DLBHC and SSDH is publicly available in https://github.com/kevinlin311tw/caffe-cvprw15 and https://github.com/kevinlin311tw/Caffe-DeepBinaryCode

[2] The same use of the VOC 2012 dataset can also be found in [12]

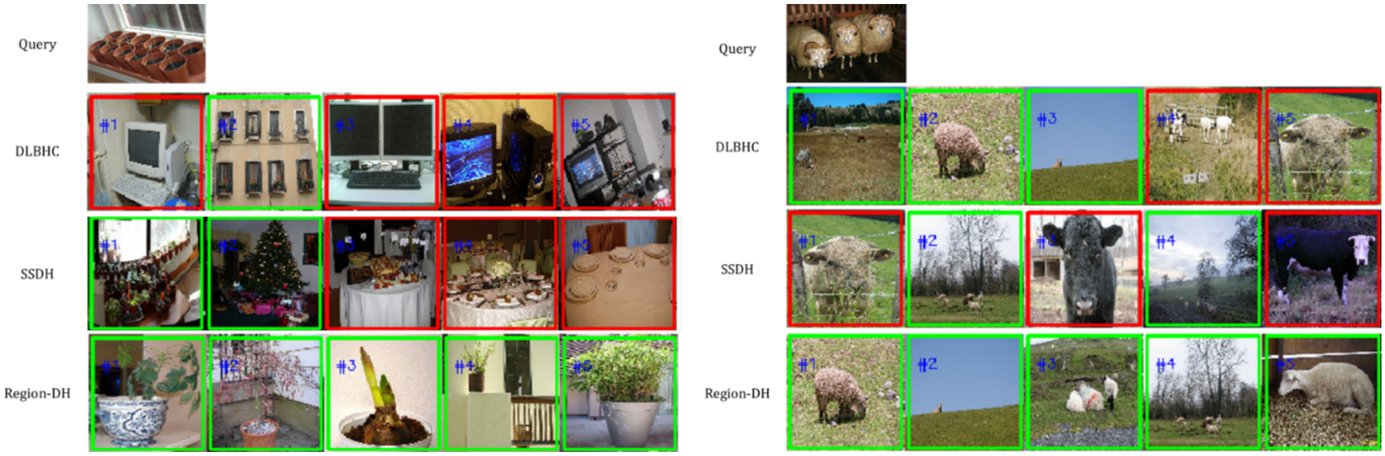

Fig. 3. Visual comparison of retrieved images for Region-DH and the baselines (DLBHC & SSDH). Left: results for a query image with label "pottedplant", Right: results for a query image with label "sheep". Each row represents the top-5 retrieved images (rank: from left to right). "red" highlights no shared labels (dissimilar images), "green" highlights at least one shared label (similar images).

3) Visualization of results: Following the evaluation process described for VOC 2007 and VOC 2012, we run queries and select the top-5 of the best matches to show the performance of the proposed method Region-DH visually. Figure 4 shows a visual comparison of Region-DH against the baselines DLBHC and SSDH. From these examples, we can see that Region-DH leverages the instance information well and retrieves similar images by building compact binary codes. Specifically, from Figure 3 (left), we query an image with label "pottedplant" and get top-5 correct retrieval with Region-DH, while we get respectively 4 and 3 incorrect on DLBHC and SSDH. In Figure 3 (right), we query an image with label "sheep", and we get accurate results with our method, while DLBHC and SSDH show significant mistakes on the top-5 returned images.

4) Summary of Evaluation: Overall, Region-DH shows a relative improvement of performance over the baselines selected, i.e. DLBHC, SSDH and IAH. We demonstrate its efficacy and robustness via the metrics (ACG, MAP) in various datasets (VOC 2007, VOC 2012).

V. Conclusion

In this paper, we have described a region-based deep hashing architecture (Region-DH), which unifies object detection, classification and hashing tasks for multi-label image retrieval. Region-DH can leverage semantic information and patterns hidden into objects to build efficient binary codes. The proposed Region-DH relies on a multi-task loss that aims to improve the hashing process while correcting the locations of objects. For future work, we plan to investigate techniques to stabilize the learning process with additional objectives (e.g., segmentation) into the multi-task loss. In addition, we will assess the impact of the hashing objective on the object detection task.

References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.

[2] R. S. Choras. Image feature extraction techniques and their applications for cbir and biometrics systems. *International journal of biology and biomedical engineering*, 1(1):6–16, 2007.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[6] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[8] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2916–2929, 2012.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

[10] B. Kulis and T. Darrell. Learning to hash with binary reconstructive embeddings. In *Advances in neural information processing systems*, pages 1042–1050, 2009.

[11] B. Kulis and K. Grauman. Kernelized locality-sensitive hashing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(6):1092–1104, 2011.

[12] H. Lai, P. Yan, X. Shu, Y. Wei, and S. Yan. Instance-aware hashing for multi-label image retrieval. *IEEE Transactions on Image Processing*, 25(6):2469–2479, 2016.

[13] X. Li, G. Lin, C. Shen, A. v. d. Hengel, and A. Dick. Learning hash functions using column generation. *arXiv preprint arXiv:1303.0339*, 2013.

[14] K. Lin, H.-F. Yang, J.-H. Hsiao, and C.-S. Chen. Deep learning of binary hash codes for fast image retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 27–35, 2015.

[15] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang. Supervised hashing with kernels. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2074–2081. IEEE, 2012.

[16] W. Liu, J. Wang, S. Kumar, and S.-F. Chang. Hashing with graphs. 2011.

[17] M. Norouzi and D. M. Blei. Minimal loss hashing for compact binary codes. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 353–360. Citeseer, 2011.

[18] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[19] R. Salakhutdinov and G. Hinton. Semantic hashing. *International Journal of Approximate Reasoning*, 50(7):969–978, 2009.

[20] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[21] J. Singh, J. S. Kaleka, and R. Sharma. Different approaches of cbir techniques. *Int. J. Comput. Distributed Syst*, 1:76–78, 2012.

[22] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *Advances in neural information processing systems*, pages 1753–1760, 2009.

[23] D. Wu, Z. Lin, B. Li, M. Ye, and W. Wang. Deep supervised hashing for multi-label and large-scale image retrieval. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, pages 150–158. ACM, 2017.

[24] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[25] H.-F. Yang, K. Lin, and C.-S. Chen. Supervised learning of semantics-preserving hash via deep convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 40(2):437–451, 2017.

[26] Z. Zhang, Q. Zou, Q. Wang, Y. Lin, and Q. Li. Instance similarity deep hashing for multi-label image retrieval. *arXiv preprint arXiv:1803.02987*, 2018.