# CNN-AWARE BINARY MAP FOR GENERAL SEMANTIC SEGMENTATION

*Mahdyar Ravanbakhsh*[⋆1], *Hossein Mousavi*[*2], *Moin Nabi* [3], *Mohammad Rastegari* [4], *Carlo Regazzoni* [1]

[1] University of Genova [2] Istituto Italiano di Tecnologia [3] University of Trento [4] Allen Institute for AI

## ABSTRACT

In this paper we introduce a novel method for general semantic segmentation that can benefit from general semantics of Convolutional Neural Network (CNN). Our segmentation proposes visually and semantically coherent image segments. We use binary encoding of CNN features to overcome the difficulty of the clustering on the high-dimensional CNN feature space. These binary codes are very robust against noise and non-semantic changes in the image. These binary encoding can be embedded into the CNN as an extra layer at the end of the network. This results in real-time segmentation. To the best of our knowledge our method is the first attempt on general semantic image segmentation using CNN. All the previous papers were limited to few number of category of the images (e.g. PASCAL VOC). Experiments show that our segmentation algorithm outperform the state-of-the-art non-semantic segmentation methods by large margin.

***Index Terms***— Image Segmentation, Convolutional Neural Networks.

## 1. INTRODUCTION

Image segmentation is a challenging task in computer vision that can specify the visual elements in an image. These elements can be used as the building blocks for any image understanding method. Traditionally, these image segments are optimized to be semantic (e.g. be an object, part of an object, or part of a scene) and visually coherent; This means that nearby pixels in each segment must have similar intensity [1, 2, 3]. Semantic image segmentation has been proposed in several articles [4, 5, 6, 7, 8]. All of these methods are limited to a narrow scope of semantics. They can only find the segments belong to *few categories* of objects (e.g. 20 categories in PASCAL VOC dataset). In this paper a method is proposed that can find *general* semantic segments.

Recently there has been a remarkable progress in computer vision through Deep Neural Networks. More specifically, with Convolutional Neural Networks (CNNs) an end-to-end object recognition has been created [9, 10, 11] that outperformed all of the previous recognition systems. These learning methods recently became more popular than traditional statistical learning techniques. CNN is a multi-layer neural network; in each layer the weights are in the shape of filters. The output of each layer is the result of convolution of filters on that layer with the input. After several layers of convolution the output can be used as a feature representation of an image. For example, in AlexNet [9] architecture the output of *fc7* layer has been used extensively as a generic image descriptor. Moreover, [12] showed that these features are so powerful that can be used for a variety of tasks in computer vision. Given an image as input we can apply a fully-convolutional neural network to obtain a feature vector per each receptive-field in the image[5]. Since these features carry semantical information about the input image, they can be used to find image segments that are semantically coherent. In this paper, we show how these segments can be extracted from such CNN features.

CNN features are very high-dimensional (namely, 4096). Traditional segmentation approaches that are mainly based on clustering techniques [13] are not feasible. Since we want each segments corresponds to a meaningful visual element, large number of cluster centers are essential. That makes the

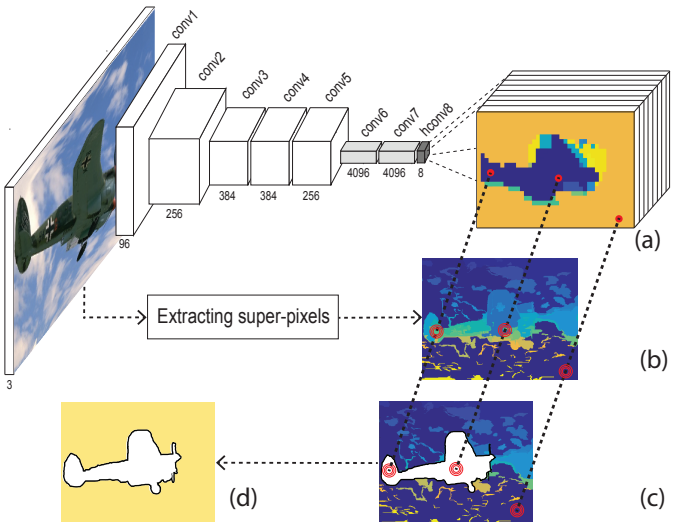



**Fig. 1**. Method overview: given an image the semantic binary map is extracted by forward-pass of image through the net (a), a low-level superpixel is extracted (b), the binary code of each superpixel is assigned using the corresponding region on the binary map (c), finally semantic segmentation is generated merging superpixels with similar binary patterns (d).

*Equal contribution.

segmentation process further more complex. To overcome such computational complexities, binary encoding of CNN features has proposed instead. A CNN feature is converted to a short binary code: each bit pattern represents a cluster center in the original CNN feature space. For example, an 32-bit binary code can generate $2^{32}$ clusters. Each bit corresponds to a visual attribute. Nearby pixels should have similar binary patterns unless they undergo a large semantical change. This is a perfect property to be used for semantic segmentation. Iterative-Quantization [14] is employed to learn these binary codes. A powerful feature of the ITQ is that it generates bits in a simple way and the transformation is linear. This is perfect setting to be embedded in the CNN networks as a new layer. Once the binary map of the CNN features is available, a low-level superpixel extraction method is applied on the whole image and then the superpixels with the similar binary patterns (under Hamming distance) are merged together.

Our major contributions in this work can be summarized as; *first:* We proposed a semantic segmentation which can be used in a general setting, unlike the all previous methods that are limited to specific categories. *second:* We introduced a compact representation of high-dimensional CNN features in the form of binary codes, to preserve semantic information, thus can be used for semantic segmentation. Hence, we present a binary encoding layer in our network, which can updates using back-propagation. This new layer is able to be attached to any other deep-nets for encoding purposes.

## 2. RELATED WORK

Despite a large body of works on low-level segmentation, there few works target semantic segmentation, and to the best of our knowledge, there is no work doing *general* semantic segmentation utilizing high-level CNN features.

**Low-level segmentation:** Low-level segmentation refers to partitioning an input image into a set of perceptually meaningful atomic regions, considering the low-level image features, like intensity, edge, or texture. This step is usually considered as a pre-processing step which can effectively be employed to reduce complexity of subsequent visual recognition tasks. In literature, apart from the core low-level feature used, a substantial debate has been mainly posed over the optimization algorithms employed to efficiently solve this partitioning problem. In this context, two classes of approaches can be identified [2]. On one hand, *graph-based* methods treat pixels as nodes in graph, connected each other via edges reflecting their similarity in the feature space. Then, the graph is partitioned into a set of sub-graphs corresponding to image segments by minimizing a cost function. Among the best performing methods, Normalized-Cuts [13], Super-pixel Lattices [15], and Efficient Graph-based Segmentation (EGS) [1] can be quoted. Here in this work, our method is compared with the last work, selected as one of the best performing non-semantic graph-based segmentation methods.

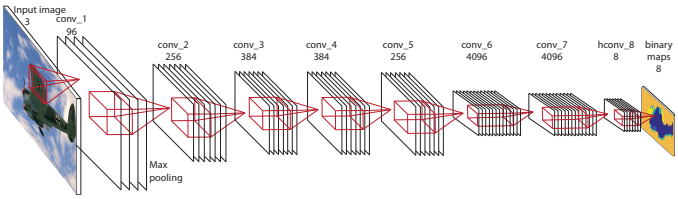



**Fig. 2**. Architecture of our proposed segmentation network.

On the other hand, a different set of methods, named *gradient-ascent-based* approaches, starts with an initial clusters of pixels, then refines iteratively until convergence in visual consistency. In this line of research, Mean-shift [16], Turbo-pixel [17] and state-of-the-art SLIC [2] should be mentioned. Note that, SLIC has picked as one of our baselines and compare our method with.

**Semantic segmentation:** Recently, visual recognition task has come to rely increasingly on segmentation, and region extraction, accordingly, emerged to play a key role in object detection [18] and activity recognition[19, 20]. Due to the fact that the quality of initial segmentation affects significantly the subsequent tasks, providing segments capturing higher-level semantics became crucial. Semantic segmentation often formulated as combining low-level segments with region-based object detectors either in a cascade [21, 22, 23] or joint [24, 25, 26] manner. Convolutional Neural Networks have recently resurfaced as a powerful tool for learning to segment semantically [4, 5, 7, 6]. Nevertheless, learning such supervised deep structures for higher number of categories (and samples) is so supervision-demanding and computationally-expensive. Very recently, ADE20K [27] has been introduced in which a wider variety of scenes and objects are annotated. Even in this case, extending the current supervised DNNs to work in a zero-shot fashion (namely, the categories other than the ones exist in the dataset) is not trivial. In this work, however, a completely different perspective to semantic segmentation is picked out. We specifically propose a method to narrow down the semantic gap (between pixels and concepts) in images, namely, trying to inject semantic inherited from generic CNN representations, so leading to more general semantic segmentation while maintaining the method complexity to a manageable level.

## 3. CNN-AWARE BINARY MAP OF IMAGE SEGMENTS

In this section, the two major parts of our proposed method are described in details: *1) Spatial-aware fully convolutional network*, *2) Binary map encoding layer* and *3) Semantic segmentation using binary maps*. Figure 1 illustrates a general work-flow of the method.

### 3.1. Spatial-aware fully convolutional network

Early convolutional layers in CNNs represent more local information of the image, while deeper ones contain more global information. The fully-connected layers capture higher-level information and usually employed for recognition purposes. It has been shown that the deep nets which trained on ImageNet [28], are rather semantic; they can address wide range of recognition problems [29, 12]. Fully convolutional Nets also can preserve relative spatial coordinates between input image and output feature map. These properties motivated us to use such structures for general semantic segmentation.

For the sake of generalization, we adopted a pre-trained network (Namely, AlexNet [9]) and simply converted it to a fully convolutional net. It provides us with general, yet spatially-consistent semantic representations. Due to the semantic power of the *fc7* layer in case of AlexNet, we exploited the corresponding layer in our network (denoted by $conv7$) to extract feature maps.

### 3.2. Binary map encoding layer

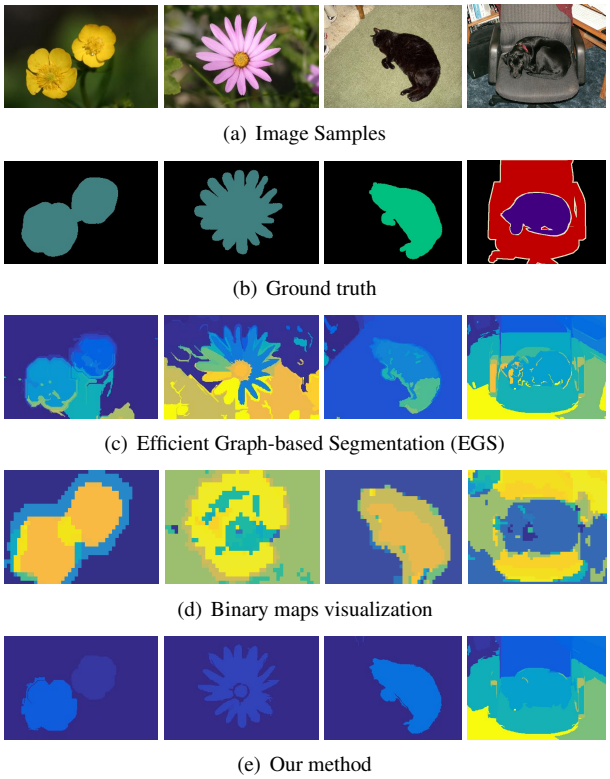
Clustering the extracted high-dimensional feature maps from $conv7$ comes with high computational cost. It leads to converge to a limited number of clusters. One possible solution to avoid this problem is partitioning high-dimensional features into a set of buckets (instead of clusters) using hashing techniques. It provide use with generating small binary codes for each feature vector, taking into account their distance simultaneously. Assuming 24-bits of binary code can address $2^{24}$ buckets. Moreover, this binary map can be represented as a 3-channels RGB image, providing a better illustration on partitions. Obviously, dealing with binary codes comes with lower computational cost and higher efficiency with respect to other clustering methods. However, the most advantage of hashing comparing to clustering, is the capability of embedding it simply as a layer inside the network.

**Binary encoding layer:** Encoding feature maps to binary codes is computed by Iterative Quantization Hashing (ITQ)[30]. It is a unsupervised binary codes method, which projects each high-dimensional feature vector into a binary space. The last layer of our network (denoted by $hconv8$) is built by the hashing linear transformation, learned initially by ITQ. During testing, for each input image to the network, a spatial-aware binary map would be generated. For this purpose, we forward-pass the image through all the convolutional layers as well as the final binary encoding layer. Such binary map is eventually used for the segmentation purpose.

### 3.3. Semantic segmentation using binary map

The generated binary maps have two important aspects: *first;* it preserves the spatial relation between input image and output features. In other word, each region on the binary map corresponds to a patch on the input image. *second;* binary maps are generated using the convolutional feature maps of the deep-net, hence they capture the semantics of the scene. Binary pattern with different values represent areas with different semantics. It specifically interesting because, any changes in the binary code patterns on binary maps can be interpreted as a semantics change on the corresponding areas on the image.

We take advantage of these two properties to employ the binary map for semantic segmentation. To this purpose, we initialized the segmentation by low-level superpixels, then merged superpixels with the similar binary codes in the binary map. This simple yet effective criteria on semantic features has shown to be much more powerful compares to the previous state-of-the-art methods which utilized the sophisticated partitioning algorithms but relaying only on low-level visual information.



(a) Image Samples

(b) Ground truth

(c) Efficient Graph-based Segmentation (EGS)

(d) Binary maps visualization

(e) Our method

**Fig. 3**. Our segmentation method compare with EGS [1]

## 4. EXPERIMENT

A set of experiments has designed to evaluate our method. This section, explains the experimental setup, evaluation protocol, datasets, and finally elaborate on the results.

**Experimental Setup:** Set of comparative experiments has been done to demonstrate the advantage of using our binary map for semantic segmentation compares to two best-

performing low-level segmentation methods: *Efficient Graph-Based Segmentation (EGS)*[1] and *gradient-ascent-based SLIC*[2]. Since there is no pre-processing step or parameter setting in our method, For the baselines, we used the publicly available codes with the default parameters ($\sigma = 1.0, k = 100$). In this means, we aims at showing the strength of the semantic segmentation compares to low-level segmentation without any parameter tuning.

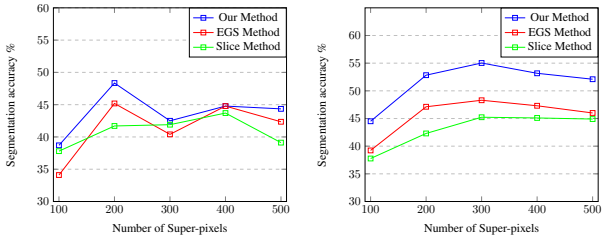
**Dataset:** We choose two datasets; The first dataset is the Berkeley Segmentation Dataset (BSDS500), includes 300 training samples as and 200 images for testing. The other, Microsoft Research Cambridge database (MSRC), includes 510 images. Both evaluation has been performed with the original setups of datasets.

**Evaluation Protocol:** we adopted the Segmentation Intersection over Union (IoU) as one of the most commonly used evaluation measure for segmentation task. IoU describe as:

$$IoU(P_m, P_{gt}) = \frac{|P_m \bigcap P_{gt}|}{|P_m \bigcup P_{gt}|}$$

Where $P_{gt}$ is ground truth segment annotation, and $P_m$ is predicted segment. As the predicted segments, we select the segments with the maximum IoU with each segment in $P_{gt}$. The final value of *Segmentation-IoU* is computed as the average over all the segments of all the images of the dataset.

**Segmentation Network Details:** The designed deep-net consist of two major parts; *1) Fully convolutional network:* As reviewed in 3.1 at first we utilized a pre-trained AlexNet model on ImageNet. Original AlexNet, contains 5 convolutional layers and two fully connected layers. In order to obtain spatial-aware feature maps, we convert the last two fully connected layers into convolutional layers. By transforming fully connected layers into convolutional layers we could enable the net to output a multi dimensional feature map disregard to input image size and produce an efficient model for spatial-aware patch pooling. In our experiment we used images with higher size to produce finer feature maps. High dimensional feature maps(28x44x4096) extracted from the last convolutional layer feed to a new layer which we called *Binary bit-map Layer* to compress into a lower dimension binary bit map (28x44x8). *2) Binary Bit-map layer:* Bit-map layer is designed for convolutional feature map quantization. In order to build the layer, we first extract convolutional maps from $conv7$ over PASCAL 2007 images to train an unsupervised ITQ hash to model 4096 dimensional feature maps into 8-bits binary codes. Hashing weights obtain from ITQ applied into a Depth Normalization Layer. We embedded the Depth Normalization Layer with pre-trained weights to the network to build *Binary Bit-map layer*. Output of the network is a set of 8-bit binary maps. Figure 1 shown a visualization example of extracted binary map in the form of grayscale image, which is spacial aware and contains semantic information about the image. This bit-mat image is eventually utilized for image segmentation.




(a) Segmentation-IoU on Berkeley (b) Segmentation-IoU on MSRC

**Fig. 4**. Segmentation-IoU over superpixel variation

| MSRC | | Berkeley | |
|---|---|---|---|
| Method | IoU | Method | IoU |
| EGS [1] | 50.3% | EGS [1] | 45.19% |
| SLIC [2] | 48.7% | SLIC [2] | 43.70% |
| Our method | **55.03 %** | Our method | **48.35%** |

**Table 1**. Quantitative results on MSRC and Berkeley datasets.

**Segmentation Strategy:** For segmentation, we first extract the binary maps for each input image. For each superpixel a binary code is assigned by the corresponding region on binary map. Then we merged the superpixels with the similar binary codes on the bit maps (i.e., zero distance in Hamming space), Figure 3(c). The final segmentation is obtained as the result of such merging of superpixels, Figure 3(e).

**Experimental Result:** Our proposed semantic segmentation significantly outperform previous low-level segmentation methods, both quantitatively, and qualitatively. A comparison results on two datasets demonstrated in Table 1. The first column shows a comparison of average segmentation-IoU for the algorithms in the MSRC dataset, and the second columns compares the algorithms on BSDS500. Qualitative comparison with EGS method, Figure 3, also shows the better results achieved by our approach. Figure 3 visualize the results of our method and EGS method. We outperform both baseline methods by large margins in term of $segmentation - IoU$ over different superpixel sizes (Figure 4). Such evaluation shows the robustness of the proposed method to the number of super-pixels.We observe that segmentation on images containing "things" (objects) are significantly better as compared to images containing "stuffs" (scenes). It also supports our hypothesis that binary patterns preserved semantic information and the understanding objects in the scene.

## 5. CONCLUSION

In this work a novel approach to general semantic-aware image segmentation has been presented which does no require category-specific training a deep-net. We employed AlexNet as pre-trained model and convert fully connected layers into convolutional layers. An efficient ITQ hashing layer is at-

tached as the final layer to the net to quantity high dimensional feature maps in form of binary code representation. Such model provides both spatial consistency as well as low dimensional semantic embedding. Our experimental results shown using these binary maps can improve the performance of the segmentation comparing to several low-level segmentation methods. As future work, we will study fine tuning the hashing layer with back-propagation and end-to-end training of semantic segmentation net with recent Region Proposal Network (RPN) in a joint manner.

## 6. REFERENCES


[1] P. F. Felzenszwalb and D. P Huttenlocher, "Efficient graph-based image segmentation," in *IJCV*. 2004, Springer.

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," in *PAMI*, 2012.

[3] A. Khoreva, R. Benenson, F. Galasso, M. Hein, and B. Schiele, "Improved image boundaries for better video segmentation," *arXiv preprint arXiv:1605.03718*, 2016.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.

[5] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015.

[6] LC Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," in *arXiv:1412.7062*, 2014.

[7] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. HS Torr, "Conditional random fields as recurrent neural networks," in *ICCV*, 2015.

[8] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *ICCV*, 2015.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

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

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015.

[12] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *arXiv:1310.1531*, 2013.

[13] J. Shi and J. Malik, "Normalized cuts and image segmentation," in *PAMI*. 2000, IEEE.

[14] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," in *PAMI*, 2013.

[15] A. P. Moore, JD Prince, J. Warrell, U. Mohammed, and G. Jones, "Superpixel lattices," in *CVPR*, 2008.

[16] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," in *PAMI*, 2002.

[17] A. Levinshtein, A. Stere, K. N Kutulakos, D. J Fleet, S. J Dickinson, and K. Siddiqi, "Turbopixels: Fast superpixels using geometric flows," in *PAMI*, 2009.

[18] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," in *PAMI*, 2016.

[19] M. Jain, J. Gemert, H. Jégou, P. Bouthemy, and C. Snoek, "Action localization with tubelets from motion," in *CVPR*, 2014.

[20] G. Gkioxari and J. Malik, "Finding action tubes," in *CVPR*, 2015.

[21] P. Arbeláez, B. Hariharan, Chunhui Gu, S. Gupta, L. Bourdev, and J. Malik, "Semantic segmentation using regions and parts," in *CVPR*, 2012.

[22] J. Carreira, F. Li, and C. Sminchisescu, "Object recognition by sequential figure-ground ranking," in *IJCV*, 2012.

[23] K. EA. Van de Sande, J. RR. Uijlings, T. Gevers, and A. WM Smeulders, "Segmentation as selective search for object recognition," in *ICCV*, 2011.

[24] J. Dong, Q. Chen, S. Yan, and A. Yuille, "Towards unified object detection and semantic segmentation," in *ECCV*, 2014.

[25] R. Mottaghi, X. Chen, X. Liu, N. Cho, SW. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *CVPR*, 2014.

[26] J. Yao, S. Fidler, and R. Urtasun, "Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation," in *CVPR*, 2012.

[27] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba, "Semantic understanding of scenes through the ade20k dataset," *arXiv preprint arXiv:1608.05442*, 2016.

[28] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.

[29] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *CVPRW*, 2014.

[30] Y. Gong and S. Lazebnik, "Iterative quantization: A procrustean approach to learning binary codes," in *CVPR*, 2011.