UNSUPERVISED IMAGE SEGMENTATION USING CONVOLUTIONAL AUTOENCODER WITH TOTAL VARIATION REGULARIZATION AS PREPROCESSING

Chunlai Wang, Bin Yang and Yiwen Liao

Institute of Signal Processing and System Theory, University of Stuttgart, Germany

ABSTRACT

Conventional unsupervised image segmentation methods use color and geometric information and apply clustering algorithms over pixels. They preserve object boundaries well but often suffer from over-segmentation due to noise and artifacts in the images. In this paper, we contribute on a preprocessing step for image smoothing, which alleviates the burden of conventional unsupervised image segmentation and enhance their performance. Our approach relies on a convolutional autoencoder (CAE) with the total variation loss (TVL) for unsupervised learning. We show that, after our CAE-TVL preprocessing step, the over-segmentation effect is significantly reduced using the same unsupervised image segmentation methods. We evaluate our approach using the BSDS500 image segmentation benchmark dataset and show the performance enhancement introduced by our approach in terms of both increased segmentation accuracy and reduced computation time. We examine the robustness of the trained CAE and show that it is directly applicable to other natural scene images.

***Index Terms*—** Image segmentation, unsupervised, convolutional autoencoder, total variation loss

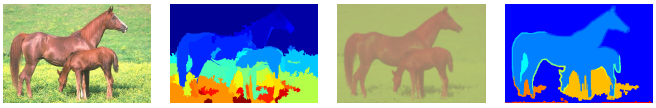
1. INTRODUCTION

Image segmentation is a process to divide a given image into several non-overlapping regions, where each region is expected to be an object or a part of an object. Unsupervised image segmentation is a challenge topic which serves as the first step for many high-level computer vision tasks [1, 2]. Classical approaches use variations of clustering algorithms [3, 4] or growing and merging of regions [5]. They are simple to use and require only basic color and geometrical information of pixels. While the true object boundaries are well preserved, these methods often suffer from over-segmentation due to noise and artifacts in the images.

Several properties are expected in a segmented region. For instance, the inter-region areas should be smooth. One common assumption for this is the piece-wise constancy of the pixel intensity values. To achieve this, the total variation (TV) is one of the most used regularization terms. Among the best known and most influential examples are the Rudin-Osher-Fatemi (ROF) image denoising model [6] and the Mumford-Shah image segmentation model [7]. As all these methods are based on the calculus of variations and partial differential equations (PDEs), different relaxation conditions and their approximate solutions were significantly under research. However, solving the optimization problem still remains to be more efficient.

In recent years, the machine learning techniques experience an explosion of progress, mainly due to the advance of deep neural networks. Among them, the convolutional neural network (ConvNet) is most attractive in vision problems due to its convenience of being stacked into deep networks and its ability of representing image features with high robustness, rotation invariance and many other properties. Recent works [8, 9] illustrate the application of fully convolutional network for semantic segmentation, where a large number of human annotations are required as ground truth for training.

In this paper, we aim at unsupervised image segmentation. While ConvNet has been proved to be powerful in high-level image recognition/classification, our idea is to utilize the efficiency of ConvNet and its ability of vision modeling for low-level image segmentation. We model the objective function by a convolutional autoencoder (CAE) using a reconstruction loss and a total variation loss (TVL). We introduce TVL as a regularization term and thus, enforce pixel-wise smoothness. We show that this objective function can be straightforwardly optimized by gradient descent methods with back-propagation as used in deep learning. After that, the smoothed image undergoes an unsupervised image segmentation. Fig. 1 shows an example of our approach.



(a) (b) (c) (d)

Fig. 1. An example of our approach. The segmentation of an image (a) using a conventional unsupervised image segmentation method [5] is shown in (b). We propose to smooth (a) using our CAE-TVL into (c) before segmentation. The new segmentation result using the same unsupervised image segmentation method is shown in (d).

Our work is related to prior works [10, 11] which use

ConvNet for low-level image processing. In [10], ConvNet was used to restore electron microscopic images. This work demonstrated that complex ConvNet outperforms simple MRF/CRF models while being more efficient during training and inference. However, they required ground truth restorations provided by human experts for supervised training. Jain and Seung [11] proposed to use ConvNet for image denoising. They assumed a noise process on the input image and training the network by minimizing the reconstruction error subject to the noise process. None of these works used the idea in this paper of combining both the reconstruction loss and the TVL for unsupervised training of a CAE.

In the next section, we introduce the proposed approach for unsupervised image segmentation in details. The experimental results are presented in section 3 and a short conclusion is given in section 4.

2. THE APPROACH

The main part of our approach is a convolutional autoencoder called CAE-TVL, which is trained in an unsupervised manner and used for preprocessing of segmentation. Given a test image, the CAE-TVL network outputs a smoothed version of it, where the noise and artifacts are significantly reduced. Thus, the smoothed image is more suitable for segmentation.

2.1. Model description

Traditional autoencoders are fully connected and learn the identity mapping. In the case of image data, the traditional autoencoders ignore the 2D image structure. However, the trend in vision and object recognition is to discover localized features that repeat themselves over the input. We hence replace the fully connected layers by convolutional layers. The convolutional kernels are shared all over the image, reducing the redundancy in the parameters.

Let $\mathbf{W}^{(l)} \in \mathbb{R}^{P_l \times Q_l \times S_l \times T_l}$ be a 4D tensor containing all convolutional kernels of the l-th convolutional layer. The first two dimensions indicate spatial coordinates, the third dimension is the number of channels of the source layer and the fourth the number of channels of the target layer. We define the convolution $\mathbf{W}^{(l)} * \mathbf{X}^{(l-1)}$ of the two tensors $\mathbf{W}^{(l)}$ and $\mathbf{X}^{(l-1)} \in \mathbb{R}^{M \times N \times S_l}$ as

$$(\mathbf{W}^{(l)} * \mathbf{X}^{(l-1)})_{mnt} = \sum_{p=1}^{P_l} \sum_{q=1}^{Q_l} \sum_{s=1}^{S_l} \mathbf{W}^{(l)}_{pqst} \mathbf{X}^{(l-1)}_{m+p-1,n+q-1,s}. \tag{1}$$

This is a 3D convolution with a stride of 1. To keep the spatial size, we pad each channel of $\mathbf{X}^{(l-1)}$ with zeros as needed (padding of $(P_l - 1) \times (Q_l - 1)$) in the calculation. The product $\mathbf{W}^{(l)} * \mathbf{X}^{(l-1)} \in \mathbb{R}^{M \times N \times T_l}$ has T_l channels. Thus, the activation $\mathbf{X}^{(l)}$ of the l-th convolutional layer is

$$\mathbf{X}^{(l)} = \Phi\left(\sigma(\mathbf{W}^{(l)} * \mathbf{X}^{(l-1)} + \mathbf{B}^{(l)})\right). \tag{2}$$

$\mathbf{B}^l \in \mathbb{R}^{1 \times 1 \times T_l}$ denotes the bias terms for T_l output channels. $\Phi(\cdot)$ is an element-wise nonlinear activation function (we use ReLU [12]) and $\sigma(\cdot)$ denotes a normalization (we use batch normalization proposed in [13] which speeds up learning).

The activation $\mathbf{X}^{(l)}$ computed by Eq. (2) can be downsampled by a pooling layer. We use max-pooling with a window size of 2×2 to keep the most significant activation and reduce the spatial resolution. The downsampled activation can be resized to the original image size by unpooling (upsampling). We do this by repeating each value once over both spatial dimensions. We stack a deconvolutional layer after unpooling. The deconvolutional layer corresponds to the convolutional one and does transposed convolution. Let $\mathbf{Z}^{(l)}$ be the output of the l-th deconvolutional layer, we define

$$\mathbf{Z}^{(l)}_{mnt} = \sum_{s=1}^{S_l} \sum_{p=0}^{P_l-1} \sum_{q=0}^{Q_l-1} \mathbf{W}^{(l)}_{p+1,q+1,s,t} \mathbf{X}^{(l-1)}_{m-p+1,n-q+1,s}, \tag{3}$$

where $\mathbf{W}^{(l)}$ and $\mathbf{X}^{(l-1)}$ are now the kernels and the input of the l-th deconvolutional layer, respectively. They are zero-padded as needed in the calculation. $\mathbf{Z}^{(l)}$ is cropped to the original size of $\mathbf{X}^{(l-1)}$.

As neural network has a strong ability of modeling nonlinear function with few hidden layers, we apply a simple CAE in this paper. Our CAE is build up by a convolutional layer (including convolution, normalization and activation), a pooling layer, an unpooling layer and a deconvolutional layer. Since we focus on low-level segmentation, we do not delve into deeper architectures which are useful for extracting high-level representations. We use 16 kernels for the convolutional layer and 3 kernels for the deconvolutional layer to project the activation maps back to 3 channels corresponding to RGB. All kernels have the spatial size of 3×3 .

2.2. Loss function

Let $\mathbf{X} \in \mathbb{R}^{M \times N \times 3}$ denote a given color image. $\mathbf{F}_{\mathbf{\Theta}}$ indicates the nonlinear function of the CAE network with parameters $\mathbf{\Theta}$. Let $\mathbf{Y} = \mathbf{F}_{\mathbf{\Theta}}(\mathbf{X})$ be the output of the network and thus, has the same dimension as $\mathbf{X}$. We define the loss function in two parts

$$J_1 = ||\mathbf{Y} - \mathbf{X}||^2 + \beta||\mathbf{W}||^2 \tag{4}$$

and

$$J_2 = |\nabla \mathbf{Y}|. \tag{5}$$

J_1 consists of the mean square error (Euclidean loss) term and the l_2-norm regularization term. β is the weight decay parameter governing the regularization term. We set it to 0.0005 in this paper. J_2 is the TVL. It penalizes the locations with large gradient of pixel intensity value. While J_1 drives the output $\mathbf{Y}$ to approach the input $\mathbf{X}$, J_2 works in the other way and enforces $\mathbf{Y}$ to be smooth. We combine these two parts and the overall loss function is defined as

$$\begin{aligned} J(\mathbf{X}, \mathbf{F}_{\mathbf{\Theta}}, \lambda, \beta) &= J_1 + \lambda J_2 \\ &= ||\mathbf{Y} - \mathbf{X}||^2 + \beta||\mathbf{W}||^2 + \lambda|\nabla \mathbf{Y}|. \end{aligned} \tag{6}$$

λ is a trade-off parameter controls the weights of the two parts. Larger λ encourages the output image to be smoother. We set $\lambda = 0.5$ in this paper.

Our loss function is related to the energy function used in TV-based image denoising and segmentation, both including the Euclidean loss and TVL. However, the standard TV-based models work on a single image and tries to evolve the input image iteratively into the final result. In comparison, we try to learn a generic smoothing function $\mathbf{F}_{\mathbf{\Theta}}$ which is suitable for arbitrary input color images. The new image $\mathbf{Y}$ is a better representation of $\mathbf{X}$ for segmentation as shown in Fig.1

2.3. Training

Given a set of N training images, our training objective is

$$\hat{\mathbf{\Theta}} = \arg\min_{\mathbf{\Theta}} \sum_{n=1}^{N} J(\mathbf{X}(n); \mathbf{\Theta}), \tag{7}$$

where the parameter vector $\mathbf{\Theta}$ of the network contains all elements of $\mathbf{W}^{(l)}$ and $\mathbf{B}^{(l)}$. Notice that we do not use labeled data and our approach is hence unsupervised.

In practice, we first train the CAE by minimizing the first part J_1 for image reconstruction. Then, we finetune the parameters by minimizing the overall loss J.

We use stochastic gradient descent (SGD) with momentum to train the network. Just as for standard networks, backpropagation algorithm is applied to compute the gradient of the loss function with respect to the parameters. As the first part J_1 of the loss function in Eq. (6) is similar to the general autoencoder, its gradient can be straightforwardly backpropagated like other neural networks. J_2 is implemented by first modeling the Nabla operator ∇ by a convolution of the output $\mathbf{Y}$ with two Sobel edge detectors (both column and row direction) and then taking the l_1-norm distance of the generated gradient maps. Without details (due to limited space), we can show that the standard backpropagation also works for the TVL term J_2.

We use mini-batch for training and one mini-batch contains 4 images. We train the network with more than 20 epochs. The learning rate is set to 10^{-7} in the first reconstruction stage and 10^{-9} in the finetuning stage. The weights in our network are initialized by the Xavier method [14]. They are drawn from a distribution with zero mean and a specific variance $\frac{1}{\sqrt{n_{\text{in}}}}$ where n_{in} is the number of input neurons of the layer. The momentum value is set to 0.9. The training is performed on a single Intel Core CPU @ 2,6 GHz. The implementation is based on the Matlab toolbox MatConvNet [15]. A complete training using 200 images takes about 1 hour.

2.4. Segmentation

After training, our CAE-TVL network is supposed to work for generic images. Given an image, we feed it into our network and get a smoothed version of it. After this preprocessing step, a conventional unsupervised image segmentation method is applied to the smoothed image to get the final segmentation.

3. EXPERIMENTAL RESULTS

Our experiments are based on the Berkeley Segmentation Data Set (BSDS500) [16]. It contains 300 images for training and validation and 200 images for testing. For each image, BSDS500 provides human-annotated segmentations as ground truth by five different subjects on average. However, these ground truth are only used for evaluation in the test phase and not used in the unsupervised training of our CAE-TVL network.

Experiment 1: In the first experiment, we use all 300 images in the training set to train the CAE-TVL network. It is then used to smooth the 200 test images before unsupervised image segmentation.

We use Mean Shift (MS) [3], Quickshift (QS) [17] efficient Graph-based Segmentation (GbS) [5] and SLIC [4] as unsupervised image segmentation methods and compare the results of segmentation over original images and our smoothed images. To our knowledge, these methods are the most widely used unsupervised image segmentation methods that achieve good results. All implementations are available in public. In particular, we use the EDISON implementation [18] of MeanShift, the VLFeat [19] implementation of Quickshift, the SLIC implementation from [20] and the GbS implementation from the original author.

Notice that supervised methods for contour detection [21, 16, 22] and segmentation [23, 16] are not considered. Besides, [24] also presents an unsupervised image segmentation method based on region growing. However, this method relies on too many prior information and requires many computations for the hand-designed features. On the contrary, the methods considered in this paper use only color and spatial information of pixels and are more efficient.

Following [16], we use the performance metrics segmentation covering, rand index (RI) and variational of information (VOI) for evaluation. For each method, we use at least 20 different parameter settings to reduce their influence on the evaluation. Both the segmentation using the optimal parameter setting for the entire dataset (ODS) and for individual images (OIS) are considered. In the case of segmentation covering, the best covering score for each segmented region from ground truth is averaged and denote as 'Best'. In addition, we also report the region numbers of the segmentation results averaged over all images.

Table 1 shows the benchmark results of the compared methods. All methods using our CAE-TVL network as preprocessing outperform the original ones. All the best scores (highlighted in bold) except for ODS covering are achieved by our approach. Surprisingly, the ODS covering and RI score (marked in red) using our CAE-TVL-GbS are worse

Table 1. Comparison of the evaluation results.

		MS	CAE-TVL-MS	GbS	CAE-TVL-GbS	QS	CAE-TVL-QS	SLIC	CAE-TVL-SLIC
Covering	ODS	0.4029	0.5006	0.5237	0.5102	0.4247	0.4339	0.4019	0.4155
	OIS	0.4681	**0.5608**	0.5466	0.5606	0.4412	0.4658	0.4272	0.4489
	Best	0.5774	0.6086	0.5987	**0.6621**	0.5715	0.5982	0.5142	0.5246
RI	ODS	0.7656	0.8000	0.8003	0.7934	0.7562	**0.8045**	0.7491	0.7621
	OIS	0.7954	0.8229	0.8171	0.8186	0.7980	**0.8362**	0.7645	0.7873
VOI	ODS	2.8120	2.2450	2.3016	**2.1168**	2.6319	2.3238	2.8205	2.4656
	OIS	2.5639	2.2537	2.2661	**2.0237**	2.4794	2.2164	2.7604	2.3945
Averaged reg. num.		209.94	60.13	114.03	58.32	119.41	64.55	619.45	278.65

than using GbS [5] directly. Since our CAE-TVL provides a smoothed image as input, it is more sensitive to the parameter setting of the segmentation methods. Assume that the range of the key parameter for GbS is from 25 to 500 with a step size of 25. The reason for the above ODS degradation might be that this parameter grid is too coarse for our approach. The averaged region numbers are significantly reduced by using our approach. This agrees with our expectation of restraining the over-segmentation by using CAE-TVL as preprocessing.

Table 2 presents the computation time we spend to segment the images (averaged for one image). With preprocessing of our approach, the segmentation methods require less time to compute the segmentation. To generate a smoothed image, our CAE-TVL takes less than 0.1 second.

Table 2. Comparison of computation time.

		MS	GbS	Quckshift	SLIC
Time (s)	w.o.	39.22	1.96	23.83	44.21
	with	36.64	1.74	21.23	34.77

Experiment 2: In this experiment, we test the generalization ability and the robustness of our CAE-TVL network. To do this, we first randomly split the whole data set into two parts, each containing 250 images. We denote the sub data sets as $D1$ and $D2$. We train our CAE-TVL on one subset D_i and compare the segmentation results using the same benchmark as in the first experiment. We only use CAE-TVL-GbS in this experiment.

Table 3 shows the benchmark scores. The segmentation accuracies in all four cases are comparable. This demonstrates that the trained CAE-TVL network is generic and can be applied once after training, to segment other new images.

To show the robustness of our CAE-TVL network, we first train it using BSDS500 and then test it on images randomly selected from Internet. Fig. 2 shows several examples. Notice that the BSDS500 contains scarcely street scene images like shown in the figure, but our approach can still generate improved segmentation.

4. CONCLUSION

In this paper, we deal with unsupervised image segmentation. We propose to train a convolutional autoencoder (CAE) with total variation loss (TVL) to smooth images. Using our CAE-TVL as preprecessing improves the performance of unsupervised image segmentation methods and reduce their computation complexity at the same time. Our approach is robust over different datasets and a pre-trained CAE-TVL network is applicable for segmentation tasks of general natural scene images from other datasets.

Table 3. Cross validation using CAE-TVL-GbS. The short notation in the first row is structured by 'training data set - test data set'

		D1-D1	D1-D2	D2-D1	D2-D2
Covering	ODS	0.5048	0.5072	0.5018	0.5176
	OIS	0.5739	0.5795	0.5705	0.5802
	Best	0.6589	0.6635	0.6560	0.6648
RI	ODS	0.7506	0.7587	0.7750	0.7812
	OIS	0.8053	0.8088	0.8239	0.8287
VOI	ODS	2.1441	2.1063	2.1876	2.1092
	OIS	1.9805	1.9542	2.0417	2.0196
Averaged reg. num.		60.724	58.052	59.417	56.960

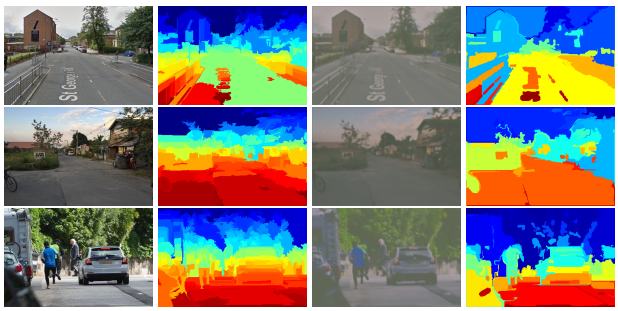


Fig. 2. Test examples using randomly selected images from Internet. From left to right: original images, segmentation results using [5], smoothed images generated by the pre-trained CAE-TVL using BSDS500, new segmentation results using the same unsupervised image segmentation method.

5. ACKNOWLEDGMENTS

The first author would like to thank CSC (China Scholarship Council) for financial support.

6. REFERENCES

[1] John Winn and Nebojsa Jojic, "Locus: Learning object classes with unsupervised segmentation," in *IEEE International Conference on Computer Vision*, 2005, vol. 1, pp. 756–763.

[2] Brian Fulkerson, Andrea Vedaldi, Stefano Soatto, et al., "Class segmentation and object localization with superpixel neighborhoods," in *IEEE International Conference on Computer Vision*, 2009, vol. 9, pp. 670–677.

[3] Dorin Comaniciu and Peter Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. PAMI*, vol. 24, no. 5, pp. 603–619, 2002.

[4] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Susstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. PAMI*, vol. 34, no. 11, pp. 2274–2282, 2012.

[5] Pedro F Felzenszwalb and Daniel P Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[6] Leonid I Rudin, Stanley Osher, and Emad Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259–268, 1992.

[7] David Mumford and Jayant Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on pure and applied mathematics*, vol. 42, no. 5, pp. 577–685, 1989.

[8] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[9] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *arXiv preprint arXiv:1511.00561*, 2015.

[10] Viren Jain, Joseph F Murray, Fabian Roth, Srinivas Turaga, Valentin Zhigulin, Kevin L Briggman, Moritz N Helmstaedter, Winfried Denk, and H Sebastian Seung, "Supervised learning of image restoration with convolutional networks," in *IEEE International Conference on Computer Vision*, 2007, pp. 1–8.

[11] Viren Jain and Sebastian Seung, "Natural image denoising with convolutional networks," in *Advances in Neural Information Processing Systems*, 2009, pp. 769–776.

[12] Vinod Nair and Geoffrey E Hinton, "Rectified linear units improve restricted boltzmann machines," in *International Conference on Machine Learning*, 2010, pp. 807–814.

[13] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[14] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks.," in *Aistats*, 2010, vol. 9, pp. 249–256.

[15] A. Vedaldi and K. Lenc, "Matconvnet – convolutional neural networks for matlab," in *ACM Int. Conf. on Multimedia*.

[16] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. PAMI*, vol. 33, no. 5, pp. 898–916, 2011.

[17] Andrea Vedaldi and Stefano Soatto, "Quick shift and kernel methods for mode seeking," in *European Conference on Computer Vision*. Springer, 2008, pp. 705–718.

[18] Christopher M Christoudias, Bogdan Georgescu, and Peter Meer, "Synergism in low level vision," in *IEEE International Conference on Pattern Recognition*, 2002, vol. 4, pp. 150–155.

[19] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," `http://www.vlfeat.org/`, 2008.

[20] Peter Kovesi, "Image segmentation using slic superpixels and dbscan clustering," `http://www.vlfeat.org/`, 2013.

[21] Jimei Yang, Brian Price, Scott Cohen, Honglak Lee, and Ming-Hsuan Yang, "Object contour detection with a fully convolutional encoder-decoder network," *arXiv preprint arXiv:1603.04530*, 2016.

[22] Piotr Dollár and C Lawrence Zitnick, "Structured forests for fast edge detection," in *IEEE International Conference on Computer Vision*, 2013, pp. 1841–1848.

[23] Zhile Ren and Gregory Shakhnarovich, "Image segmentation by cascaded region agglomeration," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2011–2018.

[24] Chunlai Wang and Bin Yang, "An unsupervised object-level image segmentation method based on foreground and background priors," in *IEEE Southwest Symposium on Image Analysis and Interpretation*, 2016, pp. 141–144.