Journal of Computer Science 4 (8): 699-705, 2008
ISSN 1549-3636

Graph Based Segmentation in Content Based Image Retrieval

[1]P.S.Suhasini, [2]K. Sri Rama Krishna and [3]I.V. Murali Krishna
[1]Asst. Professor in ECE , Daita Madhusudana Sastry Sri Venkateswara Hindu College of Engineering , Machilipatnam, A.P., India -521001
[2] Dean, R& D, Vignan's Engineering College , Vadlamudi, A.P., India – 522213
[3]Professor, Jawaharlal Nehru Technological University, Hyderabad, A.P., India –500072

Abstract: Problem statement: Traditional image retrieval systems are content based image retrieval systems which rely on low-level features for indexing and retrieval of images. CBIR systems fail to meet user expectations because of the gap between the low level features used by such systems and the high level perception of images by humans. To meet the requirement as a preprocessing step Graph based segmentation is used in Content Based Image Retrieval (CBIR). **Approach:** Graph based segmentation is has the ability to preserve detail in low-variability image regions while ignoring detail in high-variability regions. After segmentation the features are extracted for the segmented images, texture features using wavelet transform and color features using histogram model and the segmented query image features are compared with the features of segmented data base images. The similarity measure used for texture features is Euclidean distance measure and for color features Quadratic distance approach. **Results:** The experimental results demonstrate about 12% improvement in the performance for color feature with segmentation. **Conclusions/Recommendations**: Along with this improvement Neural network learning can be embedded in this system to reduce the semantic gap.

Key words: Graph Based Segmentation, Quadratic distance approach, histogram

INTRODUCTION

As processors become increasingly powerful, and memories become increasingly cheaper, the deployment of large image databases for a variety of applications have now become realizable. Databases of art works, satellite and medical imagery have been attracting more and more users in various professional fields like geography, medicine, architecture, advertising, design, fashion, and publishing. Effectively and efficiently accessing desired images from large and varied image databases is now a necessity.

Many approaches have been proposed such as the text- based retrieval and the content based image retrieval (CBIR). Text-based approach consists to attach keywords or labels to each item and then to perform searches based in on these labels. CBIR approach extracts low-level features to index image such as color, texture and shape[2]. However, these approaches are inefficient due to the gap between visual features and semantic concepts. Several systems are proposed to improve the retrieval quality.

Relevant feedback approach was used in text-based information retrieval and was introduced to CBIR to bring user in the retrieval process for reducing the semantic gap between what queries represent (low-level features) and what the user thinks[10]. In order to derive high-level semantic features, machine learning techniques have been introduced in CBIR such as neural network for concept learning.

As a preprocessing step in clustered CBIR using Neural networks Graph Based Segmentation is used. There is a considerable progress in Eigenvector-based methods of image segmentation (e.g.,[13, 15]), these methods are too slow to be practical for many applications. In contrast, the Graph based segmentation has been used in large-scale image database applications as described in[12]. While there are other approaches to image segmentation that are highly efficient, these methods generally fail to capture perceptually important non-local properties of an image. The Graph based segmentation both captures certain perceptually important non-local image characteristics and is computationally.

Principle of CBIR: Content-based image retrieval, also known as query by image content and content-based visual information retrieval is the application of

Corresponding Author: Suhasini, P.S., DMSSVH College of Engineering, Machilipatnam-521001, A.P., India computer vision to the image retrieval problem, that is, the problem of searching for digital images in large databases. Content-based means that the search makes use of the contents of the images themselves, rather than relying on human-input metadata such as captions or keywords. A *content-based image retrieval system* (CBIR) is a piece of software that implements CBIR.

There is a growing interest in CBIR because of the limitations inherent in metadata-based systems. Textual information about images can be easily searched using existing technology, but requires humans to personally describe every image in the database. This is impractical for every large databases, or for images that are generated automatically, e.g. from surveillance cameras. It is also possible to miss images that use different synonyms in their descriptions. Systems based on categorizing images in semantic classes like “cat” as a subclass of “animal” avoid this problem but still face the same scaling issues.

Different implementations of CBIR make use of different types of user queries.

- With *query by example,* the user searches with a query image (supplied by the user or chosen from a random set), and the software finds images similar to it based on various low-level criteria.
- With *query by sketch,* the user draws a rough approximation of the image they are looking for, for example with blobs of color, and software locates the images whose layout matches the sketch.
- Other methods include specifying the proportions of colors desired (e.g. “80% red, 2% blue”) and searching for images that contain an object given in a query image.

In CBIR each image that is stored in the database has its features extracted and compared to the features of the query image. It involves two steps.

Feature Extraction: The first step in this process is to extract the image features to a distinguishable extent.

Feature Matching: The second step involves matching these features to yield a result that is visually similar.

Block diagram: Basic idea behind CBIR is that, when building an image database, feature vectors from images (the features can be color, shape, texture, region or spatial features, features in some compressed domain, etc.) are to be extracted and then store the vectors in another database for future use. When given a query image its feature vectors are computed. If the distance between feature vectors of the query image and image in the database is small enough, the corresponding image in the database is to be considered as a match to the query. The search is usually based on similarity rather than on exact match and the retrieval results are then ranked accordingly to a similarity index. The block diagram of basic CBIR system is as shown in Fig. 1.

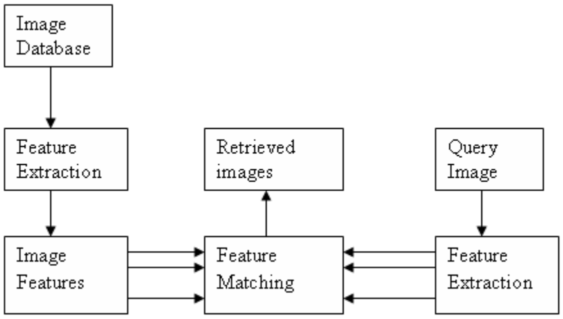

Fig. 1: Block diagram of CBIR

MATERIALS AND METHODS

Graph based segmentation in CBIR: All current CBIR techniques assume certain mutual information between the similarity measure and the semantics of the images. A typical CBIR system ranks target images according to the similarities with respect to the query and neglects the similarities between target images. For improving performance of image retrieval system considering this aspect graph based segmentation is proposed as preprocessing step in CBIR for image retrieval, retrieves segmented images instead of a set of ordered images: The query image and target images, which are selected according to a similarity measure and returned to the user.

Block diagram:

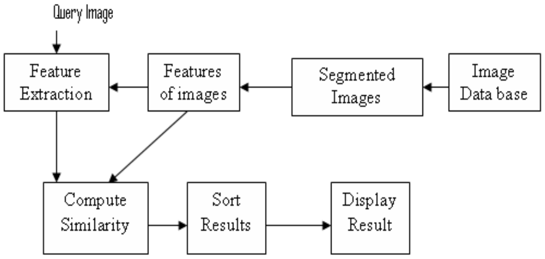

Fig. 2: Block Diagram of Graph based segmentation in CBIR

The block diagram of CBIR system with segmentation is as shown in Figure 2. The retrieval process starts with feature extraction for a segmented query image. The features for target images (images in the segmented database) are usually pre computed and stored as feature files. Using these features together with an image similarity measure, the resemblance between the query image and target images are evaluated and sorted. Next, a collection of target images that are close to the query image are selected as the neighborhood of the query image. Finally, the system displays the segmented images. The major difference between a graph based segmented image retrieval system and CBIR systems is, here we first segment the image data base and improving performance of CBIR system. The system not only improving the performance and also improve the speed of retrieval.

Principle of graph based segmentation: Let G = (V,E) be an undirected graph with vertices vi € V , the set of elements to be segmented, and edges (vi; vj) € E corresponding to pairs of neighboring vertices. Each edge (vi; vj) € E has a corresponding weight w((vi; vj)), which is a non-negative measure of the dissimilarity between neighboring elements vi and vj . In the case of image segmentation, the elements in V are pixels and the weight of an edge is some measure of the dissimilarity between the two pixels connected by that .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'), In other words, any segmentation is induced by a subset of the edges in E.

Segmentation algorithm: The input is a graph G = (V,E), with n vertices and m edges. The output is a segmentation of V into components S = (C_1, . . . ,C_r).

0. Sort E into $\pi = (o_1 , . . . , o_m)$, by non-decreasing edge weight.

1.

- Start with a segmentation S^0, where each vertex v_i is in its own component.
- Repeat step 3 for q = 1, . . . ,m.
- Construct S^q given S^{q-1}as follows. Let v_i and v_j denote the vertices connected by the q-th edge in the ordering, i.e., $o_q = (v_i, v_j)$. If v_i and v_j are in disjoint components of $S^{q-1}$and $w(o_q)$ is small compared to the internal difference of both those components, then merge the two components otherwise do nothing. More formally, let $C_i^{\ q-1}$be the component of S^{q-1}containing v_i and $C^{q-1}{}_j$ the component containing v_j . If $C_i^{\ q-1} = C^{q-1}{}_j$ and $w(o_q) \leq M\ Int(C_i^{\ q-1},C^{q-1}{}_j)$ then Sq is obtained from $S^{q-1}$by merging $C_i^{\ q-1}$ and $C^{q-1}{}_j$. Otherwise $S^q = S^{q-1}$.
- Return $S = S^m$.

Feature extraction:

Color: One of the most important features that make possible the recognition of images by humans is color. Color is a property that depends on the reflection of light to the eye and the processing of that information in the brain.

Color histogram: The color histogram serves as an effective representation of the color content of an image if the color pattern is unique compared with the rest of the data set. The color histogram is easy to compute and effective in characterising both the local and global distribution of colors in an image. In addition, it is robust to translation and rotation about the view axis and changes only slowly with the scale, occlusion and viewing angle. Since any pixel in the image can be described by three components in a certain colour space (for instance, red, green and blue components in RGB space or hue, saturation and value in HSV space), a histogram, i.e., the distribution of the number of pixels for each quantized bin, can be defined for each component. Clearly, the more bins a color histogram contains the more discrimination power it has. However, a histogram with large number of bins will not only increase the computational cost, but will also be in appropriate for building efficient indexes for image data base.

From the color map each row represents the color of a bin. The row is composed of the three coordinates of the color space. The first coordinate represents hue, the second saturation, and the third, value, thereby giving HSV. The percentages of each of these coordinates are what make up the color of a bin. Also one can see the corresponding pixel numbers for each bin, which are denoted by the blue lines in the histogram.

Quantization in terms of color histograms refers to the process of reducing the number of bins by taking colors that are very similar to each other and putting them in the same bin. By default the maximum number of bins one can obtain using the histogram function in MatLab is 256. For the purpose of saving time when trying to compare color histograms, one can quantize the number of bins. Obviously quantization reduces the information regarding the content of images but as was mentioned this is the tradeoff when one wants to reduce processing time.

Texture: Texture is that innate property of all surfaces that describes visual patterns, each having properties of homogeneity. It contains important information about the structural arrangement of the surface, such as; clouds, leaves, bricks, fabric, etc. It also describes the relationship of the surface to the surrounding environment. In short, it is a feature that describes the distinctive physical composition of a surface.

Pyramid structured wavelet transform for texture feature extraction: The pyramid-structured wavelet transform is used for texture classification. Its name comes from the fact that it recursively decomposes sub signals in the low frequency channels. It is mostly significant for textures with dominant frequency channels. For this reason, it is mostly suitable for signals consisting of components with information concentrated in lower frequency channels. Due to the innate image properties that allows for most information to exist in lower sub-bands, the pyramid-structured wavelet transform is highly sufficient.

Energy level algorithm:

- Decompose the image into *four* sub-images
- Calculate the energy of all decomposed images at the same scale, using:
-

$$E = \frac{1}{MN}\sum_{i=1}^{m}\sum_{j=1}^{n}\left|X(i,j)\right|$$

Where:
M and N = The dimensions of the image
X = The intensity of the pixel located at row i and column j in the image map.

- Repeat from step 1 for the low-low sub-band image, until 5 for five levels of decomposition.

Using the pyramid-structured wavelet transform, the texture image is decomposed into four sub images, in low-low, low-high, high-low and high-high sub-bands. At this point, the energy level of each sub-band is calculated. This is first level decomposition. Using the low-low sub-band for further decomposition, we reached fifth level decomposition, for our project. The reason for this is the basic assumption that the energy of an image is concentrated in the low-low band. For this reason the wavelet function used is the Daubechies wavelet.

Feature matching:

Similarity measure for color: content-based image retrieval calculates visual similarities between a query image and images in a database. Accordingly, the retrieval result is not a single image but a list of images ranked by their similarities. The result is not a single image, but a list of images that have been developed for image retrieval based on empirical estimates of the distribution of features in recent years. Different *similarity/distance measures* will affect retrieval performances of an image retrieval system significantly.

Minkowski- form distance: If each dimension or image features vector is independent of each other and is of equal importance, the Minkowski- form distance Lp is appropriate for calculating the distance between two images.,

Let D(I, J) be the distance measure between the query image I and the image J in the database; and fi(I) as the number of pixels in bin i of I .This distance is defined as:

$$D\ (I,J) = (\sum \mid f_i\ (I) - f_i(J) \mid^p)^{1/p}$$

When p=1, 2,…..∞, D(I, J) is the L1, L2 (also called Euclidean distance and L∞ distance respectively. Minkowski-form distance is the mot widely used metric for image retrieval.

Quadratic Form (QF) distance: The Minkowski distance treats all bins of the feature histogram entirely independently and does not account for the fact that certain pairs of bins correspond to features which are perceptually more similar than other pairs. To solve this problem, quadratic form distance is introduced:

$$\overline{D(I,J) =\sqrt{}\ (F_I\text{-}F_J)^T\ A\ (F_I}\text{-}F_J)$$

Where:
A = $[a_{ij}]$ is a similarity matrix,
a_{ij} = denotes the similarity between bin I and j
Fi and Fj = Vectors that list all the entries in fi (I) and fi (J).

Quadratic form distance has been used in many retrieval systems for color histogram-based image retrieval. It has been shown that quadratic form distance can lead to perceptually more desirable results than Euclidean distance and histogram intersection method as it considers the cross similarity between colors.
A simple distance metric involving the subtraction of the number of pixels in the 1st bin of one histogram

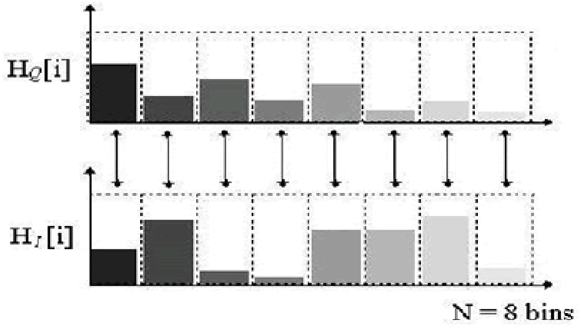

Fig. 3: Minkowski distance approach

from the 1st bin of another histogram and so on is not adequate. This metric is referred to as a Minkowski-Form Distance Metric, shown below in Fig..3, which only compares the same bins between color histograms.

This is the main reason for using the quadratic distance metric. More precisely it is the middle term of the equation or similarity matrix ***A*** that helps us overcome the problem of different color maps. The similarity matrix is obtained through a complex algorithm:

$$a_{q,i} = 1 - \frac{\left[\left(v_q - v_i\right)^2 + \left(s_q \cos\left(h_q\right) - s_i \cos\left(h_i\right)\right)^2 + \left(s_q \sin\left(h_q\right) - s_i \sin\left(h_i\right)\right)^2\right]^{\frac{1}{2}}}{\sqrt{5}}$$

which basically compares one color bin of H_Q with all those of H_I to try and find out which color bin is the most similar, as shown below in Fig. 4:

This is continued until we have compared all the color bins of H_Q. In doing so we get an N x N matrix, N representing the number of bins. What indicates whether the color patterns of two histograms are similar is the diagonal of the matrix. If the diagonal entirely consists of ones then the color patterns are identical. The farther the numbers in the diagonal are from one, the less similar the color patterns are. Thus the problem of comparing totally unrelated bins is solved.

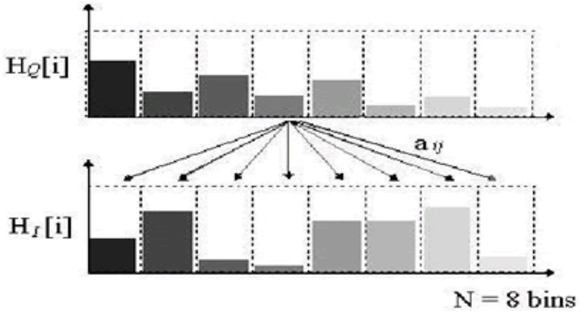

Fig. 4: Quadratic distance approach

Similarity measure for texture: Euclidean Distance Algorithm:

- Decompose query image.
- Get the energies of the first dominant k low frequency components.
- For image i in the database obtain the k energies.
- Calculate the Euclidean distance between the two sets of energies, using

$$D_i = \sum_{k=1}^{k} \left(x_k - y_{i,k}\right)^2$$

Increment i. Repeat from step 3: Using the above algorithm, the query image is searched for in the image database. The Euclidean distance is calculated between the query image and every image in the database.

RESULTS

Database:

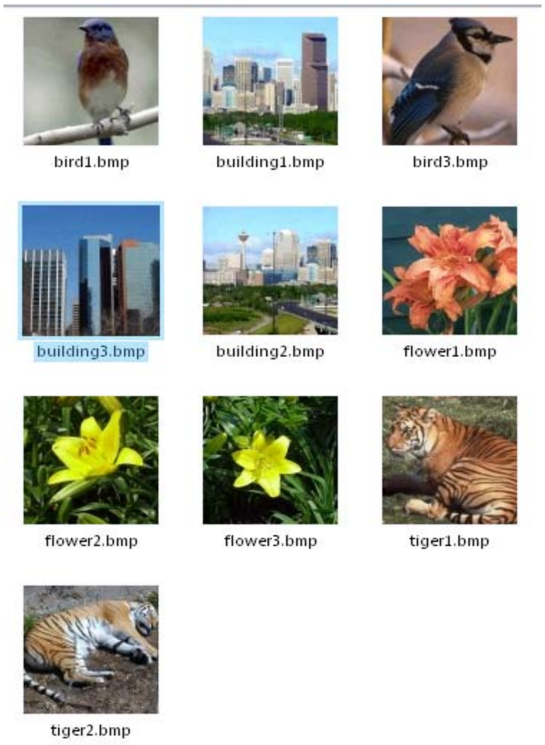

Fig. 5: Sample Database on which the experiment is performed

Results of retrieval of CBIR without segmentation:

Query

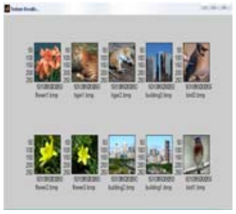

Fig. 6(a): Retrieved images based on color

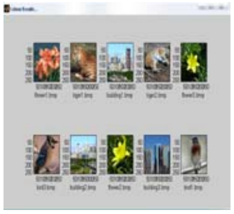

Fig. 6(b): Retrieved Image based on texture

Results of retrieval of CBIR with Segmentation:

Fig. 6(c): Retrieved Images Based on Color

Fig. 6(d): Retrieved Images based on texture

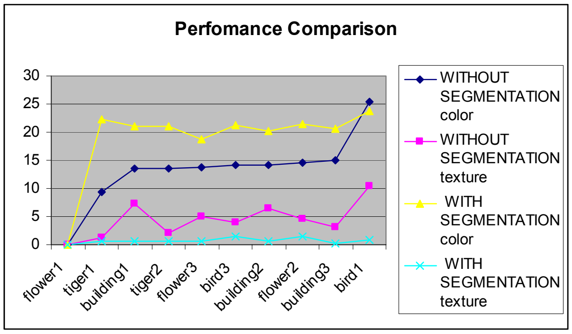

Fig.7: Performance comparison of CBIR & CBIR with Segmentation

DISCUSSION

Figure 6(a) and Fig. 6(b) are the retrieval results respectively based on color feature and texture feature without segmentation and Fig. 6(c) and Fig. 6(d) are the retrieval results respectively based on color feature and texture feature with segmentation. From the performance graph the distance measure is zero for the given query and the similar image in the database and there is increased distance for the dissimilar images particularly for CBIR with segmentation with color feature.

CONCLUSION

Graph based segmentation is used as pre-processing step in CBIR and then color and texture features are extracted. Color, texture, shape, spatial relationship and other single low-level features can only describe parts of image content; sometimes the retrieval results are not satisfied. Combining low-level features in retrieval has a lot of advantages: different features can complement each other; can enhance the system retrieval precision, make CBIR system more agile.

In this study a simple color-based search in an image database for an input query image, is performed using color histograms. It then compares the color histograms of different images using the Quadratic Distance Equation. Further enhancing the search, the application performs a texture-based search in the color results, using wavelet decomposition and energy level calculation. It then compares the texture features obtained using the Euclidean Distance measure and the results are compared with CBIR system without segmentation. With segmentation the performance is found to be increased especially using color feature.

In conventional CBIR systems, similarities among target images are usually ignored. As future work, clustering using Neural networks after segmentation for fully exploiting similarity information. Semantic gap is a challenging task in CBIR since the features from image data are low level visual characteristics which have very limited ability in representing and analyzing the high level semantic content of the image. Neural network learning can be embedded in this system to reduce the semantic gap.

REFERENCES

1. Ruan, N., N. Huang and W. Hong, 2006. Semantic-Based Image Retrieval in Remote Sensing Archive: An Ontology Approach, Geosciences and Remote Sensing Symposium, IGARSS 2006. pp: 2903-2906. DOI:10.1109/IGARSS.2006.746
2. Fuhuri Long, Hongjiang and David Dagan Feng, 2003. Fundamental of content-based ImageRetrieval; Chapter1, http://www.cse.iitd.ernet.in/~pkalra/siv864/Projects/ch01_Long_v40-proof.pdf
3. Rui, Y. and T.S. Huang, 2000. Optimizing learning in image retrieval. Proceedings of the IEEE International Conference on Computer Vision and PatternRecognition, pp: 1236-1243. Doi: 10.1016/j.patcog.2006.04.045
4. Rui, Y., T.S. Huang and S.F. Chang, 1999. Image retrieval: current techniques, promising directions, and open issues. J. Visual Commun. Image Represent., pp: 39-62. Article ID jvci.1999.0413, http://www.idealibrary.com

5. Vailaya, A., M.A.T. Figueiredo, A.K. Jain and H.J. Zhang, 2001. Image classification for content-based indexing, IEEE Trans. Image Process, pp: 117-130. DOI: 10.1109/83.892448
6. Th. Gevers and A.W.M. Smeulders, 2004. Emerging Topics in Computer Vision, chapter Image Search Engines: An Overview. Prentice Hall. http://mmlab.ceid.upatras.gr/courses/AIS_SITE/files/projects2004/SearchEngines/Image_Search_Engines_Survey(2003).pdf
7. Zheng, W., Y. Ouyang, J. Ford and Fillia S. Makedon, 2003. Ontology-based Image Retrieval, WSEAS MMACTEE-WAMUS-NOLASC 2003, Vouliagmeni, Athens, Greece, December29-31. http://devlab.dartmouth.edu/labpapers/accepted/WSEAS03/ontologyImage.pdf
8. Liu, Y., D. Zhang, G.Lu and W.Y. Ma, 2007. A survey of content-based image retrieval with high-level semantics. J. Recognition Soci., pp: 262- 282 DOI:10.1016/j.patcog.2006.04.045
9. Chen, Y., J.Z. Wang and R. Krovetz, 2003. An unsupervised learning approach to content-based image retrieval. IEEE Proceedings of the International Symposium on Signal Processing and its Applicationspp, pp: 197-200. DOI: 10.1109/ISSPA.2003.1224674
10. Rui, T.S. Huang, M. Ortega and S. Mehrotra, 1995. Relevance feedback: A power tool for interactive content-based image retrieval. IEEE Trans. Circuits Video Technol., 8: 644-655. DOI: 10.1109/76.718510
11. Pedro F. Felzenszwalb and Daniel P. Huttenlocher, 2004. Efficient graph-based image segmentation. Int. J. Comput. Vision, 59: 167-181. DOI: 10.1023/B:VISI.0000022288.19776.77
12. Ratan, A.L., O. Maron, W.E.L. Grimson and T. Lozano-Perez, 1999. A framework for learning query concepts in image classi¯cation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp: 423-431. DOI: 10.1109/CVPR.1999.786973
13. Shi, J. and J. Malik, 1997. Normalized cuts and image segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp: 731-737. DOI: 10.1109/34.868688
14. Urquhart, R., 1982. Graph theoretical clustering based on limited neighborhood set. Pattern Recognition, 15: 173-187.
15. Weiss, Y., 1999. Segmentation using eigenvectors: A unifying view, Proceedings of the International Conference on Computer Vision (2), pp: 975-982. www.cs.huji.ac.il/~yweiss/iccv99.pdf