W-Net: A Deep Model for Fully Unsupervised Image Segmentation

Xide Xia
Boston University
xidexia@bu.edu

Brian Kulis
Boston University
bkulis@bu.edu

Abstract

While significant attention has been recently focused on designing supervised deep semantic segmentation algorithms for vision tasks, there are many domains in which sufficient supervised pixel-level labels are difficult to obtain. In this paper, we revisit the problem of purely unsupervised image segmentation and propose a novel deep architecture for this problem. We borrow recent ideas from supervised semantic segmentation methods, in particular by concatenating two fully convolutional networks together into an autoencoder—one for encoding and one for decoding. The encoding layer produces a k-way pixelwise prediction, and both the reconstruction error of the autoencoder as well as the normalized cut produced by the encoder are jointly minimized during training. When combined with suitable postprocessing involving conditional random field smoothing and hierarchical segmentation, our resulting algorithm achieves impressive results on the benchmark Berkeley Segmentation Data Set, outperforming a number of competing methods.

1. Introduction

The image segmentation problem is a core vision problem with a longstanding history of research. Historically, this problem has been studied in the unsupervised setting as a clustering problem: given an image, produce a pixelwise prediction that segments the image into coherent clusters corresponding to objects in the image. In classical computer vision, there are a number of well-known techniques for this problem, including normalized cuts [9, 29], Markov random field-based methods [31] , mean shift [8], hierarchical methods [2], and many others.

Given the recent success of deep learning within the computer vision field, there has been a resurgence in interest in the image segmentation problem. The vast majority of recent work in this area has been focused on the problem of *semantic segmentation* [3, 5, 25, 27, 20, 32], a supervised variant of the image segmentation problem. Typically, these methods are trained using models such as fully convolutional networks to produce a pixelwise prediction, and supervised training methods can then be employed to learn filters to produce segments on novel images. One such popular recent approach is the U-Net architecture [27], a fully convolutional network that has been used to achieve impressive results in the biomedical image domain. Unfortunately, existing semantic segmentation methods require a significant amount of pixelwise labeled training data, which can be difficult to collect on novel domains.

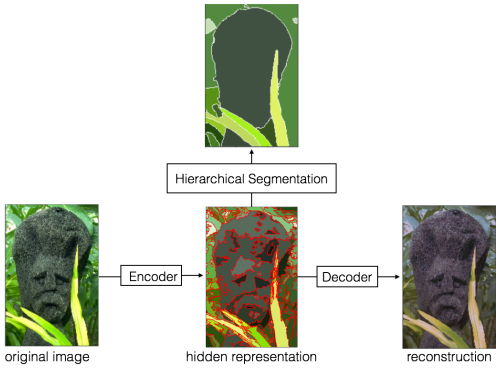

Figure 1. **Overview of our approach.** A fully convolutional network encoder produces a segmentation. This segmentation is fed into a fully convolutional network decoder to produce a reconstruction, and training jointly minimizes the normalized cut of the encoded segmentation and the reconstruction of the image. The encoded image is then post-processed to produce the final segmentation.

Given the importance of the segmentation problem in many domains, and due to the lack of supervised data for many problems, we revisit the problem of unsupervised image segmentation, utilizing recent ideas from semantic segmentation. In particular, we design a new architecture which we call W-Net, and which ties two fully convolutional network (FCN) architectures (each similar to the U-Net architecture) together into a single autoencoder. The first FCN encodes an input image, using fully convolutional layers, into a k-way soft segmentation. The second FCN reverses this process, going from the segmentation layer back to a reconstructed image. We jointly minimize both the reconstruction error of the autoencoder as well as a "soft" nor-

malized cut loss function on the encoding layer. In order to achieve state-of-the-art results, we further appropriately postprocess this initial segmentation in two steps: we first apply a fully conneceted conditional random field (CRF) [20, 6] smoothing on the outputted segments, and we second apply the hierarchical merging method of [2] to obtain a final segmentation, showed as Figure 1.

We test our method on the Berkeley Segmentation Data set benchmark. We follow standard benchmarking practices and compute the segmentation covering (SC), probabilistic Rand index (PRI), and variation of information (VI) metrics of our segmentations as well as segmentation of several existing algorithms. We compare favorably with a number of classical and recent segmentation approaches, and even approach human-level performance in some cases—for example, our algorithm achieves 0.86 PRI versus 0.87 by humans, in the optimal image scale setting. We further show several examples of segments produced by our algorithm as well as some competing methods.

The rest of this paper is organized as follows. We first review related works in Section 2. The architecture of our network is described in Section 3. Section 4 presents the detailed procedure of the post-processing method, and experimental results are demonstrated in Section 5. Section 4 discusses the conclusions that have been drawn.

2. Related Work

We briefly discuss related work on segmentation, convolutional networks, and autoencoders.

2.1. Unsupervised Segmentation

Most approaches to unsupervised image segmentation involve utilizing features such as color, brightness, or texture over local patches, and then make pixel-level clustering based on these features. Among these schemes, the three most widely-used methods include Felzenszwalb and Huttenlocher's graph-based method [14], Shi and Malik's Normalized Cuts [9, 29], and Comaniciu and Meer's Mean Shift [8]. Arbelaez et al. [1, 2] proposed a method based on edge detection that has been shown to outperform the classical methods. More recently, [26] proposed a unified approach for bottom-up multi-scale hierarchical image segmentation. In this paper, we adopt the hierarchical grouping algorithm described in [2] for postprocessing after we get an initial segmentation prediction from W-Net encoder.

2.2. Deep CNNs in Semantic Segmentation

Deep neural networks have emerged as a key component in many visual recognition problems, including supervised learning for semantic image segmentation. [22, 13, 10, 15, 16] all make pixel-wise annotations for segmentation based on supervised classification using deep networks.

Fully convolutional networks (FCNs) [21] have emerged as one of the most effective models for the semantic segmentation problem. In a FCN, fully connected layers of standard convolutional neural networks (CNNs) are transformed as convolution layers with kernels that cover the entire input region. By utilizing fully connected layers, the network can take an input of arbitrary size and produce a correspondingly-sized output map; for example, one can produce a pixelwise prediction for images of arbitrary size. Recently a number of variants of FCN have been proposed and studied and that perform semantic segmentation [24, 3, 5, 25, 27, 20, 32]. In [20], a conditional random field (CRF) is applied to the output map to fine-tune the segmentation. [32] formulates a mean-field approximate inference for the CRF as a Recurrent Neural Network (CRF-RNN), and then jointly optimize both the CRF energy as well as the supervised loss. [27] presents a U-shaped architecture consisting of a contracting path to capture context and a symmetric expanding path that enables precise localization. In this paper, we modify and extend the architecture described in [27] to a W-shaped network such that it reconstructs the original input images and also predicts a segmentation map without any labeling information.

2.3. Encoder-decoders

Encoder-decoders are one of the most widely known and used methods in unsupervised feature learning [17, 18]. The encoder **Enc** maps the input (*e.g.* an image patch) to a compact feature representation, and then the decoder **Dec** reproduces the input from its lower-dimensional representation. In this paper, we design an encoder such that the input is mapped to a dense pixelwise segmentation layer with same spatial size rather than a low-dimensional space. The decoder then performs a reconstruction from the dense prediction layer.

3. Network Architecture

The network architecture is illustrated in Figure 2. It is divided into an U_{Enc} (left side) and a corresponding U_{Dec} (right side); in particular, we modify and extend the typical U-shaped architecture of a U-Net network described in [27] to a W-shaped architecture such that it reconstructs original input images as well as predicts the segmentation maps without any labeling information. The W-Net architecture has 46 convolutional layers which are structured into 18 modules marked with the red rectangles. Each module consists of two 3×3 convolutional layers, each followed by a ReLU [23] non-linearity and batch normalization [19]. The first nine modules form the dense prediction base of the network and the second 9 correspond to the reconstruction decoder.

The U_{Enc} consists of a contracting path (the first half) to capture context and a corresponding expansive path (the

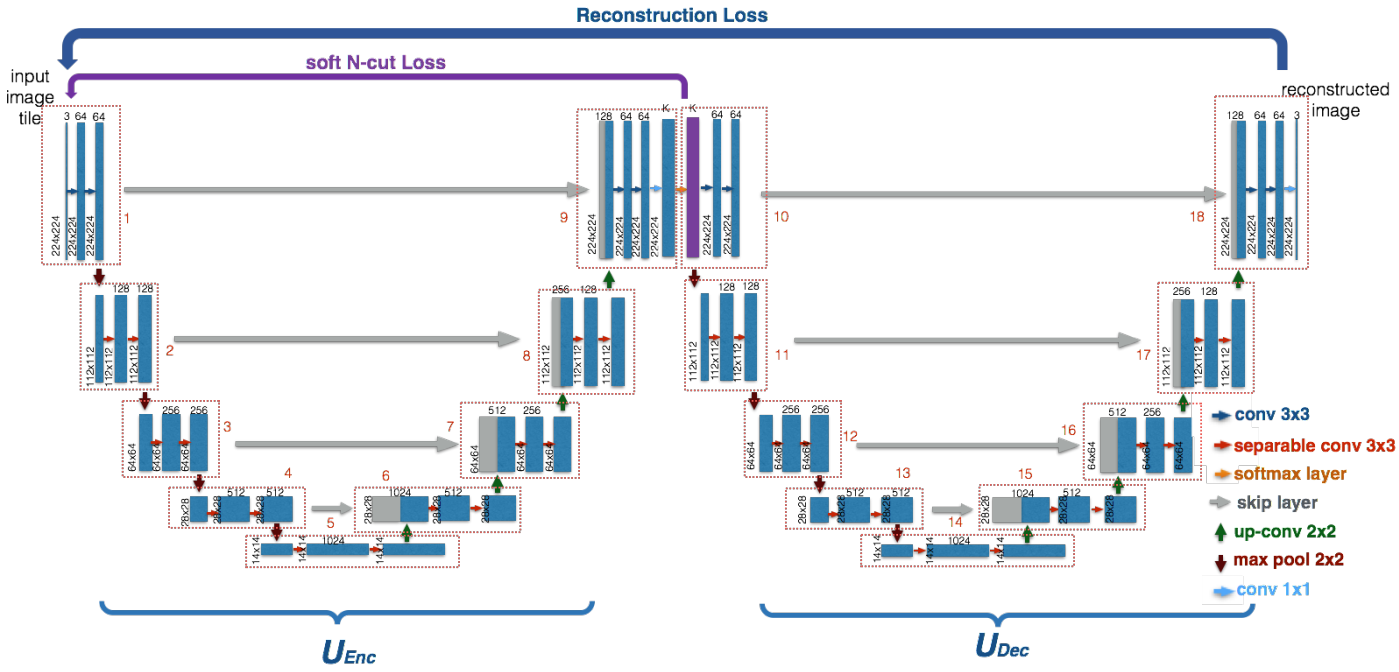

Figure 2. **W-Net architecture.** The W-Net architecture is consist of an U_{Enc} (left side) and a corresponding U_{Dec} (right side). It has 46 convolutional layers which are structured into 18 modules marked with the red rectangles. Each module consists of two 3×3 convolutional layers. The first nine modules form the dense prediction base of the network and the second 9 correspond to the reconstruction decoder.

second half) that enables precise localization, as in the original U-Net architecture. The contracting path starts with an initial module which performs convolution on input images. In the figure, the output sizes are reported for an example input image resolution of 224×224. Modules are connected via 2×2 max-pooling layers, and we double the number of feature channels at each downsampling step. In the expansive path, modules are connected via transposed 2D convolution layers. We halve the number of feature channels at each upsampling step. As in the U-Net model, the input of each module in the contracting path is also bypassed to the output of its corresponding module in the expansive path to recover lost spatial information due to downsampling. The final convolutional layer of the U_{Enc} is a 1×1 convolution followed by a softmax layer. The 1x1 convolution maps each 64-component feature vector to the desired number of classes K, and then the softmax layer rescales them so that the elements of the K-dimensional output lie in the range (0,1) and sum to 1. The architecture of the U_{Dec} is similar to the U_{Enc} except it reads the output of the U_{Enc} which has the size of $224 \times 224 \times K$. The final convolutional layer of the U_{Dec} is a 1×1 convolution to map 64-component feature vector back to a reconstruction of original input.

One important modification in our architecture is that all of the modules use the depthwise separable convolution layers introduced in [7] except modules 1, 9, 10, and 18. A depthwise separable convolution operation consists of a depthwise convolution and a pointwise convolution. The idea behind such an operation is to examine spatial correlations and cross-channel correlations independently—a depthwise convolution performs spatial convolutions independently over each channel and then a pointwise convolution projects the feature channels by the depthwise convolution onto a new channel space. As a consequence, the network gains performance more efficiently with the same number of parameters. In Figure 2, blue arrows represent convolution layers and red arrows indicate depth-wise separable convolutions. The network does not include any fully connected layers which allow it to learn arbitrarily large images and make a segmentation prediction of the corresponding size.

3.1. Soft Normalized Cut Loss

The output of the U_{Enc} is a normalized $224 \times 224 \times K$ dense prediction. By taking the argmax, we can obtain a K-class prediction for each pixel. In this paper, we compute

the normalized cut ($Ncut$) [29] as a global criterion for the segmentation:

$$\begin{aligned} Ncut_K(V) &= \sum_{k=1}^{K} \frac{cut(A_k, V - A_k)}{assoc(A_k, V)} \\ &= \sum_{k=1}^{K} \frac{\sum_{u \in A_k, v \in V - A_k} w(u,v)}{\sum_{u \in A_k, t \in V} w(u,t)}, \end{aligned} \tag{1}$$

where A_k is set of pixels in segment k, V is the set of all pixels, and w measures the weight between two pixels.

However, since the argmax function is non-differentiable, it is impossible to calculate the corresponding gradient during backpropagation. Instead, we define a *soft* version of the $Ncut$ loss which is differentiable so that we can update gradients during backpropagation:

$$\begin{aligned} J_{soft-Ncut}(V, K) &= \sum_{k=1}^{K} \frac{cut(A_k, V - A_k)}{assoc(A_k, V)} \\ &= K - \sum_{k=1}^{K} \frac{assoc(A_k, A_k)}{assoc(A_k, V)} \\ &= K - \sum_{k=1}^{K} \frac{\sum_{u \in V, v \in V} w(u,v) p(u = A_k) p(v = A_k)}{\sum_{u \in A_k, t \in V} w(u,t) p(u = A_k)} \\ &= K - \sum_{k=1}^{K} \frac{\sum_{u \in V} p(u = A_k) \sum_{u \in V} w(u,v) p(v = A_k)}{\sum_{u \in V} p(u = A_k) \sum_{t \in V} w(u,t)}, \end{aligned} \tag{2}$$

where $p(u = A_k)$ measures the probability of node u belonging to class A_k, and which is directly computed by the encoder. By training U_{Enc} to minimize the $J_{soft-Ncut}$ loss we can simultaneously minimize the total normalized disassociation between the groups and maximize the total normalized association within the groups.

3.2. Reconstruction Loss

As in the classical encoder-decoder architecture, we also train the W-Net to minimize the reconstruction loss to enforce that the encoded representations contain as much information of the original inputs as possible. In this paper, by minimizing the reconstruction loss, we can make the segmentation prediction align better with the input images. The reconstruction loss is given by

$$J_{reconstr} = \|\mathbf{X} - \mathbf{U_{Dec}}(\mathbf{U_{Enc}}(\mathbf{X}; W_{Enc}); W_{Dec})\|_2^2, \tag{3}$$

where W_{Enc} denotes the parameters of the encoder, W_{Dec} denotes the parameters of the decoder, and $\mathbf{X}$ is the input image. We train W-Net to minimize the $J_{reconstr}$ between the reconstructed images and original inputs. We simultaneously train U_{Enc} to minimize $J_{soft-Ncut}$ in order to maximize the association within segments and minimize the disassociation between the segments in the encoding layer. The procedure is formally presented in Algorithm 1. By iteratively applying $J_{reconstr}$ and $J_{soft-Ncut}$, the network balances the trade-off between the accuracy of reconstruction and the consistency in the encoded representation layer.

Algorithm 1 Minibatch stochastic gradient descent training of W-Net.

procedure W-NET($\mathbf{X}$;U_{Enc}, U_{Dec})
  for number of training iterations **do**
    Sample a minibatch of new input images x
    Update U_{Enc} by minimizing $J_{soft-Ncut}$
      ▷ Only update U_{Enc}
    Update whole W-Net by minimizing $J_{reconstr}$
      ▷ Update both U_{Enc} and U_{Dec}
  return U_{Enc}

4. Postprocessing

After obtaining an initial segmentation from the encoder, we perform two postprocessing steps in order to obtain our final result. Below we describe these steps, namely CRF smoothing and hierarchical merging.

4.1. Fully-Connected Conditional Random Fields for Accurate Edge Recovery

While deep CNNs with max-pooling layers have proven their success in capturing high-level feature information of inputs, the increased invariance and large receptive fields can cause reduction of localization accuracy. A lack of smoothness constraints can result in the problem of poor object delineation, especially in pixel-level labeling tasks.

To address this issue, while the soft normalized cut loss and skip layers in the W-Net can help to improve the localization of object boundaries, we find that it improves segmentations with fine-grained boundaries by combining the responses at the final U_{Enc} layer with a fully connected Conditional Random Field (CRF) model [6]. The fully connected CRF model employs the energy function

$$E(\mathbf{X}) = \sum_u \Phi(u) + \sum_{u,v} \Psi(u,v) \tag{4}$$

where u, v are pixels on input data $\mathbf{X}$. The unary potential $\Phi(u) = -\log p(u)$, where $p(u)$ is the label annotation probability computed by the softmax layer in U_{Enc}. The pairwise potential $\Psi(u,v)$ measures the weighted penalties when two pixels are assigned different labels by using two Gaussian kernels in different feature spaces.

Figure 3 presents an example of the prediction before and after the fully connected CRF model. The output of the softmax layer in the fully convolutional encoder U_{Enc}

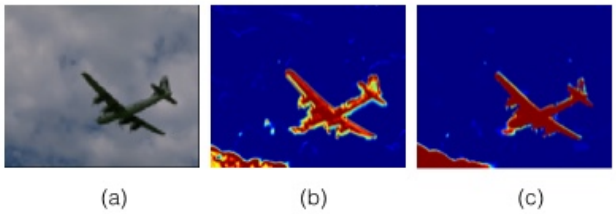

Figure 3. **Belief map (output of softmax function) before and after a fully connected CRF model.** (a) Original image, (b) the responses at the final U_{Enc} layer, (c) the output of the fully connected CRF.

predicts the rough position of objects in inputs with coarse boundaries. After the fully connected CRF model, the boundaries are sharper and small spurious regions have been smoothed out or removed.

4.2. Hierarchical Segmentation

After taking the argmax on the output of the fully connected CRF, we still typically obtain an over-segmented partition of the input image. Our final step is to merge segments appropriately to form the final image segments. Figure 4 shows examples of such initial regions with boundaries in red lines for original input images in (a). In this section, we discuss an efficient hierarchical segmentation that first converts the over-segmented partitions into weighted boundary maps and then merges the most similar regions iteratively.

We measure the "importance" of each pixel on the initial over-segmented partition boundaries by computing a weighted combination of multi-scale local cues and global boundary measurements based on spectral clustering [1, 2]:

$$gPb(x, y, \theta) = \sum_s \sum_i \beta_{i,s} G_{i,\sigma(s)}(x, y, \theta) + \gamma sPb(x, y, \theta), \tag{5}$$

where s indexes scales, i indexes feature channels including brightness, color, and texture, and $G_{i,\sigma(s)}(x, y, \theta)$ measures the dissimilarity between two halves of a disc of radius $\sigma(s)$ center at (x, y) in channel i at angle θ. The mPb signal measures all the edges in the image and the sPb signal captures the most salient curves in the image. Figure 4 (c) shows the corresponding weighted boundary maps of the initial boundaries produced by WNet-CRF in Figure 4 (b).

We then build hierarchical segmentation from this weighted boundaries by using the countour2ucm stage described in [1, 2]. This algorithm constructs a hierarchy of segments from contour detections. It has two steps: an Oriented Watershed Transform (OWT) to build an initial over-segmented region and an Ultrametric Contour Map (UCM), which is a greedy graph-based region merging algorithm.

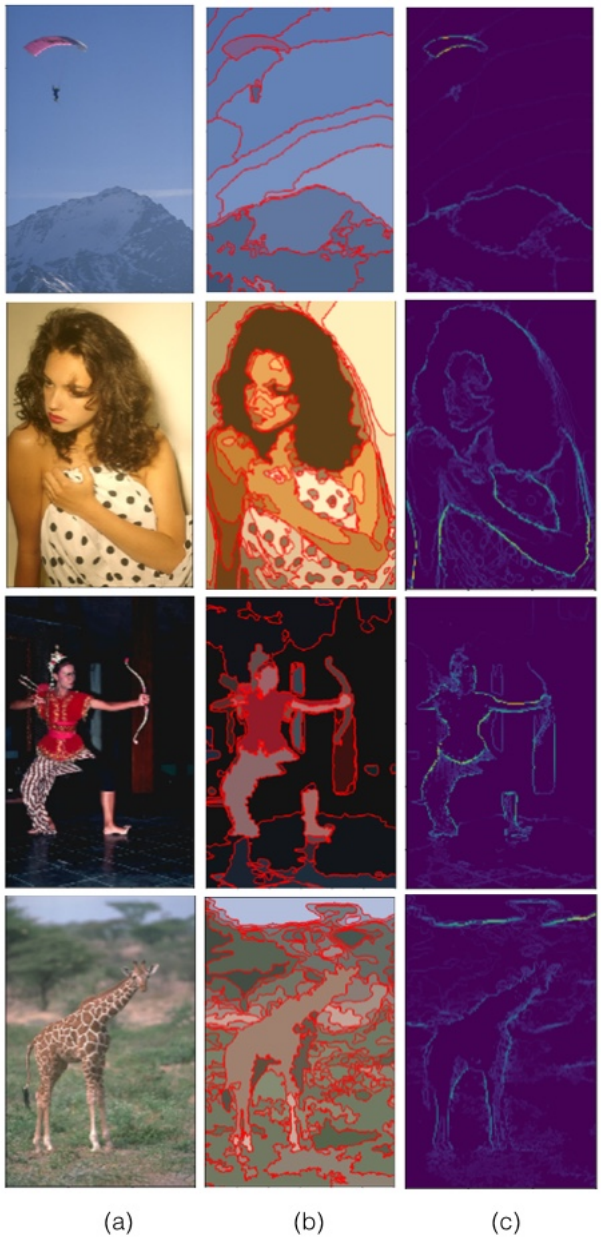

Figure 4. **The initial partitions for the hierarchical merging and the corresponding weighted boundary maps.** (a) Original inputs. (b) The output (argmax) of the fully connected CRF with boundaries showed in red lines. (c) The corresponding weighted boundary maps of the red lines in (b).

5. Experiments

We train our proposed W-Net on the PASCAL VOC2012 dataset [12] and then evaluate the trained network using the Berkeley Segmentation Database (BSDS300 and BSDS500). The PASCAL VOC2012 dataset is a large visual object classes challenge which contains 11,530 images and 6,929 segmentations. BSDS300 and BSDS500

Algorithm 2 Post processing

procedure POSTPROCESSING(x;U_{Enc}, CRF, Pb)
  $x' = U_{Enc}(x)$
    ▷ Get the hidden representation of x
  $x'' = \text{CRF}(x')$
    ▷ fine-grained boundaries with a fully CRF
  $x''' = Pb(x'')$
  ▷ compute the probability of boundary only on the edge detected in x''
  $S = countour2ucm(x''')$
    ▷ hierarchical segmentation
  return S

have 300 and 500 images, respectively. For each image, the BSDS dataset provides human-annotated segmentation as ground truth. Since our proposed method is designed for unsupervised image segmentation, we do not use any ground truth labels in the training phase; we use the ground truth only to evaluate quality of our segmentations.

We resize the input images to 224 $\times$ 224 during training, and the architecture of the trained network is shown in Figure 2. We train the networks from scratch using mini-batches of 10 images, with an initial learning rate of 0.003. The learning rate is divided by ten after every 1,000 iterations. The training is stopped after 50,000 iterations. Dropout of 0.65 was added to prevent overfitting during training. We construct the weight matrix W for $J_{soft-Ncut}$ as:

$$w_{ij} = e^{\frac{-\|F(i)-F(j)\|_2^2}{\sigma_I^2}} * \begin{cases} e^{\frac{-\|X(i)-X(j)\|_2^2}{\sigma_X^2}} & \text{if } \|X(i)-X(j)\|_2 < r \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where $X(i)$ and $F(i)$ are the spatial location and pixel value of node i, respectively. $\sigma_I = 10$, $\sigma_X = 4$, and $r = 5$.

The plots of $J_{reconstr}$ and $J_{soft-Ncut}$ losses during training are shown in Figure 5. We examine the $J_{reconstr}$ loss with and without considering $J_{soft-Ncut}$ during training. From Figure 5, we can see that $J_{reconstr}$ converges faster when the $J_{soft-Ncut}$ is not considered. When we add the $J_{soft-Ncut}$ during training, the $J_{reconstr}$ decreases slowly and less stably. At convergence of $J_{reconstr}$, the blue line (training with $J_{soft-Ncut}$) is still higher than the red one (training without); this is because the hidden representation space is forced to be more consistent with a good segmentation of the image when the $J_{soft-Ncut}$ loss is introduced, so its ability to reconstruct the original images is weakened. Finally, both $J_{reconstr}$ and $J_{soft-Ncut}$ converge which means our approach balances trade-offs between minimizing the reconstruction loss in the last layer and maximizing the total association within the groups in the hidden layer.

Figure 6 illustrates the comparison with and without considering $J_{soft-Ncut}$ loss during back-propagation. In order to make a better visualization of the output on the softmax

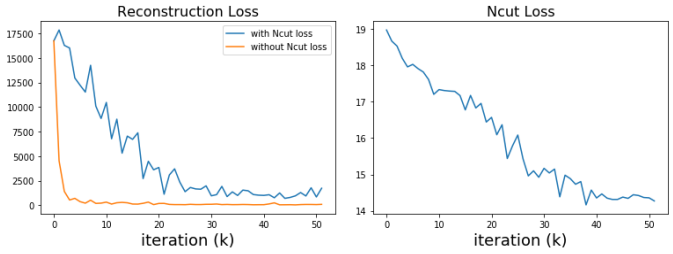

Figure 5. **$J_{reconstr}$ and $J_{soft-Ncut}$ losses in the training phase. Left:** Reconstruction losses during training (red: training without $J_{soft-Ncut}$, blue: training with $J_{soft-Ncut}$). **Right:** Soft-Ncuts loss during training.

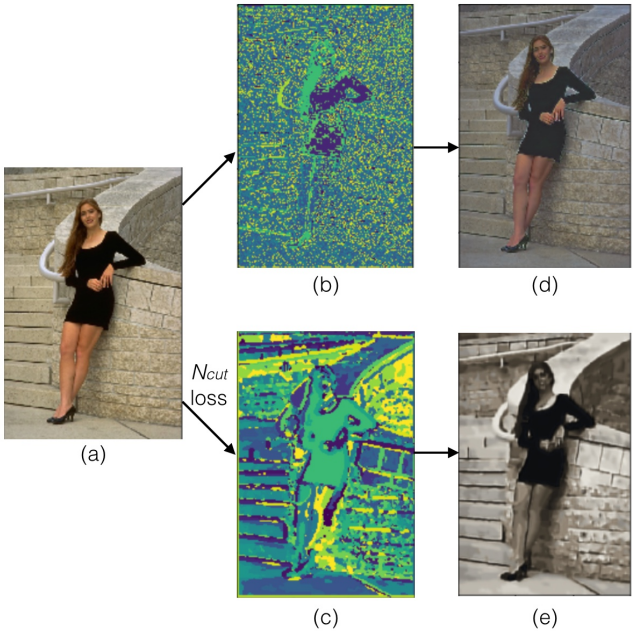

Figure 6. **A comparison with and without considering $J_{soft-Ncut}$ loss during back-propagation.** (a) Original image. (b) Visualization of the output of the final softmax layer in U_{Enc} without adding $J_{soft-Ncut}$ loss during back-propagation. (c) Visualization of the output in the final softmax layer in U_{Enc} when adding $J_{soft-Ncut}$ loss during back-propagation. (d) The corresponding reconstructed image of (b). (e) The corresponding reconstructed image of (c).

layer in U_{Enc}, we take the argmax function on the prediction and use different colors to visualize different pixel-wise labels. We can see that the pixel-wise prediction is smoothed when we consider the $J_{soft-Ncut}$ during back-propagation. When we remove the $J_{soft-Ncut}$ loss from the W-Net, the model become a regular fully convolutional encoder-decoder which makes a high-quality reconstruction; however, the output of softmax layer is more noisy and discrete. On the other hand, by adding the $J_{soft-Ncut}$

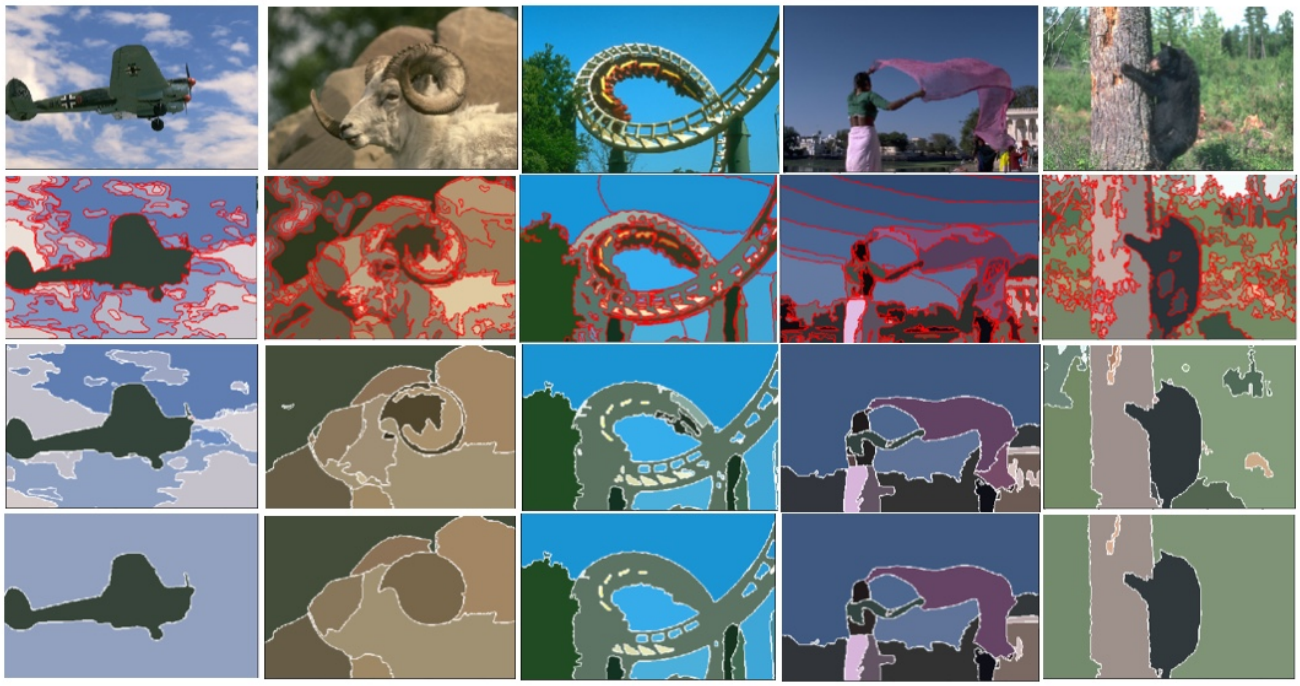

Figure 7. **Results of hierarchical segmentation using the output of WNet with CRF smoothing as initial boundaries, on the BSDS500.** From top to bottom: Original image, the initial over-segmented partitions showed in red lines obtained by the fully connected CRF, segmentations obtained by thresholding at the optimal dataset scale (ODS) and optimal image scale (OIS).

Figure 8. **Results of hierarchical segmentation using the combination of the output of WNet with CRF smoothing and UCM as the initial boundaries, on the BSDS500.** From top to bottom: Original image, the initial over-segmented partitions showed in red lines, segmentations obtained by thresholding at the optimal dataset scale (ODS) and optimal image scale (OIS).

Method	SC		PRI		VI	
	ODS	OIS	ODS	OIS	ODS	OIS
Quad Tree	0.33	0.39	0.71	0.75	2.34	2.22
Chan Vese [4]	0.49	-	0.75	-	2.54	-
NCuts [9]	0.44	0.53	0.75	0.79	2.18	1.84
SWA [28]	0.47	0.55	0.75	0.80	2.06	1.75
Canny-owt-ucm [2]	0.48	0.56	0.77	0.82	2.11	1.81
Felz-Hutt [14]	0.51	0.58	0.77	0.82	2.15	1.79
Mean Shift [8]	0.54	0.58	0.78	0.80	1.83	1.63
Taylor [30]	0.56	0.62	0.79	0.84	1.74	1.63
W-Net (ours)	**0.58**	**0.62**	**0.81**	**0.84**	**1.71**	**1.53**
gPb-owt-ucm [2]	0.59	0.65	0.81	0.85	1.65	1.47
W-Net+ucm (ours)	**0.60**	**0.65**	**0.82**	**0.86**	**1.63**	**1.45**
Human	0.73	0.73	0.87	0.87	1.16	1.16

Table 1. **Results on BSDS300**. The values are reproduced from the tables in [30].

Method	SC		PRI		VI	
	ODS	OIS	ODS	OIS	ODS	OIS
NCuts [9]	0.45	0.53	0.78	0.80	2.23	1.89
Canny-owt-ucm [2]	0.49	0.55	0.79	0.83	2.19	1.89
Felz-Hutt [14]	0.52	0.57	0.80	0.82	2.21	1.87
Mean Shift [8]	0.54	0.58	0.79	0.81	1.85	1.64
Taylor [30]	0.56	0.62	0.81	0.85	1.78	1.56
W-Net (ours)	**0.57**	**0.62**	**0.81**	**0.84**	**1.76**	**1.60**
gPb-owt-ucm [2]	0.59	0.65	0.83	0.86	1.69	1.48
DC-Seg-full [11]	0.59	0.64	0.82	0.85	1.68	1.54
W-Net+ucm (ours)	**0.59**	**0.64**	**0.82**	**0.85**	**1.67**	**1.47**
Human	0.72	0.72	0.88	0.88	1.17	1.17

Table 2. **Results on BSDS500**. The values are reproduced from the tables in [30] and [11].

loss, we get a more consistent hidden representation shown in (d), although the reconstruction is not as good as the one in a classical encoder-decoder architecture. From this comparison, we can see the trade-off between the consistency in the hidden representation and the quality of reconstruction, and it justifies our use of a soft normalized cut loss during training.

5.1. Segmentation Benchmarks

To compare the performance of W-Net with existing unsupervised image segmentation methods, we compare with the following: DC-Seg-full [11], gPb-owt-ucm [2], Taylor [30], Felzenszwalb and Huttenlocher (Felz-Hutt) [14], Mean Shift [8], Canny-owt-ucm [2], SWA [28], Chan Vese [4], Multiscale Normalized Cuts (NCuts) [9], and Quad-Tree. As has become standard, we evaluate the performance on three different metrics: Variation of Information (VI), Probabilistic Rand Index (PRI), and Segmentation Covering (SC). For SC and PRI, higher scores are better; for VI, a lower score is better. We also report human performance on this data set. For a set of hierarchical segmentations S_i corresponding to different scales, we report the result at Optimal Dataset Scale (ODS) and Optimal Image Scale (OIS).

Table 1 and Table 2 summarize the performance of the proposed method noted as **W-Net** on BSDS300 and BSDS500 respectively. Since the U_{Enc} encoder followed by a fully connected CRF provides an initial boundaries detection, we compute the multi-scale local cues only on the detected edges instead of the whole input image. As can be seen, our proposed approach has competitive performance compared to the high computation demanding gPb-owt-ucm method. We further consider combining the boundaries produced by our W-Net model after CRF smoothing with the ultrametric contour map produced by the gPb-owt-ucm method before applying the final postprocessing step; we denote this variant as **W-Net+ucm**. We can see that with this variant, our results outperform the other methods.

Figure 7 illustrates results of running the proposed method **W-Net** on images from the BSDS500. The first row shows the original inputs; the second row shows that results of initial boundaries detection produced by the U_{Enc} encoder followed by a fully connected CRF. The third and the fourth rows show the ultrametric contour maps produced by the contours2ucm stage at the ODS and OIS respectively. Figure 8 illustrates results of running the **W-Net+ucm** on images from the BSDS500.

6. Conclusion

In this paper we introduced a deep learning-based approach for fully unsupervised image segmentation. Our proposed algorithm is based on concatenating together two fully convolutional networks into an encoder-decoder framework, where each of the FCNs are variants of the U-Net architecture. Training is performed by iteratively minimizing the reconstruction error of the decoder along with a soft normalized cut of the encoder layer. As the resulting segmentations are typically coarse and over-segmented, we apply CRF smoothing and hierarchical merging to produce the final outputted segments. On the Berkeley Segmentation Data Set, we outperform a number of existing classical and recent techniques, achieving performance near human level by some metrics.

We believe our method will be useful in cases where it is difficult to obtain labeled pixelwise supervision, for instance in domains such as biomedical image analysis where new data sets may require significant re-labeling for semantic segmentation methods to work well. Further, our approach may be further refined in the future by utilizing different loss functions or postprocessing steps. Ideally, we would like to design an architecture where additional postprocessing is not necessary. Finally, designing variants of our architecture when a small amount of supervision is available would also be useful in many domains.

7. Appendix

We show additional results of running the proposed method **W-Net** on images from the BSDS500 in Figure [9] and Figure [10]. Further, Figure [11] and Figure [12] illustrates more results of running the **W-Net+ucm** on images from the BSDS500.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2294–2301. IEEE, 2009.

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2011.

[3] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.

[4] L. Bertelli, B. Sumengen, B. Manjunath, and F. Gibou. A variational framework for multiregion pairwise-similarity-based image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(8):1400–1414, 2008.

[5] A. Chaurasia and E. Culurciello. Linknet: Exploiting encoder representations for efficient semantic segmentation. *arXiv preprint arXiv:1707.03718*, 2017.

[6] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016.

[7] F. Chollet. Xception: Deep learning with depthwise separable convolutions. *arXiv preprint arXiv:1610.02357*, 2016.

[8] D. Comaneci and P. M. M. Shift. A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5), 2002.

[9] T. Cour, F. Benezit, and J. Shi. Spectral segmentation with multiscale graph decomposition. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 1124–1131. IEEE, 2005.

[10] J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3992–4000, 2015.

[11] M. Donoser and D. Schmalstieg. Discrete-continuous gradient orientation estimation for faster image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3158–3165, 2014.

[12] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[13] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1915–1929, 2013.

[14] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004.

[15] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision*, pages 297–312. Springer, 2014.

[16] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 447–456, 2015.

[17] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[18] F. J. Huang, Y.-L. Boureau, Y. LeCun, et al. Unsupervised learning of invariant feature hierarchies with applications to object recognition. pages 1–8, 2007.

[19] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[20] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011.

[21] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[22] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich. Feedforward semantic segmentation with zoom-out features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3376–3385, 2015.

[23] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[24] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1520–1528, 2015.

[25] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[26] J. Pont-Tuset, P. Arbelaez, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. *IEEE transactions on pattern analysis and machine intelligence*, 39(1):128–140, 2017.

[27] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[28] E. Sharon, M. Galun, D. Sharon, R. Basri, and A. Brandt. Hierarchy and adaptivity in segmenting visual scenes. *Nature*, 442(7104):810–813, 2006.

Figure 9. **Results of hierarchical segmentation using the output of WNet with CRF smoothing as initial boundaries, on the BSDS500.** From top to bottom: Original image, the initial over-segmented partitions showed in red lines obtained by the fully connected CRF, segmentations obtained by thresholding at the optimal dataset scale (ODS) and optimal image scale (OIS).

[29] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[30] C. J. Taylor. Towards fast and accurate segmentation. In *Pro-*

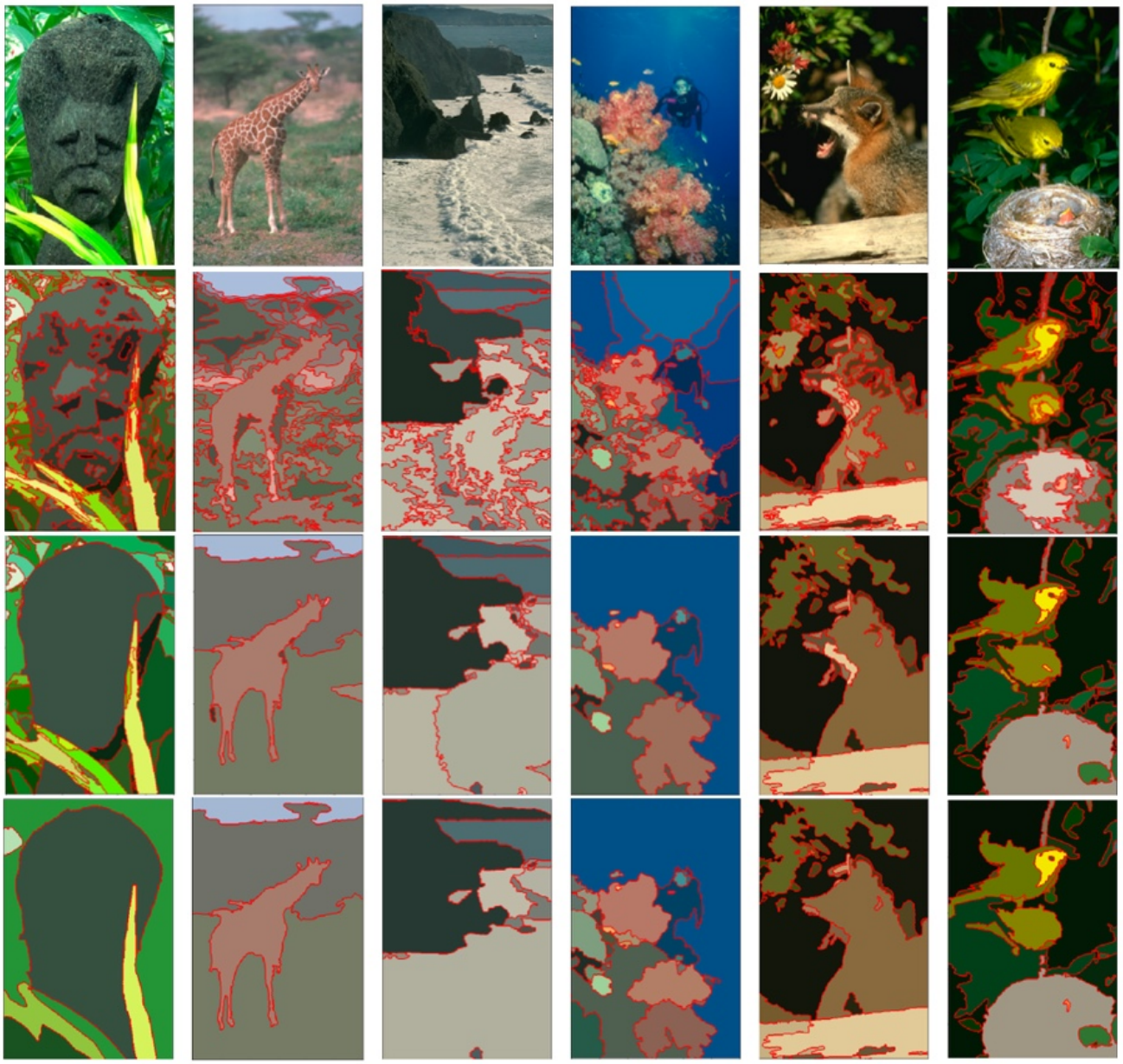

Figure 10. **Results of hierarchical segmentation using the output of WNet with CRF smoothing as initial boundaries, on the BSDS500.** From top to bottom: Original image, the initial over-segmented partitions showed in red lines obtained by the fully connected CRF, segmentations obtained by thresholding at the optimal dataset scale (ODS) and optimal image scale (OIS).

ceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1916–1922, 2013.

[31] Y. Zhang, M. Brady, and S. Smith. Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE transactions on medical imaging*, 20(1):45–57, 2001.

[32] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537, 2015.

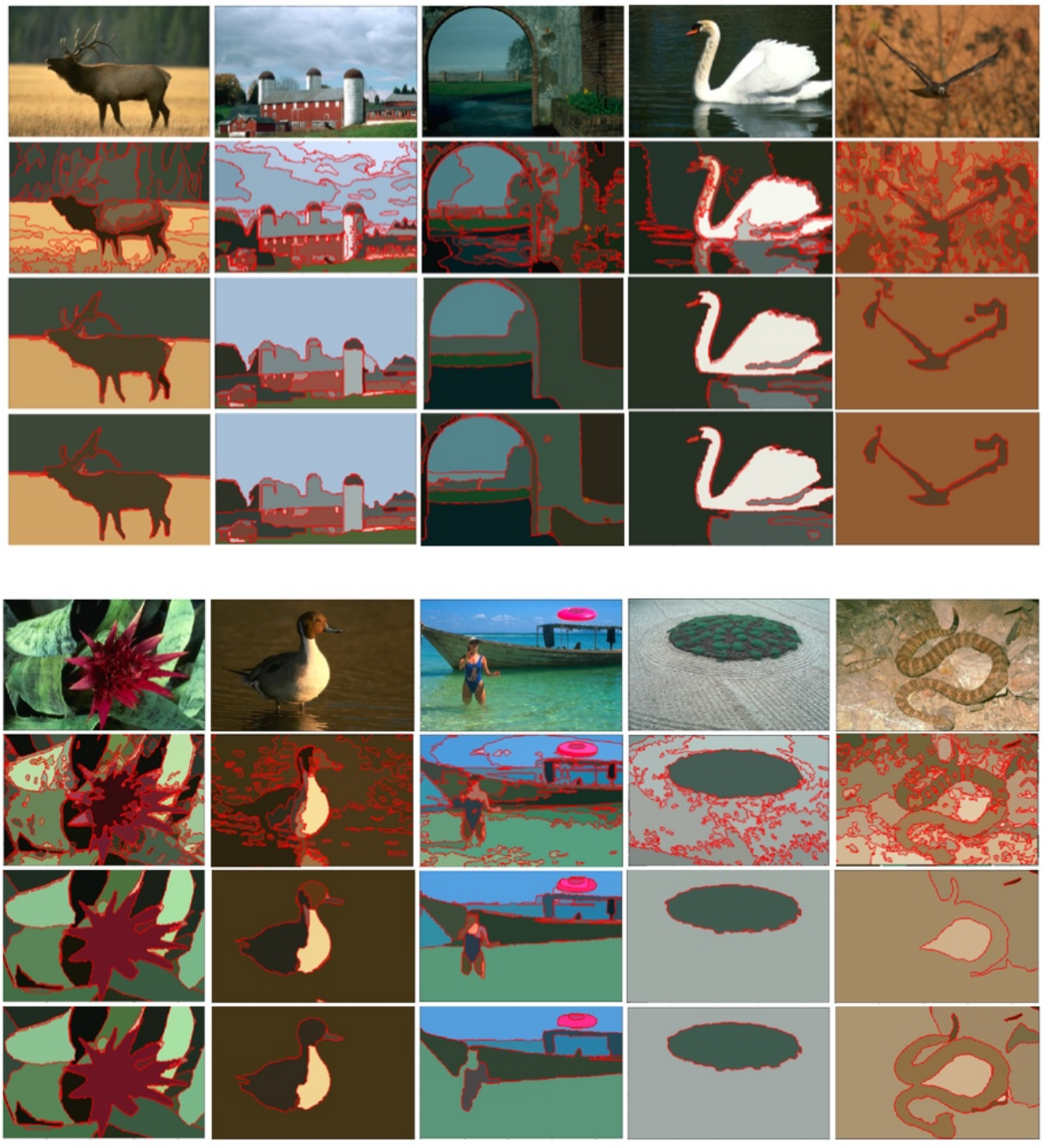

Figure 11. **Results of hierarchical segmentation using the combination of the output of WNet with CRF smoothing and UCM as the initial boundaries, on the BSDS500.** From top to bottom: Original image, the initial over-segmented partitions showed in red lines, segmentations obtained by thresholding at the optimal dataset scale (ODS) and optimal image scale (OIS).

Figure 12. **Results of hierarchical segmentation using the combination of the output of WNet with CRF smoothing and UCM as the initial boundaries, on the BSDS500.** From top to bottom: Original image, the initial over-segmented partitions showed in red lines, segmentations obtained by thresholding at the optimal dataset scale (ODS) and optimal image scale (OIS).