A Reliability Object Layer for Deep Hashing-based Visual Indexing*

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

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

Abstract. Nowadays, time-efficient search and retrieval of visually similar content has emerged as a great necessity, while at the same time it constitutes an outstanding research challenge. The latter is further reinforced by the fact that millions of images and videos are generated on a daily basis. In this context, deep hashing techniques, which aim at estimating a very low dimensional binary vector for characterizing each image, have been introduced for realizing realistically fast visual-based search tasks. In this paper, a novel approach to deep hashing is proposed, which explicitly takes into account information about the object types that are present in the image. For achieving this, a novel layer has been introduced on top of current Neural Network (NN) architectures that aims to generate a reliability mask, based on image semantic segmentation information. Thorough experimental evaluation, using four datasets, proves that incorporating local-level information during the hash code learning phase significantly improves the similar retrieval results, compared to state-of-art approaches.

Keywords: Deep hashing · Hash codes · Deep learning· Image segmentation · Neural networks.

1 Introduction

Over the recent years, the amount of visual content that is generated on a daily basis grows exponentially, mainly due to the widespread use of portable devices (e.g. smart-phones, tablets, etc.) that typically feature high-quality camera sensors. This results in the generation of extremely large visual databases, where the tasks of accurate and time-efficient search/retrieval comprise a great challenge. To address this, hashing methods have been proposed, in order to realize an efficient way of visually relevant information retrieval, in terms of both retrieval accuracy and computational time. Generally, hashing methods have very low storage requirements and exhibit fast responses, compared to other traditional retrieval approaches (e.g. image descriptors). The merits of hashing methods

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

stem out from the efficient mapping of high dimensional feature vectors to corresponding significantly low dimensional binary codes, which are subsequently used for 'query-by-example' image retrieval [18]. These mappings are also known as 'hash functions' and the generated binary vectors are typically termed 'hash codes'.

Different hashing methods have been proposed so far that can generally be divided in two main categories, namely data-independent and data-dependent ones [18, 28], as presented in Fig. 1. Data-independent approaches are not taking into account 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 space to a lower dimensional one. In the next step, quantization is applied for generating a compact binary vector that robustly encodes the original one [30, 22]. Indicative approaches of this category are the Locality Sensitive Hashing (LSH) method [6] and its variants, which are selecting projection matrices to lower-dimensional spaces and threshold the vectors to compute binary codes. On the contrary, 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 [24]. Representative data-depended methods are Spectral Hashing (SH) [23], Binary Reconstructive Embedding (BRE) [14] and Iterative Quantization (ITQ) [7].

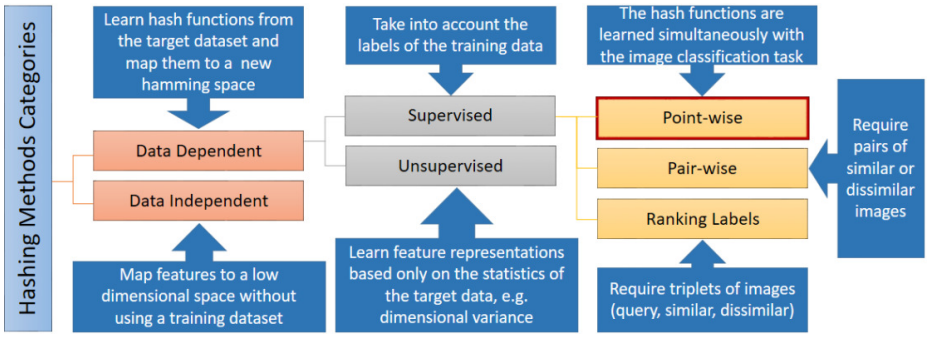

Fig. 1. A visual representation of the different hashing methods categories.

Data-dependent methods can generally be divided into two sub-categories namely supervised and unsupervised ones (Fig. 1). Unsupervised approaches aim at learning feature representations based only on pure statistics of the target data, e.g. the variance of the values in each dimension or their cardinality [20]. In other words, unsupervised methods do not take into account semantic information [22]. 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 [7]. Additionally, Isotropic Hashing (IsoHash) learns projection functions, which can produce dimensions with isotropic variance [11].

Furthermore, 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 [23].

On the contrary, supervised methods make use of semantic information during the hash functions learning phase. The advantage of using labeled data to guide the learning process enables supervised methods to generate hash codes that represent more accurately the original data and with fewer bits (i.e. smaller hash code length), compared to the ones obtained by the application of unsupervised techniques. Small hash code length is desirable for building efficient image retrieval frameworks, with respect to the required computational resources [15, 22]. Supervised information is typically considered in three different forms, namely point-wise, pair-wise and ranking labels [16] (Fig. 1). When point-wise information is used, the model simultaneously handles both the problems of hash functions and image classification learning. Methods that make use of pair-wise information generally require pairs of similar or dissimilar images for learning hash codes. Moreover, methods that make use of information in the form of ranked labels are typically generate triplets of images based on their estimated classification labels, where one image constitutes the query and the remaining two are similar/dissimilar to the query, such as DBC [19].

The above-mentioned hashing methods make use of traditional hand-crafted visual descriptors, such as HOG [3]. However, these hand-engineered descriptors (and consequently the corresponding hash codes) do not always efficiently model the original images and their semantics; thus, failing to provide a retrieval mechanism of high accuracy. However, the break-through introduced by 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, which typically correspond to end-to-end systems (i.e. receive images and input and generate hash codes) are termed deep hashing. Point-wise methods is the most commonly met category [17, 26]. Pairs of images are used by methods under the pairwise category [16]. Moreover, ranking labels methods make use of image triplets [15, 22].

In this paper, a novel deep hashing framework for fast image retrieval is proposed, which takes into account the reliability of the objects that appear in an image. In particular, semantic segmentation masks are used to determine the object. Then a novel layer is introduced, which penalizes image regions where objects are not detected with high confidence. The main contributions of this work are as follows:

- The fusion of semantic information in the form of image segmentation masks, in order to generate more expressive and robust hash codes that will combine image-level features with discriminative object-level information cues. Current deep hashing techniques are only limited to image-level analysis.

– The introduction of a Reliability Object Layer (ROL), which generates a binary mask, denoting image pixels that correspond to an object with high degree of confidence.

The remainder of the paper, is organized as follows: Related work, is discussed in Section 2. In Section 3 the proposed method is detailed while experimental results are presented in Section 4. Finally, conclusions are drawn in Section 5.

2 Related Work

In this section, analysis regarding the state-of-the-art of deep-hashing techniques is provided. As already described in Section 1, supervised methods generate more efficient and accurate respective hash functions, compared to the unsupervised techniques.

Point-wise deep hashing methods do not make use of data augmentation techniques and can resolve the hashing problem in conjunction with the classification one. In [17] a method which learns the hash functions and the classification layer at the same time is proposed. Specifically, a latent layer, placed before the classification one, learns both image features and the corresponding hash code in an end-to-end fashion. The latent layer is added between the last fully connected layer (classification layer) and after the semi-final fully connected layer that consists of 4096 hidden nodes. The authors also propose a two-step retrieval framework. In the first step, termed coarse-level search, the framework is fed with a query image and retrieves the top-k similar images calculating the hamming distances for the whole dataset. In the next step, (namely fine-level search), the ranked list of similar retrieved images is computed, while calculating the Euclidean distance from the candidate images of the previous step. An extension of the aforementioned method is proposed from Yang et al. [26], termed Supervised Semantics-preserving Deep Hashing (SSDH). The SSDH method also learns the hash codes and the image representation at the same time. In order to generate more efficient hash codes, the loss function has been enriched, by adding two more functionalities. Specifically, a mechanism forcing the outputs to be 0 or 1 is applied after the latent layer. This enables the model to minimize the quantization error. Additionally, a component that fires at each bit location with probability equal to 0.5 is included.The latter leads to the production of more discriminative hash codes.

Deep hashing methods comprising the pair-wise category have also been widely investigated. An indicative method is the so called Deep Pairwise - Supervised Hashing (DPSH) [16]. Specifically, DPSH learns hash codes in a pairwise manner within an end-to-end framework. Initially, a pair is defined as similar if both input images belong to the same class, otherwise it is defined dissimilar. Then, a siamese network architecture is implemented, in order to pass pairs of images simultaneously across the network. It is worth noting, that the two CNNs have the same structure and share the same weights. A latent layer is added on top of the network, as a common practice in deep hashing approaches, so as to

be able to learn the hash functions. Eventually, the framework also takes into account if an input pair is similar or dissimilar. More specifically, the loss function aims to minimize the distance of the real-valued vectors for similar pairs or to maximize this distance for pairs that are dissimilar, respectively. Moreover, Cao et al. [2] have introduced the so called 'HashNet'. The method relies on an architecture which exhibits improved performance in imbalanced datasets. More specifically, a weighted pairwise cross-entropy loss function is used, in order to learn similarity/dissimilarity scores between pairs of images from sparse data. For preventing the vanishing gradient problem, the tanh activation function is applied.

The ranking labels category refers to the methods that employ triplets of images. In particular, these approaches almost always receive as input three images, namely a query image a similar and a dissimilar one. Lai et al. [15] propose a triplet-based deep hashing method, in order to learn more discriminative hash codes. In more details, a divide-and-encode module that splits each image (query, positive, negative) feature representation into parts is included. In particular, the transformation from long length real vectors to short binary codes is performed by adding one node for each binary bit after an average pooling layer. Also a triplet ranking loss function is implemented that aims to regularize the distance between similar/dissimilar images from the query one to a minimum/maximum, respectively. Additionally, Zhang et al. [27] propose the framework where the binary codes are scalable. The size of hash code is generated by the addition of a new layer that learns the weight of each bit. During the test phase, the bits which contribute more are taken into account to extract the binary vector.

In all cases of supervised learning, the use of supervised information is advantageous towards learning robust 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 has also resulted into significantly improved retrieval results. However, explicitly 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. Regarding implementation complexity, the methods of the point-wise category are easier to be materialized, as a result of lack of data augmentation requirements. However, the possible merits of using local-level information can favor both pair-wise and ranking labels approaches.

3 Proposed Method

In this section, the proposed framework is explained in details. In particular, the developed architecture consists of three main components: a) The classification stream that learns the hash functions while also solving the classification problem. b) The object level stream that handles the pixel-wise classification problem, based on the use of image segmentation masks. c) The Reliability Object

Layer that penalizes pixel areas that are not associated with object classification decisions with high confidence.

3.1 Classification Stream

The fundamental consideration of the proposed framework is to reinforce features that correspond to certain semantic object categories (with a relatively high degree of confidence) during the hash code learning phase. For achieving this, the RGB input image is fed into a two stream architecture, as presented in Fig. 2. The output of the bottom stream corresponds to a semantic segmentation mask of the input image, while the upper stream handles the classification problem. As can be seen in Fig. 2 (top) the classification stream of the proposed architecture comprises of a Neural Network pre-trained using the ImageNet dataset [13]. In the current work, this base network is selected to be the VGG network with configuration 'C' [21], 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 using semantic segmentation information, in order to generate more efficient hash codes. To this end, different well-known base network architectures, such as ResNet [9] or VGG with different configurations, can be utilized.

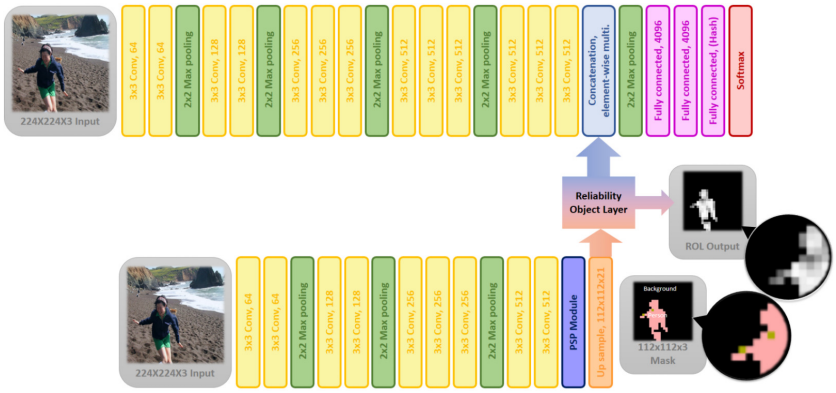

Fig. 2. Proposed deep hashing framework for directly incorporating local-level information in the form of a semantic segmentation mask using a reliability object layer.

3.2 Local-level Stream

The lower part of the architecture in Fig. 2 is responsible for incorporating semantic segmentation related information. It is worth noting that local-level information in the form of segmentation masks can be encoded using a variety of different implementations, such as [10, 1, 8]; however, the aim of this work is

not excessively evaluating the different available implementations, but to investigate the usefulness of incorporating semantic segmentation information during the hash code learning phase. For that purpose, the well-known Pyramid Scene Parsing (PSP) network [29], which exhibits satisfactory segmentation performance and relatively decreased module integration requirements, is incorporated. Specifically, the PSP architecture receives as input the feature map of the semifinal VGG convolution layer. Then, average pooling layers of different sizes are applied. Subsequently, convolution layers with kernel size 1×1 are used, followed by corresponding up-sampling layers. Eventually, the generated features are stacked with the original ones. Information for supervised training of this part of the network is given in the form of an image segmentation mask. As mentioned in table 1, 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 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, Rectied Linear Unit (ReLU) 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 then stacked. Subsequent activation of convolutional, non-linear and up-sampling layers lead to the restoration of the original (ground truth) image segmentation mask dimensions.

Table 1. PSP module configuration.

Vgg16 semi-final conv. layer output : $28 \times 28 \times 512$			
Pool 28×28	Pool 14×14	Pool 9×9	Pool 7×7
Conv 1×1, 128			
Batch normalization			
Relu ($1 \times 1 \times 128$)	Relu ($2 \times 2 \times 128$)	Relu ($3 \times 3 \times 128$)	Relu ($4 \times 4 \times 128$)
Up-sample $\times 28$	Up-sample $\times 2$	Up-sample $\times 10$	Up-sample $\times 4$
-	-	Crop2D 1×1	-
Stacked $28 \times 28 \times 1024$			

3.3 The Reliability Object Layer

In this sub-section, the functionality of the proposed reliability object layer is discussed. The developed layer aims to penalize the regions of the objects that appear in the image, taking into account the predicted segmentation masks. In particular, the generated segmentation mask is actually composed of a set of binary masks, one for each semantic object class. For each pixel, a probability is computed that denotes how possible is a certain pixel to belong to a given class. The probability corresponding to the class 'background' is neglected. Thus, the generated gray-scale segmentation mask contains pixel-level object classification information for all classes, with values close to 1 to be considered more reliable

and values close to 0 to be less reliable. Specifically, let $M = \{x_1, x_i, ..., x_N\}$ be the set of the N predicted binary segmentation masks for a given image. Initially, class 'background' is removed from M; hence, $M' = \{x_1, x_i, ..., x_{N-1}\}$ includes the pixel-level predicted probabilities for all classes except the background one. Then, the highest probability for each pixel is considered, resulting to the 2D probability matrix R. As already mentioned, the aim of integrating local-level information is to boost visual features that are selected taking into account semantic information, i.e. object classification decisions. For achieving this, element-wise multiplication is applied between the last feature map of the classification stream and matrix R.

4 Experimental Results

4.1 Employed Datasets

In order to evaluate the performance of the proposed approach in different domains and scales, the following datasets were used:

Terrorist related dataset that has been generated using visual content of the highly challenging domain of on-line terrorist propaganda videos. Specifically, the collected dataset consists of 9191 annotated images, divided into training, gallery and test sets that comprise 5000, 7407 and 1784, respectively.

PASCAL-VOC2012 [5] dataset is used in order to train the semantic segmentation architecture (Fig. 2). This dataset contains approximately 2912 images with pixel-level ground truth annotation and supports 20 semantic classes (plus one for background). 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. In the hash code learning phase, the labeled images of the training and validation sets are used. In order to maintain a fair comparison, images that have been used to train the PSP module have now been excluded. In particular, the dataset consists of 9267 images in total; 5000 of them are used in the training set, 1000 are randomly selected as query images and the gallery set contains 8267 instances.

CIFAR-10 [12] dataset is also used. This dataset consists of approximately 60000 images. 5000 images are randomly selected (500 per class) for defining the training set. The query set consist of 1000 images (100 per class) while the gallery set consists of 59000 images.

AWA2 [25] dataset consists of a total of 37322 images , belonging to 50 classes is used. The framework was trained using a set of 10000 images and 1000 images were selected for the query set; the gallery set consists of 36322 images.

4.2 Implementation Details

For training the semantic segmentation stream (Fig. 2), the AdaGrad optimizer [4] was used. The total number of epochs was set to 40 and the defined batch size was set to 64. For the classification stream (Fig. 2), 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 learning rate was initially selected equal to 10^{-3} and was subsequently decreased to 10^{-4} after 20 epochs. The total number of epochs was set to 40 and the defined batch size was set equal to 40. All input images were resized to 256×256 and then were cropped to 224×224, using a square window placed at the center of the image. All implementation activities were carried out using the Keras [1] framework and a Nvidia GTX 1070 GPU with 8GB memory.

4.3 Evaluation Results

For evaluation, the metric defined in [17] was adopted. In particular, a ranking Mean Average Precision (MAP) value was estimated 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 computed for the top-1000 retrieved images.

The SSDH method was selected in order to have a comparison between the proposed architecture and the state-of-art approaches. It should be noted that the SSDH method not only exhibits state-of-art results, but also it is characterized by relative implementation simplicity. Table 2 illustrates the obtained retrieval results from the application of the proposed method in each dataset, while the performance of the SSDH approach is also given. Additionally, different hash code length experiments are also given. Specifically, experiments with hash code length equal to 12, 24, 32 and 48 bits have been carried out. From the obtained results, it can be seen that the proposed method outperforms the SSDH one in most cases. In particular, the proposed method outperforms significantly the SSDH approach when the length of the hash code is 12 bits.

Table 2. Comparative evaluation results using the MAP@1000 metric

Hash code length	Method	Dataset			
		Terrorist related dataset	PASCAL-VOC2012	CIFAR-10	AWA2
12 bits	SSDH	55.93%	56.83%	40.25%	57.16%
	Proposed	**60.16%**	**62.44%**	**51.95%**	**60.80%**
24 bits	SSDH	68.00%	**66.27%**	69.27%	73.83%
	Proposed	**68.71%**	64.36%	**70.80%**	**77.30%**
32 bits	SSDH	69.05%	**70.30%**	**76.46%**	79.08%
	Proposed	**70.94%**	68.98%	75.60%	**81.11%**
48 bits	SSDH	69.13%	69.55%	76.59%	**83.91%**
	Proposed	**72.13%**	**71.65%**	**78.48%**	83.71%

Indicative retrieval results of the 12 bits experiments for the AWA2 and PASCAL-VOC2012 datasets are shown in Fig. 3. From a detailed examination

[1] https://github.com/fchollet/keras

of the provided results, it can be seen that the proposed method exhibits significantly better results when the visual content of the images needs to be compressed in few bits. It is worth noting here that the CIFAR-10 dataset consists of tiny images that contain single objects; so, the proposed architecture acts like a background removal algorithm, aiming at focusing on a single object. Moreover, a significant improvement is shown (in table 2) when the AWA2 dataset is used. Since, the predicted segmentation masks and consequently the binary masks, which have been generated from the ROL ones, typically contain animals (such as 'cat', 'dog', 'bird', 'cow', 'horse', 'sheep', etc), it is obvious that the respective ROL masks would exhibit high confidence scores (i.e. robust object classification decisions). This is mainly due to multiple animal classes are also present during the local-level information (segmentation) learning phase. In other words, the semantic classes of this particular dataset coincide with classes of the PASCALVOC-2012 one; hence, more accurate ROL masks can be generated. This suggests that incorporating local-level information (semantic segmentation) in the hash code learning phase and boosting features that correspond to objects with high classification confidence can also improve the retrieval performance of a deep hashing framework.

Fig. 3. Indicative retrieval results in AWA2 (top) and PASCAL-VOC2012 (bottom) datasets. The query image, for each dataset, is shown on the left and the top-10 retrieved images (for both the SSDH and the proposed method) are illustrated on the right.

In order to provide a deeper insight on the obtained results, Fig. 3 (top) shows the top-10 retrieved images for both the SSDH and the proposed method, when the same image is used as query. Additionally, similar obtained results are given for the case of the PASCALVOC-2012 dataset (Fig. 3, bottom). From the illustrated results, the superior performance of the proposed approach is demonstrated. It needs to be highlighted that for both queries, not only the

number of relevant returned images, but also their ranking, is improved for the case of the proposed approach. This is mainly due to the proposed method paying more attention to the objects of the image, such as animals and airplanes, while providing decreased importance to the features of that belong to the background.

5 Conclusions

In this work, a novel deep learning layer was proposed in order to construct binary hash codes which take into account local-level semantic information. The proposed framework was evaluated in four different and diverse datasets. The proposed architecture exhibited significantly improved performance, compared to the baseline approach (SSDH) that makes use of only image level information. The experimental results also demonstrated that boosting local-level features using local semantic information, in the form of ROL 2D masks is advantageous. Future work includes the investigation of alternative ways of using local-level information for generating more expressive ROL masks, which will consequently be used in deep hashing schemes.

References

1. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence **39**(12), 2481–2495 (2017)
2. Cao, Z., Long, M., Wang, J., Yu, P.S.: Hashnet: Deep learning to hash by continuation. arXiv preprint arXiv:1702.00758 (2017)
3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. vol. 1, pp. 886–893. IEEE (2005)
4. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research **12**(Jul), 2121–2159 (2011)
5. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html
6. Gionis, A., Indyk, P., Motwani, R., et al.: Similarity search in high dimensions via hashing. In: Vldb. pp. 518–529 (1999)
7. Gong, Y., Lazebnik, S., Gordo, A., Perronnin, F.: 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)
8. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Computer Vision (ICCV), 2017 IEEE International Conference on. pp. 2980–2988. IEEE (2017)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
10. Hong, S., Noh, H., Han, B.: Decoupled deep neural network for semi-supervised semantic segmentation. In: Advances in neural information processing systems. pp. 1495–1503 (2015)

11. Kong, W., Li, W.J.: Isotropic hashing. In: Advances in neural information processing systems. pp. 1646–1654 (2012)
12. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images (2009)
13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
14. Kulis, B., Darrell, T.: Learning to hash with binary reconstructive embeddings. In: Advances in neural information processing systems. pp. 1042–1050 (2009)
15. Lai, H., Pan, Y., Liu, Y., Yan, S.: Simultaneous feature learning and hash coding with deep neural networks. arXiv preprint arXiv:1504.03410 (2015)
16. Li, W.J., Wang, S., Kang, W.C.: Feature learning based deep supervised hashing with pairwise labels. arXiv preprint arXiv:1511.03855 (2015)
17. Lin, K., Yang, H.F., Hsiao, J.H., Chen, C.S.: Deep learning of binary hash codes for fast image retrieval. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2015 IEEE Conference on. pp. 27–35. IEEE (2015)
18. Liong, V.E., Lu, J., Wang, G., Moulin, P., Zhou, J., et al.: Deep hashing for compact binary codes learning. In: CVPR. vol. 1, p. 3 (2015)
19. Rastegari, M., Farhadi, A., Forsyth, D.: Attribute discovery via predictable discriminative binary codes. In: European Conference on Computer Vision. pp. 876–889. Springer (2012)
20. Semertzidis, T., Rafailidis, D., Strintzis, M.G., Daras, P.: The influence of image descriptors dimensions value cardinalities on large-scale similarity search. International Journal of Multimedia Information Retrieval **4**(3), 187–204 (2015)
21. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
22. Wang, X., Shi, Y., Kitani, K.M.: Deep supervised hashing with triplet labels. In: Asian Conference on Computer Vision. pp. 70–84. Springer (2016)
23. Weiss, Y., Torralba, A., Fergus, R.: Spectral hashing. In: Advances in neural information processing systems. pp. 1753–1760 (2009)
24. Xia, R., Pan, Y., Lai, H., Liu, C., Yan, S.: Supervised hashing for image retrieval via image representation learning. In: AAAI. vol. 1, p. 2 (2014)
25. Xian, Y., Lampert, C.H., Schiele, B., Akata, Z.: Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. arXiv preprint arXiv:1707.00600 (2017)
26. Yang, H.F., Lin, K., Chen, C.S.: Supervised learning of semantics-preserving hash via deep convolutional neural networks. IEEE transactions on pattern analysis and machine intelligence **40**(2), 437–451 (2018)
27. Zhang, R., Lin, L., Zhang, R., Zuo, W., Zhang, L.: 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)
28. Zhao, F., Huang, Y., Wang, L., Tan, T.: Deep semantic ranking based hashing for multi-label image retrieval. In: Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on. pp. 1556–1564. IEEE (2015)
29. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 2881–2890 (2017)
30. Zhong, G., Xu, H., Yang, P., Wang, S., Dong, J.: Deep hashing learning networks. In: Neural Networks (IJCNN), 2016 International Joint Conference on. pp. 2236–2243. IEEE (2016)