Incorporation of Semantic Segmentation Information in Deep Hashing Techniques for Image Retrieval

Konstantinos Gkountakos, Theodoros Semertzidis, Georgios Th. Papadopoulos, Member, IEEE, and Petros Daras, Senior Member, IEEE

Information Technologies Institute, Centre for Research and Technology Hellas, Greece
Email: {gountakos,theosem,papad,daras}@iti.gr

***Abstract*—Extracting discriminative image features for similarity search in nowadays large-scale databases becomes an imperative issue of paramount importance. To address the so called task of Approximate Nearest Neighbor (ANN) search in large visual dataset, deep hashing methods (i.e. approaches that make use of the recent deep learning paradigm in computer vision) have recently been introduced. In this paper, a novel approach to deep hashing is proposed, which incorporates local-level information, in the form of image semantic segmentation masks, during the hash code learning step. The proposed framework makes use of pixel-level classification labels, i.e. following a point-wise supervised learning methodology. Experimental evaluation in the significantly challenging domain of on-line terrorist propaganda video analysis, i.e. a highly diverse and heterogeneous application case, demonstrates the efficiency of the proposed approach.**

***Keywords*—deep hashing; binary codes; deep learning; segmentation mask; training; neural networks**

I. Introduction

The tremendous increase that has been observed in the recent years in the amount of the visual content that is generated and stored on a daily basis has rendered the need for search in the corresponding databases a big challenge. Hashing methods constitute the dominating approach for efficient image retrieval in terms of accuracy and computation time. Hashing methods have very low memory requirements and fast response compared to other approaches [1]. The merits of hashing methods come from the efficient mapping of high dimensional feature vectors to corresponding significantly low-dimensional binary codes, which are subsequently used for time-efficient image retrieval [2]. These mappings are also known as hash functions and the generated binary vectors are typically found as hash codes.

Numerous hashing methods have been proposed so far and can generally be divided in two main categories, namely data-independent and data-dependent methods [2], [3]. Data-independent approaches are not using a training dataset sampled from the target data and thus apply generic approaches to learn or randomly select a mapping of the high dimensional input feature vector to a lower dimensional one. Next, a quantization step follows to result in a compact binary vector that encodes the original vector [4], [5]. Representative method of this category is the Locality Sensitive Hashing (LSH) method [6] and its variants [2], which are selecting projection matrices to lower-dimensional spaces and threshold the vectors to binary codes. On the other side, data-dependent methods aim at learning hash functions from the target dataset to generate more efficient mappings of the input data to the new hamming space [7]. The methods of the data-dependent category can be further divided into supervised and unsupervised ones [2]. The unsupervised approaches aim at learning feature representations based only on the statistics of the target data e.g. the variance of each dimension or its cardinality [8]. Additionally to the statistics of the high-dimensional vectors data, the supervised approaches take also into account the labels of the training data, so that the semantics of the data are also incorporated in the learned hash functions. The advantage of using labeled data to guide the learning process leads supervised methods to generate hash codes that represent better the original data with fewer bits (i.e. smaller hash code length), compared to the ones attained by unsupervised techniques. Small hash code length is desirable for building efficient image retrieval frameworks, with respect to the required computational resources [9], [5]. Representative data-depended methods are Spectral Hashing (SH) [10], Binary Reconstructive Embedding (BRE) [11] and Iterative Quantization (ITQ) [12].

The above-mentioned hashing methods make use of traditional hand-crafted visual descriptors, such as GIST [13] or HOG [14]. However, these hand-engineered descriptors (and consequently the corresponding hash codes) do not efficiently model the original images and their semantics and thus fail to provide a retrieval mechanism of high accuracy. Fortunately, the break-through of Deep Learning (DL) techniques in the computer vision community affected also the binary hashing methodologies, by replacing the hand-crafted descriptors with learnable features extracted directly from deep neural networks, typically Convolutional Neural Networks (CNNs). The corresponding methods that learn end to end representations from the image to feature vectors and finally hash codes are termed deep hashing [3], [5], [15]. Although multiple deep hashing approaches have recently been proposed [16], [1], [4], [2], [3], [5], [9], [15], all presented methods make use of image-level features, i.e. they do not directly incorporate locality and semantic information of the individual objects that are present in an image. Performing the latter would inevitably lead to the generation of more expressive and robust hash codes that would combine image-level information with

978-1-5386-0774-9/17/$31.00 ©2017 IEEE

discriminative object-level information cues.

In this paper, a novel approach to deep hashing for image retrieval is proposed, which takes into account object locality information as well as cues from semantic segmentation of the image objects, during the hash functions learning procedure. In particular, the fundamental consideration of the proposed approach is, apart from global-level features, to incorporate object-level information, so that the estimated hash codes encode better the images' content. In the current work, deep semantic image segmentation techniques are used for providing local-level cues and object classification. More specifically, the proposed approach is essentially composed of two consecutive steps. In the first step, a particular subnetwork is integrated to the overall deep architecture for estimating semantic segmentation maps of the input images. Then, in the second step, the network learns discriminative hash codes that incorporate both global and local level information. Experimental results from the application of the proposed approach in the domain of on-line terrorist video content demonstrate the merits of incorporating semantic segmentation information in deep hashing schemes.

The remaining of the paper is organized as follows: related work is discussed in Section 2. The proposed deep semantic hashing approach is detailed in Section 3. Experimental results are presented in Section 4 and Section 5 concludes the paper.

II. Related Work

This section discusses the state-of-art in hashing techniques, including both supervised and unsupervised learning schemes, while also investigating both hand-crafted and deep methods.

Hashing methods can generally be divided into two main categories, namely supervised and unsupervised ones. Unsupervised hashing methods make use of raw features extracted directly from the image, i.e. without exploiting semantic information [5]. For instance, Iterative Quantization (ITQ) aims at preserving the locality structure of the projected data that have been processed using Principal Component Analysis (PCA), by performing rotation so as to minimize the discretization error [12]. Additionally, Isotropic Hashing (IsoHash) learns projection functions, which can produce dimensions with isotropic variance [17]. Spectral Hashing (SH) initially applies PCA on the original data, then calculates the analytical Laplacian eigenfunctions along the principal directions and eventually hash codes are generated based on the projections of these eigenfunctions [10].

Supervised methods make use of semantic information during the hash function learning step. Supervised information can be considered in three different forms, namely as point-wise, pair-wise and ranking labels [16]. When point-wise supervised information is used, the model simultaneously handles both the problems of hash functions and image classification learning. The method of [18], which learns the hash functions and the classification layer at the same time, is representative of the aforementioned category. More specifically, a latent layer, placed before the classification layer, learns both image features and the corresponding hash code in an end-to-end fashion. Methods that make use of pair-wise supervised information generally require pairs of similar or dissimilar images for learning hash codes. The similarity or not of image pairs is assessed on the basis of the estimated classification label of each image. For example, Deep Pairwise-Supervised Hashing (DPSH) [16] learns hash codes in a pairwise manner within an end-to-end framework [16]. A similar approach that utilizes pair-wise information for learning hash functions in two steps is Convolutional Neural Network Hashing (CNNH) [7]. The latter method learns hash codes using supervised information in the first step and then, in a second step, estimates simultaneously both hash functions and image feature representations using supervised information originating from the computed hash codes (stage one) and the estimated image classification labels. Moreover, methods that make use of supervised information in the form of ranking labels typically generate triplets of images based on their estimated classification labels [9],[5], where one image constitutes the query and the remaining two are similar/dissimilar to the query one.

As in all cases of supervised learning, the use of supervised information is advantageous in learning hash functions, with the cost of depending on labeled data that are not always available. Additionally, the recent trend of simultaneously learning both hash functions and classification labels (deep hashing methods) has also resulted into significantly improved retrieval results. However, to the best of our knowledge, incorporating object-level information in deep hashing schemes has not been investigated so far, while it is very likely to further reinforce the expressiveness and the discriminative power of the estimated hash codes.

III. Proposed Method

In this section, the deep hashing approach using point-wise labels is initially outlined and subsequently the proposed framework is detailed.

A. Point-wise Deep Hashing

Let $X = \{x_1, x_i, ..., x_N\} \epsilon \mathbb{R}^{d \times N}$ be the set of training images. Deep hashing methods aim at learning a set of L hash functions that estimate the desired binary hash codes. Given an image x and it's classification label the network learns the corresponding class which the image x belongs. An individual hash code $\mathbf{b}_i$, which is a L-length binary vector, is computed for each input image x_i. L is the number of hash functions that export the L-length binary vector b for each x image. The ultimate goal is to learn hash functions that will extract low-dimensional and discriminant $\mathbf{b}_i$ vectors. For achieving this, the target during the training phase is to produce hash codes that are as close as possible in the hamming space for images of the same class and as far as possible for images belonging to different semantic classes. Vectors $\mathbf{b}_i$ are computed by applying a binarization step to the real-valued output of the corresponding hash functions. The binarization step is typically implemented using the sign function, which maps all input real values to the two discrete ones $\{-1, 1\}$, according to the following equation:

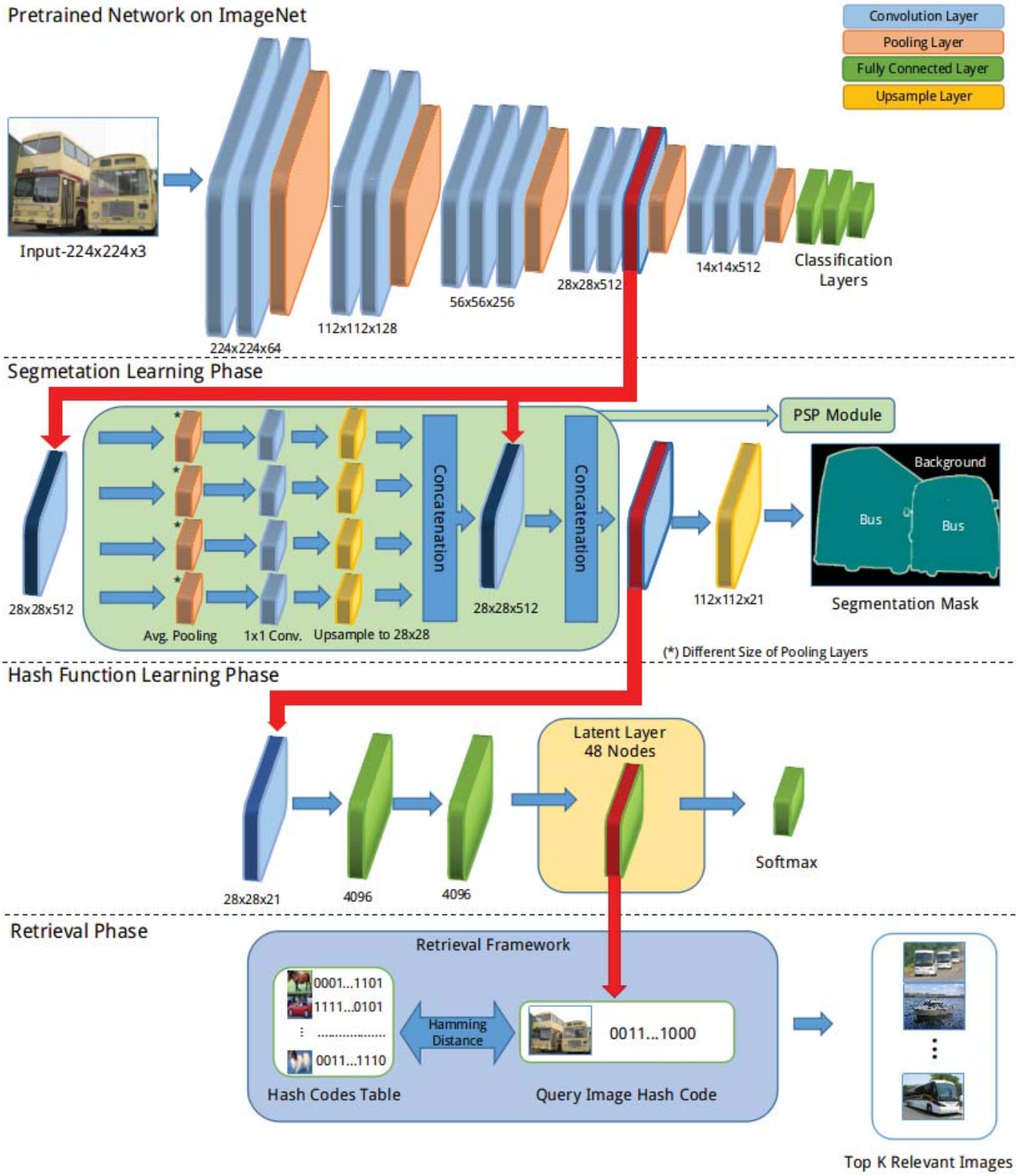

Fig. 1. Graphical representation of the proposed framework.

$$sgn(\psi) = \begin{cases} -1 & , if\ \psi < 0 \\ 1 & , if\ \psi \geqslant 0 \end{cases} \quad (1)$$

B. Proposed Framework

In this sub-section, the proposed framework for incorporating object-level (semantic segmentation) information in deep hashing schemes for image retrieval is detailed. The fundamental idea of using local-level information for increasing the discriminative power of the generated hash codes can be applied so as to extend any already proposed deep hashing method. Nevertheless, since a particular scheme needs to be used for experimental evaluation, the method of [18] is selected in this work, due to its relative implementation simplicity that is, however, coupled with increased retrieval performance. The proposed deep hashing framework comprises four distinct parts and it is graphically illustrated in Fig. 1.

The first part of the proposed framework comprises of a Neural Network pre-trained on the ImageNet dataset [19]. In the current wok, this base network is the VGG with configuration 'C' [20], which consists of a total of 16 layers. The primary goal of this work, as already discussed, is not to focus on particular base network architectures, but it is on directly incorporating semantic information regarding the objects present in the image. To this end, different well-known base network architectures, such as ResNet [21] or VGG with different configurations, can also be utilized.

The second part of the framework is responsible for integrating semantic segmentation information. In the current implementation, the well-known Pyramid Scene Parsing (PSP) network [22] is incorporated for that purpose. In particular, the PSP architecture receives as input the feature map of the semifinal VGG convolution layer. Then, average pooling layers of different size are applied to the feature map. Subsequently, convolution layers with kernel size 1×1 are used, followed by respective up-sampling layers. Eventually, the generated features are stacked with the original ones, as can be seen in Fig. 1. Information for supervised training of this part of the network is given in the form of an image segmentation mask. More specifically, the PSP module receives as input a feature map of size $28 \times 28 \times 512$. Then, four average pooling layers with bin size 28×28, 14×14, 9×9 and 7×7 are again applied. Each pooling layer is followed by a convolution layer with kernel size 1×1 and outputs $N/4$ features, where N is the number of features in the input feature map. Sequential application of batch normalization, ReLU (Rectified Linear Unit) activation and up-sampling layers over each pooling stream enables the reconstruction of the input feature map. The original feature map and the four reconstructed ones are stacked. Subsequent activation of convolutional, non-linear and up-sampling layers lead to the restoration of the original (ground truth) image segmentation mask dimensions. In the current implementation, the spatial dimension of the PSP module input is equal to 28×28.

In the hash code learning phase (third part in Fig. 1), the network learns the hash codes and the classification labels simultaneously. For achieving this, the softmax layer of the PSP module is removed and four new fully-connected layers are added; the first two comprise 4096 nodes, the next one 48 (for extracting the hash codes) and the last one is a softmax layer (which has as many nodes as the number of supported semantic classes). ReLU layers have been added after each fully-connected one, except from the hash layer (48 nodes) which is followed by a sigmoid activation function. The sigmoid layer outputs are in the range $[0, 1]$, which facilitates the extraction of the binary hash codes.

The final part of the framework is responsible for the task of retrieving relevant images. Having trained the network architecture that corresponds to the first three framework parts, a binary hash code can be generated for each query image and can be used here for retrieval purposes. In particular, as an input image x_i passes through the developed network, the latent layer in the third part outputs a vector $\mathbf{r}_i$ of real numbers in the range $[0, 1]$. For generating the respective binary hash code H_i, the following operator is used:

$$H_i = sgn(r_i - 0.5),\ H_i\ \epsilon\ \{-1, 1\} \quad (2)$$

Passing all the images of the employed training set through the developed framework generates results in the generation of a table of hash codes. During the retrieval step, the estimated hash code of the query image is compared with the aforementioned hash code table entries using hamming distance and the top-K most similar images are returned.

TABLE I
SUPPORTED LOCAL AND GLOBAL SEMANTIC CONCEPTS

Global	Local
Battlefield	Barrel
City Scape	Book
Crowd	Building
Desert	Electrical Device
Graphics	Fire
Indoors	Furniture
Interview	Gun/Rifle
Monuments	Logo
Mountain	Person
Terrorist Campus	Prisoner
	Sign
	Sky
	Smoke
	Truck
	Vegetation

IV. Experimental Results

A. Employed Datasets

The proposed framework is generic and can be directly introduced to any relevant deep hashing application case. However, in order to demonstrate its efficiency, particular datasets need to be employed for training and evaluation.

In order to train the semantic segmentation architecture (second part of the framework), the PASCAL-VOC2012 [23] dataset is used. This dataset contains approximately $2,912$ images with pixel-level ground truth annotation and supports

20 semantic classes. It was selected on the basis that the defined semantic classes correspond to commonly met real-world object categories, such as person, car, TV/monitor, etc.

For learning the hash code functions (third part of the framework), the CIFAR-10 [24] dataset is used. This dataset consists of approximately $60,000$ images. The training set ($50,000$ images) was used for modeling the hash functions and $1,000$ (100 for each supported class) images (out of the $10,000$ instances of the test set) were used for cross-validation purposes.

The overall proposed framework is evaluated in the highly challenging domain of on-line terrorist propaganda video analysis, i.e. a highly diverse and heterogeneous application domain. For that purpose, a large-scale real-world video dataset has been collected from on-line sources, where keywords or phrases commonly met in propagandistic videos have been used for identifying the relevant video content. The collected dataset consists of several hundreds of hours of video material. For processing the formed dataset and enabling search/retrieval operations, a set of approximately $27,000$ key-frames was formed, which were uniformly selected. For the experimental evaluation, two set of concepts, namely global- and local-level ones, were defined. In particular, 10 global and 15 local concepts were considered, as can be seen in Table I. The global concepts are used for describing the whole image (e.g. 'Battlefield', 'City Scape', etc.), while the local ones correspond to the different object types depicted in local regions of the image (e.g., 'Barrel', 'Book', etc.). Indicative key-frames of the formed dataset are given in Fig. 2.

B. Implementation Details

For training the part of the proposed framework that corresponds to the semantic segmentation step (second part in Fig. 1), learning rate equal to 10^{-3} was initially selected and was subsequently decreased to 10^{-4} after 20 epochs. The negative log-likelihood criterion was used during training, along with Stochastic Gradient Descent (SGD) for implementing back-propagation with momentum equal to 0.9. The total number of epochs was 30 and the defined batch size was set equal to 32. For hash code learning (third part in Fig. 1), the same training configuration as above was followed; the difference being that a batch size equal to 96 being used. All input images were cropped, using a square window placed at the center of the image with spatial dimension equal to the smaller image dimension, and then resized to 224x224 pixels. All implementation activities were carried out using the Keras [25] framework and a Nvidia GTX 1070 GPU with 8GB memory.

C. Evaluation Metrics

For evaluation, the metric defined in [18] was used. In particular, a ranking Mean Average Precision (MAP) value was calculated for each query image. For the calculations, the retrieved images that belonged to the same semantic class with the query image were considered relevant. MAP values were calculated for the top-10 and top-50 retrieved images.

TABLE II
GLOBAL CONCEPTS RETRIEVAL RESULTS

Concept	Top-10		Top-50	
	Proposed	Baseline	Proposed	Baseline
Battlefield	**50.40%**	32.60%	**36.60%**	31.20%
City Scape	**73.48%**	54.80%	**42.80%**	38.38%
Crowd	44.20%	**47.00%**	**37.40%**	35.80%
Desert	**67.26%**	55.60%	**55.64%**	49.60%
Graphics	**54.80%**	30.20%	**30.00%**	28.00%
Indoors	27.60%	**30.00%**	**23.00%**	18.00%
Interview	**78.60%**	60.80%	**61.60%**	48.00%
Monuments	20.00%	**22.60%**	**21.00%**	15.00%
Mountain	**58.20%**	48.40%	39.00%	**40.60%**
Terrorist Campus	**10.20%**	6.20%	**11.20%**	8.80%
Overall	**48.47%**	38.82%	**35.82%**	31.33%

TABLE III
LOCAL CONCEPTS RETRIEVAL RESULTS

Concept	Top-10		Top-50	
	Proposed	Baseline	Proposed	Baseline
Barrel	**36.66%**	4.00%	**28.30%**	3.88%
Book	5.47%	**15.61%**	8.87%	**13.81%**
Building	72.29%	**82.88%**	47.32%	**58.27%**
Electrical Device	**28.41%**	26.66%	**30.54%**	23.66%
Fire	34.24%	**45.10%**	**27.52%**	26.00%
Furniture	**39.43%**	15.00%	**33.72%**	14.78%
Gun/Rifle	**43.58%**	11.19%	**42.02%**	11.67%
Logo	**35.56%**	28.19%	**25.37%**	20.37%
Person	**89.71%**	75.75%	**78.43%**	73.27%
Prisoner	**44.50%**	41.16%	**39.27%**	22.74%
Sign	**56.33%**	39.60%	**42.62%**	19.68%
Sky	**92.62%**	85.74%	**84.09%**	73.49%
Smoke	**61.19%**	55.20%	22.98%	**37.29%**
Truck	**42.93%**	42.66%	32.10%	**37.81%**
Vegetation	48.10%	**55.73%**	39.19%	**42.50%**
Overall	**48.73%**	41.63%	**38.82%**	31.92%

D. Evaluation Results

The proposed framework was evaluated using the global and local semantic concepts defined in Table I. Tables II and III illustrate the obtained retrieval results for each semantic concept, while the performance of the baseline method of [18] (which does not make use of semantic segmentation information for estimating the image hash codes) is also given. Additionally, an average MAP value over all defined concepts (which are equally represented in the conducted retrieval experiments) is also estimated.

Global Concepts: From the results presented in Table II, it can be seen that the proposed framework outperforms the baseline model by approximately 9.65% and 4.49% for the top-10 and top-50 cases, respectively. From a detailed examination of the estimated results, it can be seen that for certain concepts (such as "Battlefield", "City Scape", "Crowd" and "Interview') where local concepts have an increased role (e.g. clearly visible human figures) the proposed method achieves to introduce significant performance gains, compared to the baseline method. This suggests that incorporating local-level information (semantic segmentation) in the hash code learning process can significantly boost the retrieval performance of global concepts. Additionally, it can be seen that this improve-

Fig. 2. Indicative key-frames of the formed dataset.

ment in performance is greater for the more challenging top-10 retrieval case.

Local Concepts: Table III illustrates the top-10 and top-50 retrieval results obtained by the application of the proposed framework. From the provided results, it can be seen that the proposed framework outperforms the baseline approach by approximately 7.10% and 6.90% for the top-10 and top-50 image retrieval cases, respectively. More specifically, for particular local concepts (such as "Furniture", "Person" and "Prisoner"), which exhibit well-defined appearance partners, the proposed framework introduces a significant performance increase over the baseline approach. The above observations suggest that incorporating local-level information, regarding the objects that are present in the image, during the hash code learning phase is advantageous also for the cases of local concepts. It needs to be mentioned that there is no significant difference in the performance improvement over the baseline for the cases of top-10 and top-50 evaluation for the local concepts, as opposed to the case of global concepts. Additionally, it can also be observed that the proposed framework performs on average better for the cases of local concepts.

V. Conclusion

In this paper, a novel deep hashing architecture for constructing binary hash codes that incorporate local-level information in the form of semantic segmentation masks was proposed. The introduced framework was evaluated in the challenging domain of on-line terrorist propaganda video analysis and exhibited significant image retrieval performance gains over a corresponding baseline method that makes use of only whole image information for estimating hash codes. The experimental evaluation showed that the introduced framework is advantageous both for global and local concepts. Future work includes the investigation of alternative ways for incorporating local-level information in deep hashing schemes.

VI. Acknowledgment

The work presented in this paper was supported by the European Commission under contract H2020-700367 DANTE.

References

[1] Ruimao Zhang, Liang Lin, Rui Zhang, Wangmeng Zuo, and Lei Zhang. Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification. *IEEE Transactions on Image Processing*, 24(12):4766–4779, 2015.

[2] Venice Erin Liong, Jiwen Lu, Gang Wang, Pierre Moulin, and Jie Zhou. Deep hashing for compact binary codes learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2475–2483, 2015.

[3] Fang Zhao, Yongzhen Huang, Liang Wang, and Tieniu Tan. Deep semantic ranking based hashing for multi-label image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1556–1564, 2015.

[4] Guoqiang Zhong, Hui Xu, Pan Yang, Sijiang Wang, and Junyu Dong. Deep hashing learning networks. In *Neural Networks (IJCNN), 2016 International Joint Conference on*, pages 2236–2243. IEEE, 2016.

[5] Xiaofang Wang, Yi Shi, and Kris M Kitani. Deep supervised hashing with triplet labels. *arXiv preprint arXiv:1612.03900*, 2016.

[6] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. Similarity search in high dimensions via hashing. In *VLDB*, volume 99, pages 518–529, 1999.

[7] Rongkai Xia, Yan Pan, Hanjiang Lai, Cong Liu, and Shuicheng Yan. Supervised hashing for image retrieval via image representation learning. In *AAAI*, volume 1, page 2, 2014.

[8] Theodoros Semertzidis, Dimitrios Rafailidis, Michael Gerassimos Strintzis, and Petros Daras. The influence of image descriptors dimensions value cardinalities on large-scale similarity search. *International Journal of Multimedia Information Retrieval*, 4(3):187–204, 2015.

[9] Hanjiang Lai, Yan Pan, Ye Liu, and Shuicheng Yan. Simultaneous feature learning and hash coding with deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3270–3278, 2015.

[10] Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. In *Advances in neural information processing systems*, pages 1753–1760, 2009.

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

[12] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent 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, 2013.

[13] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001.

[14] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[15] Haomiao Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. Deep supervised hashing for fast image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2064–2072, 2016.

[16] Wu-Jun Li, Sheng Wang, and Wang-Cheng Kang. Feature learning based deep supervised hashing with pairwise labels. *arXiv preprint arXiv:1511.03855*, 2015.

[17] Weihao Kong and Wu-Jun Li. Isotropic hashing. In *Advances in Neural Information Processing Systems*, pages 1646–1654, 2012.

[18] Kevin Lin, Huei-Fang Yang, Jen-Hao Hsiao, and Chu-Song 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.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

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

[21] 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.

[22] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. *arXiv preprint arXiv:1612.01105*, 2016.

[23] 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.

[24] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[25] François Chollet. Keras. https://github.com/fchollet/keras, 2015.