Original Research Paper

Volume-9 | Issue-4 | April-2019 | PRINT ISSN No 2249-555X

Computer Science

OBJECT EXTRACTION AND FACE DETECTION BASED ON GEOMETRIC FEATURES OF IMAGE IN A REGION

Cheekatla. SwapnaPriya	Department of Computer science and Engineering, Vignan's institute of information Technology, Visakhapatnam-530049, India.
Koppula. Prameela*	Department of Computer science and Engineering, Vignan's institute of information Technology, Visakhapatnam-530049, India. *Corresponding Author
Kothapalli Sameer	Department of Computer science and Engineering, Vignan's institute of information Technology, Visakhapatnam-530049, India.
Banavathu. Mounika	Department of Computer science and Engineering, Vignan's institute of information Technology, Visakhapatnam-530049, India.
Pilaka. Anusha	Department of Computer science and Engineering, Vignan's institute of information Technology, Visakhapatnam-530049, India.

ABSTRACT Image detection and recognition is challenging due to the Wide variety of faces and the complexity of noises and image backgrounds. In this paper, to detect image we propose methods like region based segmentation methods, Data clustering, Thresholding and Edge-based segmentation methods. The proposed method has good performance good recognition rate.

KEYWORDS : Image Detection, Segmentation, Clustering, Thresholding, Region Of Interest.

1. Introduction

Image processing refers, to reprocessing of digital images by using digital computers. It mainly focuses on 2 tasks:

- To improve the pictorial information for human interpretation
- Processing of image data for storage

An image is a two dimensional function *f(x, y)*, where x and y are spatial (plane) coordinates, and the amplitude of *f* at any pair of coordinates*(x, y)*is called the intensity of the image at that level. If *x ,y* and the amplitude values of *f* are finite and discrete quantities, we call that image as a "digital image"[1]. It is a representation of two dimensional images as a finite set of digital values called pixels or picture elements. Pixel has a particular location and value. To denote the elements of digital image we are using pixels. In 8bit representation pixel intensity values changes between 0(black) and 255(white).And pixel values represent gray level, colours etc. The principal source for the images is the Electro Magnetic energy spectrum.

- The spectral bands are grouped according to energy per photon ranging from the Gamma rays (highest energy) to the radio waves (lowest energy).

Gamma Ray Imaging: It is used in nuclear medicine and astronomical observation. X-Ray Imaging: It is used in medical diagnostic, industrial applications and Astronomy. In this we will process an image and doing some applications on an image like sharpening, brightening, highlighting edges is called as "Image Enhancement".

Dividing image into the subparts is known as "Image Segmentation" like finding lines, circles, identifying the things in an image like cars, trees, roads etc., In fig1 we can see the types of the image enhancement.

2. Methodology

Image detection: To extract selected image from the image we are going for these following methods. They are face detection, image segmentation, feature extraction, pattern recognition and face segmentation

Face detection: Face detection is a computer technology that determines the locations and sizes of human faces in digital images. It detects face and ignores anything else, such as buildings, trees and bodies. Face detection can be regarded as a more general case of face localization. In face localization, the task is to find the locations and sizes of a known number of faces (usually one). In face detection, face is processed and matched bitwise with the underlying face image in the database. Any slight change in facial expression, e.g. smile, lip movement, will not match the face.

Detection methods These categories may overlap, so an algorithm could belong to two or more categories. This classification can be made as follows:

- Knowledge-based methods. Ruled-based methods that encode our knowledge of human faces.
- Feature-invariant methods. Algorithms that try to find invariant features of a face despite its angle or position.
- Template matching methods. These algorithms compare input images with stored patterns of faces or features.
- Appearance-based methods. A template matching method whose pattern database is learnt from a set of training images.

Image Segmentation:

It is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze.[5][6] Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics.

Thresholding method:

The simplest method of image segmentation is called the thresholding method. This method is based on a clip-level (or a threshold value) to turn a gray-scale image into a binary image. There is also a balanced histogram thresholding.

Clustering methods:

The K-means algorithm is an iterative technique that is used to partition an image into K clusters. In this case, distance is the squared or absolute difference between a pixel and a cluster centre. The difference is typically based on pixel color, intensity, texture, and location, or a weighted combination of these factors. K can be selected manually, randomly, or by a heuristic. This algorithm is guaranteed to converge, but it may not return the optimal solution. The quality of the solution depends on the initial set of clusters and the value of K.

Histogram-based methods

Histogram based methods are very efficient when compared to other image segmentation methods because they typically require only one pass through the pixels. In this technique, a histogram is computed from all of the pixels in the image, and the peaks and valleys in the histogram are used to locate the clusters in the image.[5] color or intensity can be used as the measure.

A refinement of this technique is to recursively apply the histogram-seeking method to clusters in the image in order to divide them into smaller clusters. This is repeated with smaller and smaller clusters until no more clusters are formed.[5][7]

Edge detection:
Edge detection techniques have therefore been used as the base of another segmentation technique. The edges identified by edge detection are often disconnected. To segment an object from an image however, one needs closed region boundaries. The desired edges are the boundaries between such objects.

Feature extraction:
When the input data to an algorithm is too large to be processed and it is suspected to be notoriously redundant (e.g. the same measurement in both feet and meters) then the input data will be transformed into a reduced representation set of features (also named features vector). Transforming the input data into the set of features is called feature extraction.

Face segmentation:.
Especially, face segmentation is an essential step of face recognition system since most face classification techniques tend to only work with face images. Therefore face segmentation has to correctly extract only face part of given large image. However, because of lots of variations of image appearance, such as pose variation, occlusion, image orientation, illuminating condition and facial expression, it generates difficulties for implementing such algorithm**.**

3. Approach
The input pattern and the distribution-based class representation in the chosen feature space. Algorithms like PCA or Fisher's Discriminant can be used to define the subspace representing facial patterns. In this work, we assumed that image will always have at least one face and size of face can be as small as 50 x 50 pixel wise. With this assumption, we've tried to use both feature-based and image-based approaches to maximize the probability of segmentation. Specifically, the process begins by converting a color image into a binary image of skin likelihood regions. This binary image is further processed using morphological operations in order to optimize it for face segmentation. Template matching is subsequently employed in order to find face candidates in the image.

Skin Color Analysis:
There are many ways to represent digital images in color space model**.** *RGB*: The name of the model and the abbreviation 'RGB' come from the three primary colors, red, green, and blue. *HSV* (Hue, Saturation, Value): And value represents brightness of the color.

YCrCb: In this model, luminance and chrominance components of the image are separated. Y is the luma component, and Cb and Cr are the blue and red chromed components.

In this work, we choose to work with YCrCbcolor model because of its effectiveness in our application. In this space, chrominance of the image can provide effective information for human skin color than any of two color space described above.

B. Analysis of YCrCbcolor space on human skin color: Any RGB digital image can be converted into YCrCBcolor space using following equation:

$$Y = 0.299R + 0.587G + 0.114B$$
$$Cb = -0.169R - 0.332G + 0.500B$$
$$Cr = 0.500R - 0.419G - 0.081B$$

Fig 4:Original Image

Fig:5 Skin color filter

Fig: Skin Color Map and Regularization
Skin Color Filter
As stated in previous section, means and standard deviations of Cr and Cb of 750 images are calculated. These values are used to set min and max allowance of given pixel intensity value. Means and standard deviations calculated are following:

$$Cb\_mean = -11.098$$
$$Cb\_STD = 4.265$$
$$Cr\_mean = 21.927$$
$$Cr\_STD = 4.143$$

With these numbers, min and max are calculated as following:

$$Cb\_min = Cb\_mean - Cb\_STD * 2$$
$$Cb\_max = Cb\_mean + Cb\_STD * 2$$
$$Cr\_min = Cr\_mean + Cr\_STD * 2$$
$$Cr\_max = Cr\_mean + Cr\_STD * 2$$

Addition of standard deviation to mean maximizes boundary conditions. After min and max are obtained, skin color filter can be easily created by simple operation. Consider an input image of M x N pixels. Since only Cr and Cb values are considered in this step, the output filter is the binary matrix of M/2 x N/2 size

$$init\_filter(x, y) = \{1, if[Cr(x,y) \varepsilon Rcr] \cap [Cb(x,y) \varepsilon Rcb]\}$$
$$RCr = [Cr\_min\ Cr\_max]$$
$Cb = [Cb\_min\ Cb\_max]$ Example:

Figure 4 and Figure 5 displays first step of implementation. As it can be seen from figure 4, it correctly spotted face part of image, but it also extracted other skin parts (neck and shoulder) of person and background noises. Also parts of face (eyes, eyebrows, and mouth) were not exactly. Finally, it can be noticed that her hair separated right ear from whole face because hair color doesn't belong to skin color space.

Regularization Process – Dilation and Erosion
The filter obtained from skin color analysis can be co corrupted by noise. The noise may appear as small holes on the facial region due to undetected facial features (eyes, mouth, or eyebrows), or background objects may appear as skin regions. Therefore, simple morphological operations such as dilation to fill in any small holes in the facial area and erosion to remove any small objects in the background area. To dilate and erode input image, the density distribution of initial filter has to be calculate.

$$D(x, y) = \sum\sum init\_filter(4x + i, 4y + j)$$
$$i = 0\ j = 0$$

According to the density value, each pixel in input image cacan be classified into three types: zero (D = 0), intermediate (0 < D < 16), and full (D = 16). After density distribution of initial filter is calculated, density regularization process begins.

1) Init - Set boundaries of density map to zero.
2) Dilate - If any point of either zero or intermediate density has two or more full density point in its local 3 x 3 neighbourhood, set that point equal to 16.
3) Erode - If any full density point is surrounded by less than five other full density points in its local 3 x 3 neighbourhood, set that point equal to zero.

Then this density map can again be converted to second filter.
That we can see in figure6 and figure7.

Fig 6 :Filled small hole

Fig 7:Eroded after filling

Already the map has reduced lots of noise initial filter contained. However, this step only removed small noises in either background and within face region. The filter still can't separate neck from face region.

Regularization Process – Edge Detection:

Sometimes skin color theory can extract too much information from input image if original image already contains high number of skin color pixels. In this case, edge detection on original image can be used to separate face region from connected skin region of person. In this work, we've used Roberts Cross Edge Detection algorithm to find edges of original image. Reference to this algorithm can be found in reference section.

Basic, idea of edge detection is following

$\text{tmp1} = \text{image}(x, y) - \text{image}(x+1, y+1)$

$\text{tmp2} = \text{image}(x+1, y) - \text{image}(x, y+1)$

$\text{edge}(x, y) = |\text{tmp1}| + |\text{tmp2}|$

This edge is then applied to filter with operator.

Fig 8:Edges of original image **Fig 9:Edge and Filtered combined**

Again combined filter must be eroded to maximize thin edge lines and small holes created from this step are removed by dilate operation.

Fig 9: Eroded image of figure 8 **Fig 10:Dilated image of figure 9**

Clearly, edge detection along with density regularization has improved initial filter to segment only skin part of input image. However, it is noticed that edge detection layer may create additional noise. In case of figure 10, we see that noise reoccurred in eye and eyebrows. Therefore, further studies are required to decide when to use edge detection.

1.1 Region-Based Segmentation Methods:

Region-based methods mainly have the assumption that the neighbouring pixels within one region have similar value. The common procedure is to compare one pixel with its neighbours. If a similarity criterion is satisfied, the pixel can be set belong to the cluster as one or more of its neighbours. The selection of the similarity criterion is significant and the results are influenced by noise in all instances. In this chapter, we discuss four algorithms: the Seeded region growing, the unseeded region growing, the Region splitting and merging, and the Fast scanning algorithm.

Seeded Region Growing:

The seeded region growing algorithm performs a segmentation of an image with examine the neighbouring pixels of a set of points, known as seed points, and determine whether the pixels could be classified to the cluster of seed point or not [24]. The algorithm procedure is as follows.

Step1: We start with a number of seed points which have been clustered into n clusters, called C1, C2, …,Cn. And the positions of initial seed points is set as p1, p2, …, p3.

Step:2: To compute the difference of pixel value of the initial seed point pi and its neighbouring points, if the difference is smaller than the threshold (criterion) we define, the neighbouring point could be classified into Ci, where i = 1, 2, …,n.

Step 3: Recompute the boundary of Ci and set those boundary points as new seed points pi (s). In addition, the mean pixel values of Ci have to be recomputed, respectively.

Step 4: Repeat Step2 and 3 until all pixels in image have been allocated to a suitable cluster.

Unseed Region Growing:

The unseeded region growing (URG) algorithm is a derivative of seeded region growing proposed by Lin et al. [25]. Their distinction is that no explicit seed selection is necessary. In the segmentation procedure, the seeds could be generated automatically. So this method can perform fully automatic segmentation with the added benefit of robustness from being a region-based segmentation. The steps of URG are as below

Step1: The process initializes with cluster C1 containing a single image pixel and the running state of the process compose of a set of identified clusters, C1, C2… Cn.

Step2: We define the set of all unsigned pixels which borders at least one of those clusters as:

$$S = \left\{ x \notin \bigcup_{i=1}^{n} C_i \wedge \exists k : N(x) \cap C_k \neq \varnothing \right\},$$

where N(x) are current neighbouring pixels of point x. Moreover, let

$$\delta(x, C_i) = \left| g(x) - \operatorname*{mean}_{y \in C_i} \left[g(y) \right] \right|,$$

measure

where g(x) denotes the pixel value of point x, and i is an index of the cluster such that N(x) intersect Ci

Step3: To choose a point z∈S and cluster Cj where j∈[1,n] such that

$$\delta(z, C_j) = \min_{x \in S, k \in [1,n]} \{ \delta(x, C_k) \}.$$

If δ (z,Cj) is less than the predefined threshold t, the pixel clustered to Cj. Else, we must select the most considerable similar cluster C such that

$$C = \arg\min_{C_k} \{ \delta(z, C_k) \}.$$

If δ (z,C)<t , then we can allocate the pixel to C. If neither of two conditions conform, it is obvious that the pixel is substantially from all the clusters found so far, so that a new cluster Cn+1 would be generated and initialized with point z.

Ste4: After the pixel has been allocated to the cluster, the mean pixel value of the cluster must be updated.

Step5: Iterate Step2 to 4 until all pixels have been assigned to a cluster.

Region Splitting and Merging: The main goal of region splitting and merging is to distinguish the homogeneity of the image [26]. Its concept is based on quadtrees, which means each node of trees has four descendants and the root of the tree corresponds to the entire image. Besides, each node represents the subdivision of a node into four descendant nodes. The instance is shown in Fig. 13(a), and in the case of Fig. 13(b), only R4 was subdivided further. The basics of splitting and merging are discussed below.

Let R represent the entire image region and decide a predicate P. The purpose is that if

P(R)=FALSE , we divide the image R into quadrants. If P is FALSE for any quadrant, we subdivide that quadrant into sub quadrants, and so on. Until that, for any region Ri, P (Ri`0=TRUE. After the process of splitting, merging process is to merge two adjacent regions Rj and Rk

if p(R_j∪R_k)=TRUE The summarized procedure is described as follows:

Step1: Splitting steps: For any region Ri, which which P(Ri)= FALSE, we split it into four disjoint quadrants.

Step2: Merging steps: When no further splitting is possible, merge any adjacent regionsRj and Rk for which $p(R\_j \cup R\_k)$=TRUE.

Step3 : Stop only if no further merging is possible.

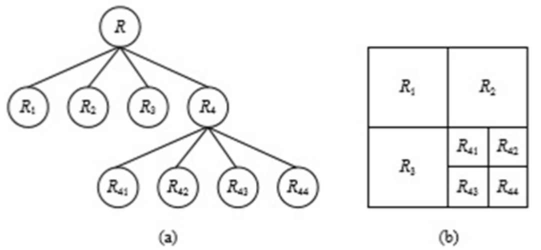

Fig. 13 (a) The structure of quad tree, where R represents the entire image region.
(b) Corresponding partitioned image. [1]

Fast Scanning Algorithm
Unlike region growing, fast scanning algorithm do not need seed point. The concept of fast scanning algorithm [27] is to scan from the upper-left corner to lower-right corner of the whole image and determine if we can merge the pixel into an existed clustering. The merged criterion is based on our assigned threshold. If the difference between the pixel value and the average pixel value of the adjacent cluster is smaller than the threshold, then this pixel can be merged into the cluster. The threshold usually chooses 45. We describe the steps of the fast scanning algorithm as below.

Step1: Let the upper left pixel as the first cluster. Set the pixel (1, 1) in the image as one cluster Ci and the pixel which we are scanning as Cj. We give an example in Fig. 14 and assume that the threshold is 45 here
Step2:In the first row, we scan the next pixel (1, 1+1) and determine if it can be merged into the first cluster or become a new cluster according to the threshold. The judgments are in the following, where mean represents the average pixel value of cluster Ci.

If |C_j-mean(C_i)|≤threshold then we merge C_j into C_i and recalculate the mean of C_i.Fig. 14(b) shows this case.
If|C_j-mean(C_i)|>threshold then we set C_j as a new cluster C_(i+1). This case is shown in Fig. 14©.

Step3: Repeat Step 2 until all the pixels in the first row have been scanned.

Step4: To scan the pixel(x+1,1) in the next row and compare this pixel with the cluster Cu which is in the upside of it. And determine if we can merge the pixel (x+1, 1) into the cluster Cu. (In the 2nd row, x is 1 and it increases with iteration).
If |C_j-mean(C_u)|≤threshold then we merge C_j into C_u and recalculate the mean of C_u .
If|C_j-mean(C_u)|>threshold then we set C_j as a new cluster C_n.
Where n is the cluster number so far. Fig. 14(e) shows above two situations.

Step5: Scan the next pixel (x+1, 1+1) and compare this pixel with the cluster Cu and Cl, which is in the upside of it and in the left side of it, respectively. And decide if we can merge the pixel (x+1, 1+1) into anyone of two clusters.
If|C_j-mean(C_u)|≤threshold&|C_j-mean(C_l)|≤threshold,
1. We merge Cj into Cu or merge Cj into Cl.
2. Merge the cluster Cu and Cl into cluster Cn, where n is the cluster number so far.
3. Recompute the mean of Cn.

This case is shown in Fig. 14(f).
If |C_j-mean(C_u)|≤threshold&|C_j-mean(C_l)|>threshold,we merge Cj into Cu and recalculate the mean of Cu.
If|C_j-mean(C_u)|>threshold&|C_j-mean(C_l)|≤threshold, we merge Cj into Cl and recalculate the mean of Cl.

Otherwise, set Cj as a new cluster Cn, where n is the cluster number so far.

Step6: Repeat Step 4 to 5 until all the pixels in the image have been scanned

Step7: Remove small clusters. If the number of C_m<Δ , we remove cluster m and assign the pixels in cluster m into adjacent clusters. The assignment is according to the smallest differences between the pixel and its mean of adjacent clusters. Fig. 14(g)(h) shows the small cluster case.

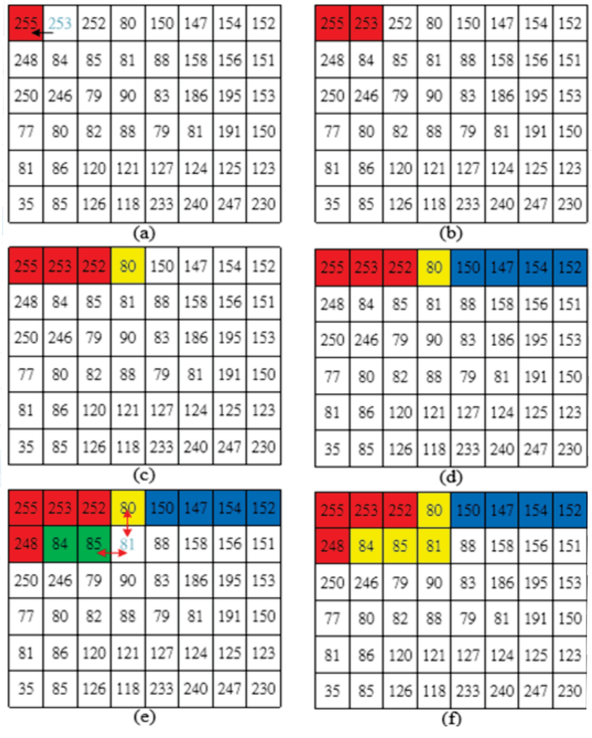

Fig. 9 Applying the fast scanning

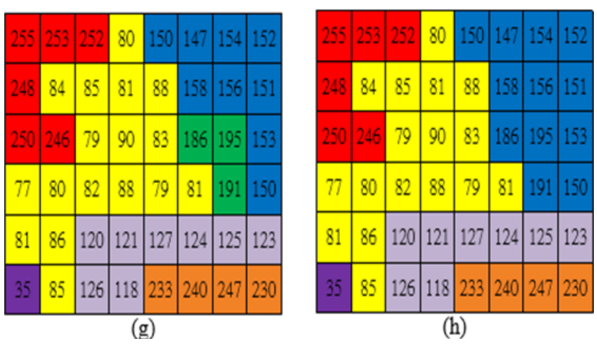

Fig. 14 applying the fast scanning algorithm to an example image

Data clustering:
The main concept of data clustering is to use the centroid to represent each cluster and base on the similarity with the centroid of cluster to classify.

Partitional Clustering:
The Partitional clustering algorithm obtains a single partition of the data. It is useful to implement in large data sets. The problem of Partitional clustering is that we have to select the number of desired output clusters before we start to classify data.
K-means Clustering Algorithm

The most famous Partitional clustering algorithm is k-means clustering. The steps of k-means clustering are as below.
Step1: Determine the number of clusters we want in the final classified result and set the number as N. Randomly select N patterns in the whole data bases as the N centroids of N clusters

Step2: Classify each pattern to the closest cluster centroid. The closest usually represent the pixel value is similarity, but it still can consider other features

Step3: Recompute the cluster centroids and then there have N centroids of N clusters as we do after Step1.

Step4: Repeat the iteration of Step 2 to 3 until a convergence criterion is met. The typical convergence criteria are: no reassignment of any pattern from one cluster to another.

Thresholding is the simplest and most commonly used method of segmentation. Given a single threshold, t,

the pixel located at lattice position (i,j), with grey scale value $f_{i,j}$ is allocated to category **if$f_{i,j} \leq$ t.**

Histogram-based thresholding
Intermeans algorithm:
Let us denote the histogram of pixel values by h0,h1,...,hN,

Where h, k specifies the number of pixels in an image with greyscale value k and N is the maximum pixel value (typically 255). the algorithm can be specified as follows.

Make an initial guess at t: for example, set it equal to the median pixel value, that is, the value for which

$$\sum_{k=0}^{t} h_k \geq \frac{n^2}{2} > \sum_{k=0}^{t-1} h_k,$$

Where $n^2$is the number of pixels in the n×n image.
2. Calculate the mean pixel value in each category. For values less than or equal to t, this is given by:

$$\mu_1 = \sum_{k=0}^{t} kh_k \Big/ \sum_{k=0}^{t} h_k \,.$$

Whereas, for values greater than t, it is given by:

$$\mu_2 = \sum_{k=t+1}^{N} kh_k \Big/ \sum_{k=t+1}^{N} h_k \,.$$

3.Re-estimate t as half-way between the two means, i.e t=[(μ1+μ2)/2] where, [ ] denotes 'the integer part of' the expression between the brackets.

4. Repeat steps (2) and (3) until t stops changing value between consecutive evaluations.

1.2 minimum-error algorithm:
This algorithm is based on this threshold and can be regarded as generalization of the inter means algorithm.
The algorithm can be specified as follows.
1. Make an initial guess at a value for t.
2. Estimate p1, μ1 and σ2 1 for pixels with values less than or equal to t, by

$$p_1 = \frac{1}{n^2} \sum_{k=0}^{t} h_k,$$

$$\mu_1 = \frac{1}{n^2 p_1} \sum_{k=0}^{t} kh_k,$$

$$\text{and } \sigma_1^2 = \frac{1}{n^2 p_1} \sum_{k=0}^{t} k^2 h_k \quad - \quad \mu_1^2.$$

Similarly, estimate p2, μ2 and σ2^2for pixels in the range t + 1 to N.
3.Re-estimate t by where A, B, C and [ ] have already been defined

$$t = \left[\frac{B + \sqrt{B^2 - AC}}{A} \right],$$

4. Repeat steps (2) and (3) until t converges to a stable value.

1.2 Edge-based segmentation:
The algorithm operates on a raster scan, in which each pixel is visited in turn, starting at the top-left corner of the image and scanning along each row, finishing at the bottom-right corner. For each non-edge pixel, (i,j), the following conditions are checked. If its already visited neighbours — (i−1,j) and (i,j −1) in the 4-connected case, also (i−1,j−1) and (i−1,j+ 1) in the 8-connected case — are all edge pixels, then a new category is created and (i,j) is allocated to it. Alternatively, if all its non-edge neighbours are in a single category, then (i,j) is also placed in that category. The final possibility is that neighbours belong to two or more categories, in which case (i,j) is allocated to one of them and a note is kept that these categories are connected and therefore should from then on be considered as a single category. More formally, for the simpler case of 4-connected regions:
Step1: Initialize the count of the number of categories by setting K = 0.

Step2: Consider each pixel (i,j) in turn in a raster scan, proceeding row by row (i =1,...,n), and for each value of i taking j =1,...,n .

Step3: One of four possibilities apply to pixel (i,j):

Step4: If (i,j) is an edge pixel then nothing needs to be done.

Step5: If both previously-visited neighbours, (i−1,j) and (i,j −1), are edge pixels, then a new category has to be created for (i,j):

Step4:K → K +1,h K = K, gij = K,

Step5: where the entries in h1,...,hK are used to keep track of which categories are equivalent, and gij records the category label for pixel (i,j).

Step6: If just one of the two neighbours is an edge pixel, then (i,j) is assigned the same label as the other one:

$$g_{ij} = \begin{cases} g_{i-1,j} & \text{if } (i, j-1) \text{ is the edge pixel,} \\ g_{i,j-1} & \text{otherwise.} \end{cases}$$

The final possibility is that neither neighbour is an edge pixel, in which case (i,j) is given the same label as one of them:

$g_{ij}=g_{i-1,j}$
and if the neighbours have labels which have not been marked as equivalent, i.e.h_(gi-1,j)≠h_(gi,j-1) ,then this needs to be done (because they are connected at pixel (i,j)). The equivalence is recorded by changing the entries in h1,...,hK, as follows:

Set l_1= min($h_{g_{i-1,j}}$,j,$h_{g_{i,j-1}}$) and l_2 = max($h_{g_{i-1,j}}$,j,$h_{g_{i,j-1}}$)

For each value of k from 1 to K, ifhk =l_2 then hk → l_1

Finally, after all the pixels have been considered, the array of labels is revised, taking into account which categories have been marked for amalgamation:
gij → hgij for i,j =1,...,n.

After application of the labelling algorithm, superfluous edge pixels — that is, those which do not separate classes — can be removed: any edge-pixel which has neighbours only of one category is assigned to that category.

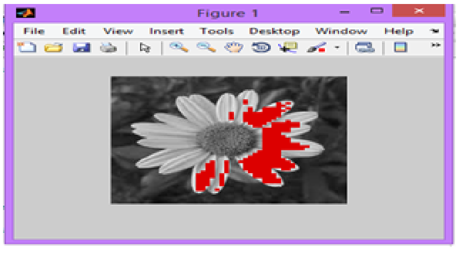

Fig15 split and merge for an image

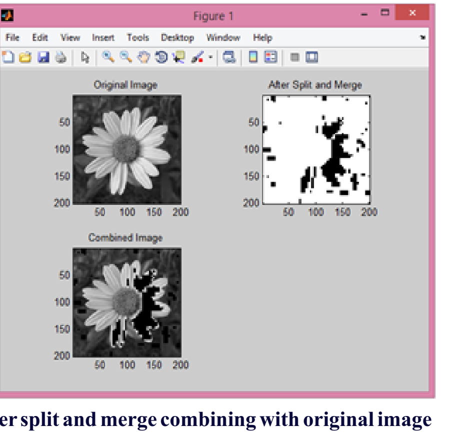

Fig 16 After split and merge combining with original image

Conclusion

In this work, to detect selected image from the image we are using many techniques like Knowledge-based methods, Feature invariant methods, template matching and appearance based methods. And to extract a image here many methods are using different methods they are mainly Image segmentation, Feature extraction and Pattern recognition.

Image detection technology has been integrated into a wide range of products and services, including online social networks, digital billboards, and mobile apps. One recent example: Face book has launched new facial recognition technology to help you troll through vast stores of pictures to tag people in the photos you post – or to help others tag you in the photos they post. Google rolled out its newest offering, Find My Face, at the event. The technology, which is now available to all Google+ users, scans photos that are uploaded to its social networking site and prompts the user to tag the faces of friends in the photo. And this report we shown face segmentation to detect face as parts with color ,dilation and erosion with some methods .

REFERENCES

[1] Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing", Prentice Hall, Upper Saddle River, New Jersey 07458

[2] http://homepages.inf.ed.ac.uk/rbf/HIPR2/filtops.htm

[3] http://www.cyanogen.com/help/maximdl/Low-Pass_Filtering.htm

[4] E. Davies Machine Vision: Theory, Algorithms and Practicalities, Academic Press, 1990, Chap. 9.

[5] Linda G. Shapiro and George C. Stockman (2001): “Computer Vision”, pp 279- 3 2 5 , New Jersey, Prentice-Hall, ISBN 0-13-030796-3

[6] Barghout, Lauren, and Lawrence W. Lee. "Perceptual information processing system."Paravue Inc. U.S. Patent Application 10/618,543, filed July 11, 2003.

[7] Ohlander, Ron; Price, Keith; Reddy, D. Raj (1978)."Picture Segmentation Using a Recursive Region Splitting Method".Computer Graphics and Image Processing 8 (3): 313–333. doi:10.1016/0146-664X(78)90060-6.

[8] .R. Louban. Image Processing of Edge and Surface Defects Theoretical Basis of Adaptive Algorithms with Numerous Practical Applications, volume 123, chapter Edge Detection, pages 9–29. Springer Berlin Hei- delberg, 2009.

[9] M.-H. Yang, D. Kriegman, and N. Ahuja. Detecting faces in images: A survey. IEEE Transactions on Pattern Analysis and Machine Intel- ligence, 24(1):34–58, January 2002.

[10] S. K. Singh, D. S. Chauhan, M. Vatsa, and R. Singh. A robust skin color based face detection algorithm. Tamkang Journal of Science and Engineering, 6(4):227–234, 2003.

[11] M. C. Nechyba, L. Brandy, and H. Schneiderman. Lecture Notes in Computer Science, volume 4625/2009, chapter PittPatt Face Detection and Tracking for the CLEAR 2007 Evaluation, pages 126–137. Springer Berlin / Heidelberg, 2009.

[12] C.-C. Han, H.-Y. M. Liao, K. chung Yu, and L.-H. Chen. Lecture Notes in Computer Science, volume 1311, chapter Fast face detection via morphology-based pre-processing, pages 469–476. Springer Berlin / Heidelberg, 1997.

[13] K. Wan-zeng and Z. Shan-an. Multi-face detection based on downsam- pling and modified subtractive clustering for color images. Journal of Zhejiang University SCIENCE A, 8(1):72–78, January 2007.

[14] R. Brunelli, Template Matching Techniques in Computer Vision: Theory and Practice, Wiley, ISBN 978-0-470-51706-2, 2009

[15] WANG, CHING YANG, Ph.D. "EDGE DETECTION USING TEMPLATE MATCHING (IMAGE PROCESSING, THRESHOLD LOGIC, ANALYSIS, FILTERS)". Duke University, 1985, 288 pages; AAT 8523046

[16] M. Turk and A. Pentland.Eigenfaces for recognition. Journal of Cog- nitiveNeurosicence, 3(1):71–86, 1991.

[17] K.-K. Sung.Learning and Example Selection for Object and Pattern Detection.PhD thesis, Massachusetts Institute of Technology, 1996.

[18] H. A. Rowley, S. Baluja, and T. Kanade.Neural network-based face detection. IEEE trans. Pattern Analysis and Machine Intelligence, 20(1):23–38, January 1998.

[19] F. Raphael, B. Olivier, and C. Daniel. A constrained generative model applied to face detection. Neural Processing Letters, 5(2):11–19, April 1997.

[20] http://ditto.us.com/what-is-image-recognition/

[21] R. Chellappa, C. L. Wilson, and S. Sirohey, “Human and machine recognition of faces: a survey,” Proc. IEEE, vol. 83, no. 5, pp. 705-740, 1995.

[22] Yu-Hsiang Wang , “Image Segmentation” Graduate Institute of Communication Engineering National Taiwan University, Taipei, Taiwan, ROC

[23] Reza Oji , ” An automatic algorithm for object recognition and detection based on asift key points” Department of Computer Engineering and IT, Shiraz University Shiraz, Iran.

[24] R. Adams, and L. Bischof, “Seeded region growing,” IEEE Trans. Pattern Anal. Machine Intell., vol. 16, no. 6, pp. 641-647, June, 1994.

[25] Z. Lin, J. Jin and H. Talbot, “Unseeded region growing for 3D image segmentation,” ACM International Conference Proceeding Series, vol. 9, pp. 31-37, 2000.

[26] S. L. Horowitz and T. Pavlidis, “Picture segmentation by a tree traversal algorithm,” JACM, vol. 23, pp. 368-388, April, 1976

[27] J. J. Ding, C. J. Kuo, and W. C. Hong, “An efficient image segmentation technique by fast scanning and adaptive merging,” CVGIP, Aug. 2009.

[28] Lifeng Liu and Stan Sclarof, “Region Segmentation via Deformable Model- Guided Split and Merge” , Boston University 11 1 Cummington St., Boston, MA 02215

[29] P. W. Fung, G. Grebbin, Y. Attikiouzel, MODEL-BASED REGION GROWING SEGMENTATION OF TEXTURED IMAGES , The University of Western Australia, Nedlands, Western Australia 6009

[30] M. Singh and B. Landau.Parts of visual shape as primitives for categorization.Behavioral and Brain Sci., 21(1):3637, 1998.