DEEP METRIC AND HASH-CODE LEARNING FOR CONTENT-BASED RETRIEVAL OF REMOTE SENSING IMAGES

Subhankar Roy[1], Enver Sangineto[1], Begüm Demir[2] and Nicu Sebe[1]

[1]Dept. of Information Engineering and Computer Science, University of Trento, Trento, Italy
[2]Faculty of Electrical Engineering and Computer Science, TU Berlin, Berlin, Germany

ABSTRACT

The growing volume of Remote Sensing (RS) image archives demands for feature learning techniques and hashing functions which can: (1) accurately represent the semantics in the RS images; and (2) have quasi real-time performance during retrieval. This paper aims to address both challenges at the same time, by learning a semantic-based metric space for content based RS image retrieval while simultaneously producing binary hash codes for an efficient archive search. This double goal is achieved by training a deep network using a combination of different loss functions which, on the one hand, aim at clustering semantically similar samples (i.e., images), and, on the other hand, encourage the network to produce final activation values (i.e., descriptors) that can be easily binarized. Moreover, since RS annotated training images are too few to train a deep network from scratch, we propose to split the image representation problem in two different phases. In the first we use a general-purpose, pre-trained network to produce an intermediate representation, and in the second we train our hashing network using a relatively small set of training images. Experiments on two aerial benchmark archives show that the proposed method outperforms previous state-of-the-art hashing approaches by up to 5.4% using the same number of hash bits per image.

Index Terms— deep hashing, metric learning, content based image retrieval, remote sensing

1. INTRODUCTION

In recent years there has been a tremendous increase in the volume of remote sensing (RS) image archives due to the continuous development of satellite technology. Thus, one of the most important research topics in RS is the development of scalable content-based RS retrieval (CBIR) methods, which aim at retrieving the most similar images to a query image from massive archives in an accurate and efficient manner. A CBIR system generally consists of a two-step procedure: (1) characterization of the content of each image by its descriptor; and (2) computation of similarities between the query image and the archive images based on the extracted descriptors.

One of the essential requirements for large-scale CBIR is the fast similarity search. The conventional similarity search methods, such as nearest neighbour search, are impractical for large scale RS image archives, particularly when the dimension of the image descriptor is high. To achieve efficient similarity search, hashing-based approximate nearest neighbour search methods have been recently introduced in RS [1], [2]. Hashing methods encode high-dimensional image descriptors using compact binary hash codes that significantly reduce the storage cost and improve the computational efficiency. To this end, hash functions are initially generated and then applied to each image descriptor. In [1], the kernel-based locality sensitive hashing techniques that learn hash functions in the kernel space from hand-crafted features (e.g., the bag-of-visual-words based on the scale invariant feature transform) are applied to RS CBIR problems. However, hand-crafted features may not accurately represent the high level semantic content of RS images. This leads to inaccurate retrieval results under complex RS image retrieval tasks.

To address this problem, inspired by the progress of deep convolutional neural networks (CNNs), a deep hashing method has been recently introduced in the framework of RS image retrieval problems [2]. This method jointly learns the deep image features (which efficiently characterize the rich semantic content of RS images and outperforms hand-crafted features) and hash codes (which represent those deep features with binary bits). To this end, their CNN architecture adopts the cross-entropy loss to formulate the objective function. The cross-entropy loss does not define any separation between positive and negative images due to the absence of a margin threshold. This leads to poor generalization capability, and thus long hash codes and a high number of annotated training images are required to reach a high retrieval accuracy.

To address these issues, in this paper we present an approach that learns a semantic-based metric space, while simultaneously producing binary hash codes for fast and accurate retrieval of RS images in large archives. Differently from [2] the proposed approach provides more compact binary hash codes with a small number of annotated training images.

2. METRIC AND HASH-CODE LEARNING

The lack of large annotated training images makes it challenging to train deep networks from scratch in RS. We propose to solve this problem using two different stages. In the first stage we use a pre-trained network (Inception Net [3]), trained on ImageNet, in order to extract an intermediate image representation. In the second stage, this intermediate representation is fed to our Metric and Hash-Code Learning Network (MHCLN). The latter is a smaller network which can be trained *from scratch* using a relatively small dataset. This network is trained using a combination of different losses, which simultaneously aim at clustering similar images while producing an easy-to-binarize final representation. Specifically, we use the *triplet loss* to learn a *metric space* where the Euclidean distance between pair of points corresponds to the semantic distance between the corresponding images. Moreover, we use two other losses: (1) a representation penalty which pushes the final network activations toward 0 and 1 and; (2) a balancing loss which incentivizes a balanced number of 0s and 1s in the final hashing code. We provide below all the details.

Let $\mathcal{I} = \{X_1, ..., X_P\}$ be the training set of RS images where X_i is associated with a class label $y_i \in \mathcal{Y} = \{y_1, y_2, ...\}$ (e.g., "airplane", "parkinglot", etc.). Our goal is to learn a hashing function $h : \mathcal{I} \rightarrow \{0, 1\}^K$, which maps images to binary hash codes of length K such that the generated binary codes embed the semantics of the corresponding images. Using these codes, at testing time, retrieving the most similar images to a given query image X_q is done by (efficiently) comparing bitwise their binary hash codes.

In the first stage of the proposed approach, each image in $\mathcal{I}$ is fed to the pre-trained Inception Net [3] and a feature vector composed of the 2048 neuron activations of the layer just before the softmax layer ($pool_3$) is extracted. Let $\mathcal{G} = \{g_1, g_2, ..., g_P\}$, $g_i \in \mathbb{R}^{2048}$, be the extracted features corresponding to the set of images in $\mathcal{I}$. Although the Inception Net was trained on a completely different set of images (ImageNet), its high-level features capture general-purpose visual semantics and we use this representation as a starting point of our representation process. Note that the Inception Net is *not* fine-tuned but used as a black-box to extract g_i from X_i: g_i is used as input of our MHCLN (Fig. 1).

In the second stage of the proposed approach we train (from scratch) our MHCLN aiming at mapping each g_i into a semantically significant metric space: $f : \mathbb{R}^{2048} \rightarrow \mathbb{R}^K$. The final hashing function is obtained by means of a quantization of $\mathbb{R}^K$. In order to learn the metric space we adopt a triplet loss. The intuition behind the triplet loss (see Fig.2) is that similar images should be clustered together in the target metric space, while different images should be pushed far apart. To achieve this result, we use the class label (y_i) associated with each X_i and we impose that images of the same class should be closer to each other than images of different classes. More specifically, from $\mathcal{G}$ we extract a set of triplets

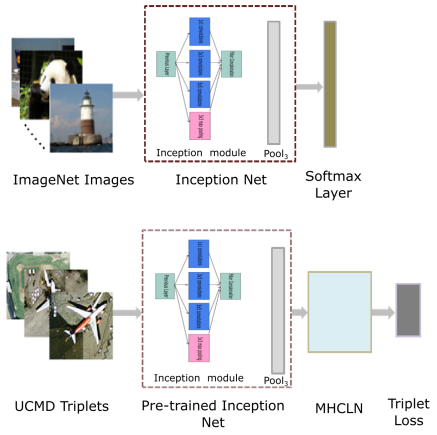

Fig. 1. Our two-step image representation process. Top: Inception Net, pretrained on ImageNet images. Bottom: Inception Net is used to extract an intermediate image representation, which is then fed to our MHCLN.

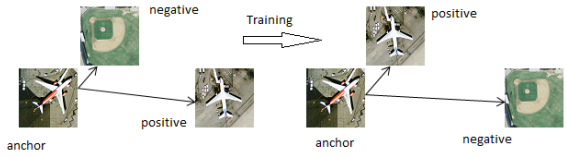

Fig. 2. The intuition behind the triplet loss: after training, a positive sample is "moved" closer to the anchor sample than the negative samples of the other classes.

$\mathcal{T} = \{(g_i^a, g_i^p, g_i^n)\}$, where g_i^a (called *anchor*), is a randomly sampled feature vector associated with label y_i, g_i^p is a *positive* sample (i.e., an image associated with the same class label y_i) and g_i^n is a *negative* sample (i.e., an image associated with a different class label $y_j \neq y_i$). Using $\mathcal{T}$ and a mini-batch of cardinality M, randomly extracted from $\mathcal{T}$, our triplet loss is defined as follows:

$$\mathcal{L}_{Metric} = \sum_{i=1}^{M} max\Big(0, ||f(g_i^a) - f(g_i^p)||_2^2 - ||f(g_i^a) - f(g_i^n)||_2^2 + \alpha\Big), \quad (1)$$

where α is a minimum margin that is imposed between the positive and the negative distances.

Our hashing network is composed of 3 fully-connected layers, composed of 1024, 512, and K neurons respectively, where K depends on the number of desired bits in the final hashing-based image representation. We use Leaky ReLU non-linearities in the first two layers and a sigmoid in the last layer, which produces neuron activations in $[0, 1]$. In order to push the latter toward the extremes of the range, similarly to [4], we use a second loss which aims at maximizing the sum of the squared errors between the last layer activations and 0.5:

Table 1. mAP and average retrieval time for the KSLSH [1] and the proposed MHCLN for the UCMD archive.

Methods	Image Features		# Hash Bits *K*					
	mAP	Time (in ms)	*K*=16		*K*=24		*K*=32	
			mAP	Time (in ms)	mAP	Time (in ms)	mAP	Time (in ms)
SVM	0.556	92.3	-	-	-	-	-	-
KSLSH [1]	-	-	0.557	25.3	0.594	25.5	0.630	25.6
Our **MHCLN**	-	-	**0.875**	25.3	**0.890**	25.5	**0.904**	25.6

$$\mathcal{L}_{Push} = -\frac{1}{K}\sum_{i=1}^{P} ||f(g_i) - 0.5\mathbf{1}||^2, \quad (2)$$

where $\mathbf{1}$ is the *K*-dimensional vector with all elements 1.

Finally, inspired by [4], we use a third loss function which aims at balancing the number of 1s and 0s in the binary code of each image representation:

$$\mathcal{L}_{Balancing} = \sum_{i=1}^{P}(mean(f(g_i)) - 0.5)^2, \quad (3)$$

where $mean(f(g_i))$ computes the mean of the activation values in the last layer of the hashing function.

The three losses are combined in our final objective:

$$\mathcal{L} = \mathcal{L}_{Metric} + \lambda_1\mathcal{L}_{Push} + \lambda_2\mathcal{L}_{Balancing}, \quad (4)$$

with $\lambda_1 = 0.001$ and $\lambda_2 = 1$ that are selected using cross-validation.

Once the network is trained, the final hashing function $h()$ is obtained by binarizing the values in $\mathbb{R}^K$. Specifically, given a test image X, which corresponds to an Inception-Net feature vector g, we compute a binary code $b = h(X)$, where, for each $1 \leq n \leq K$:

$$b_n = (sign(v_n - 0.5) + 1)/2, \quad (5)$$

where $v = f(g)$ and v_n is the n-th component of v. Finally, in order to retrieve an image X_j semantically similar to a query image X_q, we use the Hamming distance between $h(X_q)$ and $h(X_j)$.

3. EXPERIMENTAL RESULTS

Experiments were conducted on two different benchmark archives. The first one is the widely used UC Merced (UCMD) [5] containing 2100 images from 21 different categories, where each category includes 100 images (each of size 256 × 256 pixels with a spatial resolution of 30cm). The second one is the Aerial Images Dataset (AID) [6] containing 10000 aerial images from 30 different categories, where each category includes 220 to 420 images (each of size 600 × 600 pixels with a spatial resolution ranging from 50cm to 8m).

The proposed MHCLN[1] was trained by choosing a mini-batch of triplets of cardinality M=30, which comprises of 30 anchors, positives and negatives each. The value of the threshold margin, α, was set to 0.2. For the loss function optimization, Adam Optimizer was used with a small learning-rate η = 10^{-4}. The other two hyper-parameters of the Adam Optimizer, β_1 and β_2 were set to 0.5 and 0.9 respectively.

The performance of the proposed approach is evaluated through the mean average precision (mAP) score, also used in [2]. For the UCMD archive results achieved with the proposed MHCLN are compared with those obtained by: 1) Support Vector Machine (SVM) classifier; 2) the Kernel-based Supervised Locality Sensitive Hashing (KSLSH) [1]; and 3) the Deep Hashing Neural Networks (DHNN) [2]. Results of each method are provided in terms of computational time and the mAP score that are evaluated for the top-20 retrieved images. In the experiments: i) Gaussian Radial Basis Function kernel was used for the SVM and KSLSH; and ii) the number *K* of hash bits is varied in the range [16-64] with a step size increment of 8. In the experiments, we have considered two different scenarios. In the first scenario, we did not include any data augmentation and have selected 60% of images associated to each category as training images (which are used to train the MHCLN), while the rest is considered as test images (which are used to evaluate the retrieval performance).

Table 1 shows the results of the first scenario obtained by the SVM, the KSLSH and the proposed MHCLN when *K*=16, 24 and 32. We would like to point out that due to lack of availability of code of the DHNN we could not report its results within this scenario. From Table 1, one can see that the proposed MHCLN provides 31.8% higher mAP compared to the KSLSH for *K*=16 under the same retrieval time. In addition, the proposed MHCLN yields a mAP 31.9% higher with respect to the SVM with significantly reduced retrieval time. From our analysis, we have also seen that increasing the number *K* of hash bits leads to higher mAP at the cost of slightly increasing the retrieval time.

Fig 3 shows a single trial of the retrieval results with the query image selected from the airplane category by applying the KSLSH [1] and the proposed MHCLN. The retrieval order of each image is given below the related image. By visually analyzing the results, one can see that the proposed method retrieves semantically more similar images from the archive. As an example, the 4th and 19th retrieved images by the KSLSH method belongs to the freeway and storage tanks categories, respectively, whereas those by the proposed

[1]Code is available at `https://github.com/MLEnthusiast/MHCLN`

method belong to the airplane category.

In the second scenario, we have considered data augmentation as suggested in [2] and compared the proposed MHCLN with the DHNN [2]. To have a fair comparison 2100 images are rotated by 90°, 180° and 270°, producing 8400 images. Then, among these images, 1000 images are randomly chosen as test images while the remaining 7400 images are selected as training and searching images as suggested in [2]. Table 2 shows the results obtained by the proposed MHCLN and the DHNN when K=32 and K=64 for the top-50 retrieved images. By analyzing the results one can see that hash codes obtained by the proposed method are generally more distinctive than those of the DHNN when small values of hash bits are considered. As an example, the proposed method yields 5.4% better mAP when K=32. It is worth noting that the data augmentation approach adopted in [2] may lead to the presence of rotated but identical images in both the train and test sets. This causes the network to memorize the test samples (rotated variants) during training. However, we have adopted this evaluation approach to fairly compare with their method.

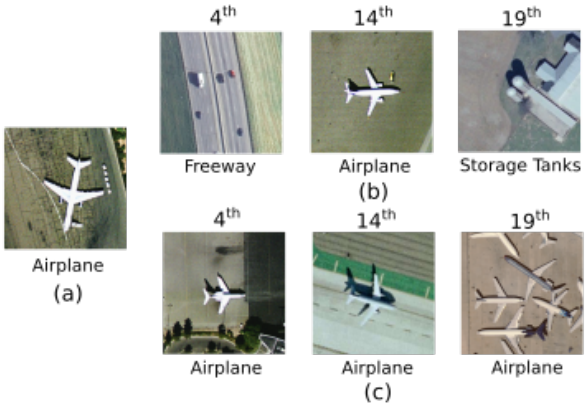

Fig. 3. (a) Query image, (b) images retrieved by KSLSH [1] and (c) images retrieved by the proposed MHCLN.

For the AID archive we have chosen a 60:40 split of images from each category, similar to the UCMD archive. For this archive, results achieved with the proposed MHCLN are compared with the SVM and the KSLSH. From the results, we observed that the same relative behavior with respect to the UCMD archive is obtained. As an example, for the proposed MHCLN we obtain 40.66% higher mAP, for K=32, when compared to the KSLSH under the same retrieval time. In addition the proposed MHCLN yields a mAP of 91.14%, which is 0.37% higher than that obtained by the SVM with retrieval time reduced by one order of magnitude.

4. CONCLUSION

In this paper, we have introduced a deep metric and hash-code learning approach for fast and accurate image search and retrieval in large RS data archives. The proposed approach is defined based on two main stages. In the first stage, an intermediate representation is obtained for each image in the archive by exploiting the pre-trained network (Inception Net), while in the second stage a hashing network is trained by considering different losses (e.g., triplet loss, representation penalty and a balancing loss) to represent each image by binary hash code. Experimental results point out that the hash codes obtained by the proposed approach: 1) efficiently characterize the complex content of RS images; 2) enable fast image search and retrieval through compact hash codes; and 3) can be learnt using a relatively few annotated training images. As a future development of this work, we will extend our approach to the framework of other deep networks.

Table 2. mAP obtained by the DHNN [2] and the proposed MHCLN with data augmentation for the UCMD archive.

	# Hash Bits K	
Methods	K = 32	K = 64
DHNN [2]	0.939	0.971
Our **MHCLN**	**0.993**	**0.995**

5. ACKNOWLEDGEMENTS

This work was supported by the European Research Council under the ERC Starting Grant BigEarth-759764. We also want to thank the NVIDIA Corporation for the donation of the GPUs used in this project.

6. REFERENCES

[1] B. Demir and L. Bruzzone, "Hashing-based scalable remote sensing image search and retrieval in large archives," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 2, pp. 892–904, 2016.

[2] Y. Li, Y. Zhang, X. Huang, H. Zhu, and J. Ma, "Large-scale remote sensing image retrieval by deep hashing neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, 2017.

[3] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *CVPR*, 2016.

[4] 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*, 2017.

[5] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *SIGSPATIAL*, 2010, pp. 270–279.

[6] G.-S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, and X. Lu, "Aid: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, 2017.