Enhanced Graph-Based Method in Spectral Partitioning Segmentation using Homogenous Optimum Cut Algorithm with Boundary Segmentation

S. Syed Ibrahim [1†] and Dr. G. Ravi [2††]
syedibs@gmail.com ravi_govindaraman@yahoo.com
[1] Research Scholar, [2] Associate Professor and Head, Department of Computer Science
Jamal Mohamed College (Autonomous) (Affiliated to Bharathidasan University)
Tiruchirappalli, Tamil Nadu, India

Abstract
Image segmentation is a very crucial step in effective digital image processing. In the past decade, several research contributions were given related to this field. However, a general segmentation algorithm suitable for various applications is still challenging. Among several image segmentation approaches, graph-based approach has gained popularity due to its basic ability which reflects global image properties. This paper proposes a methodology to partition the image with its pixel, region and texture along with its intensity. To make segmentation faster in large images, it is processed in parallel among several CPUs. A way to achieve this is to split images into tiles that are independently processed. However, regions overlapping the tile border are split or lost when the minimum size requirements of the segmentation algorithm are not met. Here the contributions are made to segment the image on the basis of its pixel using min-cut/max-flow algorithm along with edge-based segmentation of the image. To segment on the basis of the region using a homogenous optimum cut algorithm with boundary segmentation. On the basis of texture, the object type using spectral partitioning technique is identified which also minimizes the graph cut value.

Keywords: *Image segmentation, graph-based approach, pixel, region and texture, boundary segmentation, graph cut value.*

1. Introduction

Image segmentation is processing of simply partitioning images into relevant regions, but the most challenging is to accurately describe the spatial extent of few objects which is the combination of several regions. Generally, each image is represented as a set of pixels that are partitioned based on identical features like intensity, color, texture and so on [1]. The major objective of segmentation lies in changing the image representation into more meaningful which makes analysis very easy. This process normally locates objects and detects boundaries like points, lines, arcs, curves and so on in the input image.

After detecting the boundaries, based on their similarity, regions are separated [2].

Manuscript received July 5, 2023
Manuscript revised July 20, 2023
https://doi.org/10.22937/IJCSNS.2023.23.7.8

The graph-based segmentation approach is one among several schemes with remarkable features like flexibility and efficient computation. Moreover, it is efficient and fast in generating image segments [3]. Several types of research have been carried out on the graph-based approach. The benefit of this approach is to reuse the existing theorems and procedures described for other image processing approaches. This method selects edges from a graph, in which every pixel is a node. The dissimilarity among pixels is measured based on edge weights. Intensity differences between adjacent pixels and across the boundary within every region are compared and then the boundaries between the regions are defined.

The following tasks are performed:

a. Object recognition – functions like segmenting regions, locating points and lines, verifying the results and matching the object model and image.

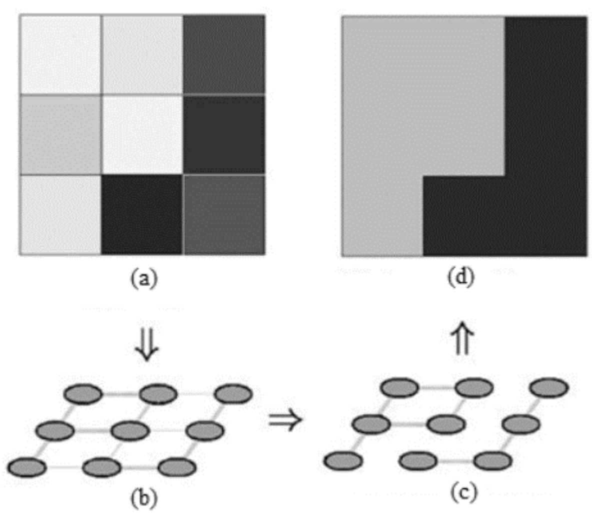
b. Object delineation – pixel classification, region growing from internal and external seeds, tracking optimal boundary, an image-graph cut etc.

Rather than computers, object models and human recognizes objects in a better way, but this is not the case for object delineation. A point is chosen inside the object easily by humans but manually tracing the same boundary many times is difficult. Most approaches are effective and interactive as the best model for object recognition and object delineation are integrated [4] which helps in automatic segmentation.

It is noteworthy that graph-based approaches present various representations of graphs with nodes as pixels, pixel vertices, regions, or user-drawn markers. Graph methods employed for solving the problem include graph matching, Prim's or Kruskal's algorithm, Dijkstra's algorithm, Random walker, etc. Moreover, a single algorithm, say Dijkstra's algorithm, are involved in the region as well as boundary-based segmentation, in addition to other operators. Hence, graph-based segmentation differs from other approaches in terms of graph model and the method used for segmentation [5].

Segmenting images results in a set of homogeneous image regions such that every pixel in a region is connected. When all the regions are combined, the entire image is obtained. Every region is a set of pixels where every pixel is identified by its location and feature vector. This graph-based approach works under the principle of partitioning a graph. Every approach considers an image as a graph G where vertices are represented as pixels. Generally, every edge is assigned a weight depending on the vertices it is related to. Each vertex of every sub-graph provides a complete set of vertices of G [6]. The association between graph partitioning and image segmentation is pictorially represented in Figure 1.

Figure 1. Association between image segmentation and graph partitioning.



(a) Image (b) Graph (c) Graph partition (d) Segmented image

Partitioning in a better way is still a challenging as it is subjective in nature. Consider that this is achieved optically by defining the criteria. Methods based on this approach are extensively examined in image analyzing and processing. Graph-based segmentation approaches are categorized as graph cut, interactive, minimum spanning tree and pyramid based approaches [7].

The remaining sections of the paper are presented as: The following section discusses few of the works related to this research. Section 3 elaborates on the proposed method. Section 4 presents the results obtained from the graph-based image representation. Section 5 briefs the achievements of this paper.

2. Existing Works

Aslanzadeh R et al. implemented a novel algorithm for segmenting images that are inspired naturally by human groups and are spread through a topographic plane. This algorithm has 4 steps: 1) Watershed algorithm: an edge detection method is used for determining primitive image segments. 2) Co-evolution: primitive segments make relations together based on their relation weights which point to similarity and proximity, then primitive segments which have high relation weights join together and form the primitive tribes. 3) Immigration-Deportation: primitive segments that have not become a member of a tribe; immigrate to the fittest tribe using an iterative process. 4) Emerging process: the resulted tribes which are similar highly combined together. These steps are independent of each other; therefore, each step could be changed autonomously to increase the overall performance of the algorithm for a specific application. Also, other features could be used instead of proposed features, for improving segmentation precision for specific applications [8].

Boykov Y et al. developed and compared the efficacy of the min-cut/max-flow method experimentally by considering the execution time. Goldberg-Tarjan style “push-relabel” approach and Ford-Fulkerson style “augmenting paths” based algorithm were examined. For several scenarios, the algorithm developed was faster than other methods [9].

Cui Y et al. designed a dynamic programming model to generate optimal constrained one-stage homogenous strip cutting pattern (OSHSCP) in which for every item, the maximum demand is stated. The aim is to optimize the total value of items that can be involved in a pattern. From the results, it is observed that the computational time was reasonable. OSHSCP comprises parallel homogenous strips of equal length.

This approach was beneficial as it can be independently utilized or used as elements to construct other cut patterns [10].

Wang S et al. introduced a cost function cut ratio to segment images based on graph-based approaches. The Cut ratio is the ratio of the total of two various edge weights on the cut boundary which models the mean affinity among segments partitioned. This cost function segments the image, ensures that there is a connection between partitioned segments, and there exist no unfairness in shape, size, boundary length or smoothness. Furthermore, performs iterated pixel as well as region-based segmentation efficiently. Obtaining a minimal cut ratio is difficult in an arbitrary graph, with connected planar graphs it is easier [11].

Wang X et al. designed a graph-cut segmentation approach in which an affinity graph was constructed using sparse representation. Initially, segmented images were computed which can be associated with every segment. Sparse representation of every feature set is determined by solving a minimization problem. The connection details among superpixels are determined as non-zero coefficient representation, and the affinity of these pixels is estimated as the relative error of representation. As a result, an affinity graph is provided that has beneficiary properties and a graph cut provides segmented images [12].

Manjula K. A. tried to alter or simplify image representation with meaningful and made analyzing easier as efficient edge detection is more important in segmentation. Often, segmentation is a complex task and segmenting the region of interest in real images is more burden in implementing robust image processing applications whose success lies in determining image segments. As detecting edges is a basic task in segmentation, the user must carefully select the appropriate edge detection technique depending on the application [13].

Ma W. Y et al. introduced a boundary detecting approach on the basis of "edge flow". A Predictive coding model is utilized for identifying the change in texture and color of every image and generated an edge flow vector. Boundaries of the image are detected by propagating the edge flow iteratively, in which two opposite directions of flow were encountered in the stable state. A user-defined scale for an image is the only control parameter required. Moreover, color and texture are integrated into a single framework for the detection of boundaries [14].

Estrada F. J et al. discovered the usage of random walks and the spectral embedding approaches related to automatically generate the required regions appropriately. Initially, a mathematical association between anisotropic image smoothing kernels and spectral embedding is derived. Then, these properties are used to select several sources and sink region pairs for min-cut. Typically, over-segmentation is introduced, and hence regions are merged to produce final image segmentation. This process was demonstrated with multiple sample images [15].

Yuan J et al. investigated and comparatively examined graph based min-cut/max-flow approaches and designed a new max-flow approach in the spatial continual setting, with or without supervised constraints described previously. The continuous max-flow representation corresponding to their min-cut representation are dual and primal issues. Hence, from max-flow/min-cut, primary concepts and terminologies are revised. It proves that the related non-convex partitioning issues, supervised or unsupervised, are globally and precisely solved through convex continuous max-flow and min-cut models. Furthermore, introduced a fast max-flow-based approach whose convergence was ensured by the typical theories of optimization **[16].**

3. Research Methodology

This section discusses about the proposed technique whose framework is depicted in figure 2.

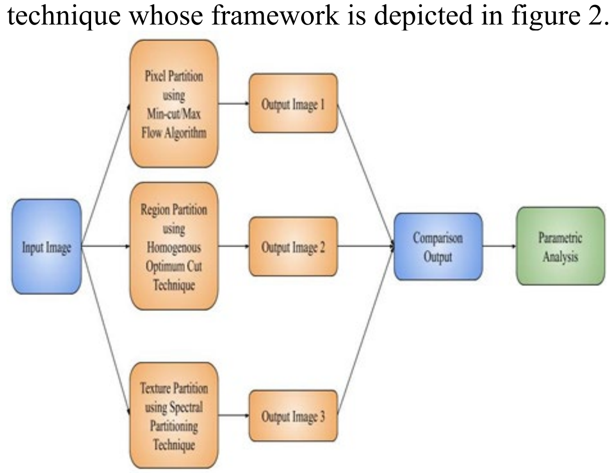


Figure 2: Framework of the Proposed Methodology

The input image has been taken initially. Then the image has been partitioned based on pixel, region and texture. The pixel-based partition has been done utilizing the min-cut/max flow technique along with edge-based segmentation. During the pixel partition, the input image has been segmented on the basis of edge. The region-based partition has been done using the homogenous optimum cut algorithm with boundary segmentation. While the region is partitioned, the boundary of the image has been segmented. The texture partition has been done using the spectral partitioning technique which also minimizes the graph cut value. When the texture has been segmented, the graph cut value of the image has been reduced. Then finally the comparison output is given and the parametric analysis has been done.

Assume an undirected graph G = (V, E) where vertices $v_i \in V$ and edges $(v_i; v_j) \in E$ which connects a pair of adjacent vertices. Each edge is assigned a non-negative weight w. The vertices V are pixels and weights are measures of dissimilarity between two vertices connected by the edge. In the graph-based method, V is partitioned into S segments where every component C ∈ S is connected in G' = (V, E'), where E^'⊆E. There exist several ways for measuring segmentation but generally, the elements are similar in a component and dissimilar in different components i.e., edges in the same component must be of low weights and that of different components must be higher.

These definitions. In the graph-based approach, a segmentation S is a partition of V into components such that each component (or region) C ∈ S corresponds to a connected component in a graph G' = (V, E'), where E^' ⊆ E. In other words, any segmentation is induced by a subset of the edges in E. There are different ways to measure the quality of segmentation but in general we want the elements in a component to be similar, and elements in different components to be dissimilar. This means that edges between two vertices in the same component should have relatively low weights, and edges between vertices in different components should have higher weights.

Pixel segmentation using min-cut/max-flow algorithm with edge-based segmentation

Here min-cut/max-flow approach is discussed which provides prominent image segments with appropriate source and sink regions. Two regions of the image are given. Pixels of one region are associated with the sink node, while that of the other region to the source. Each pixel is connected to its adjacent one with weighted edges. The weights assigned to the edges that join the pixels of source and sink regions must be large than the total weights of every other edge in the graph and hence never cut. A min-cut/max-flow approach that separates the source and sink is calculated in which the cost of any cut is the total weights of every edge that is cut. In this approach, the most promising property is that the cut is ensured to be a global minimum. Furthermore, even for large images, recent min-cut /max-flow algorithms are computationally feasible.

Two search trees at the source and sink, namely S and T, are non-overlapping and maintained with roots s and t respectively. In S, every edge from the parent to children is non-saturated, whereas edges from children to parent are non-saturated in T. The nodes which are not available either in S or T are known as "free" then

$$S \subset v, s \in S, T \subset v, t \in T, S \cap T = \phi \quad [1]$$

The nodes in S and T is either "active" or "passive". The former ones denote the outer border in every tree and allow new children to be added to the tree and thus tree grows. The passive nodes are internal and thus no growth occurs as blocked completely by the other nodes of the same tree. Active nodes can contact the nodes of other trees. When an active node of a tree detects an adjacent node of another tree then an augmenting path is said to be found.

The algorithm repeats three stages iteratively which are described below:

Growth: S and T keeps on growing until reaching s ⟶ t path

Augmentation: Path identified is augmented, S and T are broken into the forest(s)

Adoption: Restores S and T.

Trees expand in the first stage. Active nodes discover neighboring edges that are non-saturated and obtain new children that become active in the trees concerned. Once, all the neighbors of this active node have been discovered, the active node turns into passive. This stage ends when an active node come across an adjacent node belonging to the opposite tree.

In the second stage, the path identified is augmented. Few edges in the path become saturated as the largest flow possible is pushed. Hence, some nodes in S and T turn out to be "orphans", which means that the edges connecting the nodes to their parents are invalid. Actually, in this stage, there is a possibility of splitting S and T as forests. Still, s and t are roots of S and T while for all other trees, roots are formed by orphans. The adoption stage comes with the objective of restoring the structure of S and T. Here, a new parent that is valid for every orphan is found. This parent should be in S or T, as the orphan and also is connected via an edge that is non-saturated. If the parent is unqualified, the orphan is removed from S or T and it is set as a free node. Moreover, all the former children of that parent becomes orphans. This stage is terminated when there exist no orphans and restores the structures of S and T. After the completion of this stage, the algorithm enters the growth stage. The algorithm gets terminated when there is no possibility for S and T to grow and the saturated edges separate the trees. This means that a max-flow is obtained then the respective min-cut can be estimated by S = S and T = T.

Algorithm for min-cut/max flow pixel based segmentation

Input: G (V, E): |V | = n, |E| = m
Output: Components S = {C1, C2, . . . Cr}
Step 1. Sort the edges E into π = {o1, o2, . . . om}, by non-decreasing weight of the edge.
Step 2. Assume initial segmentation S0 = {{vi}} i = 1 . . . n i.e. every vertex belongs to its component.
Step 3. Do step 4 for q ranging from 1 to m.
Step 4. From the previous segmentation Sq-1, next level segmentation Sq is formed as follows
4.1 Consider vi and vj as the vertices joined by the edge oq i.e. oq = (vi, vj)
4.2 Consider C q−1 i = component in Sq−1 containing vi
4.3 Similarly, consider C q−1 j = component in Sq−1 containing vj

4.4 If C q−1 i 6= C q−1 j and w(oq) ≤ MInt(C q−1 i , Cq−1 j) then C q−1 i and Cq−1 j is merged in S q. If not S q = S q−1
Step 5. Return S = S m.

Homogenous optimum cut technique with boundary based region segmentation

Partition G=(V,E) into two separate sets, namely A and B, such that $A\cup B=V, A\cap B=\emptyset$ by just removing the boundary that connects two regions. The Dissimilarity between these two is estimated as the total weight of the boundary removed. In graph theory, this is termed a cut which is given as:

$$cut(A,B) = \sum_{u\in A, v\in B} w(u,v) \qquad [2]$$

The cut value is optimized with the optimal bi-partitioning of a graph. Even though there exist several such partitions, estimating the optimal cut is still challenging and efficient approaches are developed to handle it. Consider that the weights of the edges are inversely proportionate to the distance between two nodes, it is observed the cut which partitions nodes n1 or n2 have the least value. Actually, when nodes on the right half are cut, the cut value is smaller than the cut partitioning the nodes into the left half.

For avoiding abnormal unfairness partitioning of few points, a new disassociation is introduced between two groups. Rather than considering the value of total weights of the boundary connecting two regions, the cut cost is computed as a fraction of the total boundary that connects all nodes in G. This disassociation is known as optimum cut (O_H-cut) is given below:

$$O_H - cut(A,B) = \frac{cut(A,B)}{assoc\,(A,V)} + \frac{cut(A,B)}{assoc\,(V,A)'} \quad [3]$$

Where $assoc(A,V) = \sum_{u\in A, t\in V} w(u,t)$ is the total connection from nodes in A to all other nodes in G and likewise assoc (B,V). Cut partitioning small isolated points have no small O_H-cut value because certainly, the cut value is mostly large in percentage by the total connection of the small set to every other node. Here, it is found that the value of cut1 across node n1 is equal to the total connection from that node. A measure for the total homogenous optimum connection in a group for a specified partition is defined as:

$$O_{H_{assoc}}(A,B) = \frac{assoc(A,A)}{assoc(A,V)} + \frac{assoc(B,B)}{assoc(B,V)'} \qquad [4]$$

Where assoc (A, A) and assoc (B, B) represents the total weights of the boundary that connects nodes of A and B respectively. This balanced measure reflects on the connection on average nodes in the group. Other important property of association and disassociation of a partition is that naturally the connections are interrelated:

$$O_H - cut(A,B) = \frac{cut(A,B)}{assoc\,(A,V)} + \frac{cut(A,B)}{assoc\,(B,V)'} \; [5]$$

$$= \frac{assoc(A,V) - assoc\ (A,A)}{assoc\ (A,V)} + \frac{assoc(B,V) - assoc\ (B,B)}{assoc\ (B,V)} \quad [6]$$

$$= 2 - \left(\frac{assoc\ (A,A)}{assoc\ (A,V)} + \frac{assoc\ (B,B)}{assoc\ (B,V)}\right) \quad [7]$$

$$= 2 - O_{H_{assoc}}(A,B) \quad [8]$$

Hence, as far as this algorithm is considered, the partition criteria are to minimize the disassociation among groups and maximize the association within groups. In fact, these criteria are identical and are simultaneously satisfied. Optimum cut is used as the partition criteria in this algorithm.

Processing Steps for Homogenous optimum cut technique with boundary-based region segmentation

Step 1: Input image for segmentation is read
Step 2: Resize the given image
Step 3: Estimate the image pixel column and row
Step 4: Evaluate the image entropy with global variables.
Step 5: Estimate the Euclidean distance with consideration of space distance
Step 6: Evaluation of masking regions of the image with consideration of superpixels.
Step 7: Estimate the image boundary
Step 8: Calculate image kernel
Step 9: Image segmentation with the calculated kernel and boundary
Step 10: Calculation of elapsed time of image

Texture identification using spectral partitioning technique

Consider G as a weighted connected graph. Laplacian matrix L is constructed for G and an eigenvector v2 of L is computed which is related with the second-smallest Eigenvalue of L. Generally, components of v2 are not dual valued, nevertheless somewhat smoothly distributed over a set of real values. Hence, the Eigenvector components are rounded off. From the different existing rounding heuristics which can be chosen but nearly every general heuristic are threshold cuts. A value for threshold is chosen which isolates the vertices of G1 from that of G2.

The very simple rounding approach is the zero threshold cut or sign cut. Vertex i is assigned to G1 when the component i of v2 is positive or else to G2. G1 and G2 are connected when not even a single component of v2 is exactly zero. When any component is exactly zero, then G2 is connected assuming that the value of all the vertices are zero but this is not the case in G1. Usually, when rounded by sign, a balanced cut is not produced. The balanced constraint guarantees that the mean component of x is zero, but not whether number of positive components equals the negative ones. Perfect balance is ensured by determining the median component which is then used as the threshold to partition the graph. When the value assigned to various vertices and median components are equal, to balance the cut, those values are divided among sub-graphs which is termed as a median cut.

For few applications, near-perfect balance is necessary but applications that have no perfect balance are beneficial. Some graphs accept a very small cut when a slight imbalance is allowed. For selecting a threshold, the commonly used approach is sweep cut or criterion cut. Initially, the components of v2 are sorted. Next, try with each feasible threshold by explicitly estimating the cut weight for a chosen threshold. The threshold is chosen such that it provides the smallest cut. While sweeping through the sorted list of components, the cut incrementally changes. Hence, this is performed in O(|V| + |E|) time with standard representation of sparse matrix. Rather than the cut weight, sweep cut criteria can be employed. Criteria like Sparsity, isoperimetric ratio or normalized cut can be chosen. A related rounding approach sorts the components and searches the largest gap between two consecutive values. This partition is termed as a jump or gap cut which is easy to program but not reliable as of sweep cut. A non-connected graph deserves exceptional treatment.

Processing Steps for Texture identification using the spectral partitioning technique

Step 1: Read input image for segmentation
Step 2: Apply image filtering with estimating entropy
Step 3: Resize the input image based on calculated entropy
Step 4: Convert input image as a binary image
Step 5: Calculate the threshold value of the input image with masking
Step 6: Estimate the image texture in the masked region with consideration of the image top and bottom.
Step 7: Estimate the image boundary
Step 8: Image segmentation with calculated boundary
Step 9: Calculate the elapsed time of the image.

4. PERFORMANCE ANALYSIS

This section investigates the graph-based segmentation methods so that clusters of similar points are found. There exist several ways to determine feature points that are connected by edges with a fixed number of nearest neighbors. Moreover, all the neighbors with some fixed distance d can also be used as an alternative. Elapsed time is the major parameter considered for the proposed methodology. The obtained output has been given below.

Pixel partitioned image:

Min-Max Segmentation

Figure 3: Pixel partition image using min-cut/max flow algorithm

Above figure 3 shows the pixel partition for the input image using proposed min-cut-max algorithm. Where this image also shows the edge based segmentation with elapsed time of 4.177407 seconds. For segmenting the edges of the image along with the minimum cut/maximum flow in the image this time is optimal when compared with existing techniques. Its time T= 4.1796 and its number of iterations = 37.

Region partitioned image:

Figure 4 shows the region partitioned image using a homogenous optimum cut algorithm along with the boundary cut segmentation. The boundary cut segmented image has been shown below in figure 5. Here the region partitioning using homogenous optimum cut algorithm takes elapsed time as 2.8928 and its time is T= 2.2567 and its number of iterations is 30. As time reduces, the number of iterations decreases and this conserves time and reduces the elapsed time.

Figure 4: Region partitioned image using homogenous optimum cut algorithm

Figure 5: Boundary cut segmentation of input image

Texture partitioned image

Thresholded Texture Image

Figure 6: texture Partitioned based on threshold using spectral partitioning technique

Figure 7: Texture Partition based on segmentation of top and bottom texture

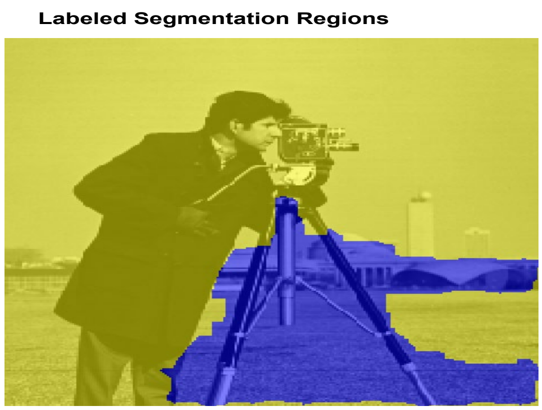


Figure 8: Labeled segmentation of Texture partitioned image

Here the texture partitioned image on the basis of threshold is shown in figure 6 and segmented based on the top texture and bottom texture of the input image is shown in figure 7. And finally figure 8 shows the labeled output. Therefore, the above images show the texture partitioned with minimum graph cut. Here, its elapsed time is 3.89 and time T=3.0245 and their number of iterations is 32.

Table 1: Comparison of Min-Max Pixel Partition

Methods	**Recall**	**F-measure**	**Precision**
Mean Shift	0.912	0.895	0.878
NCuts	0.878	0.875	0.872
Segmentation Tree	0.937	0.913	0.891
Proposed Min-Max algorithm	0.946	0.924	0.916

Above table-1 shows the comparison for the parameters recall, F- measure and precision. Existing techniques compared are mean shift, N-cuts, segmentation tree and proposed Min-max algorithm. According to comparison, the recall, F-measure, and precision have been optimized for the proposed Min-max algorithm.

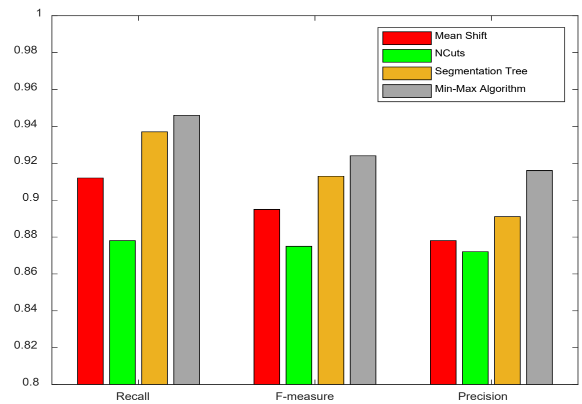


Figure 9: Comparison of Min-Max algorithm with existing algorithms

Above figure 9 shows the graphical representation for table 1 in comparing the min-max algorithm with existing techniques. As mentioned in table 1, the graph shows a higher value for proposed the min-max algorithm.

The Homogeneous Optimum Cut

Table 2: Comparison of Homogeneous Optimum Cut

Methods	**Recall**	**F-measure**	**Precision**
Mean Shift	0.912	0.895	0.878
NCuts	0.878	0.875	0.872
Segmentation Tree	0.937	0.913	0.891
Proposed Optimum Cut	0.916	0.908	0.898

Above table 2 shows the comparison for the proposed homogenous optimum cut algorithm with the existing mean shift, N-cuts and segmentation tree techniques. The parameters compared are recall, F-measure, and precision. When compared with all the existing techniques the proposed optimum cut has attained the enhanced output in segmentation of the input image.

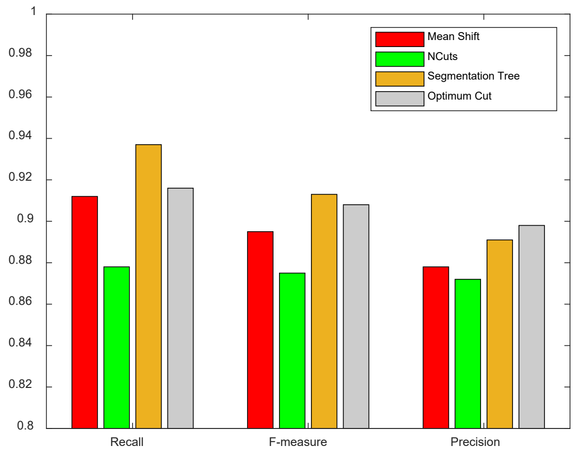


Figure 10: Comparison of Optimum Cut algorithm with existing algorithms

Above figure 10 is the graphical representation of table 2 in comparison of the proposed technique with the existing technique. The parameters plotted are recall, F-measure and precision. Since the comparison values have been enhanced by the proposed technique, the graph plotted with the optimal values of the parameters.

Table 3: Comparison of Spatial Partitioning with existing technique

Methods	Recall	F-measure	Precision
Mean Shift	0.912	0.895	0.878
NCuts	0.878	0.875	0.872
Segmentation Tree	0.937	0.913	0.891
Proposed Spectral Partitioning	0.959	0.947	0.939

Above table-3 shows the comparison measures among the existing and proposed techniques. The proposed spectral partitioning has been compared with the existing mean shift, N-cuts and segmentation tree. Parameters compared are recall, F-measure and precision. The proposed technique segmentation is optimized than existing techniques.

Below figure 11 shows the graphical representation of table 3 where the graphs have been plotted between the existing and proposed techniques for parameters recall, F-measure, and recall.

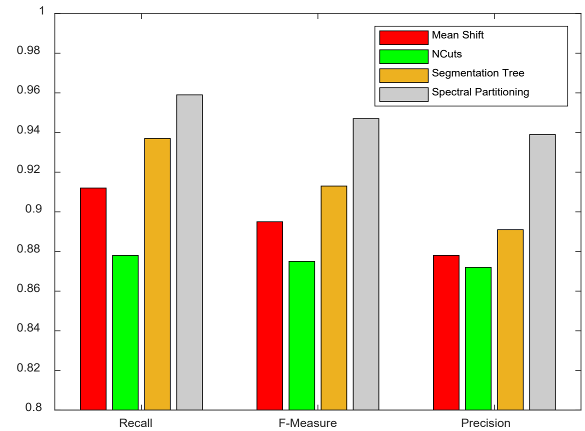


Figure 11: Comparison of Spectral Partitioning algorithm with existing algorithm

Overall Comparison Proposed Methodology

Table 4: Overall Comparison of three partitioning techniques

Methods	Recall	F-measure	Precision
Proposed Min-Max algorithm	0.946	0.924	0.916
Proposed Optimum Cut	0.916	0.908	0.898
Proposed Spectral Partitioning	0.959	0.947	0.939

Above table 4 shows the overall comparison for the proposed methodology. The parameters compared are recall, F-measure, and precision. The techniques that are compared are the proposed Min-Max algorithm, Optimum Cut and Spectral Partitioning. Figure 12 is the graphical representation of table 4. Here the graph has been plotted between proposed techniques for parameters recall, F-measure and precision.

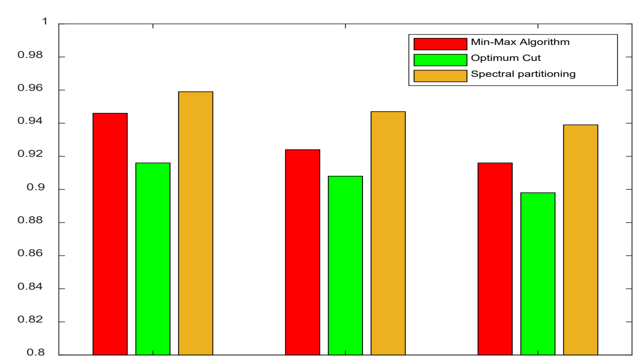


Figure 12: Overall Comparison of three pixels, region and texture partitioning

5. CONCLUSION

This section summarizes the implementation of the graph-based segmentation on basis of three motivations, namely partitioning pixel in the image using min-cut/max-flow algorithm where the edge segmentation has been done, segmentation of the image based on its region using homogenous optimum cut algorithm where the boundary of the image has been segmented with minimum elapsed time and iterations and finally segmentation of the image on the basis of texture and labeling them with minimum graph cut using spectral partitioning technique. The experimental results give optimum time consumption in segmenting the image and reducing the elapsed time during this process. The major challenge for existing techniques is elapsed time and that has been minimized using this proposed methodology.

References

[1] Ning J, Zhang L, Zhang D and Wu C, "Interactive Image Segmentation by Maximal Similarity based Region Merging", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol.43, pp.445-456,2010.

[2] Zuva T, Olugbara O. O, Ojo S. O and Ngwira S. M, "Image segmentation, available techniques, developments and open issues", *Canadian Journal on Image Processing and Computer Vision*, vol.2, no.3, pp.20-29, 2011.

[3] RajeshwarDass Priyanka and Swapna Devi, "Image Segmentation Techniques",*International Journal of Electronics and Communication Technology*,vol.3,no.1,2012.

[4] Salem Saleh Al-amri, Kalyankar N.V and Khamitkar S.D, "Image Segmentation by Using Threshold Techniques", *Journal of Computing,* vol.2, no.5, 2010.

[5] Senthilkumaran N and Rajesh R, "Edge Detection Techniques for Image Segmentation – A Survey of Soft Computing Approaches", *International Journal of Recent Trends in Engineering*, vol.1, no.2, 2009.

[6] Vicente S, Kolmogorov V and Rother C, "Graph cut based image segmentation with connectivity priors", *IEEE conference on computer vision and pattern recognition*, pp.1-8, 2008.

[7] Aslanzadeh R, Qazanfari K and Rahmati M, "An Efficient Evolutionary Based Method For Image Segmentation", *arXiv preprint arXiv:1709.04393*, 2017

[8] Peng B, Zhang L and Yang J, "Iterated graph cuts for image segmentation", *Asian Conference on Computer Vision*, pp.677-686, 2009.

[9] Boykov Y and Kolmogorov V, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision", *IEEE transactions on*

[10] *pattern analysis and machine intelligence*, vol.26, no.9, pp.1124-1137, 2004.

[11] Cui Y and Yang Y, "An algorithm for generating optimal constrained one-stage homogenous strip cutting patterns", *Engineering Optimization*, vol.42, no.10, pp.943-957, 2010.

[12] Wang S and Siskind J. M, "Image segmentation with ratio cut", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.25, no.6, pp.675-690, 2003.

[13] Wang X, Li H, Bichot C. E, Masnou S and Chen L, "A graph-cut approach to image segmentation using an affinity graph based on ℓ 0-sparse representation of features", *IEEE International Conference on Image Processing*, pp.4019-4023,2013.

[14] *pattern analysis and machine intelligence*, vol.26, no.9, pp.1124-1137, 2004.

[15] Cui Y and Yang Y, "An algorithm for generating optimal constrained one-stage homogenous strip cutting patterns", *Engineering Optimization*, vol.42, no.10, pp.943-957, 2010.

[16] Wang S and Siskind J. M, "Image segmentation with ratio cut", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.25, no.6, pp.675-690, 2003.

[17] Wang X, Li H, Bichot C. E, Masnou S and Chen L, "A graph-cut approach to image segmentation using an affinity graph based on ℓ 0-sparse representation of features", *IEEE International Conference on Image Processing*, pp.4019-4023,2013.

[18] Manjula K. A, "Edge Detection as an Effective Technique in Image Segmentation for Image