**2.1. Tiến triển của thuật toán và các công việc liên quan**

Đồng phân cụm đồng thời phân đoạn theo cả đối tượng và theo đặc trưng. Vì vậy nên cần có 2 hàm thuộc : hàm thuộc đối tượng ( hay được gọi là hàm thuộc phân vùng) và hàm thuộc của đặc trưng(hàm xếp hạng). Hàm thứ 2 phục vụ cho việc lọc ra các đặc trưng liên quan trong thời gian tính của hàm thuộc đối tượng và do đó giải quyết được vấn đề về thưa thớt dữ liệu bằng cách giảm số chiều. Thuật toán đồng phân cụm khá phù hợp với những dư liệu nhiều chiều và đã được kiểm chứng trên ảnh đa màu sắc. Vấn đề về phần tử ngoại lai cũng được tối thiểu hóa bằng cách sử dụng hàm thuộc đặc trưng.[26]. Vấn đề với việc chỉ sử dụng hàm thuộc đặc trưng là nó có thể dẫn đến trùng/chồng chéo cụm, do đó việc sử dụng cả hàm thuộc của đặc trưng và đối tượng là điều cần thiết.Ngoài ra hàm tính toán khoảng cách giữa các điểm đặc trưng và các tâm cụm được đưa thêm vào trong quá trình xử lý đồng phâm cụm để tăng thêm hiệu quả so với các thuật toán đồng phân cụm khác . Đồng phân cụm được kết hợp với các phương pháp mờ khác để đưa ra được các cụm khác nhau [24,26]. Cả 2 hàm thuộc đặc trưng và thuộc đối tượng đều mờ,ví dụ như hàm thuộc đối tượng được tính toán khi các cụm khác nhau cạnh tranh 1 điểm dữ liệu , và hàm thuộc đặc trưng được định nghĩa khi các đặc trưng khác nhau cạnh tranh nhau 1 cụm. Như vậy chúng ta sẽ có sự kết hợp giữa 2 hàm thuộc mờ (đối tượng và đặc trưng) trong phương pháp này.

Do đó mục đích là để có một thuật toán đồng phân cụm với những ưu điểm sau:

1. Không nhạy cảm khi khởi tạo và hình thành các cụm riêng biệt. (phân cụm mờ)
2. Thực hiện tốt trong tập dữ liệu nhiều chiều và xác định các cụm được rõ. (đồng phân cụm)
3. Hạn chế tối đa tác động của các giá trị ngoại lai để cải thiện tính chính xác của đồng phân cụm(hàm thuộc đặc trưng/xếp hạng)
4. Hàm mục tiêu cần tích hợp đo khoảng cách của các điểm đầu vào với các cụm ra thành entropy.

5. Thuật toán phải đủ nhanh

Một số thuật toán phân nhóm entropy tối đa và các biến thể của chúng có sẵn trong các tài liệu [34,35]. Một cách tiếp cận hiện tại là biến thể của FCM nhưng tính toán theo entropy[36]. Nó liên quan đến việc tối thiểu hàm mục tiêu sau đây:



Với điều kiện



Với C,N là số cụm và số điểm dữ liệu , uci là hàm thuộc mờ, Tu là trọng số , Dist(xi,pc) là bình phương khoảng cách Euclidean giữa điểm xi và tâm cụm pc.

Phần đầu ở vế phải của (1) biểu thị bình phương khoảng cách,phần thứ 2 đóng vai trò như phần điều chỉnh trong quá trình tối thiểu hàm. Trước hết,chúng ta hãy thử áp dụng phương pháp đồng phân cụm cho thuật toán FCM này bằng cách thay thế hàm tính toán khoảng cách Dist(xi,pc) bằng Dist(xij,pcj) trong hàm mục tiêu. Dist(xij,pcj) được tính toán cho từng đặc trưng tách biệt j=1,2,…,K .

**2.2. Xây dựng hàm mục tiêu**

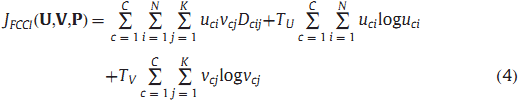
Đặt X={ x1, x2, …,xi, …, xN} Є Rk là N điểm của 1 ảnh I có kích cỡ N1xN2= N, và K là số chiều của của không gian đặc trưng gắn liền với mỗi điểm dữ liệu. Gọi xij là đặc trưng thứ j của điểm thứ i, P = {pcj} là tập hợp các tâm cụm và Dcij = Dist(xij,pcj) là bình phương khoản cách Euclidean giữa điểm xij và tâm cụm pcj tính bằng :

Dcij = d2(xij, pcj) = (xij−pcj)2(3)

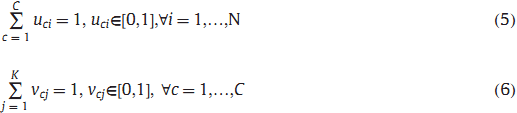
Đặt uci là hàm thuộc đối tượng của điểm thứ i cụm c, U = {uci} là ma trận CxN của hàm thuộc đối tượng của ảnh I , vcj là hàm thuộc đặc trưng của đặc trứng j với tâm cụm thứ c và V = {vcj} sẽ là ma trân CxK của hàm thuộc đặc trưng của ảnh I.

Thêm hàm thuộc đặc trưng vcj vào phần 1 của (1) và thay thế hàm khoảng cách bởi Dcij = Dist(xij, pcj) ta có ∑Cc = 1∑Ni= 1∑Kj= 1ucivcjDcij.Biểu thức entropy ∑Cc = 1∑Ni= 1uciloguci và ∑Cc = 1∑Kj= 1 vcjlogvcj của hàm thuộc đối tượng và hàm thuộc đặc trưng sẽ đóng vai trò là phần thứ 2 và 3 cấu tạo lên hàm mục tiêu JFCCI . Tối thiểu hóa 2 hàm này tương đương với tối đa hóa entropy mờ của −∑Cc = 1∑Ni= 1uciloguci và −∑Cc = 1∑Kj= 1vcjlogvcj. Các entropy được tối đa hóa khi các hàm thuộc mờ uci và vcj được phân bố đều theo các ràng buộc ,ví dụ uci = 1=C and vcj = 1=K.

Hàm mục tiêu JFCCI kết quả từ việc kết hợp tất cả các vấn đề trên là :



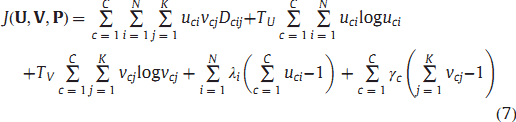
Với ràng buộc



Tối thiểu hóa phần đầu của công thức (4) gán các đối tượng có giá trị của hàm thuộc cao hơn vào cụm mà nó gần hơn. Thành phần bên trong {vcjDcij} gán các trọng số cao hơn vào hàm tính khoảng cách đối với những tính năng nổi bật và trọng số thấp hơn với những tính năng còn lại .Vì vậy phần đầu thể hiện bình phương khoảng cách hiện tại. Công thức entropy thứ 2 và thứ 3 kết hợp riêng biệt tất cả các u′ci và v′cj lại với nhau . Nó giúp thể hiện sự mờ hóa trong các cụm kết quả. TU và TV là các trọng số mờ. Tăng TU Và TV sẽ làm tăng sự mờ của các cụm lên.

**2.3. Cải tiến phương trình**

Để tối ưu hóa hàm mục tiêu FCCI từ (4) có thể áp dụng hệ số nhân Lagrange λi và γc cho các ràng buộc (5) và (6) tương ứng như sau.



Lấy đạo hàm riêng của J(U,V,P) trong công thức (7) đối với U và đặt gradient bằng 0 thì ta sẽ có,



Thay thế uci tính được trong (8) vào (5) ta có công thức tính hàm thuộc của đối tượng uci,



Tương tự như vậy,tính toán đạo hàm riêng của J(0,V,P) theo V và đặt gradient bằng 0 thì ta có,



Tính vcj từ (10) và áp dụng vào công thức (6) ta có công thức tính hàm thuộc đặc trưng vcj như sau



Lấy đạo hàm riêng của J(U,V,P) theo P và đặt gradient bằng 0 thì



Từ (12) ta tính được pcj như sau :



Lời giải cho vấn đề tối ưu hóa công thức (7) có thể tính xấp xỉ bằng dãy Picard hoặc bằng tối ưu hóa xen kẽ (Alternating Optimization-AO) [37] qua (9), (11) và (13) – là những phương trình cập nhật cho các hàm thuộc đối tượng,hàm thuộc đặc trưng và các tâm cụm trong mỗi lần lặp. Để tìm kết quả tối ưu của bài toán tối ưu có ràng buộc, thuật toán FCCI đưa ra một bước học sử dụng thuật toán Bacterial Foraging (vi khuẩn săn mồi) để tối ưu hóa giá trị của các trọng số TU và TV.

**2.4. Giã mã cho thuật toán FCCI**

1. Xác định trọng số TU, TV, độ chính xác giải thuật ε and số lần lặp lớn nhất τmax.
2. Đặt số lặp τ =1.
3. Xác định uci với 0 ≤uci ≤1.

4.Lặp

1. Tính pcj sử dụng công thức (13)
2. Tính Dcij sử dụng công thức (3).
3. Tính vcj sử dụng công thức (11).
4. Tính uci sử dụng công thức (9).
5. Xét lại τ= τ+1.
6. Dừng lại cho đến khi max(|uci(τ)−uci(τ−1)|) ≤ε hoặc τ =τmax.

Với kết quả của nhiều thử nghiệm thì thường ta đặt τmax =200 và ε = 10−2 .

**3. Phân cụm ảnh màu sử dụng FCCI**

**3.1. A. Thuật toán cho phân cụm màu**

**3.1.1. Đánh giá giá trị của cụm S theo Xie và Beni**

Theo Xie và Beni [33] công thức tính giá trị S của các cụm trong trường hợp tồi nhất là

Giá trị d min trong (14) được tính như sau :



khi dmin là khoảng cách nhỏ nhất giữa tâm cụm pcj của cụm c =1,…,C và đặc trưng j và s là là sự thay đổi lớn nhất trong số tất cả các cụm C, được tính bằng :

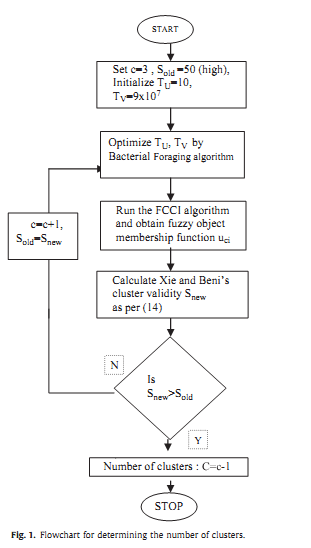


Để xác định được số lượng cụm dựa trên giá trị S trên,ta sử dụng thuật toán như hình Fig. 1 ,thuật toán này sẽ kiểm tra sự xuất hiện của cực tiểu địa phương đầu tiên của S.

**3.2. Thuật toán phân cụm ảnh màu sử dụng FCCI**

Thuật toán gồm cách bước sau :

1. Lấy dữ liệu từ ảnh 3 chiều RBG đầu vào.
2. Chuyển đổi hệ màu RGB sang hệ màu CIELAB với số chiều K=2, ví dụ. {a\*,b\*}. Không gian L\*a\*b\* bao gồm một '\* L' lớp sáng, lớp kết tủa màu 'a\*' và lớp kết tủa màu 'b\*'. Tất cả các thông tin màu sắc nằm trong các lớp ' a\*' và 'b \*'
3. Chuyển đổi dữ liệu 2 chiều sang dữ liệu 1 chiều để tạo ra các điểm xij trong chiều j, j=1,2 với mỗi điểm ảnh i=1,…,N, với N là độ lớn của dữ liệu.Bước này khá quan trọng vì việc tính toán trên mảng dữ liệu 1 chiều thì đơn giản hơn trên mảng dữ liệu 2 chiều.
4. Tính toán số cụm C dựa trên thuật toán như hình Fig. 1.
5. Chạy thuật toán FCCI cho C cụm và tính toán được hàm thuộc đối tượng uci.
6. Giải mờ uci vào các cụm.



**3.3 Thuật toán vi khuẩn săn mồi (Bacterial Foraging) cho bài toán tối thiểu hóa**

Thuật toán vi khuẩn săn mồi (Bacterial Foraging-BF) [39] dựa trên các hành vi của vi khuẩn E. Coli được sử dụng để tối ưu hóa các giá trị của tham số mờ TU và TV trong thuật toán. Thuật toán BF ban đầu chấp nhận một tập hợp các giá trị ban đầu của người dùng,sau đó tối ưu hóa những giá trị này thành cực tiểu bằng các vòng lặp liên tục. Giá trị đầu vào của thuật toán này sẽ là: TU =10, TV =9x107 được giả định cho các thí nghiệm phân cụm màu. Sự lựa chọn các giá trị ban đầu này được thực hiện bằng cách tiến hành một loạt các thí nghiệm ngẫu nhiên bằng cách thử nghiệm. Vi khuẩn khi săn mồi được coi như là 1 quá trình tối ưu [40], trong đó mỗi loại vi khuẩn tìm cách tối đa hóa năng lượng thu được trên một đơn vị thời gian dành cho việc săn mồi . Giả định rằng θ là vị trí của vi khuẩn và đặt JD(θ) là các ảnh hưởng từ môi trường, ví dụ JD(θ)<0, JD(θ)= 0, JD(θ)>0, có nghĩa rằng vi khuẩn đang ở vị trí môi trường giàu dinh dưỡng,bình thường và độc hại.

Như vậy khi vi khuẩn săn mồi,chúng có xu hướng tìm những nơi có dinh dưỡng cao(ví dụ tìm những nơi có giá trị JD(θ)<0 càng thấp càng tốt) và tránh những nơi độc hại( JD(θ)>0) và tìm cách tránh những nơi bình thường,trung lập (vị trí mà JD(θ)= 0).Người ta gọi mỗi bước như vậy là “hóa hướng” (chemotaxis step).

Cuối cùng ta đánh giá về sức khỏe của các vi khuẩn và sắp xếp chúng theo thứ tự giảm dần.Một nửa số vi khuẩn khỏe được nhân bản bằng cách gán cho cùng vị trí và một nửa khác bị loại bỏ. Hoạt động này gọi là quá trình sinh sản.

Sau khi đạt đến số lượng bước sinh sản, mỗi vi khuẩn có thể bị loại bỏ hoặc phân tán với xác suất nào đó. Phân tán có nghĩa là tổ chức lại những vi khuẩn để di chuyển chúng đến vị trí tốt hơn

Các tham số cài đặt ban đầu của thuật toàn vi khuẩn săn mồi (Bacterial Foraging)

1. Đặt số lượng vi khuẩn B =50.

1. Số lượng biến w cần tối ưu là 2: TU, TV.
2. Độ dài mỗi lần bơi (Swimming length) Ns =4.
3. Nc,đặt số lần lặp lại trong một vòng lặp hóa hướng là 100
4. Nre, đặt số lượng các bước sinh sản là 2.
5. Ned, đặt số lượng các sự kiện loại bỏ và phân tán là 2.
6. Xác suất cho mỗi vi khuẩn bị loại bỏ / phân tán được thiết lập để 0.25.
7. Vị trí của mỗi loại vi khuẩn L(w,B, Nc, Nre, Ned).
   1. **Kết quả của thuật toán**

**3.4.1. Segmentation evaluation indices**

The quality of the segmentation is generally judged by two types of indices: the goodness methods such as Liu’s F-measure which ascertains the color difference in the CIELAB color space and also penalizes the formation of large number of segments, and the discrepancy methods which ascertain the quality with respect to some reference result like ground truth images for example the RAND index. The above two types of quality measures are used together to judge the efficiency and practicality of the proposed algorithm.

1. Liu’s Evaluation Measure (F): The performance of color segmentation is evaluated using Liu and Yang’s [41] evaluation function F:



where I is the segmented image and N1xN2 is the image size, G is the number of regions of the segmented image, Ai is the area and ei is the average color error of the ith region where ei is defined as the sum of Euclidean distances between the {a\*,b\*} color vector of the pixels of region i and the color centroid attributed to region i in the segmented image. The smaller the value of F(I) the better the segmentation result. We choose Liu’s F-factor as one of our evaluation criteria since it gives an accurate measure of the color differencing achieved by the segmentation algorithm and at the same time penalizes large number of regions formed.

1. (a) Probabilistic RAND index (PR): The PR index is a generalization of the RAND index [42] introduced by Unnikrishnan et al. in [43]. It allows a comparison of the test segmentation with multiple ground truth images through soft non-uniform weighting of pixel pairs as a function of the variability in the ground truth sets.

Suppose each human k provides information about the seg-mentation in the form of binary numbers ∏(liSk = ljSk ) for each pair of pixels (xi,xj). The set of all perceptually correct segmentations defines a Bernoulli distribution giving a random variable with the expected value denoted as hij. The Probabilistic RAND index (PR) is then defined as



where mij denotes the event of a pair of pixels i and j having the same label in the test image Stest:

|  |  |
| --- | --- |
| mij = ∏(liStest = ljStest ) | (19) |

This measure takes values [0,1], where 0 means no similarities between Stest and {S1,S2,…SK}, and 1 means all segmentations are identical.

(b) Normalized Probabilistic RAND index (NPR): The Normalized PR index by Unnikrishnan et al. in [12], is an excellent means of qualitative comparison among image segmentation algorithms. Once the segmentation of all the test images for all the algorithms being compared has been compiled the Normalized PR index is calculated so that a global measure is possible.



The above equation assures that the expected value of normal-ized index is zero providing a wider range. The maximum value of PR, MaximumPR, in (20) is taken as 1.

3.4.2. Color segmentation results

In this section experimental results are presented to prove the effectiveness of the proposed color segmentation algorithm on the natural images. For these results MATLAB (ver.7.9) software is run on a Pentium-IV 1.4 GHz PC. All images are digitized to 24 bits per pixel in the RGB format. Since the distance between any two points in the RGB space is not proportional to their color difference, transformation from the RGB space to a uniform color space: CIELAB [31] is performed. The vector {a\*,b\*} of CIELAB color space contains the total chrominance color information of pixels and is the feature space for our color segmentation experiments. The vector {L\*} or luminance vector which decides the darkness or fairness of the image segments is discarded in the clustering process to ensure that the illumination effects do not affect the segmentation process. It is observed that the FCCI algorithm yields highly crisp values of object membership function uci (close to 0 and 1). On the other hand the feature membership values vcj are highly fuzzy (close to 0.5) due to averaging over the entire dataset. The vcj values however are accentuated by the high values of parameter TV (≈108) in Eq. (4) creating a considerable influence in the computation of uci, eventually leading to crisp values of uci after the iterative procedure. This helps in crisp classification during the defuzzification process.

A set of 100 test images is taken from the Berkeley segmentation database [44,45] along with 5–7 ground truth segmentations available for each image in the database for the evaluation of the results. The size of each image is either 321x481 or 481x321 and the average time taken by the FCCI algorithm for each image is approximately 35 s. The segmentation of all 100 images by the proposed FCCI algorithm is shown in Fig. 2, with the edges superimposed on original images. The corresponding graphs for Liu’s F-factor and the Probabilistic RAND index (PR) is shown in Fig. 3 for 100 images bestowing excellent values for both segmen-tation evaluation measures. The results show a good match with human ground truth segmentations as indicated by a high value of Probabilistic RAND index (PR), and also efficient color differentiation as indicated by a low value of Liu’s evaluation measure F. The number of clusters is determined from the first local minima in the cluster validity S graph (normally <7 clusters for our experiments) as demonstrated by the example shown in Fig. 4

Some observations made from the results obtained are as follows:

1. Tradeoff between color differencing and human perception: in the case of images with distinct colors (Fig. 5) there is an excellent correspondence with human perception (high NPR) but the color differentiation is not so good (high Liu’s F-factor). In the case of images with indistinct colors (Fig. 6), very good color difference is observed (low Liu’s F-factor) but the results appear to be over-segmented and hence NPR value is relatively less. The proposed technique therefore maintains a good tradeoff between the two segmentation evaluation measures.
2. Sensitivity to parameters: the algorithm is found to be more sensitive to the values of TU than TV since the values of uci

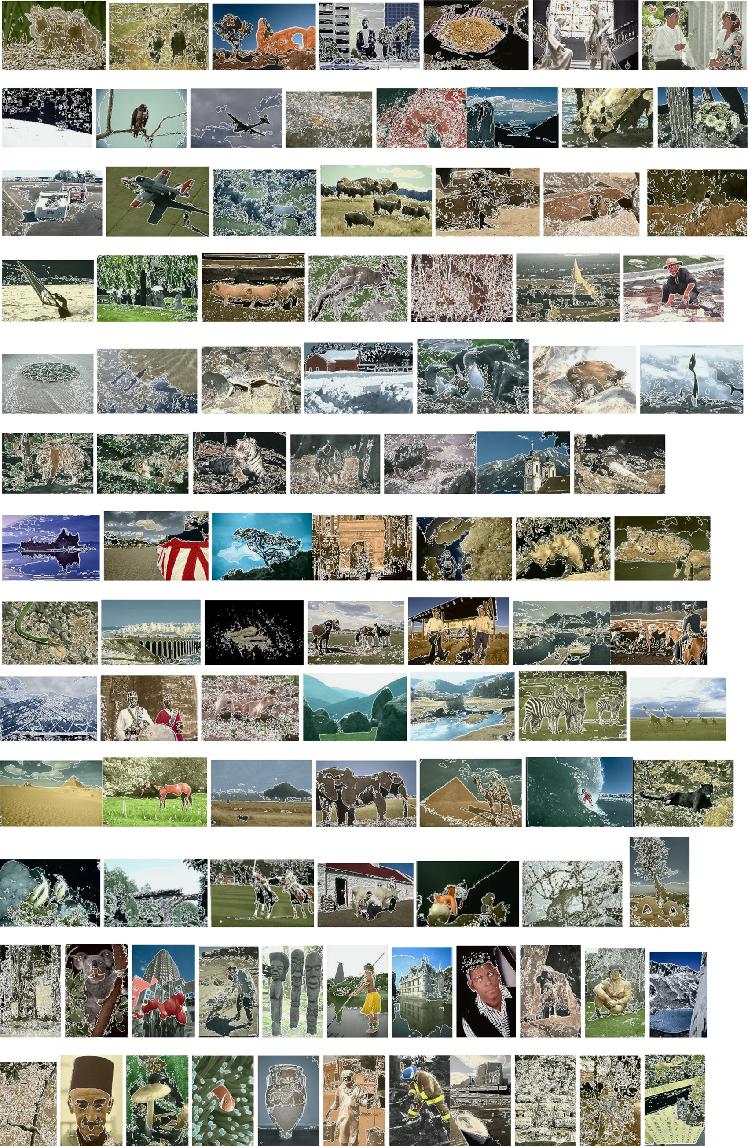


Fig. 2. Color Segmentation Results of 100 test images from Berkeley segmentation database [44] by the proposed method. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

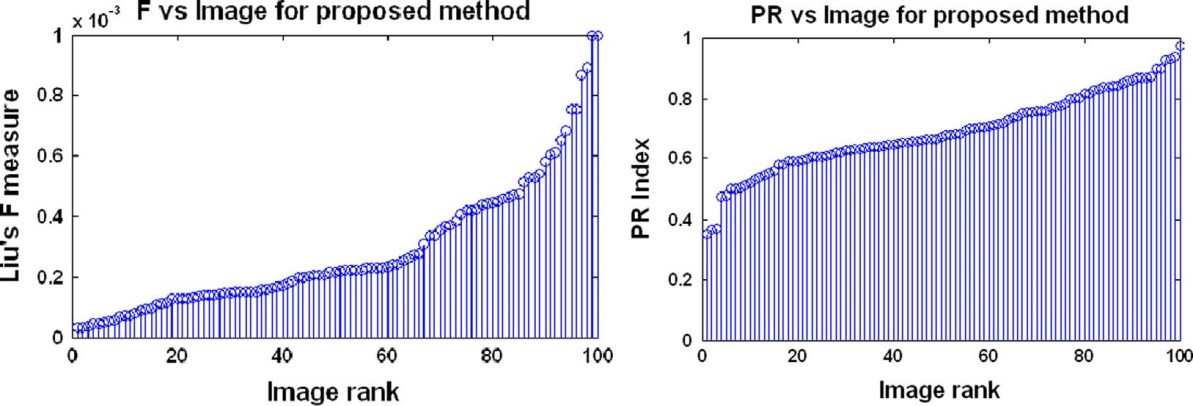


Fig. 3. Liu’s F-measure and PR index in the increasing order for segmentations of 100 test images from Berkeley dataset [44] by the proposed method.

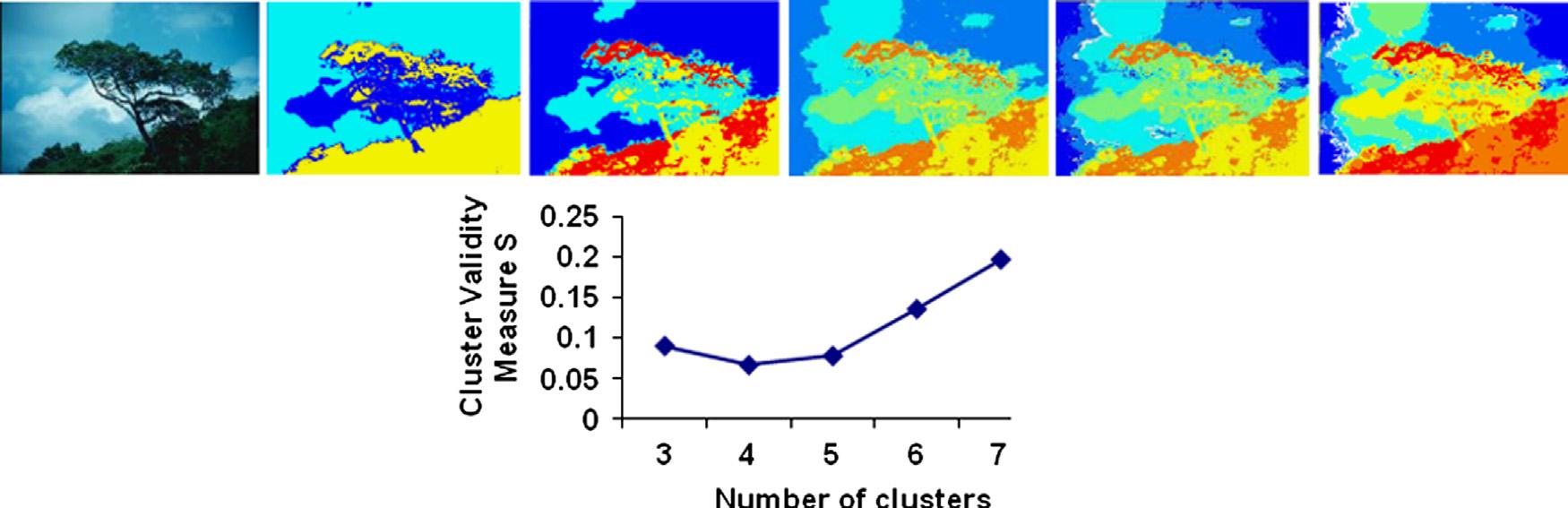


Fig. 4. Original image and Segmentation results (from left to right and top to bottom) of “Tree” image and the corresponding Clustering validity graph as the number of clusters c is varied from 3 to 7. Corresponding values of NPR and Liu’s F-factor are: c: 3, 4, 5, 6, 7, NPR: 0.5519, 0.5203, 0.1314, −0.213, −0.1332, F: 0.00044, 0.000347, 0.000322, 0.000332, 0.000342. Number of clusters is aptly determined to be C =4 (from the first local minima in the graph) as it results in the most optimum combination of NPR and Liu’s F-factor.



*Fig. 5. Example segmentations of images with distinct colors—NPR(from left to right): 0.6065, 0.7962, 0.8007, 0.9213, F(from left to right): 0.00047, 0.000138, 0.000134, 0.000085. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)*

obtained are very crisp and a careful choice of TU in (4) is required for the algorithm to converge. Replacing the Bacterial Foraging optimization by the Genetic algorithm in our experi-ments results in a large computational overhead with an initial population of 100 chromosomes required for acceptable results, while Bacterial Foraging starts giving good results for the initial population of 8 bacteria, thereby, significantly redu-cing the computational overhead.

The value of TU in our experiments is found to range from 1–30 for different images with values close to 1 for images with non-distinct colors (Fig. 6) and higher values for visually distinct colors (Fig. 5). The valid values of TV ranges widely from 106– 108 and do not have a major impact on the resulting clusters since its only function is to contribute to the computation of uci by scaling vcj.

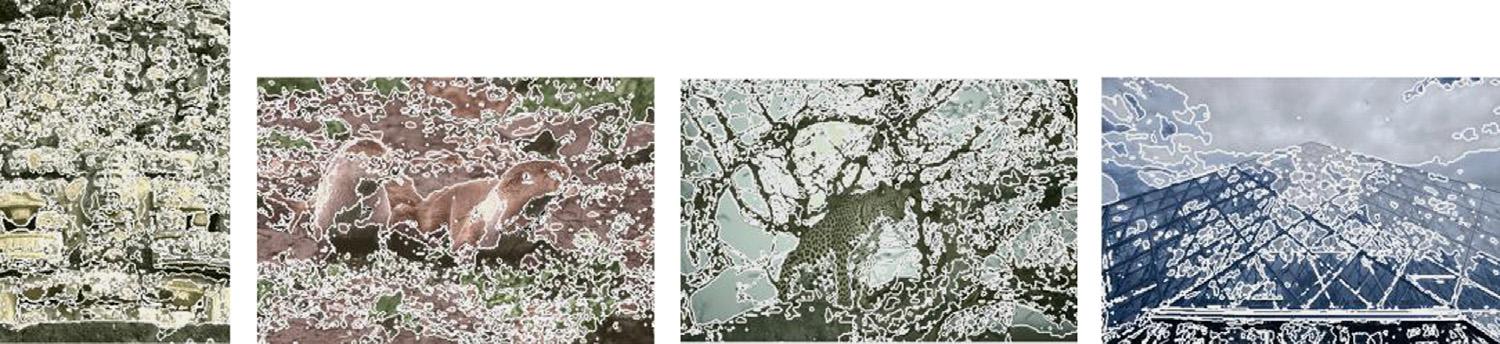
1. Complex illumination Patterns: the algorithm is able to segment natural scenes containing non-uniform illumination efficiently

(Fig. 7(a)) by segregating shadows from sunlit portions thus agreeing with human perception. However in the cases where the shadows tend to merge with the colors in the scene (Fig. 7 (b)) result tends to look over-segmented in spite of very good color differencing (low F).

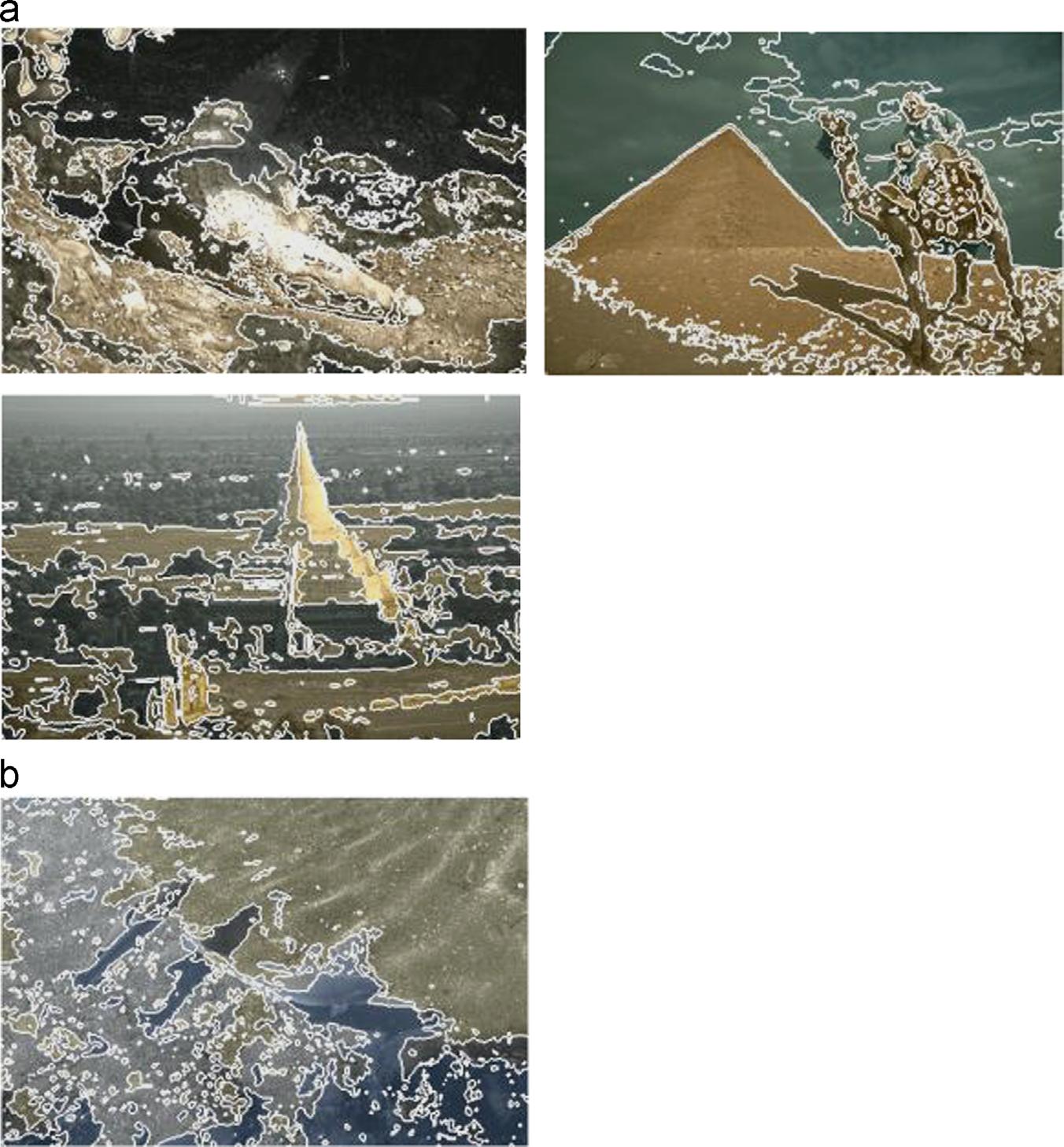
1. Under-segmentation: only in rare cases (1 out of 100) the algorithm fails to segregate extremely indistinct colors as demonstrated by the under-segmented result in Fig. 8 due to formation of highly fuzzy clusters.

3.4.3. Comparisons with other methods

The proposed color segmentation technique is compared with some well known methods in literature for unsupervised color segmentation: Fuzzy C-Means (FCM) [22,37], Normalized Graph-Cut (N-CUT) Method [10], Gaussian Mixture model (GMM) [8] and



*Fig. 6. Example segmentations of images with indistinct/similar colors—NPR (from left to right): −0.284, −0.391, −0.334, −0.098, F(from left to right): 0.000053, 0.00023, 0.000057, 0.0000701. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)*

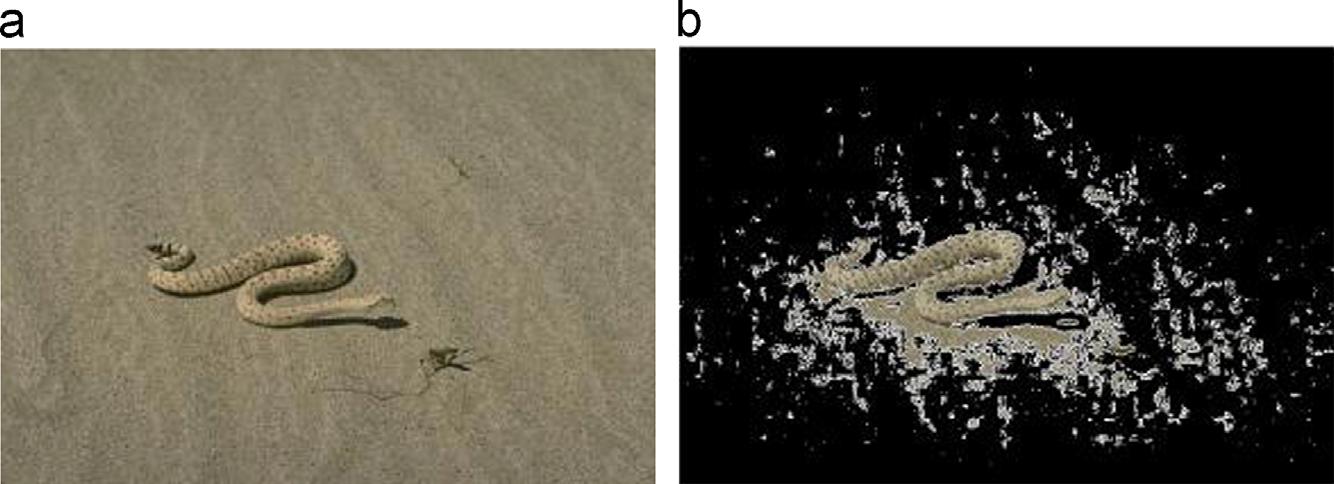


*Fig. 7. Example segmentations with shadows and non-uniform illumination. NPR(from left to right): (a)−0.301, 0.314, −0.022 and (b) −0.7609, F(from left to right):*

*(a) 0.00025, 0.00023, 0.00026 and (b) 0.00015*

Mean-Shift (MS) [9] segmentation methods. The CIELAB color space is used for all the comparisons. While both mean shift and NCUT graph based method are popular feature space clustering methods, FCM is chosen since it is a widely popular fuzzy clustering method for image segmentation. The parameters for FCM algorithm used in our experiments are: index of fuzziness m=2, maximum error limit ε = 10−2, maximum iteration =200. Normalized Cuts graph based segmentation method uses eigen-vector techniques to obtain graph partitions. It finds minimum cuts in a graph while minimizing the similarity between different patches. GMM models the color features as a mixture of Gaussian kernels using Expectation Maximization algorithm for estimation of parameters of the Gaussian mixture and is a popular method for image segmentation and retrieval. Figs. 9 and 10 show the graphs for F and NPR indices for the 100 test images from the Berkeley Segmentation Dataset in the form of histograms for the five methods. It is observed from the graphs that the proposed FCCI algorithm provides the most optimum combination of the lowest values of F-measure (of the order of 10−4) and sufficiently high values of NPR index among all the five methods proving the efficiency of the proposed color segmentation algorithm by strik-ing a neat balance between color differencing and human percep-tion standards.

Fig. 11 shows the color segmented results of the five methods for six randomly selected test images from the Berkeley Segmen-tation Dataset namely: ’Mud-Huts’, ‘Plane’, ‘Eagle’, ‘Building’, ‘Wolf’, ‘Tree’. The corresponding F and NPR segmentation evalua-tion indices are listed in Table 1 depicting the lowest values of F for the proposed method as compared to all other methods. The NPR readings are also observed from Table 1, to be overall best for the proposed method though the Mean-Shift algorithm performs better for ‘Plane’ image. It is observed from the segmentation results in Fig. 11(b) that the proposed method FCCI forms well defined and interpretable clusters even when the color difference



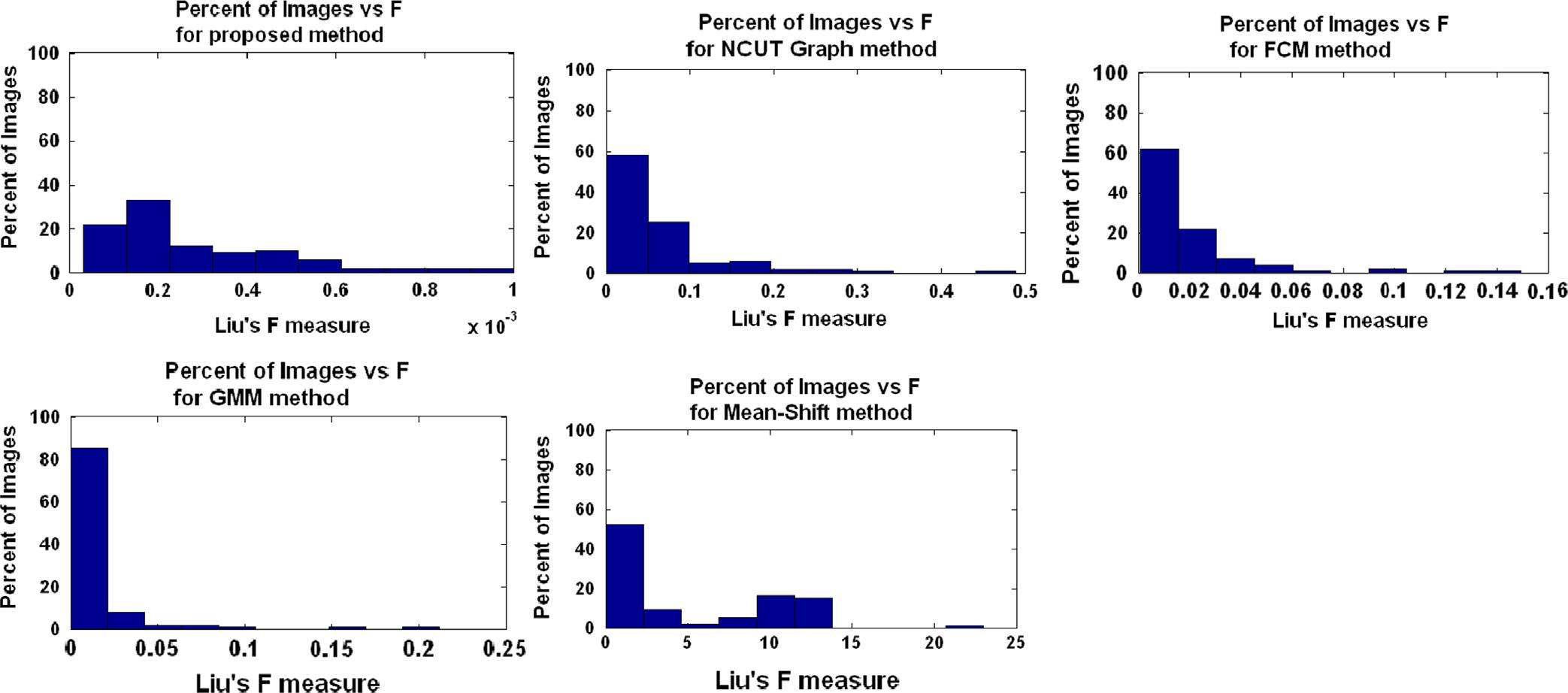


Fig. 9. Histograms of Liu’s F-measure achieved for individual images for the proposed method, NCUT Graph based method, FCM, GMM, Mean-shift segmentation methods.

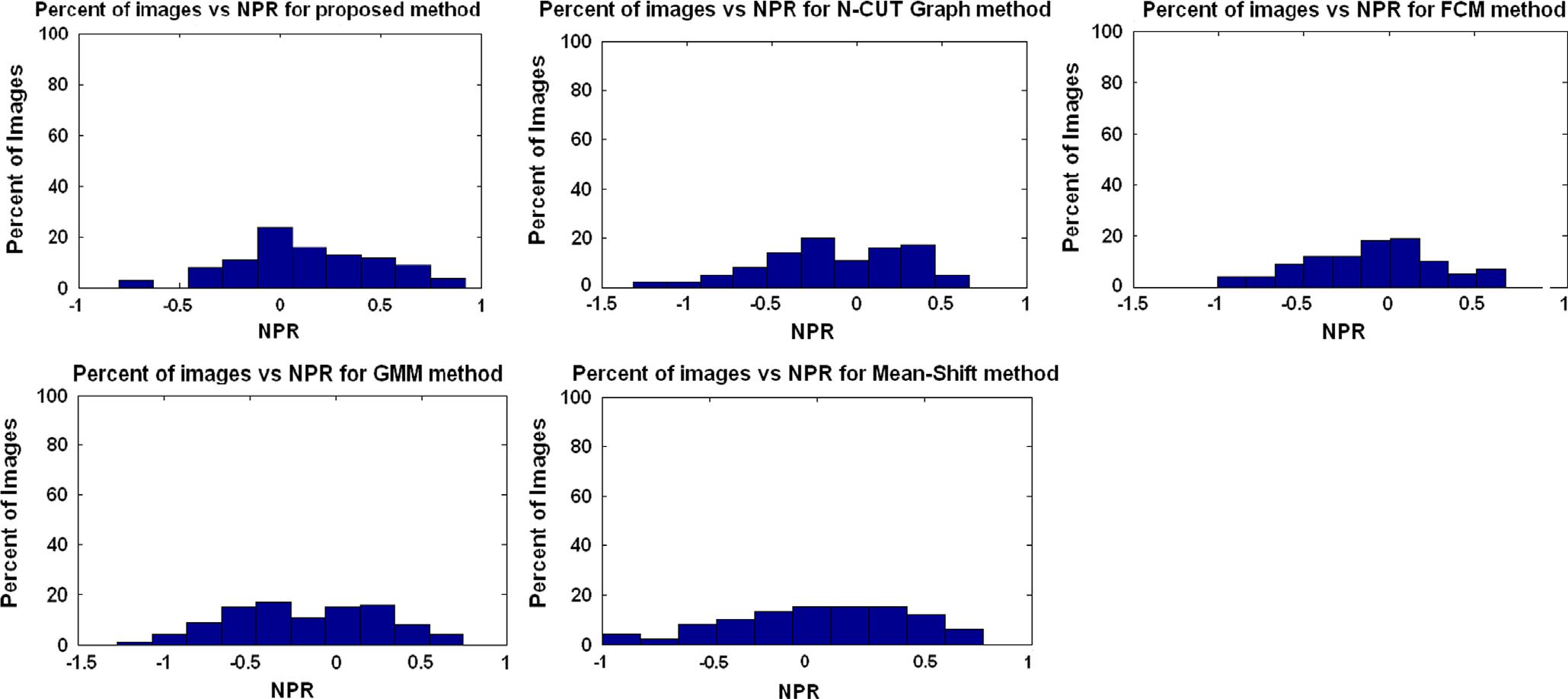


Fig. 10. Histograms of NPR Index achieved for individual images for the proposed method, NCUT Graph based method, FCM, GMM, Mean-shift segmentation methods.

between two regions is not too distinct as in the case of ‘Wolf’ image. The proposed technique is efficient in segmenting out uniform color regions and gives less fake boundaries as observed in the case of ‘Building’ image in Fig. 11(b) where the windows of the building are nicely segmented out visually as compared to other methods. An important factor that gives FCCI an edge over

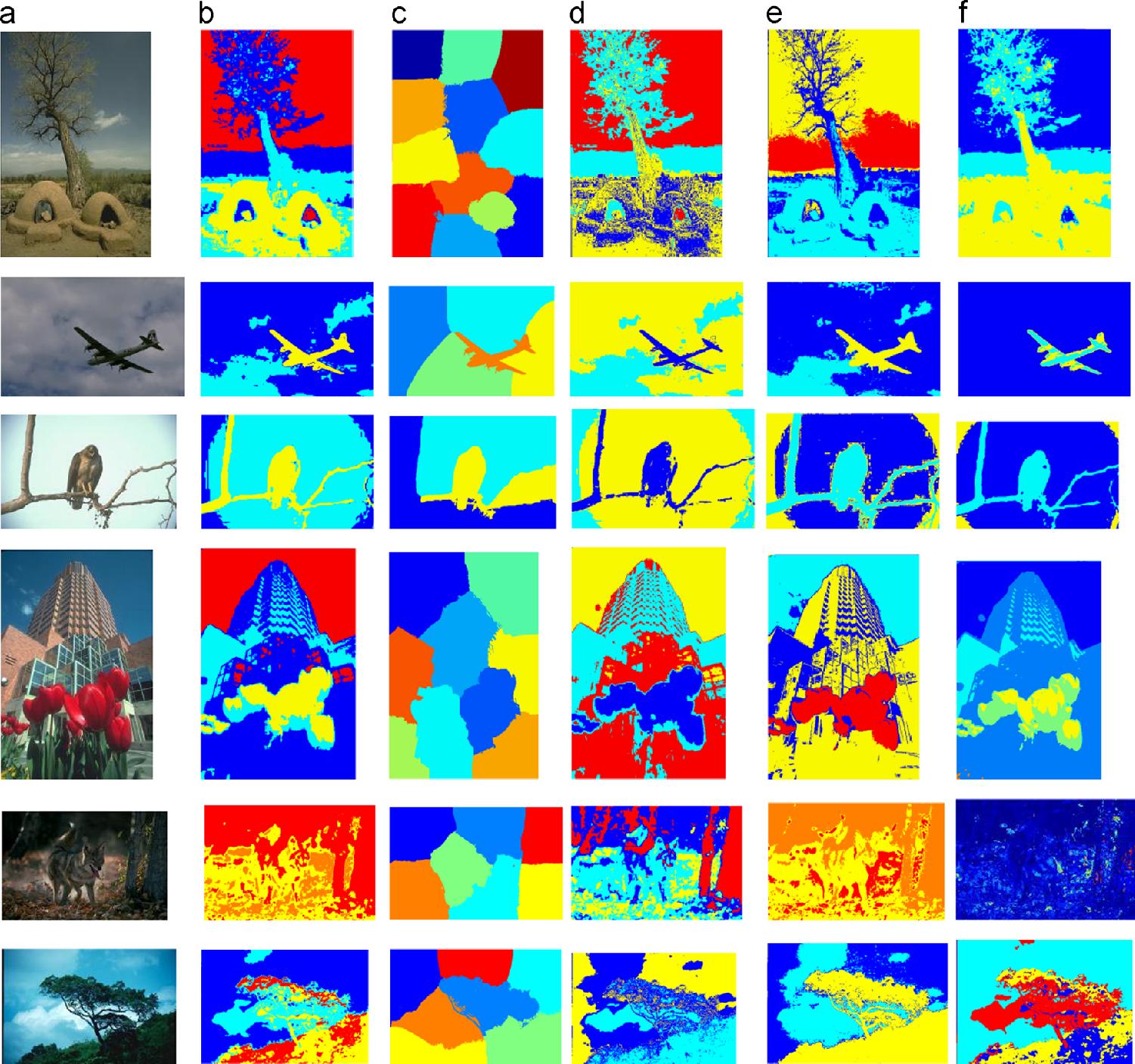
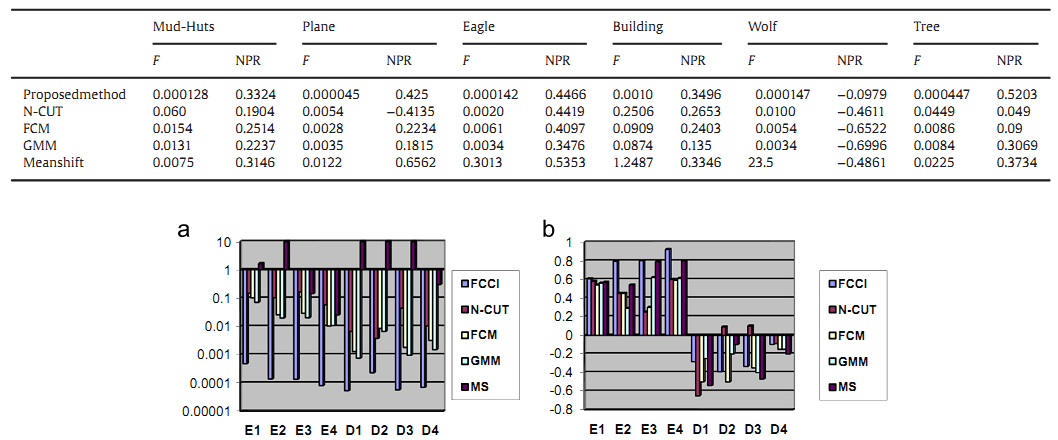


Fig. 11. Color segmentation results (a) Original images from Berkeley segmentation database [57] : ‘Mud-Huts’, ‘Plane’, ‘Eagle’, ‘Building’, ‘Wolf’, ‘Tree’ (b) Proposed method [(from top to bottom): TU ={10.96,1.15,2.05,9.52,4.81,11.79}, TV = 108] (c) N-CUT (d) FCM (e) GMM (f) Mean Shift segmentations [Space Bandwidth hs = 15, Color bandwidth hr (from top to bottom): {7,3,4,10,1,8}]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

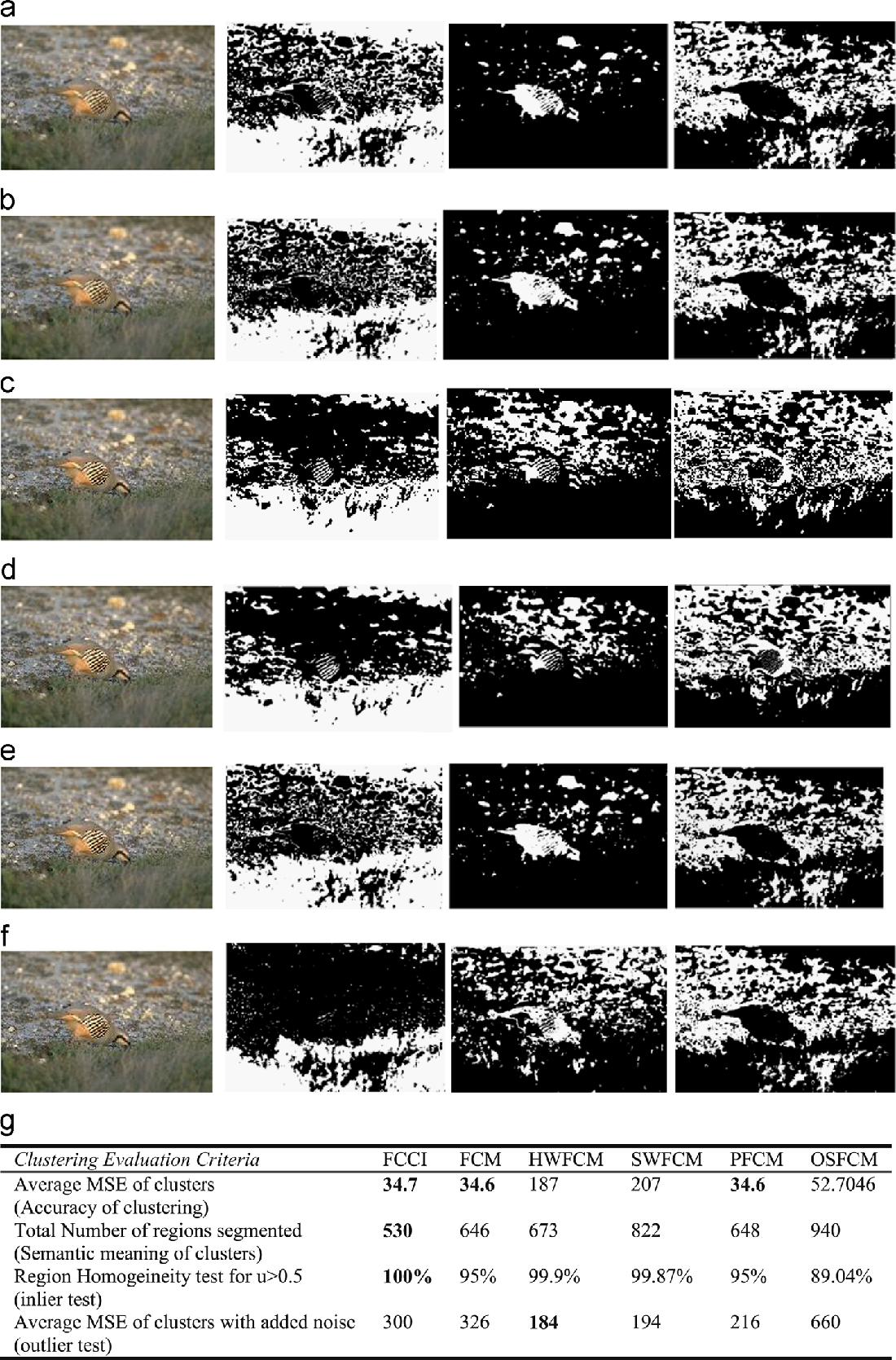
Table 1

Liu ’ s F-measure and normalized probabilistic RAND index for the six test images: ‘ Mud-Huts’ , ‘ Plane ’ , ‘ Eagle ’ , ‘ Building’ , ‘ Wolf ’ , ‘ Tree’



*Fig. 12. The plots for (a) F-value (logarithmic scale) and (b) NPR readings for 4 easy (E1 to E4) images (from Fig. 5) and 4 difficult (D1 to D4) images (from Fig. 6) are shown below.*

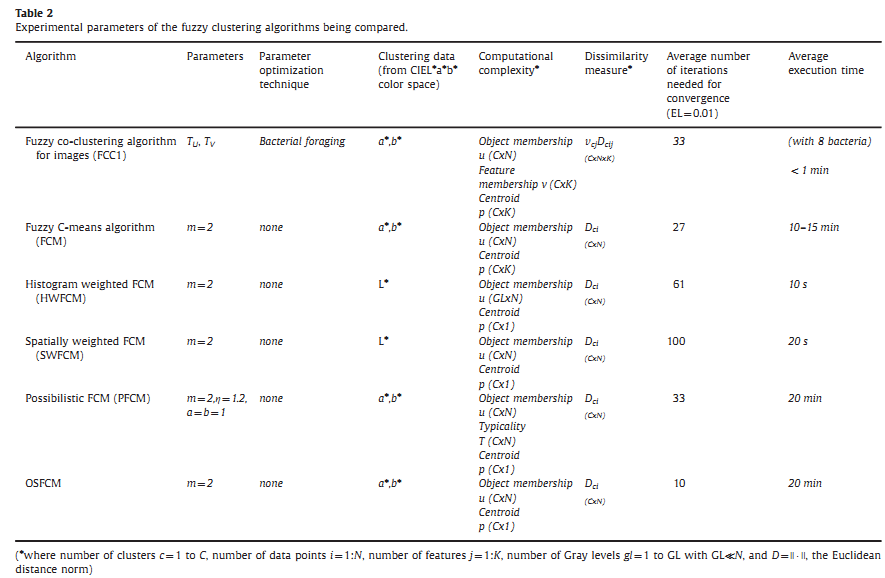
FCM clustering technique is the reduced time complexity. The time required by FCM for clustering is 10–15 min while the proposed algorithm hardly takes 1 min for the same. How fuzzy co-clustering scores over fuzzy clustering can be observed by comparing the segmentations of the proposed method (Fig. 11(b)) with that of FCM (Fig. 11(d)). The criterion for choosing number of clusters is the same as that for the proposed method (from the local minima in the clustering validity graph). As seen in the ‘Tree’ image co-clustering forms distinct and correct clusters with criteria being maximum separation between clusters/colors. Thus the green foliage and the blue sky being of distinct colors are grouped into separate clusters by the proposed method whereas in FCM, the parts of sky/clouds are clustered together with green foliage. FCM also suffers from the problem of outliers as observed from the pigmented segmentations in the ‘Mud-Huts’ example in Fig. 11(d). The co-clustering results are improved because of the grading/relevance factor (feature membership function) assigned to each of the color feature (an,bn) with respect to a particular

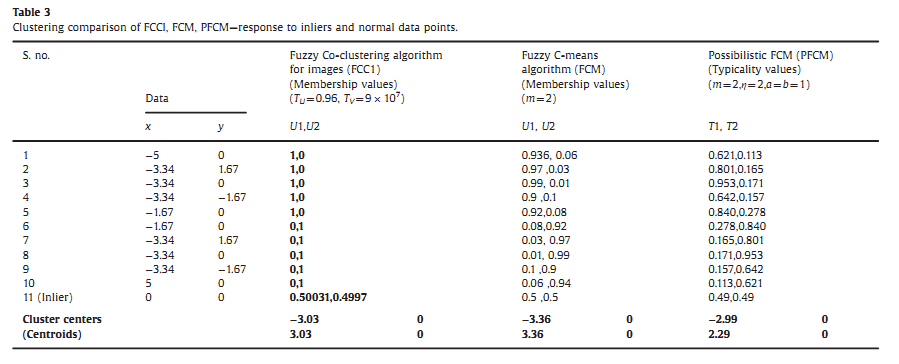


*Fig. 13. The comparison of fuzzy clustering results (with the best values in bold) on the ‘Bird’ image for three clusters (a) FCCI (b) FCM (c) Histogram Weighted FCM(d) Spatially Weighted FCM (e) PFCM (f) OSFCM (g) Segmentation Evaluation results in terms of Mean Square Error (MSE), Region Homogeneity Test (RH) and Mean Square Error (MSE) in the presence of noise.*

cluster which leads to correct evaluation of clusters and also solves the problem of outliers. The N-CUT Graph based method does not give a very good correctness with respect to human ground truth images as compared to the proposed method and mean shift method. Also it tends to segregate uniform color regions into large chunks. GMM by Expectation Maximization algorithm is an unsupervised technique resulting in blob like segments. It classi-fies the entire tree and green foliage of the example ‘Tree’ in Fig. 11(e) into 1 segment performing well with respect to human evaluation but poorly for color difference between regions. More-over, the resulting blobs contain no internal details as demon-strated in the blobs formed in ’Eagle’ image in Fig. 11(e). The mean shift algorithm performs well corresponding to human ground truths as indicated by a high value of NPR (example–‘Plane’ image in Fig. 11(f)). However it fails in the absence of any dominant colors in the scene as seen in the example of ‘Wolf’ image where the colors are visually indistinct. The Liu’s F-measure values are found to be generally high in the case of both mean shift and N-CUT graph based algorithms. Also the mean shift method is very sensitive to its color bandwidth parameter hr with a slight change causing large changes in granularity of segmentation. The hr values for the test images are in the range 1–15 while the space bandwidth is hs =15. Though some feature weighted clustering algorithms have been proposed in the past [46] and were applied for image segmentation problems [47], they do not incorporate a separate feature membership to be independently updated in each iteration along with pixel memberships and cluster centroids. Fig. 12 shows the comparison (F and NPR values) between the five methods for the mix of easy and difficult images in Figs. 5 and 6 respectively taken in the same order. The results affirm that F-values are minimum for FCCI for all images, a fact already established from the graphs in Fig. 9, while NPR values are highest for FCCI for all the easy images E1 to E4 and most of the difficult images except for the D2 difficult image for which NCUT, GMM and Mean Shift yield better results. In Fig. 13 we compare the segmentations using some recent fuzzy clustering algorithms apart from FCM namely Spatially weighted FCM [48], Histogram weighted FCM [49], Possibilistic FCM [50] and Orientation sensi-tive FCM (OSFCM) [51] on an example image ‘Bird’ containing three main clusters, the bird, the stones and the grass. The experimental parameters of these algorithms are summarized in Table 2. The results in Fig. 13 indicate that FCCI provides the most optimum segmentation with respect to human evaluation of the scene and is equivalent to FCM in terms of accuracy of the clusters formed. FCM and FCCI form the most visually acceptable clusters of the scene segmenting out the bird nicely in Cluster 2. Hence the reason that FCM among all the existing fuzzy clustering algorithms is most popular for image segmentation purpose. FCCI yields the cleaner image of the two (with lesser number of regions as compared to FCM as seen in Fig. 13) complying more with human perception of the scene.

The inlier or bridge pixels are properly clustered by FCCI as compared to all other clustering algorithms as indicated by the region homogeneity tests (criteria being uci 40.5), though the outlier problem of FCM persists. The inlier pixels are defined as those that are equidistant from all centroids and have a membership of 0.5 to all clusters as a result. Since FCCI computes feature memberships as well which evaluates the relationship of a cluster to a feature, it is taken into account the feature value distribution of the inlier to compute its membership to a cluster. However the outlier points or noise pixels are a problem since for these data points the algorithm behaves like FCM and is sensitive though some marginal improvement is noticed from the Table in Fig. 13. Table 3 evaluates the memberships obtained from FCCI, FCM and PFCM algorithms for the X11 dataset fX10∪(inlier)gin [50] where X10 contains 10 two dimensional data points. The results are found to be best for the proposed FCCI algorithm in terms of crisp values of memberships (1,0) obtained and the definitive values of inlier memberships that is indicative of the clusters (uci 40.5) to which they belong to. The only shortcoming of FCCI clustering is the response to the outliers which need to be minimized.





4. Conclusions

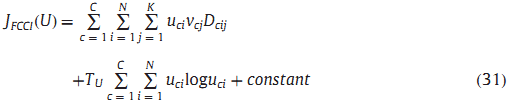
In this paper, the Fuzzy co-clustering approach based on the simultaneous clustering of both object and feature memberships is used for the color segmentation of natural images. A new objective function is formulated and the update rules are derived. The new algorithm (FCCI) is tried for color segmentation on 100 test images from the Berkeley segmentation dataset yielding precise segmen-tation. The color vectors {a\*,b\*} of CIELAB color space are the feature variables for the color segmentation algorithm. The per-formance is evaluated on the basis of Liu’s Function F, and the Normalized Probabilistic RAND (NPR) index. The number of clusters is determined from the first local minima of Xie and Beni’s cluster validity curve and is found to produce apt results (low F and high NPR). The proposed method produces accurate color differencing and at the same time adheres to the human perception in segmenting the natural scenes with non-uniform illumination and shading. It is also compared with some of the existing color segmentation techniques and is found to outperform them. The future scope of this work lies in improving the image segmentation in the presence of outliers and exploring other evolutionary algorithms for speedy solutions of the two para-meters involved.

Appendix A

The proof of convergence of the FCCI algorithm is shown below:

Theorem 1. The updated values of uci given by Eq.(9) never increase the objective function in every iteration

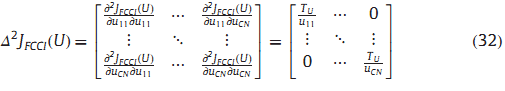
Proof. Consider the objective function as a function of uci alone.



where, constant = TV ∑Cc = 1∑Kj= 1vcjlogvcj

Also, the product vcj Dcij may be considered as constant. To prove Theorem 1 we have to prove that U\*, i.e the updated values of uci given by Eq. (9) are the local minima of the objective function JFCCI(U\*) provided that the constraints in (5) and (6) are satisfied.

For this we need to prove that the Hessian matrix ∆2JFCCI(U\*) is positive deﬁnite.



At U\*, uci≥0 and TU is always assigned a positive value. Therefore the Hessian matrix ∆2JFCCI(U\*) is positive definite. We have proved the first necessary condition (∂JFCCI (uci)=∂uci) = 0 and the second sufficient condition: ∆2JFCCI(U\*) is positive definite. There-fore ucin updated is indeed a local minima of JFCCI(U) and it never increases the objective function value.

Theorem 2. For every iteration the updated values of vcj given by Eq. (11) never increases the objective function.

Proof. Proof is similar to proof of Theorem 1.

Theorem 3. The following constraint is satisfied by JFCCI in (4):



Proof. Since the minimum value of uci and vcj is 0, and Dcij≥0, the first term of JFCCI reduces to



The second and third terms denote the entropy values and maximum value of entropy occurs when uci =1/C and vcj =1/K.

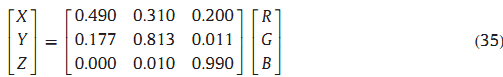
|  |  |  |
| --- | --- | --- |
| In view of these the point of minima is |  |  |



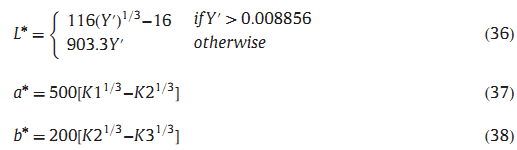
Corollary. Theorems 1–2 prove that the updated equations of FCCI point to a local minima of the Objective function and Theorem 3 indicates the lower limit of JFCCI.

Appendix B

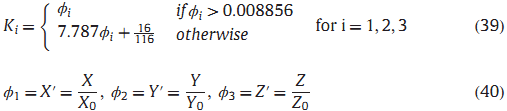
The CIELAB color space is obtained from the RGB color space by the following transformation:



The features of CIELAB are derived from



Where



The X0, Y0 and Z0 are the values of X,Y,Z for the reference white, respectively. The reference white is defined as {R =G=B =255}.