Generating superpixels with deep representations
Extended abstract – Deep Vision workshop, CVPR 2018

Thomas Verelst Maxim Berman Matthew B. Blaschko
Dept. ESAT, Center for Processing Speech and Images
KU Leuven, Belgium
thomas.verelst1@student.kuleuven.be, {maxim.berman,matthew.blaschko}@esat.kuleuven.be

Abstract

Superpixel algorithms are a common pre-processing step for computer vision algorithms such as segmentation, object tracking and localization. Many superpixel methods only rely on colors features for segmentation, limiting performance in low-contrast regions and applicability to infrared or medical images where object boundaries have wide appearance variability. We study the inclusion of deep image features in the SLIC algorithm to exploit higher-level image representations. In addition, we study ways of fine-tuning superpixel segmentations to a particular image domain, yielding an intermediate domain-specific image representation that can be applied to different tasks.

1. Introduction

Many deep learning based applications in computer vision operate on grids of pixels and use convolutions trained end-to-end. However, popular algorithms have successfully leveraged image segmentation to group pixels into superpixels, reducing the input dimensionality while preserving the semantic content needed to address the task at hand [3]. Superpixels are efficient domain-specific image priors that tend to transfer across tasks and reduce the data needed to train models, which can be very beneficial for domain adaptation and weakly supervised settings, e.g. weakly supervised image segmentation [5]. Graph-based convolutional networks [4] also allow applications of deep learning beyond grid-like inputs. In this work, we study the inclusion of superpixel priors in deep learning pipelines.

The hand-crafted design of superpixels algorithms limits our ability to tune image segmentations to a specific image domain, such as infrared, medical, of spatio-temporal data. Given the focus on efficiency, superpixels have often been designed to operate on color features only; image segmentations could however incorporate higher-level image representations. We consider extensions to a standard superpixel algorithm incorporating higher-level unsupervised or supervised image features. We also study paths to fine-tune a superpixel segmentation algorithm to a specific modality.

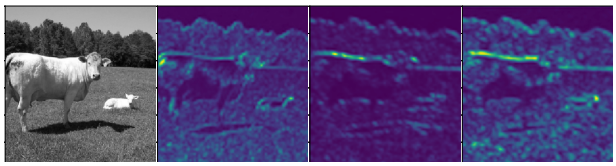


Figure 1: Original image and some scattering features

2. SLIC algorithm

The Simple Linear Iterative Clustering (SLIC) [1] image segmentation algorithm is popular both for its speed and performance; it uses a clustering approach similar to k-means, and usually operates on images in the CIELAB color space. After initialization of the cluster centers along a grid, a two-step iterative process clusters pixels until convergence. First the pixels are assigned the closest cluster center in the joint 5-dimensional space of colors (L, a and b) and spatial (x and y) components, with a weighted $L2$ distance that includes a compactness parameter σ balancing between colors and space. Second, the cluster centers are updated based om the pixel assignments. Finally, after convergence, a simple connected components algorithm enforces connectedness of the image segments.

3. Augmenting SLIC with deep representations

We experiment with SLIC beyond the original Lab features. For a particular pixel, we consider pixel-level image features extracted from a deep network $f_1, f_2, ...f_M$. These features can be unsupervised, as in the case of scattering features [6] (Fig. 1), or trained for a particular vision task. As we aim to integrate superpixels in a deep architecture, these features can be provided at no extra computational cost. We typically experiment with early layers of a CNN, which typically behave like smooth and universal filters.

To incorporate the image features into SLIC, we aug-

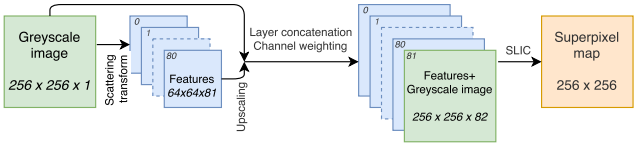
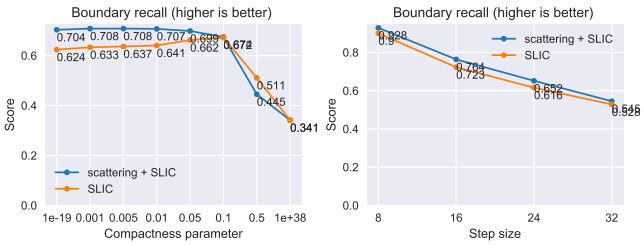


Figure 2: Integration of features into SLIC.

Table 1: Performance evaluated on 300 PASCAL VOC [2] images (superpixel size 16, compactness 0.05, 5 iterations)

	IoU	Rec	MDE	UE	CO
SLIC	0.920	0.723	1.23	0.080	0.29
Scat + SLIC	0.944	0.764	1.05	0.069	0.31



(a) Compactness parameter (b) Superpixel size

Figure 3: Influence of superpixel parameters

ment the number of image channels. The scattering features are upscaled and concatenated with the input image. The final image of size $W \times H \times (M+3)$ can be used in the SLIC algorithm, where the $Labxy$ clustering space now becomes a larger $L, a, b, f_1, f_2, ...f_M, x, y$ space. Individual feature maps are weighted with coefficients $\alpha_1 ... \alpha_M$.

We first experiment on greyscale images with features extracted from a 2-layer scattering network [6] (Figure 2). We manually define the weights α of the individual layers based on their visual appearance. Superpixel performance is evaluated using 5 metrics. Four of them are described in [7]: Boundary Recall (Rec), mean distance to edge (MDE), undersegmentation (UE) and compactness (CO). We also define an intersection over union (IoU) metric, which gives the maximum segmentation performance when superpixels are optimally labeled. Table 1 compares the performance of standard SLIC versus SLIC with scattering features. The scattering features improve all metrics by a considerable amount. Figure 3 shows the influence of the compactness and size parameters on the boundary recall scores.

4. Trainable superpixels

We further maximize the use of extra input features by integrating a trainable neural network in the SLIC algorithm. This eliminates the need to define feature weights α while also opening possibilities for better clustering.

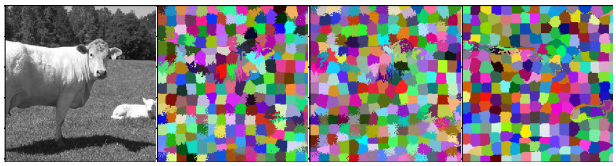


Figure 4: left-to-right: Input image, SLIC pixel clustering, trainable superpixels learned on SLIC, weakly supervised trainable superpixels

4.1. Architecture

We train a classifier assigning a pixel to one of its neighboring cluster centers. This assignment method replaces the k-means clustering of SLIC, and fits in the iterative approach of SLIC. The classification network is fed with the features of the pixel to be labeled, extended with the distances to the N closest clusters and the feature differences with these clusters. The network outputs a N-dimensional vector giving the assignment probabilities for the N closest clusters. The classifier network is small and easy to train. Parallel interference is possible since each pixel can be processed independently.

4.2. Training methods

We first train the neural classifier to replicate SLIC outputs and demonstrate a good replication of SLIC performance in our trainable framework (see Fig. 4). Beyond SLIC, we consider a supervised training of superpixels using semantic segmentation labels as a ground truth. A multi-label loss is used, where *1* is assigned to all clusters in the same segment as the pixel being labeled, and *0* to others. The training tends to have instabilities. The method does generally produce superpixels, but the generated superpixels do not always preserve object borders. Adding slightly more supervision by labeling all clusters in the same segment with *0.8* and the closest cluster with *1* gives more stable but inaccurate superpixels, as shown in Figure 4. The small receptive field of unsupervised scattering features might be a limiting factor; future experiments will explore our training strategy with more general deep features.

5. Conclusions and future work

Using a more comprehensive feature space instead of the common color space can improve superpixel algorithms. The integration of superpixels in a trainable pipeline can open the way to domain adaptation for superpixel representations. Trainable superpixels might exploit the deep representations in a better way. Future work includes a more careful design of the loss function for robust training of superpixels, and using deep features beyond unsupervised scattering representations.

Acknowledgements This work is partially funded by Internal Funds KU Leuven and an Amazon Research Award. This work made use of a hardware donation from the Facebook GPU Partnership program. We acknowledge support from the Research Foundation - Flanders (FWO) through project number G0A2716N.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

[2] 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.

[3] B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 670–677. IEEE, 2009.

[4] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[5] S. Kwak, S. Hong, B. Han, et al. Weakly supervised semantic segmentation using superpixel pooling network. In *AAAI*, pages 4111–4117, 2017.

[6] E. Oyallon, E. Belilovsky, and S. Zagoruyko. Scaling the scattering transform: Deep hybrid networks. In *International Conference on Computer Vision (ICCV)*, 2017.

[7] D. Stutz, A. Hermans, and B. Leibe. Superpixels: an evaluation of the state-of-the-art. *Computer Vision and Image Understanding*, 166:1–27, 2018.