Supplementary Materials for HERS Superpixels: Deep Affinity Learning for Hierarchical Entropy Rate Segmentation

Hankui Peng[1], Angelica I. Aviles-Rivero[1], and Carola-Bibiane Schönlieb[1]

[1]DAMTP, University of Cambridge.
{hp467, ai323, cbs31}@cam.ac.uk

This document extends the network design details and visual results presented in the main paper, which is structured as follows.

- **Section 1: Network Design.** We provide a detailed breakdown of the design of our proposed Deep Affinity Learning (DAL) network.
- **Section 2: Further Details on Borůvka's Algorithm.** We provide an explanation along with a visual example on the advantages of using Borůvka's Algorithm.
- **Section 3: Performance Measures.** In the interest of completeness, we detail the definitions of the performance measures that are used in the main paper to evaluate the superpixel segmentation results of various methods.
- **Section 4: Supplementary Qualitative Results.** We showcase various additional visual comparisons: i) demonstrating the advantage of our method against other state-of-the-art superpixel methods; and ii) showcasing the adaptiveness of our segmentation results with varying numbers of user-specified superpixel counts.
- **Section 5: Superpixels for saliency detection.** We apply our proposed DAL-HERS technique as a preprocessing step to the task of saliency detection.

1. Network Design

We provide further network design details of our Deep Affinity Learning (DAL) network in Table 1 and Table 2. We consider the setting where the given input image is of size $H \times W$, where $H = 480$ and $W = 320$. Each layer within a side output block is interleaved by a ReLU layer, which we omit in the table for ease of presentation.

In the first part of our DAL network, we obtain an intermediate affinity map (*out3c*) at the end of the three ResBlocks. This is used as input to the second stage, which mainly consists of the HED network structure (see Table 2) for further boundary information learning. Within the HED structure, the five side outputs as obtained at the end of each of the five Bilinear Interpolation (BI) steps all have size $8 \times 480 \times 320$. They are concatenated together to form a tensor of size $40 \times 480 \times 320$, which serves as input to a convolutional layer (fusion layer) that converts 40 channels down to 8 channels. Finally, the learned affinity map is obtained by applying the Sigmoid function to the output of the aforementioned fusion layer.

2. Further Details on Borůvka's Algorithm

In the main paper, we present a graph-based framework which consists of a neural network for deep affinity learning, and an efficient superpixel segmentation method HERS for obtaining highly adaptive superpixels efficiently. An illustration of HERS is displayed in Figure 1. It is clear that Borůvka's algorithm already arrived at the correct segmentation at the end of iteration 1, whereas the lazy greedy algorithm has only connected two nodes together.

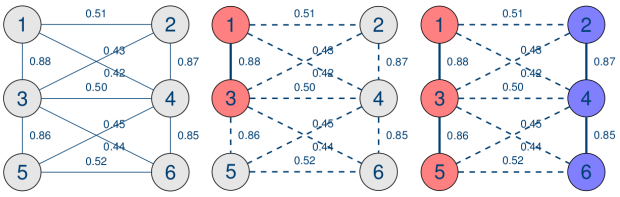


Figure 1: A simple example that illustrates the efficiency and parallelisability of Borůvka's algorithm. (a) displays an initial graph containing six nodes and their weighted edges. Borůvka's algorithm, displayed in (c), only needs one iteration to arrive at the final partitioning, whereas the lazy greedy algorithm [7] in (b) requires four iterations.

3. Performance Measures

The performance of various superpixel segmentation algorithms are commonly measured by the Under-segmentation Error (UE) [13], Achievable Segmentation Accuracy (ASA) [7] and Boundary Recall (BR) [8]. UE compares each computed superpixel with the ground truth superpixel that it overlaps with the most, and measures the "leakage" area that are not in the overlapped region. Opposite to UE, ASA quantifies the percentage of overlap between the segmented superpixels and the ground truth superpixels. That is, ASA can be directly obtained from UE as ASA=1-UE.

As such, we choose one out of these two and report ASA in our experiments. Concretely, ASA can be computed as

$$\text{ASA}(\mathcal{G}, \mathcal{L}) = \frac{1}{N} \sum_{k=1}^{K} \arg\max_{\mathcal{G}_c} |\mathcal{L}_k \cap \mathcal{G}_c|, \tag{1}$$

in which $\mathcal{G} = \{\mathcal{G}_1, \ldots, \mathcal{G}_C\}$ denotes the ground truth segmentation, and $\mathcal{L} = \{\mathcal{L}_1, \ldots, \mathcal{L}_K\}$ denotes the segmentation given by the chosen algorithm.

Boundary Recall (BR) measures the boundary adherence of the computed superpixels to the ground truth boundaries. It measures the proportion of ground truth boundary pixels that have been correctly identified by the computed superpixels. Concretely, BR can be computed as

$$\text{BR}(\mathcal{G}, \mathcal{L}) = \frac{\text{TP}(\mathcal{G}, \mathcal{L})}{\text{TP}(\mathcal{G}, \mathcal{L}) + \text{FN}(\mathcal{G}, \mathcal{L})}, \tag{2}$$

in which $\text{TP}(\mathcal{G}, \mathcal{L})$ stands for true positive, it denotes the number of ground truth boundary pixels that have been identified by the superpixel segmentation algorithm. $\text{FN}(\mathcal{G}, \mathcal{L})$ stands for false negative, which denotes the remaining number of ground truth boundary pixels that have not been identified.

Additionally, we also report the Explained Variation (EV) [9] score, which quantifies the variance within an image that is captured by the superpixels without relying on any ground truth labelling. It is calculated using the following formula

$$\text{EV}(\mathcal{L}) = \frac{\sum_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})^2}{\sum_i (\boldsymbol{p}_i - \boldsymbol{\mu})^2}, \tag{3}$$

where $\boldsymbol{p}_i$ denotes the RGB pixel values for the i-th pixel, $\boldsymbol{\mu}$ denotes the global mean of the RGB pixel values of an image, and $\boldsymbol{\mu}_i$ denotes the mean RGB pixel values for the superpixel that contains pixel $\boldsymbol{p}_i$.

4. Supplementary Qualitative Results

In this section, we provide further visual comparisons on the BSDS500 [3] and NYUv2 [11] datasets across the following state-of-the-art techniques: i) classic techniques: ERS [7], SH [14], SLIC [1], SNIC [2], SEEDS [13], ETPS [16]; and ii) deep learning techniques: SSN [6], SEAL-ERS [12] and SP-FCN [15].

4.1. Additional visual comparisons against state-of-the-art methods

Figures 2 displays an additional example from the BSDS500 dataset. We note that amongst the compared techniques, our technique presents the most visually appealing output. By visual inspection, we can notice that our superpixels clearly show better segmentations of fine details and stronger boundary adherence whilst avoiding partitioning homogeneous regions. In particular, several methods including the deep learning techniques of SEAL-ERS (see output (j)) and SP-FCN (see output (k)) often fail to capture the boundary structures. By contrast, our superpixels are better at capturing the objects of the scene including complex ones such as the shape of the flower in Figures 2. Some examples of these advantages are highlighted in the zoomed-in views.

These benefits of our technique are also observed in indoors scenes as displayed in Figures 3, which are taken from the NYUv2 test set. We selected interesting samples with complex objects of varying sizes. In Figure 3, one can observe that none of the competing techniques are able to adhere well to the boundaries of small objects (e.g. see the highlighted part in the blue zoomed-in square).

4.2. Various numbers of superpixels K

In this section, we demonstrate the adaptiveness of our superpixels with the number of user-defined superpixel counts ranging from 200 to 1200. Figures 4 and 5 showcase the results in terms of superpixel boundaries and in terms of the average RGB pixel features per superpixel on an image from the BSDS500 test set. It can be observed easily from the view with superpixel boundaries (see Figure 4) that our technique gradually focuses on segmenting the texture-rich regions of the image as the user-specified number of superpixels increases. As a result, our superpixels are able to provide a very accurate and smooth representation of the original image, even with a relatively small number of superpixels (see Figure 5).

Similarly, the same benefits of our superpixels can be observed in indoor scenes in Figures 6 and 7. We observe that our technique is able to delineate the main object boundaries in the image with 200 superpixels. With the increase of K, our technique further outlines the fine details within the identified objects. As a result, it is hard to even discern the difference visually between the original image (Figure 7 (a)) and the superpixel representations of the image (see Figure 7 (e) (f) (g)) at a first glance.

Operation	Input	Output	Kernel size	Stride size	Channel I/O	Input Res.	Output Res.
Conv.	image	out1	7	1	3/8	$3 \times 480 \times 320$	$8 \times 480 \times 320$
Ins. Norm.	out1	out2	-	-	8/8	$8 \times 480 \times 320$	$8 \times 480 \times 320$
Relu	out2	out3	-	-	8/8	$8 \times 480 \times 320$	$8 \times 480 \times 320$
ResBlock	out3	out3a	3	1	8/8	$8 \times 480 \times 320$	$8 \times 480 \times 320$
ResBlock	out3a	out3b	3	1	8/8	$8 \times 480 \times 320$	$8 \times 480 \times 320$
ResBlock	out3b	out3c	3	1	8/8	$8 \times 480 \times 320$	$8 \times 480 \times 320$
HED	out3c	out					

Table 1: Specification of the proposed Deep Affinity Learning (DAL) network structure.

Operation	Input	Output	Kernel size	Stride size	Channel I/O	Input Res.	Output Res.
Side Output 1	out3c	hed1a	3	1	8/64	$8 \times 480 \times 320$	$64 \times 480 \times 320$
	hed1a	hed1	3	1	64/64	$64 \times 486 \times 326$	$64 \times 480 \times 320$
Side Output 2	Max Pooling (Kernel size $= 2$, Stride $= 2$)						
	hed1	hed2a	3	1	64/128	$64 \times 240 \times 160$	$128 \times 240 \times 160$
	hed2a	hed2	3	1	128/128	$128 \times 240 \times 160$	$128 \times 240 \times 160$
Side Output 3	Max Pooling (Kernel size $= 2$, Stride $= 2$)						
	hed2	hed3a	3	1	128/256	$128 \times 120 \times 80$	$256 \times 120 \times 80$
	hed3a	hed3b	3	1	256/256	$256 \times 120 \times 80$	$256 \times 120 \times 80$
	hed3b	hed3	3	1	256/256	$256 \times 120 \times 80$	$256 \times 120 \times 80$
Side Output 4	Max Pooling (Kernel size $= 2$, Stride $= 2$)						
	hed3	hed4a	3	1	256/512	$256 \times 60 \times 40$	$512 \times 60 \times 40$
	hed4a	hed4b	3	1	512/512	$512 \times 60 \times 40$	$512 \times 60 \times 40$
	hed4b	hed4	3	1	512/512	$512 \times 60 \times 40$	$512 \times 60 \times 40$
Side Output 5	Max Pooling (Kernel size $= 2$, Stride $= 2$)						
	hed4	hed5a	3	1	256/512	$512 \times 30 \times 20$	$512 \times 30 \times 20$
	hed5a	hed5b	3	1	512/512	$512 \times 30 \times 20$	$512 \times 30 \times 20$
	hed5b	hed5	3	1	512/512	$512 \times 30 \times 20$	$512 \times 30 \times 20$
Conv. 1	hed1	hed1_out	1	1	64/8	$64 \times 480 \times 320$	$8 \times 480 \times 320$
	Bilinear Interpolation (BI), Output size $= 8 \times 480 \times 320$						
Conv. 2	hed2	hed2_out	1	1	128/8	$128 \times 240 \times 160$	$8 \times 240 \times 160$
	Bilinear Interpolation (BI), Output size $= 8 \times 480 \times 320$						
Conv. 3	hed3	hed3_out	1	1	256/8	$256 \times 120 \times 80$	$8 \times 120 \times 80$
	Bilinear Interpolation (BI), Output size $= 8 \times 480 \times 320$						
Conv. 4	hed4	hed4_out	1	1	512/8	$512 \times 60 \times 40$	$8 \times 60 \times 40$
	Bilinear Interpolation (BI), Output size $= 8 \times 480 \times 320$						
Conv. 5	hed5	hed5_out	1	1	512/8	$512 \times 30 \times 20$	$8 \times 30 \times 20$
	Bilinear Interpolation (BI), Output size $= 8 \times 480 \times 320$						
Conv. (Fusion Layer)	5 BI outputs	combined_out	1	1	40/8	$8 \times 480 \times 320$	$8 \times 480 \times 320$

Table 2: Specification of the HED component within the DAL network.

5. Superpixels for Saliency Detection

In this section, we present additional results on the application of our proposed DAL-HERS technique as a preprocessing tool for the downstream task of saliency detection [5, 4, 17]. The main purpose of this task is to extract the most salient object from its background in an image. One of the most classical techniques in saliency detection is Saliency Optimisation (SO) [17]. SO first segments an image into a number of superpixels, and then constructs an undirected weighted graph using the superpixels as primitives for further detecting the salient regions among them.

Here, the superpixels are produced using SLIC due to its simplicity and efficiency.

To demonstrate the advantages of our proposed DAL-HERS technique, we replace SLIC with DAL-HERS in the saliency detection process. We report both the quantitative comparison in Table 3 and the visual comparisons in Figure 8 in terms of the standard performance metrics on the ECSSD dataset [10]. It can be seen from Table 3 that our DAL-HERS technique enjoys an obvious advantage over SLIC across all three metrics. This advantage is further supported by the visual comparisons in Figure 8. It is clear that the results produced using SLIC are very non-smooth and segmented, which is due to the non-adaptive nature of SLIC superpixels. Whereas the results produced with DAL-HERS are smooth whilst highlighting the contours of the most salient object in an image.

Method	F-measure	weighted F_β	MAE
SLIC	0.885	0.432	0.286
DAL-HERS	**0.906**	**0.520**	**0.200**

Table 3: Superpixels for saliency object detection.

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine 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] Radhakrishna Achanta and Sabine Susstrunk. Superpixels and polygons using simple non-iterative clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4651–4660, 2017.

[3] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2010.

[4] Stas Goferman, Lihi Zelnik-Manor, and Ayellet Tal. Context-aware saliency detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):1915–1926, 2011.

[5] Xiaodi Hou and Liqing Zhang. Saliency detection: A spectral residual approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[6] Varun Jampani, Deqing Sun, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Superpixel sampling networks. In *Proceedings of the European Conference on Computer Vision*, pages 352–368, 2018.

[7] Ming-Yu Liu, Oncel Tuzel, Srikumar Ramalingam, and Rama Chellappa. Entropy rate superpixel segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2097–2104. IEEE, 2011.

[8] David R Martin, Charless C Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):530–549, 2004.

[9] Alastair P Moore, Simon JD Prince, Jonathan Warrell, Umar Mohammed, and Graham Jones. Superpixel lattices. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[10] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia. Hierarchical image saliency detection on extended CSSD. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(4):717–729, 2015.

[11] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *European Conference on Computer Vision*, pages 746–760. Springer, 2012.

[12] Wei-Chih Tu, Ming-Yu Liu, Varun Jampani, Deqing Sun, Shao-Yi Chien, Ming-Hsuan Yang, and Jan Kautz. Learning superpixels with segmentation-aware affinity loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 568–576, 2018.

[13] Michael Van den Bergh, Xavier Boix, Gemma Roig, Benjamin de Capitani, and Luc Van Gool. SEEDS: Superpixels extracted via energy-driven sampling. In *Proceedings of the European Conference on Computer Vision*, pages 13–26. Springer, 2012.

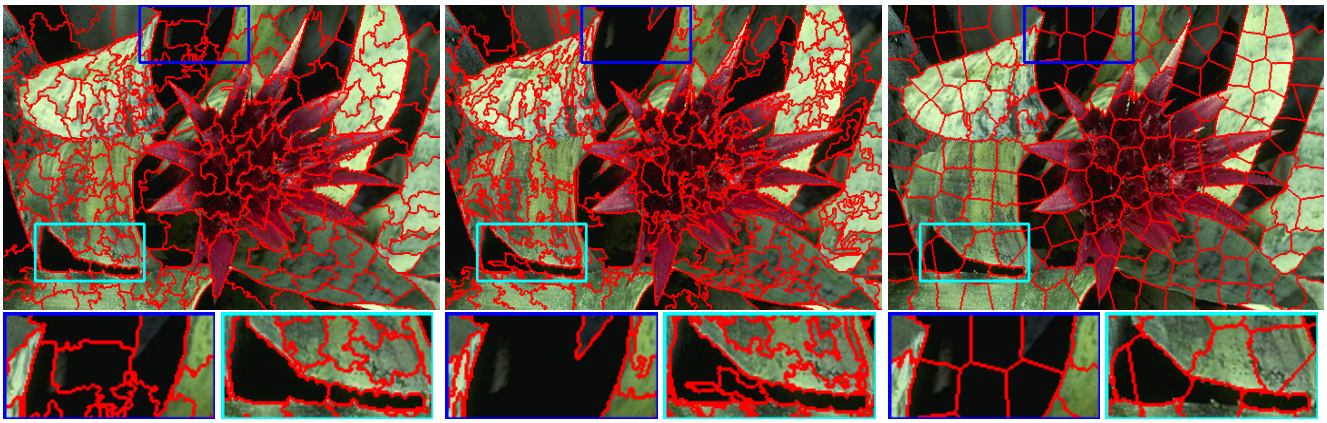
[14] Xing Wei, Qingxiong Yang, Yihong Gong, Narendra Ahuja, and Ming-Hsuan Yang. Superpixel hierarchy. *IEEE Transactions on Image Processing*, 27(10):4838–4849, 2018.

[15] Fengting Yang, Qian Sun, Hailin Jin, and Zihan Zhou. Superpixel segmentation with fully convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 13964–13973, 2020.

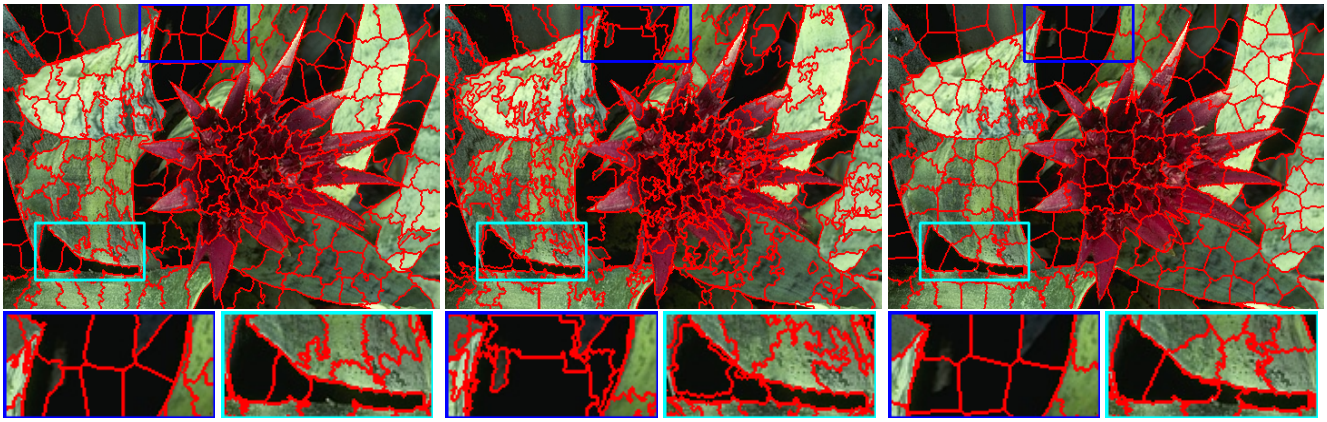
[16] Jian Yao, Marko Boben, Sanja Fidler, and Raquel Urtasun. Real-time coarse-to-fine topologically preserving segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2947–2955, 2015.

[17] Wangjiang Zhu, Shuang Liang, Yichen Wei, and Jian Sun. Saliency optimization from robust background detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2814–2821, 2014.

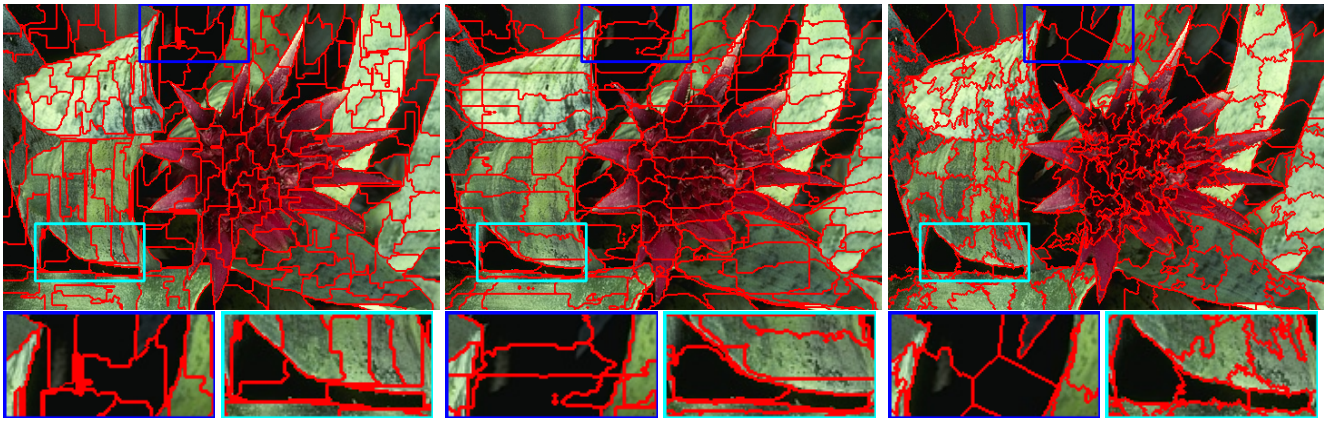
(a) Original. (b) Ground truth. (c) Ours.



(d) ERS. (e) SH. (f) SLIC.



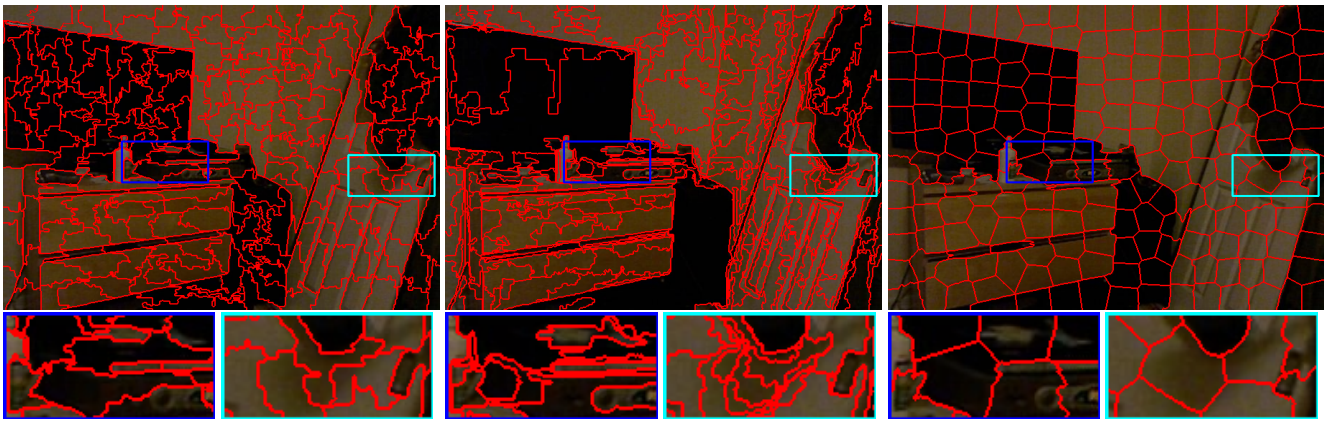
(g) SNIC. (h) SEEDS. (i) ETPS.



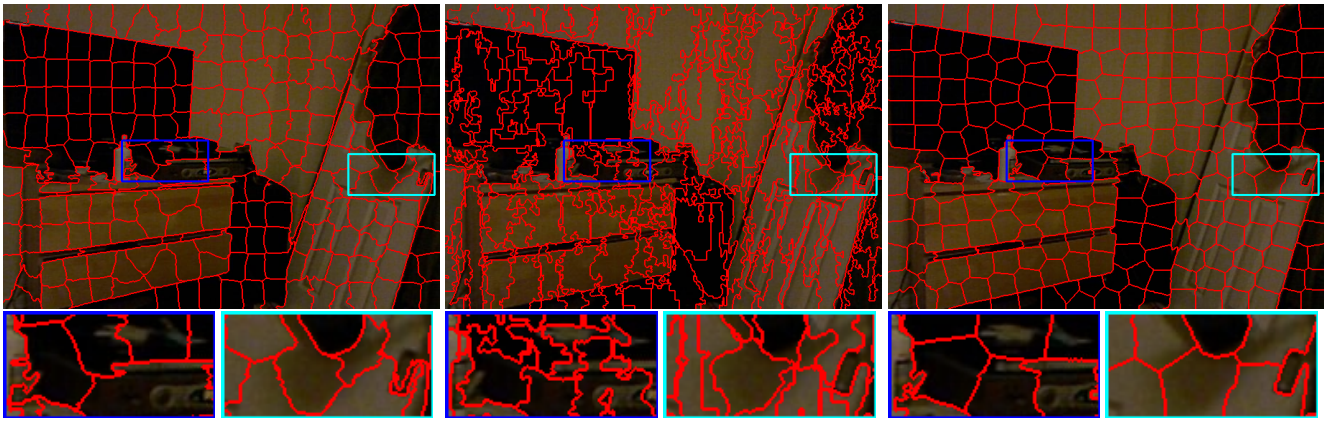
(j) SEAL-ERS. (k) SP-FCN. (l) SSN.

Figure 2: Segmentation results on a sample image from the BSDS500 test set with 200 superpixels.

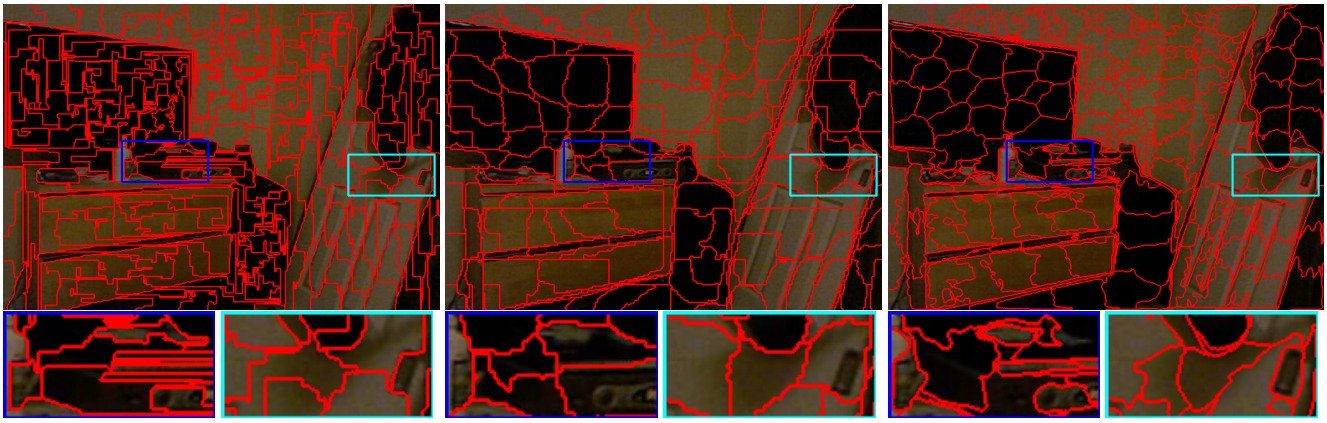
(a) Original. (b) Ground truth. (c) Ours.



(d) ERS. (e) SH. (f) SLIC.

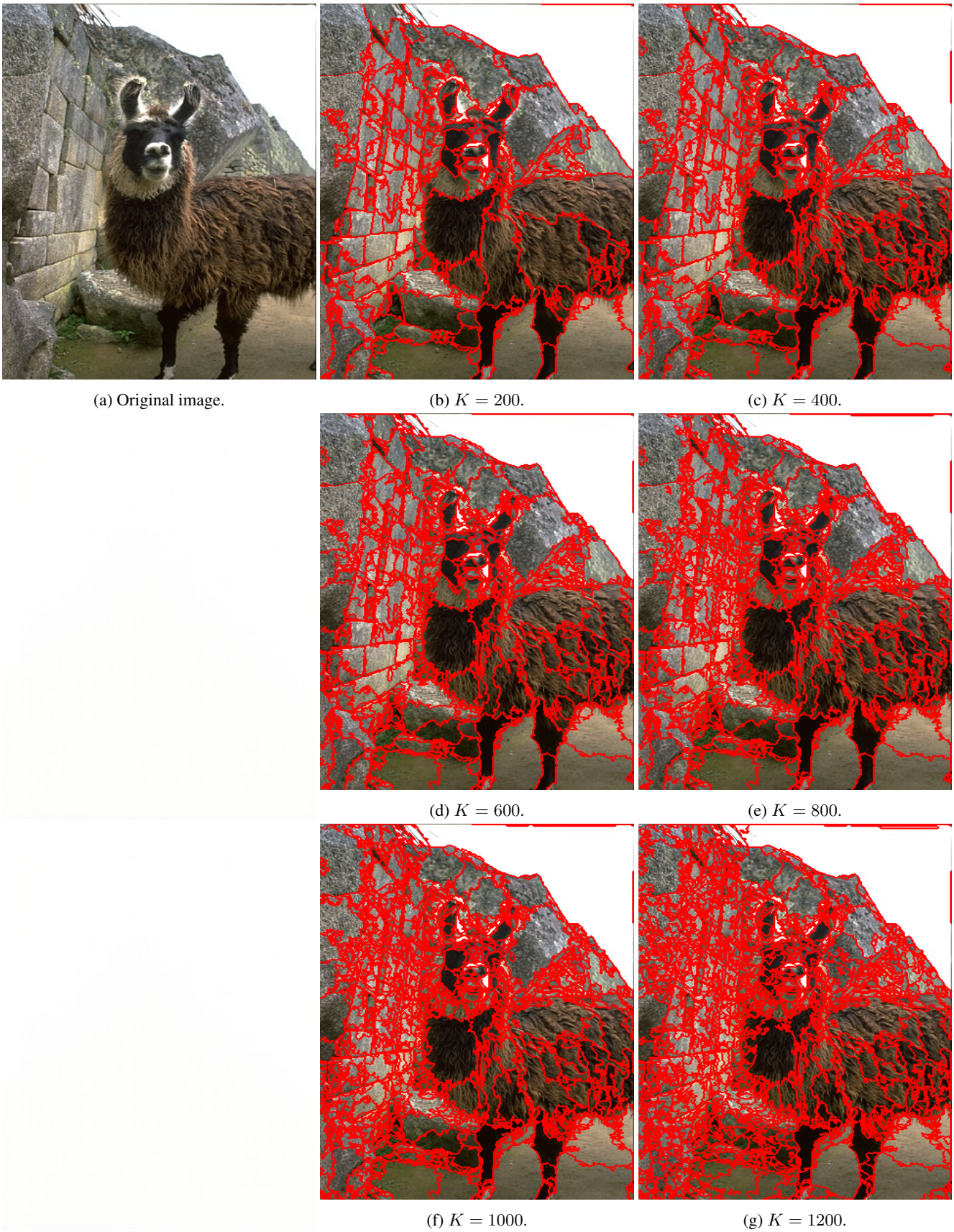


(g) SNIC. (h) SEEDS. (i) ETPS.



(j) SEAL-ERS. (k) SP-FCN. (l) SSN.

Figure 3: Segmentation results on a sample image from the NYUv2 test set with 200 superpixels.



(a) Original image. (b) $K = 200$. (c) $K = 400$.

(d) $K = 600$. (e) $K = 800$.

(f) $K = 1000$. (g) $K = 1200$.

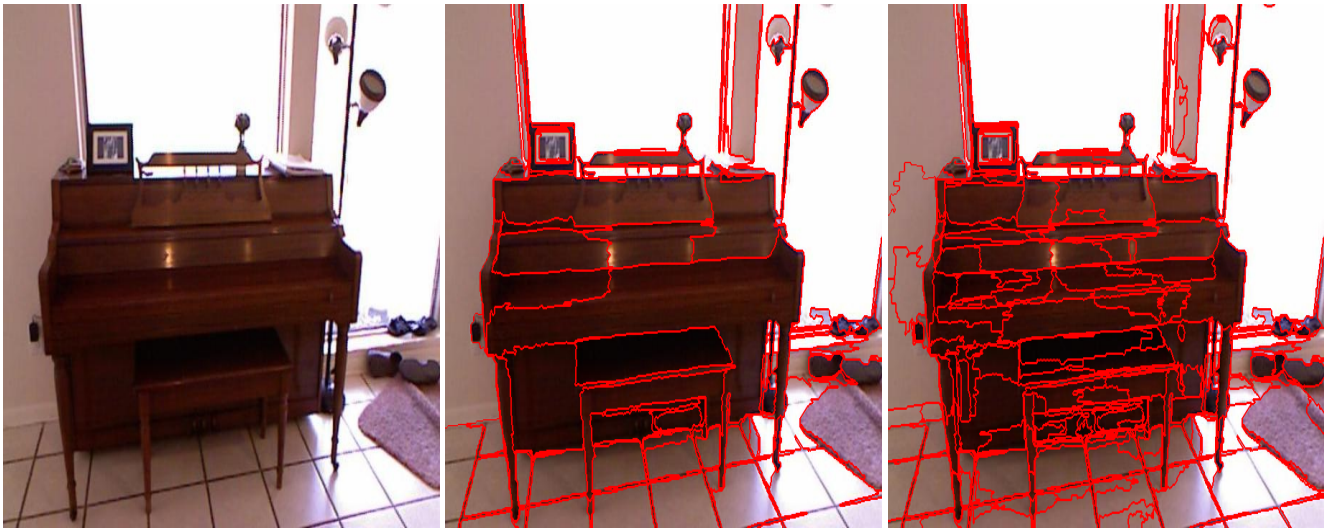
Figure 4: Segmentation results using our method on an image from the BSDS500 test set with varying number of superpixels.

(a) Original image. (b) $K = 200$. (c) $K = 400$.

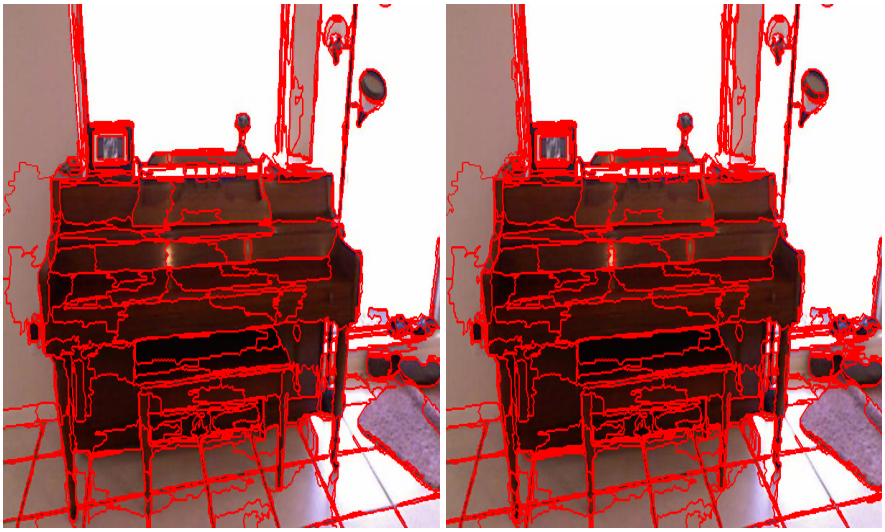
(d) $K = 600$. (e) $K = 800$.

(f) $K = 1000$. (g) $K = 1200$.

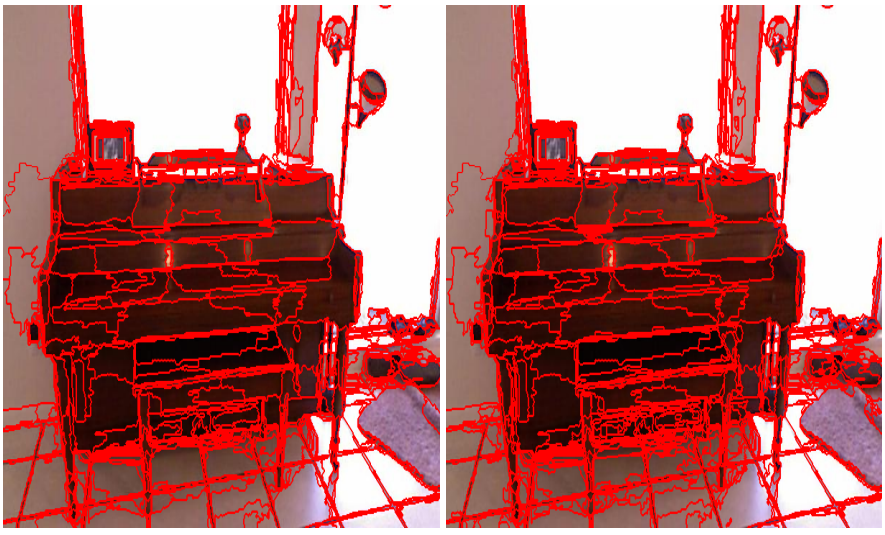
Figure 5: Segmentation results using our method on an image from the BSDS500 test set with varying number of superpixels. Each segmented image is represented with the average RGB pixel values of the corresponding superpixel.



(a) Original image. (b) $K = 200$. (c) $K = 400$.

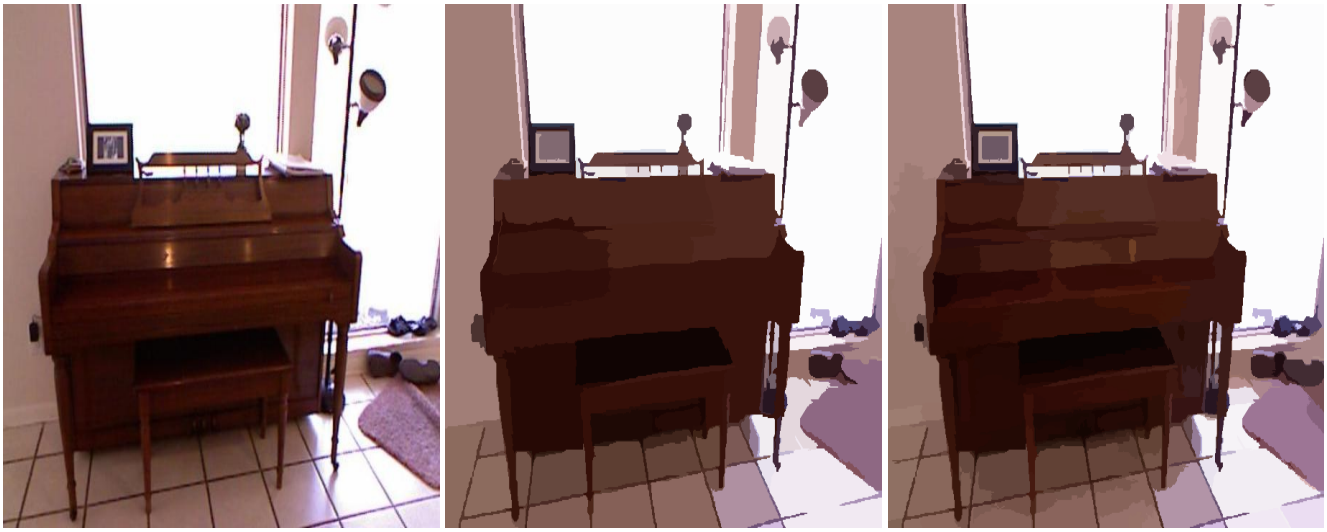


(d) $K = 600$. (e) $K = 800$.

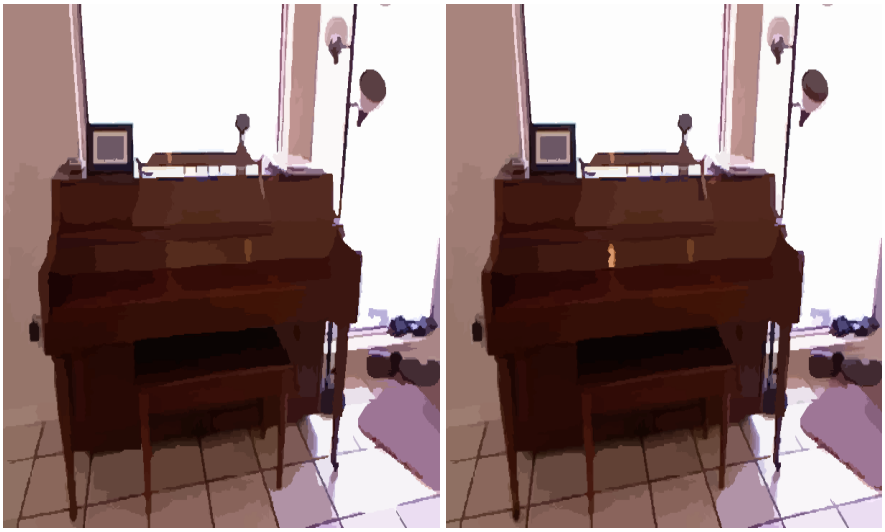


(f) $K = 1000$. (g) $K = 1200$.

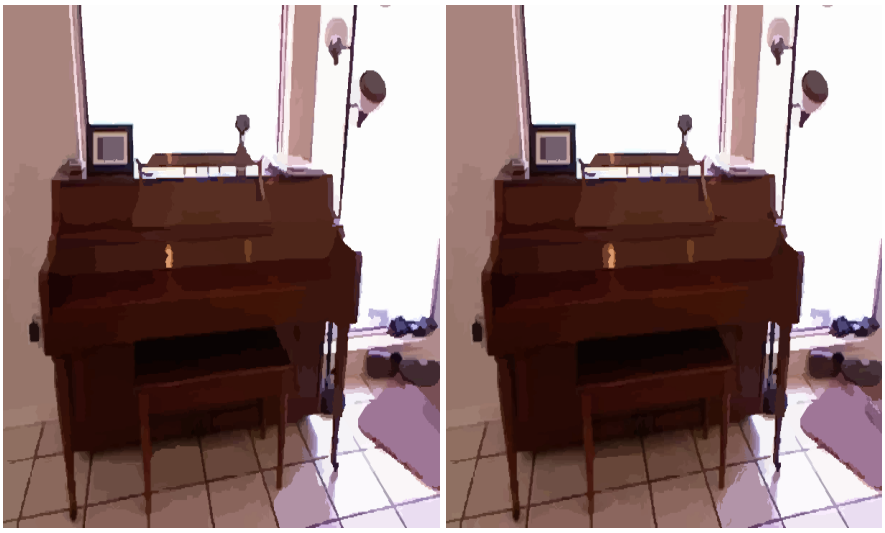
Figure 6: Segmentation results using our method on an image from the NYUv2 test set with varying number of superpixels.



(a) Original image. (b) $K = 200$. (c) $K = 400$.



(d) $K = 600$. (e) $K = 800$.



(f) $K = 1000$. (g) $K = 1200$.

Figure 7: Segmentation results using our method on an image from the BSDS500 test set with varying number of superpixels. Each segmented image is represented with the average RGB pixel values of the corresponding superpixel.

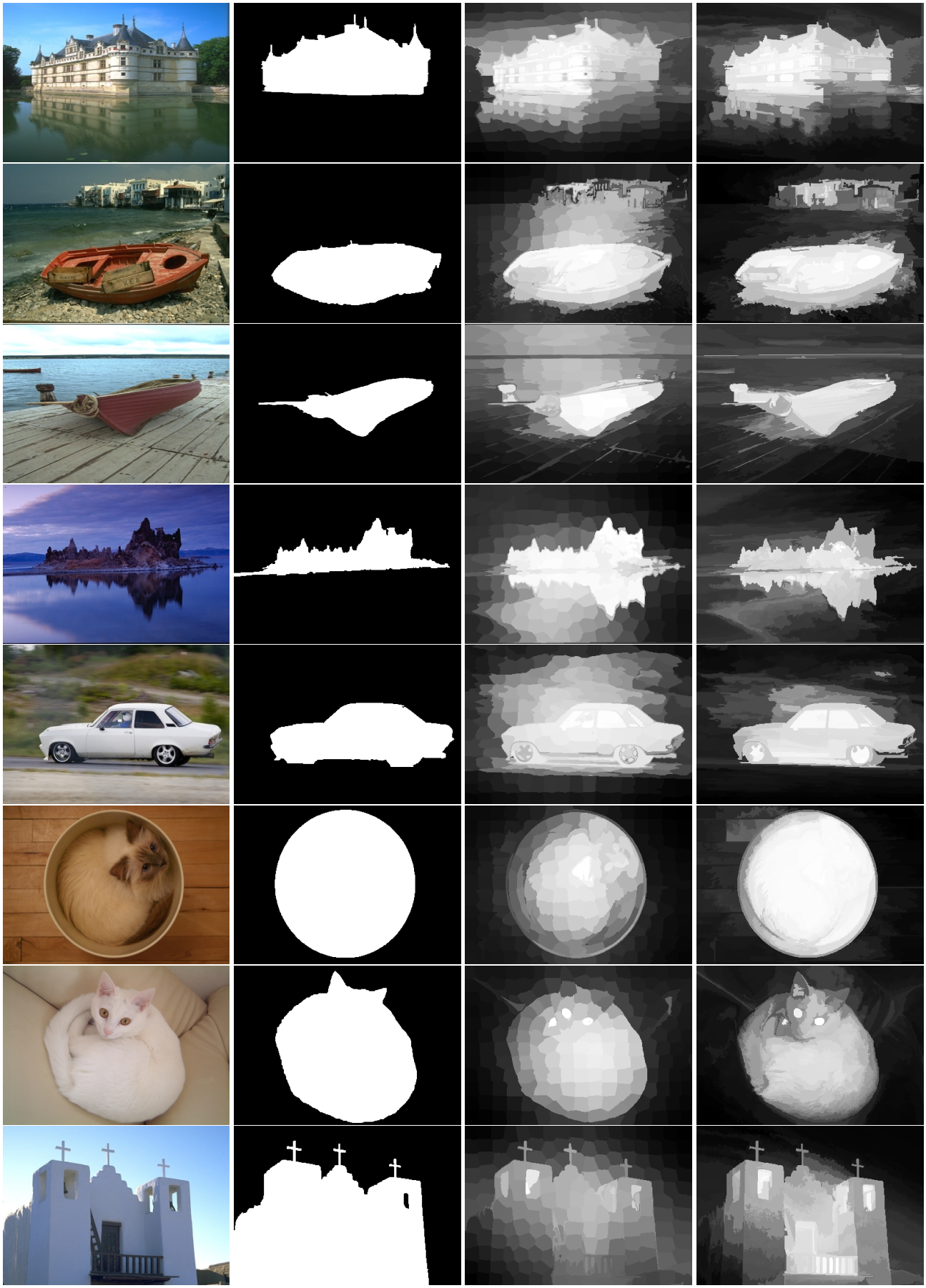


Figure 8: Saliency detection results on sample images from the ECSSD dataset [10] with 200 superpixels. From left to right: original image, ground truth segmentation mask, SLIC, DAL-HERS.