

An Evolutionary Approach for Image Segmentation

Name: Lin Huangjing SID: 1155072677

1. Background

Image segmentation is a challenging task in many fields. Image segmentation aims at partitioning an image into certain regions satisfying a homogeneity criterion that takes into account one or more pixel features, such as color, texture, intensity, and so on. Many methods have been proposed to deal with segmentation. Most of these methods can be divided into three main types: boundary detection-based approaches, region clustering-based approaches, and graph-based approaches. In this paper, the authors proposed their method named “GeNCut” based on graph-based theory. Based on the method done by Shi and Malik [1], the authors improved the method by proposing a new fitness function and introducing the concept of Genetic Algorithm (GA). With evolutionary procedure to adjust the number of division objects dynamically, this improvement overcomes the disadvantage of inevitably setting number k of division beforehand in the previous approaches.

2. Problem Definition

The image is defined as a graph $G = (V, E, w)$, where V is the set of n nodes of graph G , E is the set of edges, and $w: E \rightarrow \mathbb{R}$ is a function that assigns a value to graph edges. Each node corresponds to a pixel in the image, and a graph edge (i, j) connects two pixels i and j , provided that these two pixels satisfy some suitably defined property. The weight $w(i, j)$ associated with an edge (i, j) represents the likelihood that pixels i and j belong to the same image region. The definition of weight between two pixels is defined as follow:

$$w(i, j) = \begin{cases} e^{-\max_{x \in \text{line}(i, j)} \|\text{Edge}(x)\|_2^2 / 2a^2} & \text{if } \|X(i) - X(j)\|_2 < r, \quad i \neq j \\ 0 & \text{otherwise} \end{cases}$$

where $a = (\max_{y \in I} \|\text{Edge}(y)\|) \times \delta$, $\text{Edge}(x)$ is the image edge strength at position x , $\text{line}(i, j)$ is a straight line between i and j .

Taking a 4x4 toy image for example, the weight can be defined as a matrix as follow:

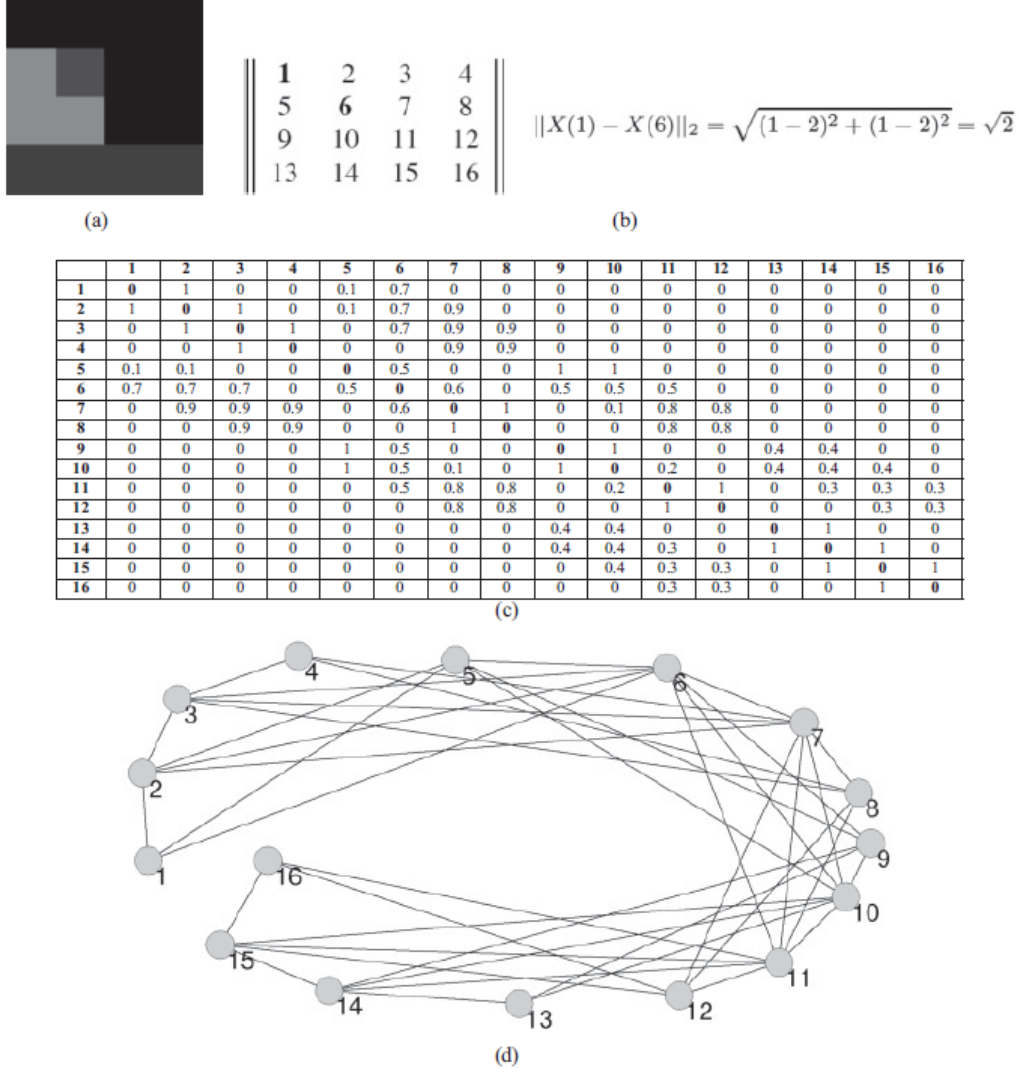


Figure 1: (a) An image I composed of 4×4 pixels. (b) The matrix corresponding to I where each pixel is labeled with increasing numbers from 1 to 16, and computation of the spatial Euclidean distance between pixel 1 and pixel 6 at position (1,1) and (2,2), respectively. (c) Adjacency weight matrix of size 16×16 , associated with I . Weights are computed by using Equation (1) with $r = 2$. (d) Graph representation corresponding to image I having 16 nodes.

3. Graph Partition for Image Segmentation

In the previous work has been done in [1], Shi and Mali used the conception of normalized cut proposed by Wu and Leahy [2], which is defined as follow:

$$\begin{aligned} cut(A, B) &= \sum_{i \in A, j \in B} w(i, j). \\ assoc(A, V) &= \sum_{i \in A, t \in V} w(i, t) \\ Ncut(A, B) &= \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}. \end{aligned}$$

The problem is formalized to find the minimal value of the normalized cut and it can be solved as a generalized eigenvalue problem. They compute an optimal partition by using the eigenvector with the second smallest eigenvalue. To perform k -way partitioning, the objective function is modified as:

$$\text{Ncut}_k = \sum_{i=1}^k \frac{\text{cut}(A_i, V - A_i)}{\text{assoc}(A_i, V)}$$

In order to get a better performance in evolution, the authors introduced an extension of the conception of normalized cut named “weighted normalized cut (WNCut)”. The WNCut is a bit different from the Ncut in the second term, and it has been defined as:

$$\text{WNCut} = \sum_{i=1}^k \frac{\text{cut}(A_i, V - A_i)}{\text{assoc}(A_i, V)} + \frac{\text{cut}(A_i, V - A_i)}{\text{assoc}(V - A_i, V)}$$

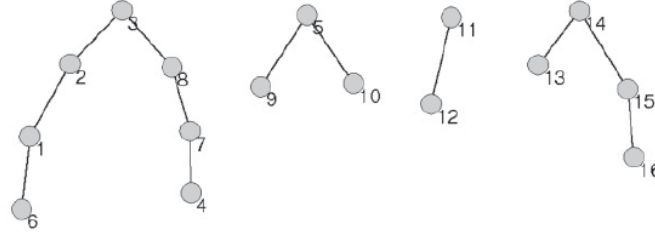
4. An Evolutionary Method to Segment Images

4.1 Representation in Genetic Algorithm

The genetic algorithm uses the locus-based adjacency representation proposed by Park and Song [3]. In the representation, an individual of the population consists of n genes g_1, g_2, \dots, g_n and each gene can assume value in the range $\{1, \dots, n\}$. For example, if node i connects with node j , the $g_i = j$ is set (As Fig. 2(a) shows).

pixel	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
neighbor	2	3	8	7	9	1	4	7	5	5	12	11	14	13	14	15

(a)



(b)

$$\begin{pmatrix} r_1 & r_1 & r_1 & r_1 \\ r_2 & r_1 & r_1 & r_1 \\ r_2 & r_2 & r_3 & r_3 \\ r_4 & r_4 & r_4 & r_4 \end{pmatrix}$$

(c)

Figure 2: (a) Locus-based representation of individual J_1 . (b) Corresponding segmentation of the graph G of Figure 1(d) in four segments composed by the pixels $r_1 = \{1, 2, 3, 4, 6, 7, 8\}$, $r_2 = \{5, 9, 10\}$, $r_3 = \{11, 12\}$, and $r_4 = \{13, 14, 15, 16\}$. (c) Image partitioning.

In the pixel connection, only pixels satisfying the h-Neighborhood are regarded as neighbors. This h-Neighborhood definition not only considers the spatial closeness, but also the pixel affinity. Within the distance radius r , only pixels with top-h highest weight are regarded as a pixel's neighbor. For example in Fig. 1(c) and Fig. 6, if we define the $h=2$, for node 3, the top-2 highest weight are 1.0 and 0.9, so the node 2, 4, 7, 8 are kept as its neighbor and node 6 ($w=0.7$) is removed.

$$\begin{aligned} 1 &\rightarrow \{(2; 1.0), (6; 0.7)\} \\ 2 &\rightarrow \{(1; 1.0), (3; 1.0), (7; 0.9)\} \\ 3 &\rightarrow \{(2; 1.0), (4; 1.0), (7; 0.9), (8; 0.9)\} \\ 4 &\rightarrow \{(3; 1.0), (7; 0.9), (8; 0.9)\} \\ 5 &\rightarrow \{(6; 0.5), (9; 1.0), (10; 1.0)\} \\ 6 &\rightarrow \{(1; 0.7), (2; 0.7), (3; 0.7), (7; 0.6)\} \\ 7 &\rightarrow \{(2; 0.9), (3; 0.9), (4; 0.9), (8; 1.0)\} \\ 8 &\rightarrow \{(3; 0.9), (4; 0.9), (7; 1.0)\} \\ 9 &\rightarrow \{(5; 1.0), (6; 0.5), (10; 1.0)\} \\ 10 &\rightarrow \{(5; 1.0), (6; 0.5), (9; 1.0)\} \\ 11 &\rightarrow \{(7; 0.8), (8; 0.8), (12; 1.0)\} \\ 12 &\rightarrow \{(7; 0.8), (8; 0.8), (11; 1.0)\} \\ 13 &\rightarrow \{(9; 0.4), (10; 0.4), (14; 1.0)\} \\ 14 &\rightarrow \{(9; 0.4), (10; 0.4), (13; 1.0), (15; 1.0)\} \\ 15 &\rightarrow \{(10; 0.4), (14; 1.0), (16; 1.0)\} \\ 16 &\rightarrow \{(11; 0.3), (12; 0.3), (15; 1)\} \end{aligned}$$

(a)

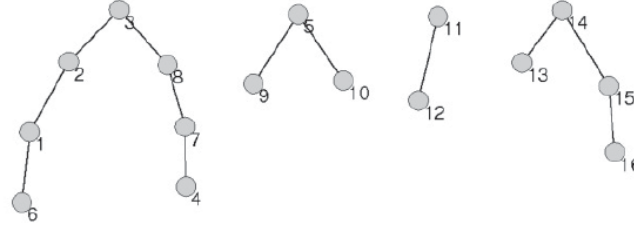
Figure 6: Nearest neighbors nn_i^h for $1 \leq i \leq 16$ and $h = 2$.

4.2 Crossover and Mutation

Provided there are two parent-individuals J_1 and J_2 as follow:

pixel	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
neighbor	2	3	8	7	9	1	4	7	5	5	12	11	14	13	14	15

(a)



(b)

$$\begin{pmatrix} r_1 & r_1 & r_1 & r_1 \\ r_2 & r_1 & r_1 & r_1 \\ r_2 & r_2 & r_3 & r_3 \\ r_4 & r_4 & r_4 & r_4 \end{pmatrix}$$

(c)

Figure 2: (a) Locus-based representation of individual J_1 . (b) Corresponding segmentation of the graph G of Figure 1(d) in four segments composed by the pixels $r_1 = \{1, 2, 3, 4, 6, 7, 8\}$, $r_2 = \{5, 9, 10\}$, $r_3 = \{11, 12\}$, and $r_4 = \{13, 14, 15, 16\}$. (c) Image partitioning.

pixel	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
neighbor	2	3	4	3	6	7	8	7	10	9	12	11	14	13	14	15

(a)



(b)

$$\begin{pmatrix} r_1 & r_1 & r_1 & r_1 \\ r_2 & r_2 & r_2 & r_2 \\ r_3 & r_3 & r_4 & r_4 \\ r_5 & r_5 & r_5 & r_5 \end{pmatrix}$$

(c)

Figure 3: (a) Locus-based representation of another individual J_2 . (b) Corresponding segmentation of the graph G of Figure 1(d) in five segments composed of the pixels $r_1 = \{1, 2, 3, 4\}$, $r_2 = \{5, 6, 7, 8\}$, $r_3 = \{9, 10\}$, $r_4 = \{11, 12\}$, and $r_5 = \{13, 14, 15, 16\}$. (c) Image partitioning.

The crossover is implemented by an exquisite definition of mask that is generated randomly (As Fig. 4(a)). If mask[i] is 0, the g_i is selected from J_1 ;

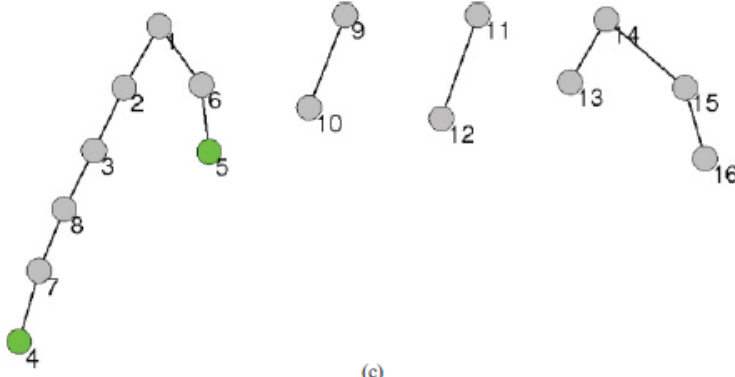
otherwise, g_i is selected from J2. As Fig. 4 show, after crossover between J1 and J2, the segmentation of image is composed of the pixels $\{1, 2, 3, 4, 5, 6, 7, 8\}$, $\{9, 10\}$, $\{11, 12\}$ and $\{13, 14, 15, 16\}$.

mask	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	1	0	0	0	1	0	0	1	1	1	0	0	0	1	1	0

(a)

pixel	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
neighbor	2	3	8	7	6	1	4	7	10	9	12	11	14	13	14	15

(b)



(c)

$$\begin{pmatrix} r_1 & r_1 & r_1 & r_1 \\ r_1 & r_1 & r_1 & r_1 \\ r_2 & r_2 & r_3 & r_3 \\ r_4 & r_4 & r_4 & r_4 \end{pmatrix}$$

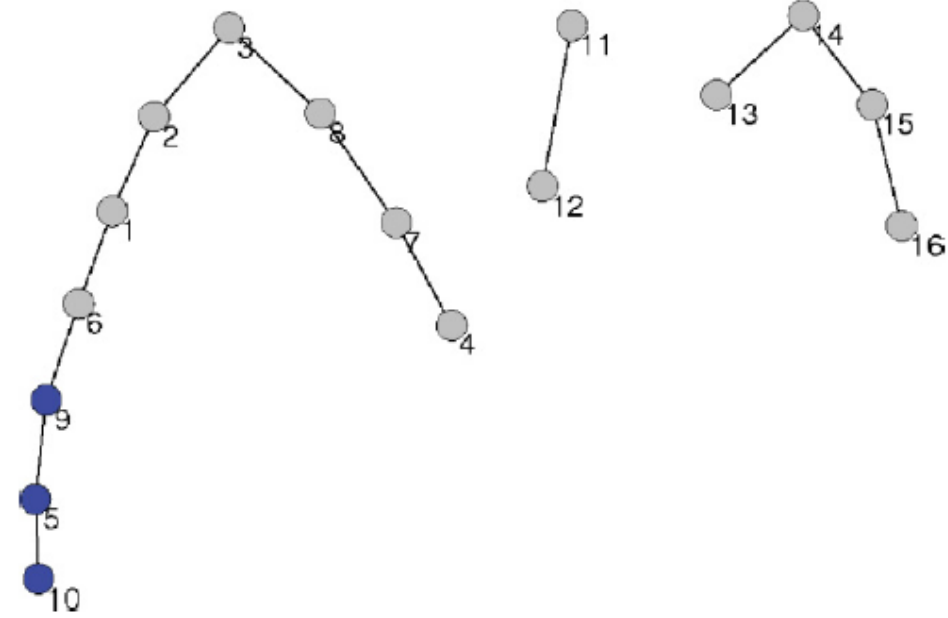
(d)

Figure 4: Crossover between the two individuals J_1 and J_2 of Figures 2–3. (a) Random mask generated. (b) Offspring generated from J_1 and J_2 . (c) Corresponding segmentation of the graph G of Figure 1(d) in four segments composed of the pixels $r_1 = \{1, 2, 3, 4, 5, 6, 7, 8\}$, $r_2 = \{9, 10\}$, $r_3 = \{11, 12\}$, and $r_4 = \{13, 14, 15, 16\}$. (d) Image partitioning.

The mutation operator, analogous to the initialization process, randomly assigns to a neighbor node i . For example in the Fig. 5, the node 9 is chosen at random and its neighbor 5 is substituted by the other neighbor 6. As Fig. 5 show, after mutation from J1, a new segmentation in three components constituted by $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, $\{11, 12\}$ and $\{13, 14, 15, 16\}$.

allele	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
gene	2	3	8	7	9	1	4	7	6	5	12	11	14	13	14	15

(a)



(b)

$$\begin{pmatrix} r_1 & r_1 & r_1 & r_1 \\ r_1 & r_1 & r_1 & r_1 \\ r_1 & r_1 & r_2 & r_2 \\ r_3 & r_3 & r_3 & r_3 \end{pmatrix}$$

(c)

Figure 5: (a) Mutation on individual J_1 of Figure 2 where 9 is the randomly chosen node. Its neighbor 5 is substituted with 6. (b) The new generated individual corresponds to the segmentation of graph G of Figure 1(d) in three segments composed of $r_1 = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, $r_2 = \{11, 12\}$, and $r_3 = \{13, 14, 15, 16\}$. (c) Image partitioning.

4.3 Algorithm

The framework of algorithm can be divided into three main phases: (1) preprocessing phase, (2) segmentation phase and (3) final post-processing phase. The framework of algorithm is presented as:

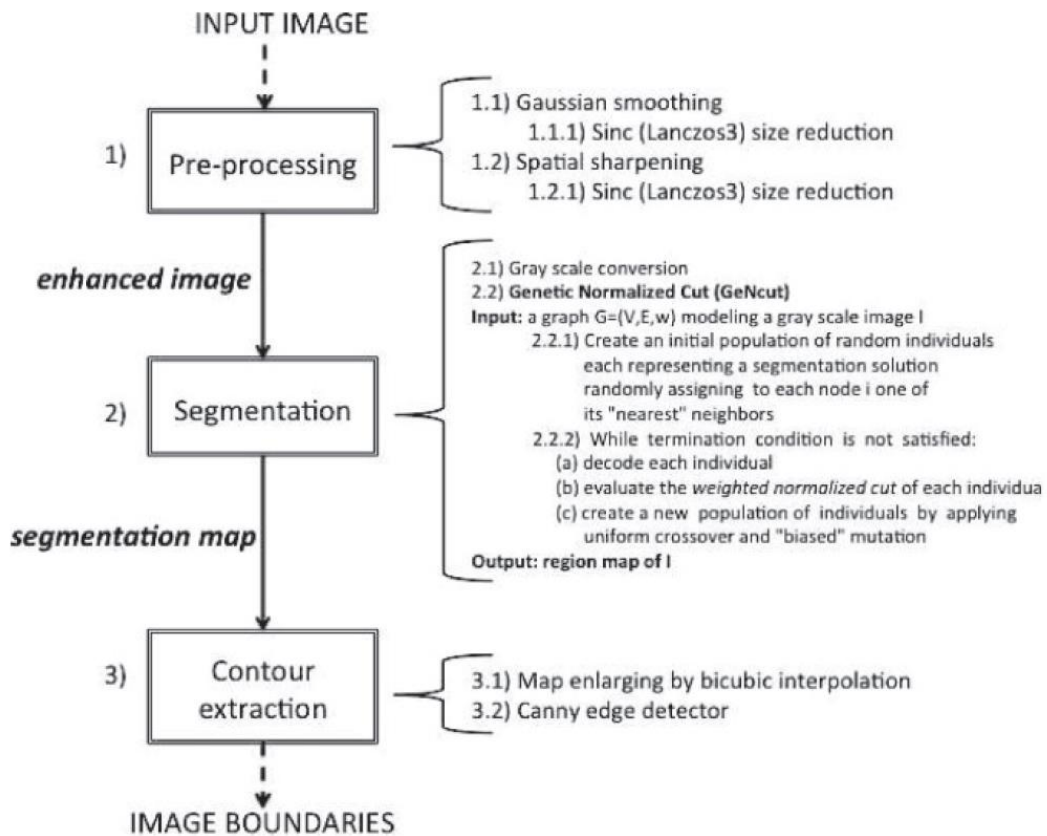


Figure 7: Main steps of the segmentation process and pseudocode of GeNCut. In the preprocessing phase, a Gaussian filter is used to smooth the input image. After that, the input image is resized to a smaller size before it being processed.

In the segmentation phase, the input image is transformed into a gray scale image. After that, the graph modeling the image is generated and given as input to GeNCut. The output of GeNCut is a segmentation map in which each pixel is assigned the label of the cluster it belongs to.

In the post-processing phase, the result will be resized to original size of input image, and then, canny edge detector is used to find segmentation boundaries.

5. Experimental Results

5.1 Synthetic Image

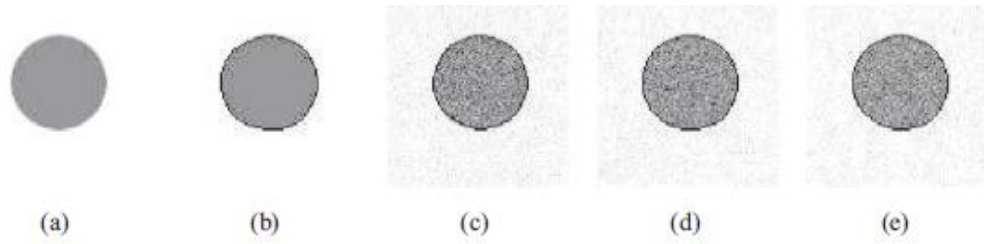


Figure 8: (a) A synthetic image and (b) its segmentation. Segmentation results on the synthetic image altered with increasing Gaussian noise having zero mean with increasing variance (c) 0.01; (d) 0.02; and (e) 0.03.

5.2 Satellite Image

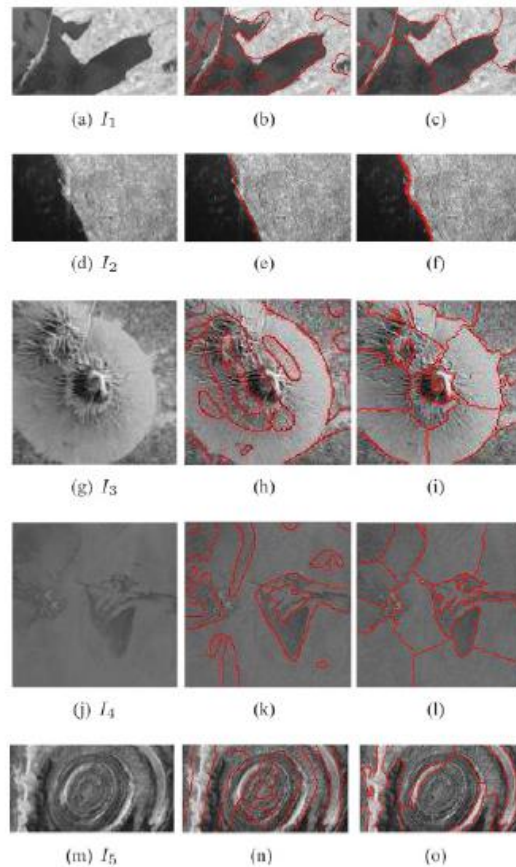


Figure 9: Segmentation on satellite images. From the first to the last image we have: Strait of Kerch, Nero Sea (image 1), Pompei area, Italy (image 2), Mount Egmont, New Zealand (image 3), oil spills in Gulf of Mexico (image 4) and geological structure of Richat, Mauritania (image 5). For each line, the original image together with the segmentation result of GeNCut and Ncut are presented.

5.3 Medical Image

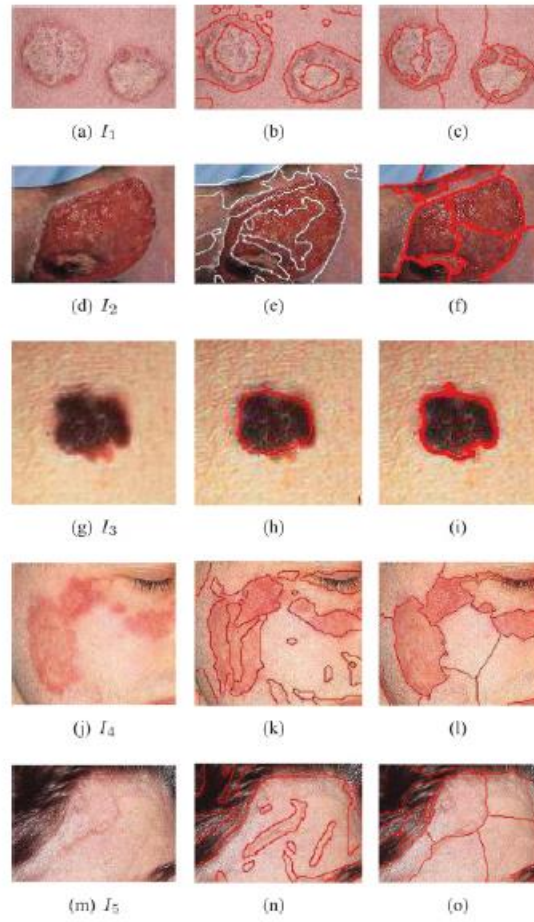


Figure 10: Segmentation on skin lesion medical images. From the first to the last image we have: psoriasis (image 1), cutaneous vasculitic ulcer (image 2), melanoma (image 3), lupus (image 4), and temporal arteritis (image 5). For each line, the original image together with the segmentation result of GeNCut and NCut are presented.

5.2 Natural and Human Image

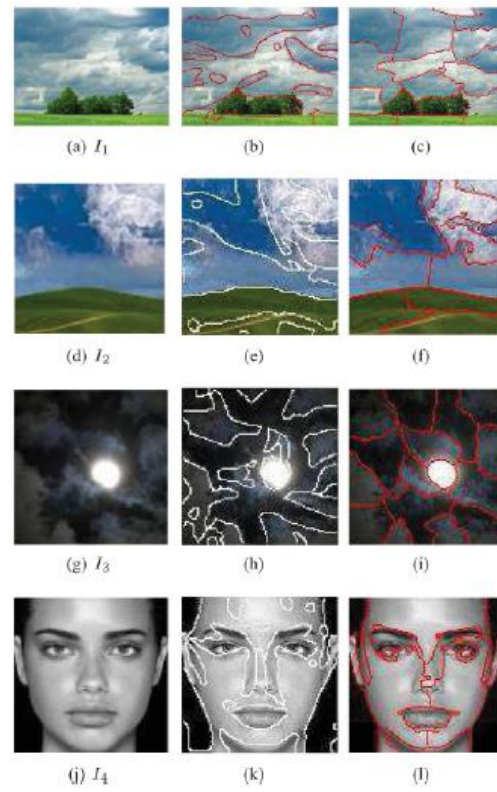


Figure 12: Segmentation of various kinds of natural and human images: pictures from nature, and human faces. From the first to the last image we have: landscapes (images 1–3), human face (image 4). For each line, the original image together with the segmentation result of GeNCut and Ncut are presented.

6. Reference

- [1] Shi, J., and Malik, J. (2000). Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8):888–905.
- [2] Wu, Z., and Leahy, R. (1993). An optimal graph theoretic approach to data clustering: Theory and applications to image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(11):1101–1113.
- [3] Park, Y., and Song, M. (1989). A genetic algorithm for clustering problems. In Proceedings of 3rd Annual Conference on Genetic Algorithms, pp. 2–9.