



HAL
open science

A General Framework for Complex Network-Based Image Segmentation

Youssef Mourchid, Mohammed El Hassouni, Hocine Cherifi

► **To cite this version:**

Youssef Mourchid, Mohammed El Hassouni, Hocine Cherifi. A General Framework for Complex Network-Based Image Segmentation. *Multimedia Tools and Applications*, 2019, 78 (14), pp.20191-20216. 10.1007/s11042-019-7304-2 . hal-02377145

HAL Id: hal-02377145

<https://hal.science/hal-02377145v1>

Submitted on 23 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A General Framework for Complex Network-Based Image Segmentation.

Youssef Mouchid · Mohammed El
Hassouni · Hocine Cherifi

Received: date / Accepted: date

Abstract With the recent advances in complex networks theory, graph-based techniques for image segmentation has attracted great attention recently. In order to segment the image into meaningful connected components, this paper proposes an image segmentation general framework using complex networks based community detection algorithms. If we consider regions as communities, using community detection algorithms directly can lead to an over-segmented image. To address this problem, we start by splitting the image into small regions using an initial segmentation. The obtained regions are used for building the complex network. To produce meaningful connected components and detect homogeneous communities, some combinations of color and texture based features are employed in order to quantify the regions similarities. To sum up, the network of regions is constructed adaptively to avoid many small regions in the image, and then, community detection algorithms are applied on the resulting adaptive similarity matrix to obtain the final segmented image. Experiments are conducted on Berkeley Segmentation Dataset and four of the most influential community detection algorithms are tested. Experimental results have shown that the proposed general framework increases the segmentation performances compared to some existing methods.

Keywords Complex networks · image segmentation · community detection

Youssef Mouchid
LRIT, URAC No 29, Faculty of Sciences, Mohammed V University in Rabat, B.P.1014 RP,
Rabat, Morocco.
E-mail: youssefmour@gmail.com

Mohammed El Hassouni
DESTEC, FLSHR, Mohammed V University in Rabat, LRIT, URAC No 29, Faculty of
Sciences, Mohammed V University in Rabat, B.P.1014 RP, Rabat, Morocco.

Hocine Cherifi
LE2I UMR 6306 CNRS, University of Burgundy, Dijon, France.

1 Introduction

Segmentation is an important task in image and video processing which plays an essential role to understand images and videos. The goal of image segmentation is to split the image into uniform and homogeneous regions with respect to some characteristic such as gray tone or texture. Image segmentation constitutes an important issue in pattern recognition due to its effect in several applications. Image segmentation application area varies from the identification of objects from remote sensing data [1] to the detection of cancerous cells. For example, it can be used to diagnosis medical imaging or to allow locating tumors and other pathologies [2] or to extract interest points for images for identifying the local features of an image such as content-based image retrieval or target recognition [32]. Its process is still a challenging problem because each image has a huge number of pixels, chiefly for images with high resolution, so the process of image segmentation is time-consuming. Several segmentation methods are presented in the literature, and there is no single method can be considered efficient for all images, nor are all methods equally good for a particular type of image. In recent years, the graph has emerged as a representation of the image. It models the image components into mathematically sound structures which makes the segmentation problem easier and the computation faster and efficient. The problem of the graph based image segmentation methods concerns the partition into several sub-graphs such that each sub-graph represents a meaningful object of interest in the image. The idea of using graphs as an approach for segmentation using a global cost function was brought by Wu et al. [3] in 1990s (A survey). From then on, the study of optimization techniques on the graph attracted much research attention.

Graphs are structured into communities which represent groups of nodes that are heavily connected among themselves, but sparsely connected with nodes in other groups [31]. Community identification can be considered as a data clustering problem. Taking into account the importance of community detection, it is not surprising that many community detection methods have been developed using techniques from different disciplines. Looking at the identification of objects in an image as a data clustering problem, community detection algorithms can be used for image segmentation. However, using them only has a practical limitation.

Motivated by graph-based methods and the application of community detection algorithms in graphs, we propose a general framework for image segmentation based on community detection algorithms. In many existing works, complex networks have been formulated as binary networks to simplify the analysis and the computation. Nevertheless, a binary network can simplify computation and analysis, but it lost some important information, which affects the network performance. Unlike these existing works, the proposed general framework uses a weighted network. Furthermore, it finds the differences between image segmentation and community detection using a texture feature which computes the number of gradient orientation occurrences in each localized part of the image and then encodes it into the image similarity matrix.

Adaptively, the similarity between regions of pixels is recalculated to avoid the over-segmentation and many small regions. The proposed algorithms detect automatically the region's number in an image compared the state of the art segmentation algorithms, and also produce sizable regions and achieve much better segmentation of the image. We summarize the proposed framework as follows: First, we apply an initial segmentation to split the image into homogeneous small regions for building a network of regions. Then, we use texture and color features for detecting meaningful objects in the image, and to encode much better the similarity between regions from the semantic point of view. After using the two features of both texture and color, we construct the similarity matrix adaptively to avoid small regions in the segmentation. At each iteration, we compute the similarity between two regions of the image, to avoid breaking visually coherent regions which have smooth changes in texture or color. Finally, efficient community detection algorithms are used.

The rest of the paper is organized as follows. Section 2 presents previous investigations related to the proposed general framework. The details of each step of the proposed framework are presented in section 3. We reported our experiments on the publicly BSDS500 (Berkeley Segmentation Data Set) in Section 4. Finally, in section 5, we present our conclusions.

2 Related work

Several graph-based techniques are used to segment an image [3–5], a lot of them are based on the optimization of a cost function. In [3], authors considered a minimum cut criterion, which seeks to partition a graph into a number of subgraphs by minimizing the sum of the edge weights.

In [4], authors proposed an approach derived from a pairwise region comparison to segment the image by defining a criterion to evaluate if an edge exists between two regions based on an iterative strategy and using an image representation based on the graph to obtain the final segmentation. Their proposed method is almost linear-time in the number of graph edges and can be employed to segment large images.

In [5], Shi and Malik proposed a technique, called normalized cut. The points to be clustered are represented by an undirected graph, where the nodes represent the points to be segmented and each edge weight represents the similarity between two points. The graph cut is measured by the weights of the total connection from vertices in a set A to all the vertices in the graph, the weight is computed by measuring a certain image quantity (e.g., color, intensity, etc.) between the two vertices connected by that edge. The Ncut measure tries to minimize the cut and to penalize partitions at the same time in which one set of nodes is only loosely connected to the graph at large.

In [6], Li et al propose an image segmentation algorithm based on graph modularity optimization. First, they start their algorithm by an initial segmentation using the superpixels method which oversegment the image into a set of small

segments. Each segment represents a region in the graph, and then the graph of regions is constructed using two features based on the LAB color, and the Histogram of state (HoS) to compute the similarity between regions, this similarity is assigned as a weight to the network. Finally, from the regions graph, they apply a community detection algorithm based on the modularity optimization. The algorithm process is repeated iteratively until no modularity increase in the community structure of the graph. However, authors use in their paper community detection algorithms that have a high computational cost and are not efficient compared to the latest published community detection algorithms.

In Abin et al [7], authors proposed the same approach as Li et al [6]. First, they start by an initial segmentation using meanshift and then they construct the network using the similarity between two regions which is computed from only color information. This similarity is used as weight to the edges of regions network. Finally, they apply community detection algorithm to obtain the segmented image. Nevertheless, authors didn't use an iterative process to avoid the over-segmentation problem, instead, they use a post-processing algorithm to merge regions with areas smaller than a predefined threshold with other regions. If a region area is smaller than the threshold t , it's merged to the most similar adjacent region in the network. Nevertheless, the proposed approach in Abin et al [7] also used a high computational cost community detection algorithm (Fast Newman) as in Li et al [6].

In [10], authors proposed also an algorithm based on community detection algorithms, they start first by an initial segmentation using superpixels method, and then, they consider pixels on the image as nodes on the graph. To construct the network, they use the CIELAB feature to compute the similarity between superpixels in the image. A connection between two nodes is considered only if the weight is smaller than a threshold t . Finally, when the graph of superpixels is constructed, they apply the fast greedy algorithm which is a hierarchical agglomerative algorithm for community detection, based on the maximization of the modularity measure. However, the proposed approach in [10] uses a binary network instead of a weighted network.

3 The Proposed Framework

Due to the inherent properties, segmentation and community detection problems are different. Using community detection algorithms only to segment an image by considering pixels as nodes on the graph, can lead to low performances. The failure of such method can be explained by several reasons. First, when we segment an image, pixels can have different properties, for example, different colors, but in community detection, nodes can share similar features. Second, we cannot take regularities and information for homogeneous segments from the image using just a single pixel. Third, compared with communities images share some information, as an example, two adjacent regions belong probably to the same community. So, to address the mentioned problems, our

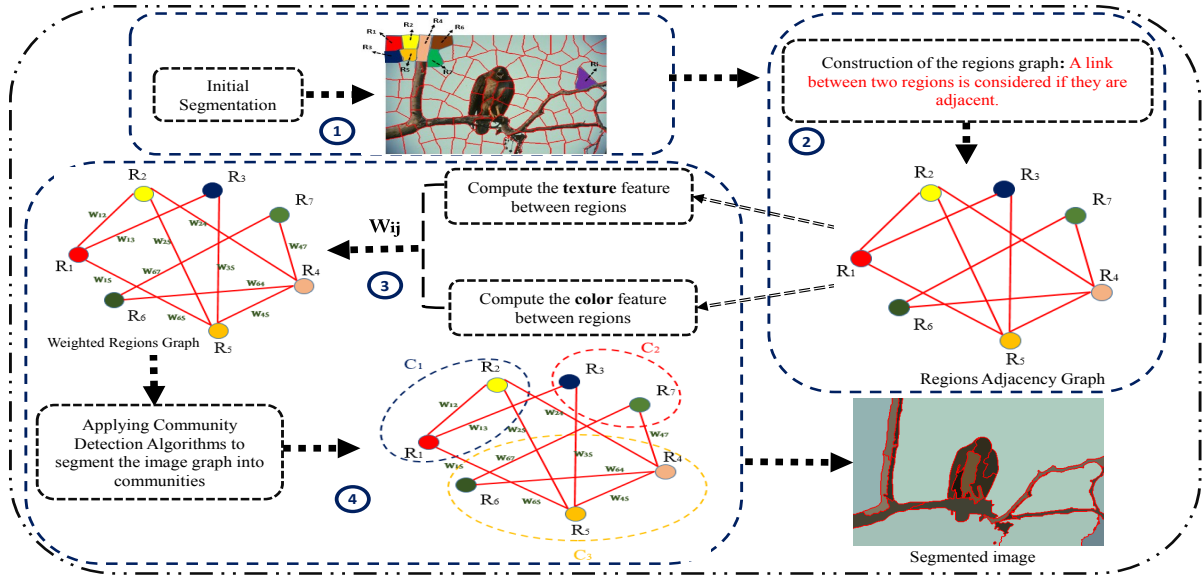


Fig. 1: Flow chart of the proposed framework for an iteration.

framework takes advantage of the inherent properties of the image and also the efficient optimization in modularity/stability using community detection algorithms. In this section, we describe our framework steps so as the reader will have a global picture of the entire framework before delving into the details. We refer to Fig.1 for the illustration of the steps of the proposed general framework. The details of each step and some technical points are explained in the next sections.

3.1 Initial Segmentation

The goal of the initial segmentation is to split the image into homogeneous, possibly small regions as shown in Fig.2. Several low-level segmentation methods can be used in this step, such as super-pixel, Meanshift, levelset, and watershed. The **super-pixels** segmentation as is used in Li et al [6] split the same perceptual region in a multitude of smaller regions. It's usually used as an initial segmentation process for reducing the pixels number and the computational complexity of subsequent tasks. Several studies exist for the extraction of super-pixels. In [8], authors propose an efficient technique that yields quasi-uniform super-pixels with low computational cost. The results obtained by the method show its efficiency in terms of computational costs and compactness of segments and over-segmentation errors. Authors use a connected K-means algorithm with convexity constraint for extracting super-pixels. According to

a number of regions desired by the user, the image is split into rectangular regions (segments) using a regular grid. Then, by minimizing a cost function, the algorithm tests pixels at the over segment boundaries and assigned them to the new segments. In [7], Abin et al used the **Meanshift** [9] algorithm, which can cluster a given set of data points. One advantage of Meanshift over other pre-segmentation techniques, is that we don't have to specify the number of segments (clusters) because the algorithm itself finds the best number of clusters for the image. The process of this algorithm follow these steps:

- Compute the joint histogram of the given RGB image: We used a number of clusters for each color to be 10. So there will be 1000 bin joint histogram, we can vary the number of bins according to the user. The Meanshift algorithm takes two inputs from the users. The first one is the color range and the second one is the spatial range. Color range defines pixels of what color to in the neighborhood are to be considered. Spatial range defines which pixels in what neighborhood are to be considered for computing the mean for the pixel of interest.
- Compute the mean: $M = \text{sum}(m(i) * p(i)) / \text{sum}(p(i))$, for example let us say that the pixel is initially falling in 5,5,5th bin in the joint histogram. and we give the color range the value 2, then the value of $m(i)$ varies from 3 to 7 for each color, $p(i)$ denotes the probability of a pixel belonging to that $m(i)$ th bin and we assign the pixel to the mean computed then we compute the mean again with the new neighborhood. Iterate this until the mean value converges. This is done by measuring the distance between the original bin to which the pixel belonged and the new bin to which it's assigned after every iteration.
- Repeat the same process for all the pixels in the image. Figure 2 shows the result of super-pixels and Meanshift segmentation in a typical image.

The reason for using the initial segmentation is because when we use directly community detection algorithms, we treat every single pixel as a community, which leads to a time-consuming detection since most community detection methods are computationally expensive. In addition, no reason exists to treat just one pixel which has no information about texture. So, starting with an initial segmentation approach provide very small regions of pixels which reduce greatly the computational costs and compactness without affecting the segmentation performance.

To achieve good segmentation results and to choose the appropriate initial segmentation, we run some experiments in a group of images. Results prove that the Meanshift approach is more appropriate than the super-pixels because it gives the best segmentation of the image and reduces the number of the small region. So in the next experiments, the Meanshift approach is chosen as an initial segmentation.

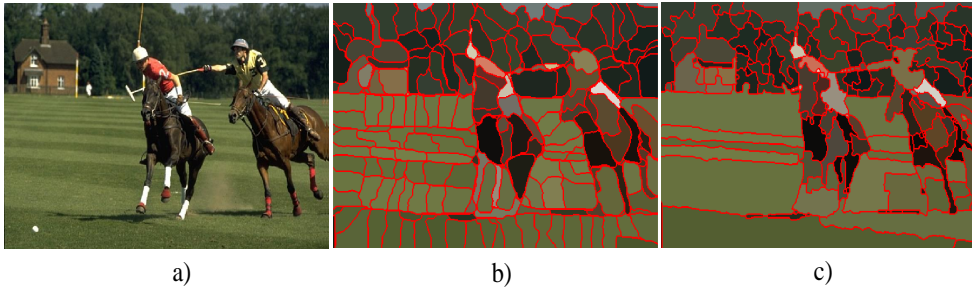


Fig. 2: Results of the initial segmentation: a) A typical color image; b) Super-pixels initial segmentation result; c) Meanshift initial segmentation result.

3.2 Regions graph construction

Unlike conventional networks, images contain spatial a priori information compared to social networks or citation networks. Adjacent regions in the image are often considered as a single image segment, than other regions which are far away. Hence, unlike community detection algorithms which consider two regions to be neighbors if the similarity weight between them is greater than zero, we constructed an image region graph using the spatial a priori information of images and by merging adjacent regions at each grouping process.

The construction is achieved by considering the adjacent regions of a current region as neighbors and the adjacency between regions if they share at least one pixel with the central region. As shown in Fig.3 the network of regions is built after the initial segmentation where each region in the image is considered as a vertex in the network. A connection between two regions is considered if they have at least one shared pixel.

3.3 Construction of the weighted graph by the similarity matrix

Unlike the proposed approach in [10] where authors built a graph by considering each node as a connected super-pixel according to a weight function. In this work, we use the initial segmentation not only for reducing nodes number in the built network, but also because information and regularities in the image are taken into consideration in regions. So, we merge adjacent regions in the image by considering the neighboring regions as neighbors for the current region if at least they share one pixel with the current region.

3.3.1 Computing the similarity using features

Color in segmentation is an important and straightforward feature. In the literature, various color spaces are proposed to capture different categories of the color, such as RGB, LAB, HSV, and YUV. Choosing an appropriate color

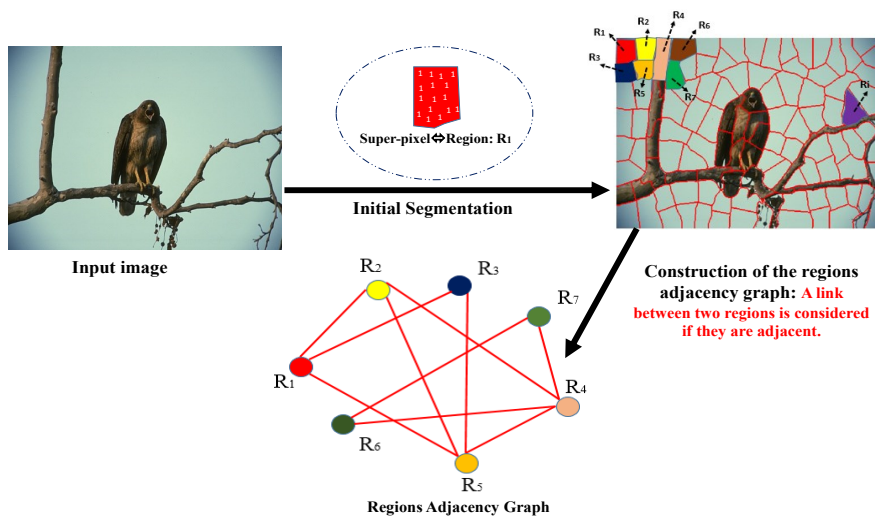


Fig. 3: The construction process of the regions adjacency graph from initially segmented images.

space to segment a color image is a crucial step in order to achieve a better segmentation performance. Thanks to its accordance with the human visual system [30], we choose LAB color space for the rest of this paper. It's a 3-axis color-opponent space with dimension L for lightness and A and B for the color opponent dimensions.

To compute the similarity between two regions, we calculate first the values of the image pixel in the L^*a^*b color space that are considered as features. Nevertheless, using only the color feature in the image cannot achieve a good segmentation result, because the color feature in some homogeneous object will decompose image regularities into different segments. Therefore, we propose a texture feature as a solution to remedy this problem. Many recent approaches use wavelet as features [11], other methods, such as [12], learn dictionaries of local structures from training images. In our work, we use a feature called Histogram of Oriented Gradients (HOG) which is well known in image processing and computer vision for detecting objects in the image. It computes the number of gradient orientation occurrences in localized parts or a segment of interest in the image detection window. To construct the Histogram of Oriented Gradients we proceed with the following steps:

- In the first step, the image is splited into small regions where for each region, the histogram of gradient directions is computed for the pixels into the region.
- In the second step, we represent each region as angular bins just as the gradient orientation where each region in the image contributes a weighted gradient to its corresponding angular bin.

- We consider groups of adjacent regions as spatial regions which can be called blocks. The block histogram is represented by a normalized group of histograms and the set of these all blocks represents the descriptor.

3.3.2 Measures for computing the similarity

Several measures can be used to compute the similarity for the proposed features. Because each pixel in the image is represented by a three-dimensional vector in the L^*a^*b color space, we measure the similarity between two regions, by considering each pixel in a region as a three-dimensional Gaussian distribution. Various measures to compute the distance between distributions are proposed in the literature such as:

Kullback-Leibler (KL) Divergence [13] which is an information criteria for measuring the difference between two probability distributions over the same random variable x . Its concept was originated in probability theory and information theory. We can use this measure to compute the distance between image regions. Nevertheless, it's inconvenient is when the two distribution $p(x)$ and $q(x)$ have different supports, then $q(x)$ can be 0 and the divergence measure becomes ill-posed.

The Earth Mover's Distance (EMD) [14] is a well-known method to evaluate if two multi-dimensional distributions of a region are dissimilar or not in different feature spaces which is called the ground distance. It's a very good metric for measuring the distance between two regions of the pixel value.

The Mean Distance (MD) is an approximation of the Earth Mover's distance, both are considered as a heuristic measures. Our proposed framework computes the distance between the two feature distributions of color for each two regions in the image, the Mean Distance is used for this computation because is faster than computing the EMD. To transform the color feature distribution distance to a similarity measure, we use a radius basis function of a Gaussian type which can be defined by:

$$c_{ij}(\text{color}) = \exp\left(\frac{-\text{dist}(R_i, R_j)}{2\sigma}\right) \quad (1)$$

Where $\text{dist}(R_i, R_j)$ is the distance between the pixel value distributions for region R_i and R_j .

To compute the Histogram of Oriented Gradients (HOG), we use a similarity function based on the cosine, for measuring the similarity between two regions. Each region can be represented as a 8 dimensional vector and for two regions R_i and R_j , $h_i, h_j \in R^8$ are the HOG feature vectors for region i and region j as defined by:

$$t_{ij}(\text{texture}) = \cos(h_i, h_j) = \frac{h_i^T h_j}{\|h_i\| \cdot \|h_j\|} \quad (2)$$

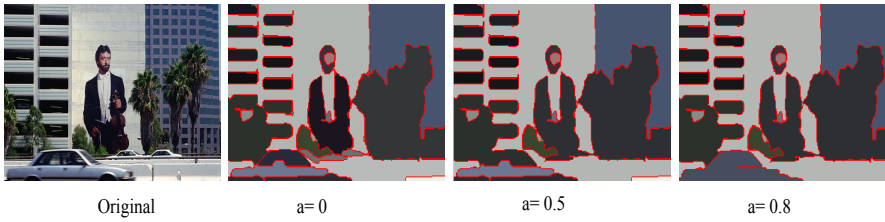


Fig. 4: Image segmentation with various values of the balancing parameter a by Fast multi-scale community detection algorithm using the criterion from Ronhovde and Nussinov (RN).

where h_i, h_j are respectively the histograms of the regions R_i and R_j .

3.3.3 Adaptive similarity matrix

To construct the similarity matrix W , an adaptive process is so proposed. We update at each iteration the adaptive similarity matrix. Using the equation (3) and (4), we recompute the similarity between each two regions again.

Because regions keep expanding when we use community detection algorithms. For this reason, an adaptive process is used to construct the similarity matrix, and to recalculate the similarity between regions at each iteration, because similarity measure given by the previous iteration is not good enough for the current iteration. This process avoids many regions in the image which should be merged on the same community according to the perspective of the human visual system.

We refer to [15] for the construction of the adaptive similarity matrix (weighted network) at each iteration. In [15], authors use a hybrid model that combine two features. In this work we choose the texture and the color features as defined in the equation below:

$$W = w_{ij} = a \times \sqrt{t_{ij}(\text{texture}) \times c_{ij}(\text{color})} + (1 - a) \times c_{ij}(\text{color}); \quad (3)$$

$$(i, j) = 1, \dots, n$$

Where n denotes regions number and a is a balancing parameter. If $a = 0$, it means that the texture information is not considered, if we increase the value of a , results show that stripe patterns are well encoded into the similarity, which means that the regularities and information in the image are better preserved as shown in Fig.4. Nevertheless, if the value a is too big, distinct objects in the image can be merged into one segment.

3.4 Extracting communities from the network

In this step, from the similarity matrix, we extract communities from the graph of regions using community detection algorithms. To find the best partition of the network that gives a maximum modularity or stability, several algorithms have been proposed in the literature. Newman [16] proposed an agglomerative hierarchical clustering approach to maximize the modularity. Vertices are grouped successively for forming larger communities, when modularity increases after the merging and the edges are not initially present, during the procedure, edges are added one by one. Unlike the proposed approaches in Li et al [6] and Abin et al [7] which use community detection algorithms that have a high computational cost, and does not always produces the best segmentation, such as Newman-Fast algorithm and Modularity optimization, our framework uses efficient community detection algorithms which strike the best balance between computational cost and segmentation performance.

In order to choose the appropriate community detection algorithms for the built network for communities extraction, synthesis papers are used to find and then to assess community detection algorithms. From [17], [18], [19], the algorithms proposed by Ronhovde and Nussinov [20], Infomap [21], Fast greedy modularity optimization algorithm [22] and Louvain [23] are judged to be able for delivering a reasonable estimator of the number of communities for different size of networks and then, outperforms all the state of the art algorithms for detecting communities. Here, we present a brief description of each algorithm.

3.4.1 Fast multi-scale community detection algorithm using the criterion from Ronhovde and Nussinov (RN)

The algorithm is an improvement of the algorithm in [20] which is based on the minimization of the Hamiltonian of a Potts-like spin model, where the spin state denotes the belonging of the node in the community. To enable one to span several community scales from very small to very large communities, we use a resolution parameter. To identify relevant scales, the algorithm checks for each given value of the resolution parameter, the stability of the obtained partitions. This is done when we compute for the same resolution parameter, the similarity of partitions obtained, but by starting from different initial conditions. Peaks in the similarity spectrum represent relevant partitions. The algorithm is rather fast and its computation complexity is slightly superlinear in the number of edges of the graph. We will refer to the method as RN in the next sections. The aim of our proposed framework is speed efficiency. To deal with that a greedy approach is used which exploits all the available information (i.e. input data and information computed as the algorithm runs).

3.4.2 Infomap

This algorithm is based on the same algorithm process of [24], which solves the problem of finding the best division structure of the network, by turn it into an optimally compressing problem of the information on the structure of the network, in order to recover the original structure when the compressed information is decoded. The method in [24] shows that, using the minimum of a function which gives the best tradeoff between both, the original and the compressed information and the maximal compression of the minimal conditional information. To optimize this function, we refer to a simulated annealing, that makes the algorithm quite slow, although one could always go for a faster and less accurate optimization. The difference between Infomap [21] and the proposed method in [24] is that before compressing the information on the network, Infomap compresses the information of a dynamic process taking place on the graph, namely, a random walk. Infomap optimizes a quality function to achieve the optimal compression, which denotes the minimum description length of the random walk. Such optimization can be achieved rather quickly by combining both greedy search and simulated annealing.

3.4.3 Fast greedy modularity optimization algorithm

The algorithm [22] represents the fast version of a previous method proposed by Newman. It starts from a set of nodes that are initially isolated and added edges between them to construct the original graph iteratively, by producing at each step the greatest possible increase of the modularity value of Newman and Girvan. First, the algorithm starts with a number of communities N , each community contains a single node, the communities are repeatedly grouped together iteratively at each step, by choosing the set that results in the largest increase (or smallest decrease) in modularity value. The algorithm runs far more quickly. For networks that have a hierarchical structure with communities at many scales and sparse networks, the algorithm has essentially linear running time. This is not only an advanced technique but it's a technique that has substantial practical implications, as it allows to study networks with a large number of nodes.

3.4.4 Louvain

The algorithm finds partitions of large graphs with high modularity value in short time and unfolds a complete hierarchical community structure for the graph [23]. It's divided into two steps which are iteratively repeated. First, the algorithm starts with a weighted graph that contains N nodes and we assign to each community of the network one node. For each node i , the algorithm considers the neighbors j of i , and evaluates the gain of modularity when i and j are grouped into the same community. The node i is then moved to the community for which this gain is maximum, but only if this gain is positive. If no positive gain is possible, the node i stays in its current community.

The process of grouping is applied iteratively for all nodes until no further maximization of the gain can be achieved. In the second step of the algorithm a new graph is constructed, whose nodes are now the communities found during the first step. Louvain offers a fair balance between the accuracy of the estimate of the modularity maximum and computational complexity. The output of the algorithm, therefore, provides several partitions. The partition found after the first step contains many communities that contain a small number of nodes. At subsequent steps, larger communities are found due to the iterative grouping mechanism. This process naturally leads to hierarchical decomposition of the graph.

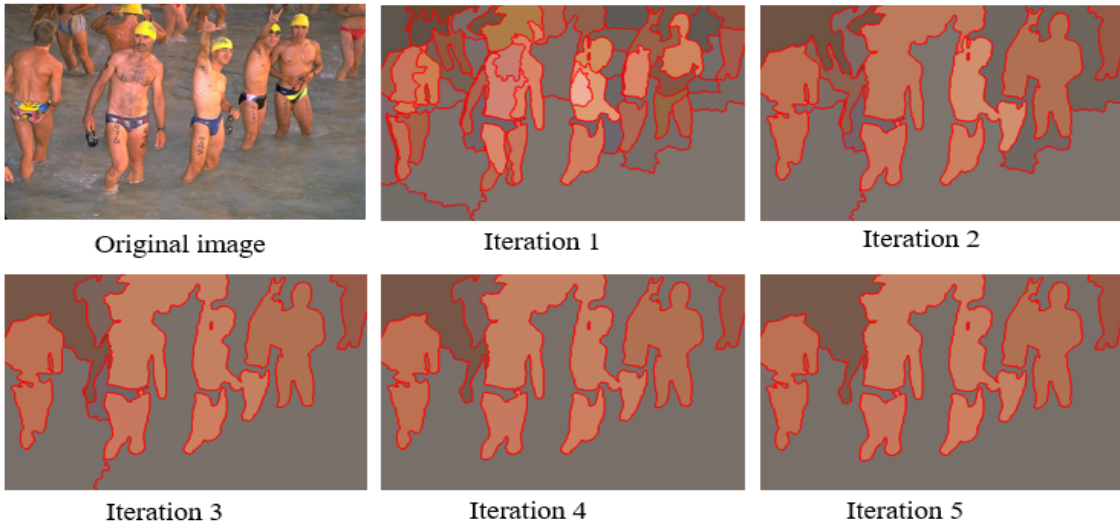


Fig. 5: Iterative region merging process for fast multi-scale community detection algorithm with the regions merging results at each iteration.

4 Experiments and results

We have discussed our proposed framework but so far not shown any results. For the sake of completeness and illustration, in this section, the performance of the proposed framework is assessed qualitatively as well as quantitatively by providing some experiments. The publicly available Berkeley Segmentation Data Set 500 (BSDS500) [25] was used to evaluate the performance of the proposed algorithms of our framework, BSDS500 contains 300 natural scene images and provides a different number of manual segmentation performed by different subjects for each of the 300 images of size 481×321 . The proposed general framework starts from an initially over-segmented image. To clearly

show the iterative process of the proposed framework, we give an example in figure 5 for a chosen image from BSDS500, in which there initial regions are merged iteratively at each iteration until no change in community structure is caused by merging any two communities in the image.

For our evaluation, we investigate for the quantitative evaluation the Probabilistic Rand Index (PRI) [26] and the Variation of Information (VOI)[27] which are a well-known evaluation metrics for segmentation. The PRI measures the probability that a segmentation and its ground truth have matched labels in the two partitions. The larger value is, the more the similarity between two segmentation are. The PRI range is in $[0,1]$.

VOI metric measures the sum of information gain and information loss between two segmentations. The VOI metric is nonnegative, the more is lower the more the similarity is greater. It's defined by the formula below:

$$\text{VOI}(C, C') = H(C) + H(C') - 2I(C, C') \quad (4)$$

where $H(C)$ and $H(C')$ denotes the entropy of the two segmentation C and C' respectively and $I(C, C')$ denote the mutual information of C and C' . The metric range is $[0, \infty]$, and the smaller the value is, the more similar the two segmentations are.

We also evaluate the performance of the proposed framework from two aspects: Precision and Recall. These two measures are attractive as measures of segmentation quality because they are sensitive to under and over-segmentation, under-segmentation leads to low recall scores, while over-segmentation leads to low precision scores.

The Precision measures the fraction of detected boundary pixels which match the ground-truth boundaries is defined as:

$$\text{Precision} = \frac{|S_{\text{test}}| \cap |S_{\text{gt}}|}{|S_{\text{test}}|} \quad (5)$$

where S_{gt} is the ground truth segmentation and S_{test} the testing segmentation and $|S|$ denotes the boundary pixels number in the segmentation S .

The Recall computes the percentage of ground-truth boundary pixels that are detected, is defined as:

$$\text{Recall} = \frac{|S_{\text{test}}| \cap |S_{\text{gt}}|}{|S_{\text{gt}}|} \quad (6)$$

F_α -measure is a quality measure based on Recall and Precision only, which measure the harmonic mean of the Precision and Recall, is defined as:

$$\text{F-measure} = \frac{\text{Precision} \cdot \text{Recall}}{(1 - \alpha) \cdot \text{Recall} + \alpha \cdot \text{Precision}} \quad (7)$$

For all our next experiments we set $\alpha = 0.5$.

Table 1: Quantitative comparison between different algorithms on **animals** category.

Algorithms	PRI (larger better)	VOI (smaller better)	Precision	Recall	F-measure	Running Time (second)
Fast multi-scale community detection [20]	0.911	1.520	0.897	0.789	0.839	4.324
Fast greedy modularity optimization [22]	0.858	1.585	0.868	0.681	0.763	8.583
Louvain [23]	0.847	1.610	0.857	0.670	0.752	5.322
Infomap [21]	0.588	2.402	0.598	0.546	0.570	2.161

4.1 Comparison between the proposed community detection algorithms

In this section, we compare the proposed community detection algorithms, to choose the best of them for the next comparison with the state of the art methods. First, all images in BSDS500 are classified into four categories, namely, people, urban scenery, animals and natural scenery. We take for each category five randomly chosen images with their ground truths as shown in figure 6. We run the proposed algorithms of our framework, to determine the best value of the parameter a , we vary a from 0 to 1 with 0.2 interval. Results show that $a = 0.4$ gives the best performance in term of both metrics PRI and VOI. In all next experiments, we use $a=0.4$.

In the first qualitative evaluations experiment, as shown in Fig.6, 7, 8 and 9, for each category Animals, People, Natural Scenery and Urban Scenery, we have applied the proposed algorithms of our framework with the adaptive similarity matrix. The results produce sizable regions and give much better results for all selected image for each category, for example, the human face in the people category, animals in natural scenery category, the castle in urban scenery category and mountains in natural scenery category. Using the HOG texture feature in the similarity matrix can successfully groups pixels image into the same segment and preserves the information, even if some of these pixels have different values into the same regions. As shown in figures and tables the Infomap algorithm doesn't always gives the best segmentation and underestimates the number of communities even though is faster than other community detection algorithms. In addition, we can observe from the figures that the Fast multi-scale community detection algorithm gives the best segmentation of the image.

We assess the performance of the proposed framework with the four segmentation techniques quantitatively. Since in BSDS500, there are multiple segmentation maps of the ground-truth, 5 segmentation maps for each image, in our experiment the mean value of the computed metrics is used between all the segmentation maps for each image and the segmentation result. In the first quantitative experiment, we show just the result of each category. Tables 1, 2, 3 and 4 present the average values of the PRI, VOI, Precision and Recall for each category. Again, the results of the proposed algorithms show their efficiency for the image segmentation task in term of PRI/VOI/Precision/Recall/F-measure.

Table 2: Quantitative comparison between different algorithms on **people** category.

Algorithms	PRI (larger better)	VOI (smaller better)	Precision	Recall	F-measure	Running Time (second)
Fast multi-scale community detection [20]	0.921	1.510	0.901	0.887	0.893	4.525
Fast greedy modularity optimization [22]	0.868	1.543	0.878	0.696	0.776	8.842
Louvain [23]	0.849	1.598	0.864	0.684	0.763	5.762
Infomap [21]	0.562	2.454	0.558	0.539	0.548	2.258

Table 3: Quantitative comparison between different algorithms on **natural scenery** category.

Algorithms	PRI (larger better)	VOI (smaller better)	Precision	Recall	F-measure	Running Time (second)
Fast multi-scale community detection [20]	0.887	1.610	0.891	0.837	0.863	4.897
Fast greedy modularity optimization [22]	0.849	1.643	0.862	0.674	0.756	8.984
Louvain [23]	0.828	1.648	0.856	0.671	0.754	5.954
Infomap [21]	0.577	2.421	0.579	0.564	0.571	2.742

Table 4: Quantitative comparison between different algorithms on **urban scenery** category.

Algorithms	PRI (larger better)	VOI (smaller better)	Precision	Recall	F-measure	Running Time (second)
Fast multi-scale community detection [20]	0.887	1.610	0.891	0.837	0.863	4.624
Fast greedy modularity optimization [22]	0.789	1.697	0.782	0.621	0.692	8.725
Louvain [23]	0.783	1.698	0.779	0.612	0.685	5.689
Infomap [21]	0.574	2.419	0.585	0.556	0.570	2.521

4.2 Comparison of the proposed framework with related works

First, we have performed a qualitative comparison of our framework with some well-known state of the art segmentation methods: Li et al [6], Abin et al [7], Lossy Compression (LC) [28] and EDISON [29].

In Li et al [6] and Abin et al [7], we preserve the same parameters used by authors. In EDISON [29] method which is based on the mean shift implementation in both boundaries extraction and noise filtering scheme, the main parameters of EDISON is the minimal region size. So, we set the parameter value to 1000, to avoid the creation of small regions. In Lossy Compression (LC) [28] we use the Gaussian Mixture Model to fit the image textures, and for finding the optimal segmentation we employ the principle of Minimum Description Length, that produces the minimum coding length under a certain distortion ratio. We use the distortion rate $\epsilon = 0.2$.

We choose Fast multi-scale community detection algorithm for our framework for the comparison because it gives the best image segmentation as shown in the previous section. As shown in Fig.10, LC, EDISON shows the different extent of over-segmentation by resulting many small regions, and also by

Table 5: Quantitative comparison between different algorithms on all Berkeley dataset images.

Algorithms	PRI (larger better)	VOI (smaller better)	Precision	Recall	F-measure
Humain	0.870	1.160	0.910	0.720	0.797
EDISON [29]	0.786	2.002	0.728	0.524	0.609
LC [28]	0.735	1.978	0.700	0.531	0.603
Abin et al [7]	0.813	1.721	0.764	0.615	0.681
Li et al [6]	0.777	1.879	0.733	0.508	0.600
Fast multi-scale community detection [20]	0.828	1.695	0.788	0.621	0.694

Table 6: COMPUTATIONAL TIME OBTAINED IN THE SEGMENTATION OF ALGORITHMS (IN UNIT OF SECONDS).

Algorithms	Computational Times				
	Superpixels	MeanShift	Graph	Community Detection Algorithm	Total
Abin et al [7]	-	0.239	0.128	11.215	11.482
Li et al [6]	3.299	-	0.231	16,870	20.40
Fast multi-scale community detection [20]	-	0.239	0.178	4.026	4.443

breaking information and regularities in some homogeneous regions of the image, compared to our proposed framework, which preserves information and regularities in the segmented image and produces sizable homogeneous regions, also our framework has the best performance as Li et al [6] and Abin et al [7]. Results indicate the superiority of our proposed framework over other methods.

In the second quantitative experiment which is applied to all of the 100 images in the Berkeley segmentation dataset. We compute the average values of the PRI, VOI, Precision and Recall for all images. As shown in Table 5 the proposed algorithms of our framework give a high value and better results for the image segmentation task, compared to all well-known segmentation algorithms EDISON, LC, Li et al [6] and Abin et al [7] in term of PRI/VOI, and also have a close performance to human visual perception with PRI=0.828 and VOI=1.695. Also, the Precision, Recall and F-measure of our proposed framework with Fast multi-scale community detection algorithm obtain the highest values with Precision=0.788, Recall=0.621 and F-measure=0.694 compared to the other algorithms which indicate that most of our segmentation has consistent labels with the ground-truth segmentation in BSDS500. As result, we can say that the proposed framework achieves better performance in terms of Precision, Recall, and F-measure compared with other state of the art algorithms.

4.3 Computational Time

We compare the running time between our proposed framework with Fast multi-scale community detection algorithm and Li et al [6] and Abin et al [7]. All algorithms are implemented in Matlab and are carried out on 4 GB of

RAM and a 2.60 GHz processor, each algorithm is tested over 100 validation images of Berkeley dataset, and then for each step (Initial segmentation, Graph generation, and Community detection) we compute the mean running time. It can be observed in Table. 6 that our algorithm runs consistently faster than Li et al [6] and Abin et al [7], specifically, about 2.5 times faster than Abin et al [7], and 5 times faster than Li et al [6]. As a conclusion, we can say that the proposed framework yields better results with shorter processing time than other algorithms of the state of the art.

5 Conclusion

This paper proposed a framework for image segmentation which takes advantages of the inherent properties of images and the optimization of modularity/stability. Efficient community detection algorithms are used to optimize modularity/stability, as a Fast multi-scale community detection algorithm using the criterion from Ronhovde and Nussinov (RN), Fast greedy modularity optimization and Louvain. All these algorithms can detect automatically the number of the region in the image. By using both, Histogram of Oriented Gradients (HOG) texture feature and color feature, the similarity matrix is constructed adaptively between different regions by optimizing the modularity/stability and merge adjacent regions iteratively. If no change occurs in modularity/stability when we merge any adjacent regions, the optimal segmentation is achieved. Our experiments have shown that our proposed framework gives a best qualitative segmentation result, as proved in the figures and achieve the best performance quantitatively compared to all state of the art methods in terms of PRI, VOI, Precision, and Recall. Since, our general framework based on three efficient community detection algorithms, it avoids the problem of having many small regions in the image and preserves information and regularities in the object. In addition, it provides a good time complexity and runs consistently faster than the state of the art algorithms.

References

1. Chen, Xiaohui, et al. "Image segmentation using a unified Markov random field model." *IET Image Processing* (2017).
2. J. A. Noble and D. Boukerroui, "Ultrasound image segmentation: A survey," *IEEE Transactions on medical imaging*, 25(8), 987-1010.
3. Wu, Zhenyu, and Richard Leahy. "An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation." *IEEE transactions on pattern analysis and machine intelligence* 15.11 (1993): 1101-1113.
4. Felzenszwalb, Pedro F., and Daniel P. Huttenlocher. "Efficient graph-based image segmentation." *International journal of computer vision* 59.2 (2004): 167-181.
5. Shi, Jianbo, and Jitendra Malik. "Normalized cuts and image segmentation." *IEEE Transactions on pattern analysis and machine intelligence* 22.8 (2000): 888-905.
6. Li, Shijie, and Dapeng Oliver Wu. "Modularity-based image segmentation." *IEEE Transactions on Circuits and Systems for Video Technology* 25.4 (2015): 570-581.

7. Abin, Ahmad Ali, Farzane Mahdisoltani, and Hamid Beigy. "WISECODE: wise image segmentation based on community detection." *The Imaging Science Journal* 62.6 (2014): 327-336.
8. Cigla, C., and Alatan, A. A. (2010, September). Efficient graph-based image segmentation via speeded-up turbo pixels. In *Image Processing (ICIP), 2010 17th IEEE International Conference on* (pp. 3013-3016). IEEE.
9. D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1271-1283, 2010.
10. Linares, Oscar AC, et al. "Segmentation of large images based on super-pixels and community detection in graphs." *arXiv preprint arXiv:1612.03705* (2016).
11. Viola, Paul, and Michael J. Jones. "Robust real-time face detection." *International journal of computer vision* 57.2 (2004): 137-154.
12. Bernstein, Elliot Joel, and Yali Amit. "Part-based statistical models for object classification and detection." *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 2. IEEE, 2005.
13. Kullback, Solomon. *Information theory and statistics*. Courier Corporation, 1997.
14. Puzicha, Jan, et al. "Empirical evaluation of dissimilarity measures for color and texture." *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*. Vol. 2. IEEE, 1999.
15. Sumengen, Baris, Luca Bertelli, and B. S. Manjunath. "Fast and adaptive pairwise similarities for graph cuts-based image segmentation." *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*. IEEE, 2006.
16. Girvan, Michelle, and Mark EJ Newman. "Community structure in social and biological networks." *Proceedings of the national academy of sciences* 99.12 (2002): 7821-7826.
17. Orman, G. K., Labatut, V., and Cherifi, H. (2011, June). Qualitative comparison of community detection algorithms. In *International conference on digital information and communication technology and its applications* (pp. 265-279). Springer Berlin Heidelberg.
18. Orman, G. K., Labatut, V., and Cherifi, H. (2011). On accuracy of community structure discovery algorithms. *arXiv preprint arXiv:1112.4134*.
19. Lancichinetti, A., and Fortunato, S. (2009). Community detection algorithms: a comparative analysis. *Physical review E*, 80(5), 056117.
20. Ronhovde, P., and Nussinov, Z. (2010). Local resolution-limit-free Potts model for community detection. *Physical Review E*, 81(4), 046114.
21. Rosvall, M., and Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4), 1118-1123.
22. Clauset, A., Newman, M. E., and Moore, C. (2004). Finding community structure in very large networks. *Physical review E*, 70(6), 066111.
23. Blondel, V. D., Guillaume, J. L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10), P10008.
24. Rosvall, M., and Bergstrom, C. T. (2007). An information-theoretic framework for resolving community structure in complex networks. *Proceedings of the National Academy of Sciences*, 104(18), 7327-7331.
25. P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898-916, 2011.
26. R. Unnikrishnan, C. Pantofaru, and M. Hebert, "Toward objective evaluation of image segmentation algorithms," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 6, pp. 929-944, 2007
27. Meila, M. (2005, August). Comparing clusterings: an axiomatic view. In *Proceedings of the 22nd international conference on Machine learning* (pp. 577-584). ACM.
28. A. Yang, J. Wright, Y. Ma, and S. Sastry, "Unsupervised segmentation of natural images via lossy data compression," *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 212-225, 2008.
29. C. M. Christoudias, B. Georgescu, and P. Meer, "Synergism in low level vision," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 4. IEEE, 2002, pp. 150-155.

-
30. Wright, W. D. "Color science, concepts and methods. Quantitative data and formulas." *Physics Bulletin* 18.10 (1967): 353.
 31. Newman, Mark. *Networks: an introduction*. Oxford university press, 2010.
 32. Zou, Qingyu, and Jing Bai. "Interest Points Detection in Image Based on Topology Features of Multi-level Complex Networks." *Wireless Personal Communications*: 1-11.

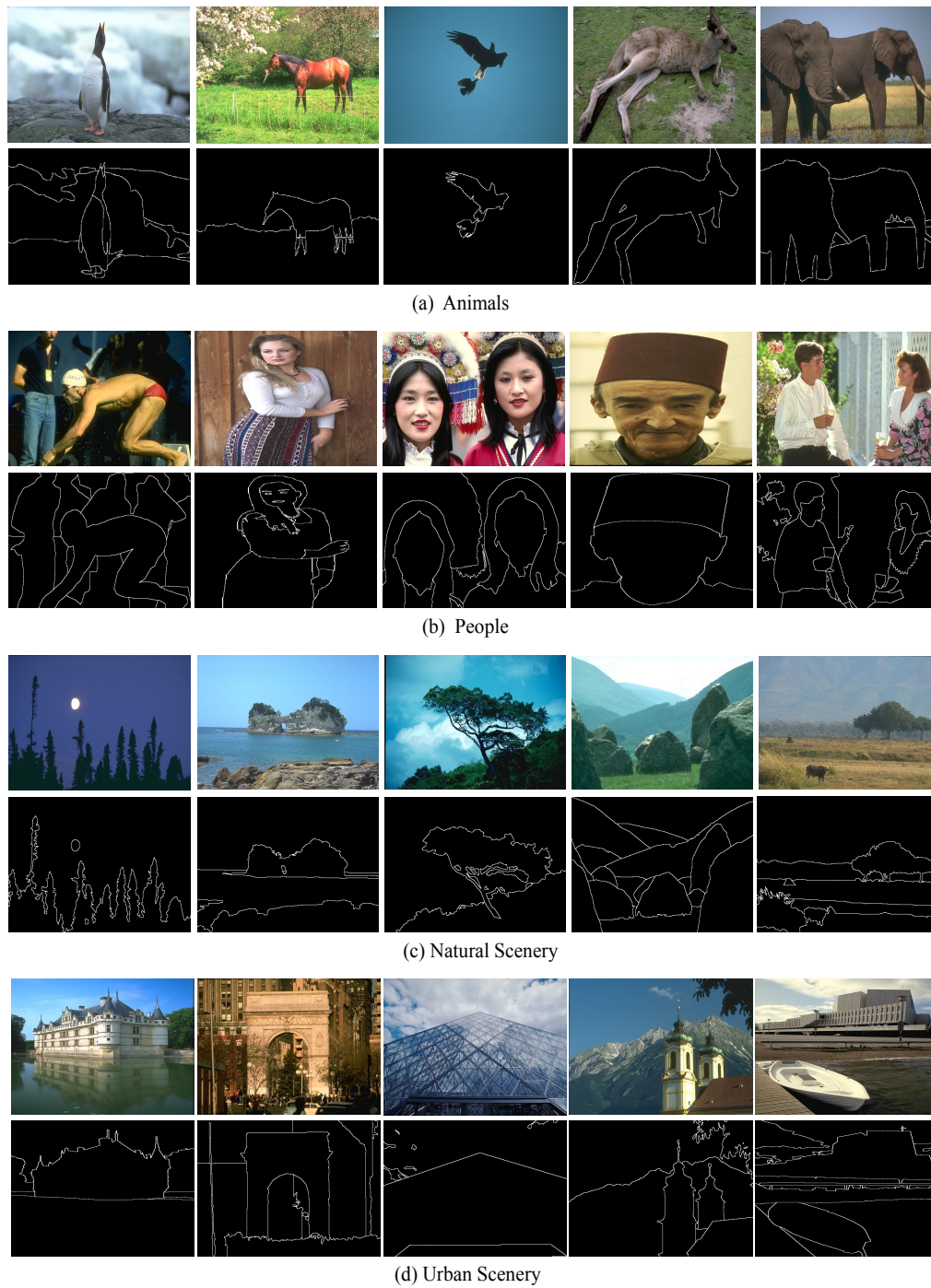


Fig. 6: BSDS500 images for different categories. For each category, Line 1: Original images. Line 2: Ground truths segmentation.

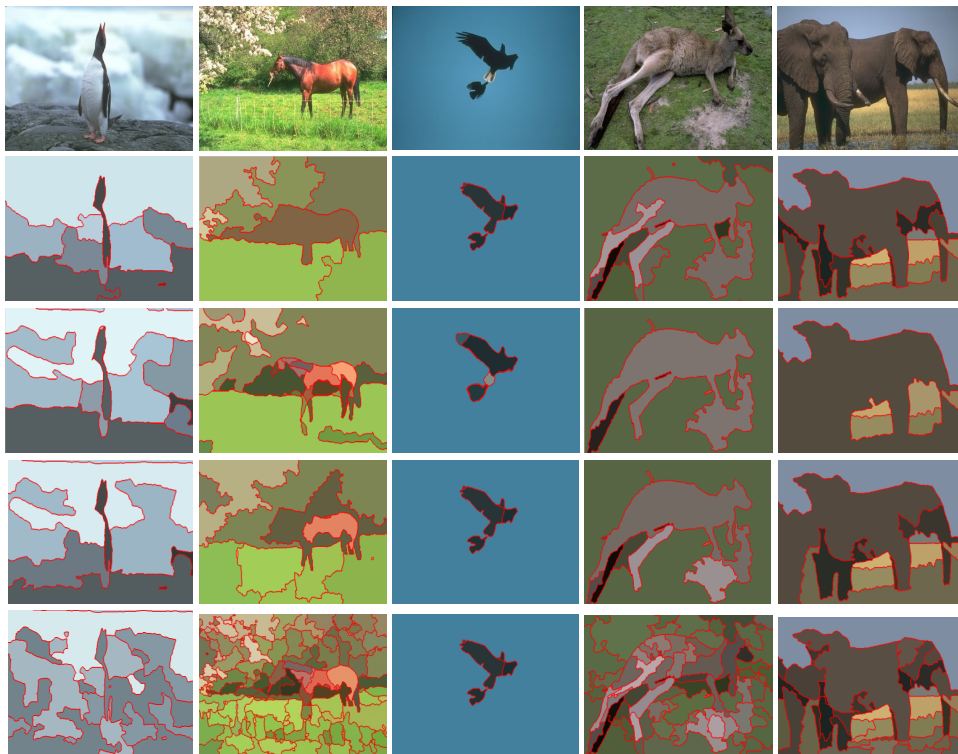


Fig. 7: Segmentation results for images from **animals** category, Line 1:Original image; Line 2: Fast multi-scale community detection algorithm; Line 3: Fast greedy modularity optimization algorithm; Line 4: Louvain; Line 5: Infomap.



Fig. 8: Segmentation results for images from **people** category, Line 1:Original image; Line 2: Fast multi-scale community detection algorithm; Line 3: Fast greedy modularity optimization algorithm; Line 4: Louvain; Line 5: Infomap.

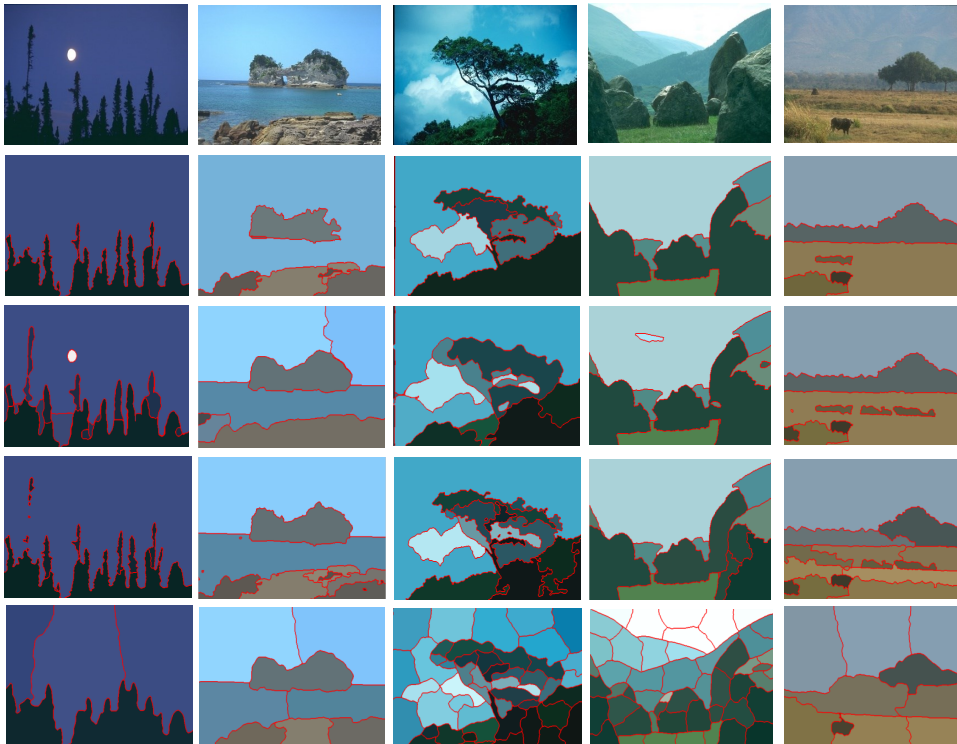


Fig. 9: Segmentation results for images from **natural scenery** category, Line 1:Original image; Line 2: Fast multi-scale community detection algorithm; Line 3: Fast greedy modularity optimization algorithm; Line 4: Louvain; Line 5: Infomap.

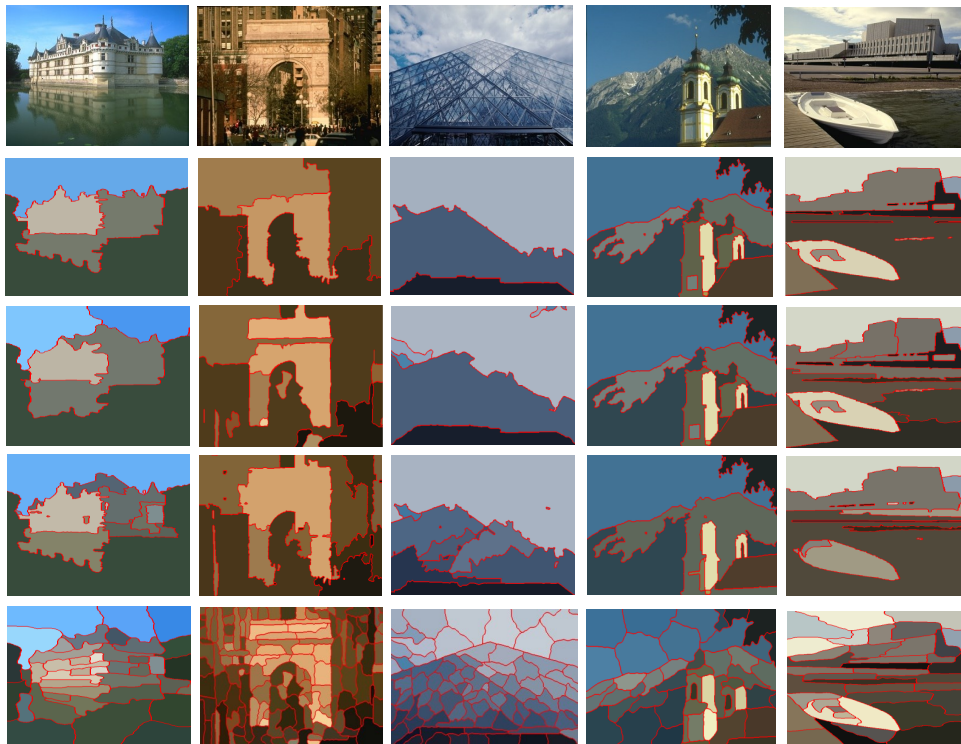


Fig. 10: Segmentation results for images from **urban scenery** category, Line 1:Original image; Line 2: Fast multi-scale community detection algorithm; Line 3: Fast greedy modularity optimization algorithm; Line 4: Louvain; Line 5:Infomap.



Fig. 11: Comparison of segmentation results of all algorithms, Line 1:Original image; Line 2:EDISON; Line 3: LC; Line 4: Abin et al [7]; Line 5:Li et al [6]; Line 6: Fast multi-scale community detection algorithm.