



Semantic Guided Deep Unsupervised Image Segmentation

Sudipan Saha^{1,2} , Swathikiran Sudhakaran^{1,2}, Biplab Banerjee³,
and Sumedh Pendurkar⁴

¹ Fondazione Bruno Kessler, Trento, Italy
{saha,sudhakaran}@fbk.eu

² University of Trento, Trento, Italy

³ Indian Institute of Technology Bombay, Mumbai, India
bbanerjee@iitb.ac.in

⁴ College of Engineering Pune, Pune, India
sumedh.pendurkar@gmail.com

Abstract. Image segmentation is an important step in many image processing tasks. Inspired by the success of deep learning techniques in image processing tasks, a number of deep supervised image segmentation algorithms have been proposed. However, availability of sufficient labeled training data is not plausible in many application domains. Some application domains are even constrained by the shortage of unlabeled data. Considering such scenarios, we propose a semantic guided unsupervised Convolutional Neural Network (CNN) based approach for image segmentation that does not need any labeled training data and can work on single image input. It uses a pre-trained network to extract mid-level deep features that capture the semantics of the input image. Extracted deep features are further fed to trainable convolutional layers. Segmentation labels are obtained using argmax classification of the final layer and further spatial refinement. Obtained segmentation labels and the weights of the trainable convolutional layers are jointly optimized in iterations in a mechanism that the deep network learns to assign spatially neighboring pixels and pixels of similar feature to the same label. After training, the input image is processed through the same network to obtain the labels that are further refined by a segment score based refinement mechanism. Experimental results show that our method obtains satisfactory results inspite of being unsupervised.

Keywords: Unsupervised image segmentation · Semantic guided · Deep learning

1 Introduction

Image segmentation refers to the process of extracting perceptually meaningful regions of the image and accordingly assigning unique labels to the image pixels [15]. Image segmentation is treated as a high level image analysis paradigm and

it acts as a precursor step for many visual inference tasks [19]. Image segmentation is a challenging problem given complex interaction between different objects present in the image. Traditionally, image segmentation problem has been dealt in unsupervised way in the literature [9]. The unsupervised segmentation methods generally exploit region based techniques [7] to obtain spatially coherent group of pixels that share the spectral characteristics. The contour based strategies have also been proposed in the literature that connect the points of interest (e.g., edge pixels or corner pixels) in the image to delineate the boundaries separating adjoining regions [1]. Unsupervised segmentation may sometimes involve complex segment threshold calculation [26]. Considering the fact that unsupervised segmentation into arbitrary number of labels is challenging, many unsupervised segmentation approaches restrict the number of labels to foreground and background [16].

In contrary to the unsupervised models, the supervised models exploit available training pixels to learn a classifier model that can be subsequently used to obtain segmentation labels. Usually, low-level shallow features are constructed exploiting the spectral properties and subsequently different feature encodings like bag of words, Vector of Locally Aggregated Descriptors (VLAD) are used for effective representation of the pixels in the feature space. Supervised algorithms, e.g., SVM, Naive Bayes, Neural Networks etc. are henceforth used to learn the classification model. However, sole application of the traditional classifiers in this respect may not preserve the image discontinuities and further may get affected by outliers. Additionally, most supervised methods impose spatial homogeneity constraints using structured learning approaches [3, 4, 12]. Supervised methods are preferred when grouping into many labels is required.

Recently, deep learning, especially, Convolutional Neural Network (CNN) based techniques have obtained state-of-the-art performance in most computer vision tasks. Inspired by this success, several approaches for image segmentation using deep learning have been proposed in the literature. These approaches belong to the category of the supervised approaches and require large training datasets with pixel-level labels. Obtaining abundant labeled data requires lot of effort and may not be available in many domains.

To circumnavigate the absence of labeled data, many image processing applications have recently successfully employed pre-trained deep networks as deep feature extractor in unsupervised [22] and semi-supervised settings [21]. Such features capture the image semantics in a more effective way than the low-level shallow features. Motivated by this, we approach the problem of deep unsupervised image segmentation by exploiting such features cascaded with a series of trainable convolutional layers. We assume a single image input and reference labels of pixels are not present. The input image is processed through a pre-trained VGG Net as deep feature extractor that are further fed to a series of trainable convolutional layers. The final convolutional layer is processed through a decision process to obtain the predicted labels. Predicted labels and the learnable parameters of the trainable convolutional layers are jointly optimized in iterations such that the deep network keeps evolving learning to identify the

unique segments present in the image. More specifically, the network learns to assign pixels having similar features and neighboring pixels to the same label. After completion of training process, same network is used to obtain the labels from the input image. To further address over-segmentation problem, the labels are refined through a segment score based refinement mechanism.

The rest of this paper is organized as follows. A number of related methods are discussed briefly in Sect. 2. We describe the proposed algorithm in Sect. 3. We present the experimental results in Sect. 4. Finally, we conclude the paper and discuss scope of future works in Sect. 5.

2 Related Works

Considering the attention of the proposed work, we mainly discuss the deep learning based approaches on image segmentation. A significant number of deep learning based segmentation algorithms have been proposed in the recent past.

2.1 Supervised Deep Segmentation

The supervised approaches for the image segmentation is implicitly same as the pixel level classification task given a set of reliable training pixels. Generally, the segmentation methods employing deep neural network relies on bottom-up region proposals generation techniques to supervise the segmentation process. There has been a number of works involving deep neural networks for image segmentation [2, 5, 6, 10, 13, 17, 20, 27]. Though not a deep learning based method, in [5] one of the first applications of pooling layer was proposed for semantic segmentation. Pooling forms indispensable part in most CNNs recently. In [10], region proposal is combined with CNN for object detection and semantic segmentation. Fully convolutional networks (FCNs) [13] that replace the fully connected layers with the convolutional layers is one of the simple yet effective model for supervised semantic segmentation problem. Such network has ability to take input of arbitrary spatial dimension and generate same size pixelwise segmentation map. [20] presents a variant of FCN that has a U-shaped architecture that supplements an usual contracting network by successive layers to capture context and a symmetric expanding path that improves the localization accuracy. Compared to FCN, U-Net has more upsampling layers and utilizes learnable weights instead of fixed interpolation strategies. SegNet [2] is another variation that uses a novel strategy to decode or upsample encoded features by storing the max-pooling indices used in pooling layer. In spite of varying architectures, all these methods require substantial amount of training data and hence their use on every application domain is not possible. In case they are used in other application domains, they are trained with the images from that domains, e.g., semantic segmentation for remote sensing images [27].

2.2 Unsupervised Deep Segmentation

As deep learning techniques are maturing, there has been an increased interest in exploiting deep learning techniques in unsupervised way. There are two major trends in this direction, transfer learning based methods that effectively transfers a network trained for a task on other tasks [23] and Generative Adversarial Network based methods that still require a lot of unlabeled data [24]. Aligned with trend of increased interest in unsupervised deep learning, very recently we observe that few works have been proposed to address the image segmentation problem in an unsupervised way [11, 28]. In [28], popular U-Net architecture is modified to a W-shaped network that is optimized to reconstruct the input images and simultaneously predicts a segmentation map without using any labeling information. However, the network is pretty complex consisting of 46 convolutional layers that are grouped into 18 modules that are further grouped into two groups of 9 modules each. The first one of these two forms the dense encoding and prediction part of the network and the second one forms the reconstruction decoder. In our opinion, such network complexity is not desired in unsupervised settings that involve zero to few training samples. Even though the method in [28] don't use label information for training, it still requires a substantially big dataset for training (trained on Pascal VOC 2012 dataset [8]). Our proposed method involves an architecture much simpler than this and still obtains reliable result. Inspired by unsupervised deep clustering [29], in [11] an unsupervised image segmentation method is proposed that uses few convolutional layers. The output label is obtained at the final layer using argmax classification that is further regulated by superpixel based refinement. Predicted pixel labels and network weights are jointly optimized in an iterative fashion by gradient descent method. Our method is related to this method in essence that we also build up on joint optimization of pixel labels along with weights of the learnable convolutional layers. However, instead of feeding the learnable layers with raw image inputs, our semantically guided method feeds the deep features extracted from an intermediate layer of a pre-trained network. Such features that carry much rich information than pixel values guides the subsequent learnable layers to optimize in more effective fashion. Moreover, we exploit mode statistics based spatial filtering along with argmax classification to obtain image labels that captures the spatial context in contrary to [11] that uses a superpixel based refinement. We also propose a segment score based refinement step to effectively reduce the over-segmented clusters.

3 Proposed Algorithm

3.1 Problem Definition

Let us assume that we have a RGB image X consisting of a set of N pixels x_n ($n = 1, \dots, N$). The label information related to any pixels in x_n ($n = 1, \dots, N$) is not known. Our goal is to assign meaningful segment labels c_n^{final} to all of the N pixels by learning a mapping function $f(x_n)$. The mapping function

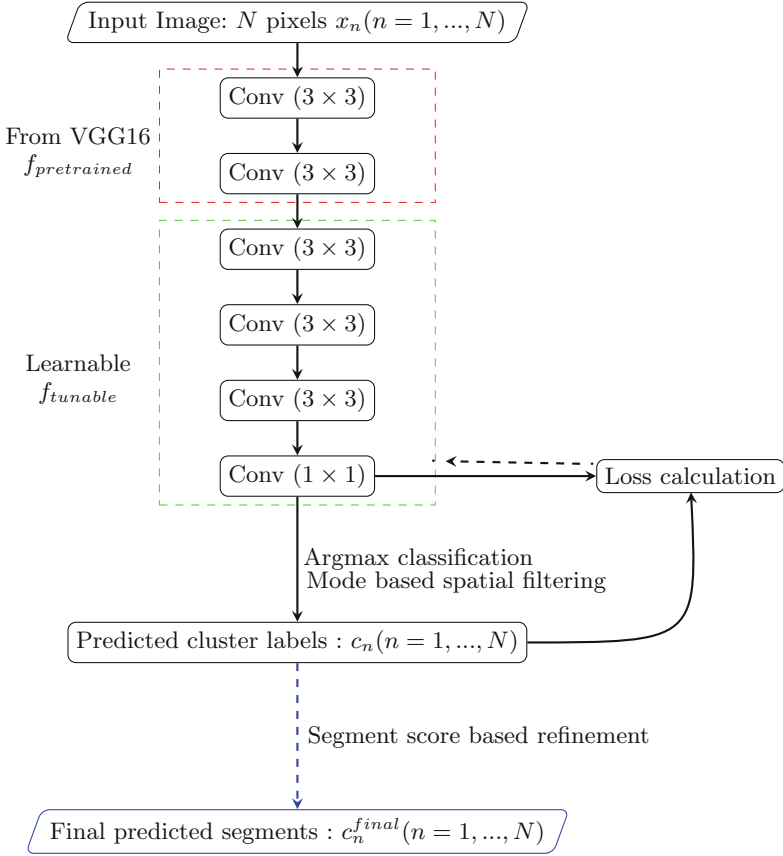


Fig. 1. Proposed unsupervised deep image segmentation framework. Input image are processed through a fixed pre-trained feature extractor $f_{pretrained}$ followed by learnable $f_{pretrained}$. Labels c_n corresponding to all input pixels x_n ($n = 1, \dots, N$) are shown obtained through a decision process consisting of argmax classification and spatial filtering. Training loss is computed using output of the final conv. (1×1) layer and the predicted labels. The segment score based refinement to obtain final predicted labels is not performed during training iterations. ReLu activation function and batch normalization layers are not shown here.

is learned using a series of convolutional layers, however in an unsupervised way, i.e., not using any label information about x_n . In designing the mapping function $f(x_n)$, we are guided by following notions:

1. **Knowledge from other:** Use of pre-trained network as a semantic guide to enhance the segmentation process, i.e., to harvest the knowledge that has been already learned by such a network for other tasks.

2. **Simple:** Keeping the network shallower, not using complex operations like superpixel segmentation that may produce varying result according to input parameters. No hard-coded intensity threshold between image segments.

3.2 Algorithm Gist

Inspired by the success of CNN features for transfer learning [23] and mid level CNN features for different visual tasks [22,30], we process the input image through a pre-trained network and obtain deep features from it. Instead of pixel values or shallow image features, deep features obtained from a pre-trained networks are most robust and hence suitable for designing the mapping function $f(x_n)$. However features in pre-trained network tend to be more inclined to the specific distribution of the training dataset and not particularly useful for segmentation task. Hence, we extract feature from an intermediate layer chosen from one of the initial layers of the network. Thus, given the deep feature extraction process, the $f(x_n)$ can be thought of composed of two different mapping functions: $f_{pretrained}(x_n)$ and $f_{tunable}(f_{pretrained}(x_n))$. The $f_{tunable}$ is composed of another set of convolutional layers. Thus input (x_n) processed through these two mapping functions, one followed by other, produces pixel labels c'_n . Given, at a particular iteration, $f(x_n)$ is fixed, it can be used to obtain c'_n that is further modified using a spatial regularization process to obtain $c(x_n)$. In some other iteration, if c_n is kept fixed, then the network $f(x_n)$ (particularly the tunable part $f_{tunable}$ can be tuned (weights adjusted) according to standard supervised learning settings. These two processes can be carried out in alternate fashion. After the training process, the trained network is used to obtain $c_n \forall n = 1 \dots, N$ that are refined using segment score based refinement to obtain c_n^{final} .

An overview of the proposed method is shown in Fig. 1.

3.3 Algorithm Details

Pre-trained Deep Feature Extraction. Here we explain in details the process of deep feature extraction using pre-trained network (i.e., the mapping function $f_{pretrained}(x_n)$). Deep feature extraction process is based on the assumption that features captured by the convolutional filter banks of the pre-trained CNN are more effective in capturing spatial context and spectral information than the raw pixel values. In particular we chose VGG16 network [25] pretrained on ImageNet dataset. We use PyTorch [18] for implementation purposes and pre-trained network is provided by it. As discussed in Sect. 3.2, deep features required for segmentation is not required to be from very deeper layers. Hence, we extract deep features from the second convolutional layer of VGG16. Thus the part of the VGG16 used for feature extraction consists of a convolutional layer followed by a ReLU activation function and another convolutional layer. Since there is no pooling layer upto second convolutional layer, hence we can input image of any size to obtain the feature map of size 64 while retaining the same spatial dimension. The mapping function $f_{pretrained}(x_n)$ transforms 3-band input pixels x_n to 64 channel feature maps. These 64 filter maps provides semantic guidance

to the subsequent steps. Note that this is a striking difference of our method from [11] that uses pixel values as input to the learnable layers.

Learnable Convolutional Layers. Here we describe the details of $f_{tunable}$. Though features extracted from pre-trained deep network is robust, they are not attuned for the given input image. Moreover, we need to obtain a response map y_n corresponding to each pixel x_n to classify the pixels into different labels. Hence, we use learnable convolutional layers designed for these purposes. More specifically, we use four convolutional layers, each followed by batch normalization. The first three convolutional layer consists of filters of size 3×3 and helps in further modulating the features obtained from the pre-trained network. The first one takes 64 dimensional input and converts it into 100 dimensions. The following two take 100 dimensional input and maintains same dimension. The last convolutional layer consists of filters of size 1×1 and is meant to obtain a response map y_n by applying a linear classifier on the features. It takes 100 dimension input and maintains the dimension. The response map is further processed using batch normalization to obtain y'_n . Such normalization helps in controlling number of clusters as described in [11]. All these four layers comprise of learnable weights.

Training Learnable Layers. Here we describe mechanism to adjust weights of $f_{tunable}$. The response map y'_n is processed to obtain the cluster label c_n for each pixel x_n . In details, c'_n for a specific pixel x_n is obtained by argmax classification, i.e., choosing the dimension in y'_n that has maximum value in y'_n [11]. Considering that spatial continuity is important in image segmentation, we use a simple sliding window mode (most common value in a window) based image filtering to further refine the prediction map and we obtain c_n . This also helps in eliminating spurious redundant cluster labels. In this way we force pixel labels to take spatial information into account, however using a very simple process instead of complicated superpixel segmentation. The training process is accomplished in iterations, where in one iteration the c_n is obtained by keeping the weights of the $f_{tunable}$ fixed. In another iteration the weights corresponding to the $f_{tunable}$ are adjusted by keeping the c_n fixed. Cross-entropy loss is calculated between c_n and y'_n and loss is propagated back to the network for weight adjustment. However, weights corresponding to first two convolutional layers (obtained from VGG16) are not modified.

Obtaining Cluster Labels. To accomplish a reasonable training process, the training is executed for 500 iterations. However, if total number of clusters prematurely (i.e., before 500 iteration) reach 3, then the training process is stopped prematurely. In each training iteration, the number of clusters reduces as smaller labels merge into bigger labels. After completion of training iterations, the same network is used to obtain cluster label map (i.e., c_n for each x_n). Note that the cluster labels obtained at this stage need not be spatially continuous and can

have same labels for two spatially disconnected region. They are further refined through a segment based refinement process to obtain final segment map c_n^{final} for each x_n .

Segment Score Based Refinement. We detect each unique label segments from the predicted map consisting of N labels c_n ($n = 1, \dots, N$). This is accomplished by using following principle:

1. Two spatially disconnected regions having same label are considered different segments.
2. Two spatially connected regions having different labels are considered different segments.

Here, we use the concept of 8-neighbor connectivity. Once M segments are detected from the from the cluster label map, we define a segment score $\alpha_{m_1 m_2}$ between each pair of segments m_1 and m_2 in M such that:

1. If segments m_1 and m_2 are not neighbor, $\alpha_{m_1 m_2} = 0$.
2. If segments m_1 and m_2 are neighbor, number of pixels in m_1 (denoted by \mathcal{A}_{m_1}) and in m_2 (denoted by \mathcal{A}_{m_2}) are calculated. We define $\alpha_{m_1 m_2} = \frac{\mathcal{A}_{m_1}}{\mathcal{A}_{m_2}}$.

If $\alpha_{m_1 m_2} \geq 50$, then segment m_2 is merged with segment m_1 . After this refinement process, we obtain the final segment map c_n^{final} ($n = 1, \dots, N$).

4 Result

4.1 Dataset

For experimental evaluation, we use the popular Berkeley Segmentation Dataset and Benchmark (BSDS500). We test on the 200 test images from the BSDS500 dataset [1, 14]. Since our method is unsupervised, we do not use the training images from this dataset.

4.2 Method

For each image, the proposed model is trained for 500 iterations and subsequently the trained model is used to predict segmentation map as described in Sect. 3.3. The number of iterations is set as equal to the same used in [11].

4.3 Qualitative Result

Sample results from our method is shown in Fig. 2. We observe that the proposed method is able to delineate meaningful object segments. We further observe that the proposed method is able to handle cases where object has significant variation in texture (e.g., in case of giraffe, Fig. 2(c) and (g)). The stairs in complex building scenario (Fig. 2(d) and (h)) are partially detected. This shows the effectiveness of the proposed method based on pre-trained deep feature extraction and learnable convolutional layers. It is evident in Fig. 2 that the proposed method is not prone to over-segmentation. This shows the efficacy of the segmentation refinement process based on segment score.

4.4 Quantitative Result

For quantitative comparison, we follow the procedures used in [11]. We calculate Intersection over Union (IoU) of the detected segments and ground truth segments. The detected segment is considered correctly detected if maximum IoU is greater than a IoU threshold 0.5. By following this procedure, for IoU threshold 0.5, proposed method obtains an average precision score of 0.1640 in comparison to 0.1394 by [11], 0.049 by kMeans clustering, and 0.1161 by [9]. Thus the proposed method outperforms the deep learning based unsupervised method in [11] that does not use pre-trained weights. Proposed method is able to obtain significant gain of average precision score (0.0246 gain from [11]). This further demonstrates that the proposed method is able to obtain satisfactory quantitative result and the usage of pre-trained weights before the trainable layers is indeed useful to obtain better segmentation.

We do not perform a direct quantitative comparison to supervised methods or those unsupervised methods that are dependent on large unlabeled dataset. Since those methods are more data driven, it is expected that those methods will possibly outperform the proposed method and they are beyond the scope of this paper.

4.5 Comments on Timing Complexity

Though we did not investigate timing requirements in detail, the method takes approximately 90 s per image (images of 481×321 pixels [1, 14]) on a computer having GPU NVidia Geforce GTX 1080 Ti, Intel I7 CPU (3.2 GHz), and 32 GB RAM.

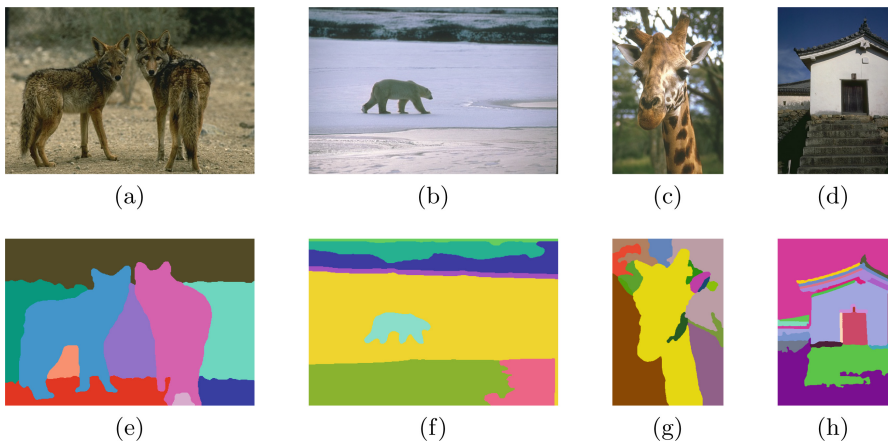


Fig. 2. Illustrations of the obtained segmentation map, input images in top row (a, b, c, d) and respective segmented images in bottom row (e, f, g, h)

5 Conclusion

In this paper, we propose an unsupervised image segmentation method. The proposed method does not require any labeled training pixels or availability of many unlabeled images. The proposed method can work on single image input using intermediate deep feature extracted from a pre-trained network followed by a series of trainable convolutional layers. The weights for the trainable layers are learned on the input image itself while optimizing the segmentation prediction map. Finally the label map is further refined using a simple refinement process to obtain final segment map. The results obtained on the benchmark dataset confirm the effectiveness of the proposed framework. The proposed method can be easily extended for image foreground extraction. The network trained by the proposed method on a particular image may be potentially reused on other images of similar content. By exploiting this assumption, we plan to extend the method for image co-segmentation.

References

1. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(5), 898–916 (2011)
2. Badrinarayanan, V., Kendall, A., Cipolla, R.: SegNet: a deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(12), 2481–2495 (2017)
3. Banerjee, B., Saha, S., Merchant, S.N.: Image foreground extraction—a supervised framework based on region transfer. In: 2016 International Conference on Signal and Information Processing (ICONSIP), pp. 1–5. IEEE (2016)
4. Besag, J.: Spatial interaction and the statistical analysis of lattice systems. *J. Roy. Stat. Soc.: Ser. B (Methodol.)* **36**, 192–236 (1974)
5. Carreira, J., Caseiro, R., Batista, J., Sminchisescu, C.: Semantic segmentation with second-order pooling. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) *ECCV 2012*. LNCS, vol. 7578, pp. 430–443. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33786-4_32
6. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(4), 834–848 (2018)
7. Cooper, M.C.: The tractability of segmentation and scene analysis. *Int. J. Comput. Vis.* **30**(1), 27–42 (1998)
8. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC 2012) Results. <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>
9. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. *Int. J. Comput. Vis.* **59**(2), 167–181 (2004)
10. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 580–587. IEEE (2014)

11. Kanezaki, A.: Unsupervised image segmentation by backpropagation. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1543–1547. IEEE (2018)
12. Li, S.Z.: Markov Random Field Modeling in Image Analysis. Springer, London (2009). <https://doi.org/10.1007/978-1-84800-279-1>
13. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
14. Martin, D., Fowlkes, C., Tal, D., Malik, J., et al.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. ICCV, Vancouver (2001)
15. Pal, N.R., Pal, S.K.: A review on image segmentation techniques. *Pattern Recogn.* **26**(9), 1277–1294 (1993)
16. Park, K., Lee, J., Moon, Y.: Unsupervised foreground segmentation using background elimination and graph cut techniques. *Electron. Lett.* **45**(20), 1025–1027 (2009)
17. Paszke, A., Chaurasia, A., Kim, S., Culurciello, E.: ENet: a deep neural network architecture for real-time semantic segmentation. arXiv preprint [arXiv:1606.02147](https://arxiv.org/abs/1606.02147) (2016)
18. Paszke, A., et al.: Automatic differentiation in PyTorch (2017)
19. Ribbens, A., Hermans, J., Maes, F., Vandermeulen, D., Suetens, P.: Unsupervised segmentation, clustering, and groupwise registration of heterogeneous populations of brain MR images. *IEEE Trans. Med. Imaging* **33**(2), 201–224 (2014)
20. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
21. Roy, S., Sangineto, E., Sebe, N., Demir, B.: Semantic-fusion GANs for semi-supervised satellite image classification. In: 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 684–688. IEEE (2018)
22. Saha, S., Bovolo, F., Bruzzone, L.: Unsupervised deep change vector analysis for multiple-change detection in VHR images. *IEEE Trans. Geosci. Remote Sens.* **57**, 3677–3693 (2019)
23. Sharif Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: an astounding baseline for recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 806–813 (2014)
24. Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., Webb, R.: Learning from simulated and unsupervised images through adversarial training. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2107–2116 (2017)
25. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014)
26. Tang, Z., Wu, Y.: One image segmentation method based on Otsu and fuzzy theory seeking image segment threshold. In: 2011 International Conference on Electronics, Communications and Control (ICECC), pp. 2170–2173. IEEE (2011)
27. Volpi, M., Tuia, D.: Dense semantic labeling of subdecimeter resolution images with convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* **55**(2), 881–893 (2017)
28. Xia, X., Kulis, B.: W-Net: a deep model for fully unsupervised image segmentation. arXiv preprint [arXiv:1711.08506](https://arxiv.org/abs/1711.08506) (2017)

29. Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis. In: International Conference on Machine Learning, pp. 478–487 (2016)
30. Yang, B., Yan, J., Lei, Z., Li, S.Z.: Convolutional channel features. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 82–90 (2015)