Recognition of Skin using Fuzzy Logic Approach

Muhammad Usman

UoB Number: 12036782

Submitted to: Dr. Malik Jahan Khan

Dated: 1st January 2016

Namal College Mianwali - Pakistan

Department of Computer Science

Abstract: Injecting self-dependent ability in the system was the rationale that gave rise to the field of 'Soft Computing'. In this area, robust systems are made to accommodate imprecision of real world scenarios. These systems are fabricated in such a way that they can learn by the passage of time and can improve their accuracy by learning. Skin recognition problem is being tacked by these systems and its solution is being vastly used as a pre-processing step in various applications. Different 'Machine Learning' approaches can be used to classify skin and non-skin cases in a given data-set, and fuzzy logic is also one of them. In this work, fuzzy inference system (FIS) is used to get the required results. To explain, Fuzzy C-mean (FCM) algorithm was used to cluster the training data-set that helped to design membership functions of the FIS. Also, rules were extracted from decision tree created using the clustered data. The proposed solution produces 97-98% accuracy on training (80%) and testing (20%) data.

Contents

1 Introduction

Making the machines intelligent enough to have accurate and precise decision power, is always a challenge for the researchers. The main idea behind these technologies is the complex perceptive capability of human brain that learns from environment and makes changes in itself. Also, our digital computers in hardware level, do the computations in 0s and 1s that is not the case in real world where we tackle 'Fuzzy' scenarios. Therefore, if we want to mimic the human brain, the ability to handle uncertain inputs must also be added to these system and 'Fuzzy logic' is being used for this purpose.

Skin segmentation is one of the most studied areas of research due to its implications in different areas. Identifying skin pixels in a given real time image could be a useful pre-processing step to reduce the workspace for face detection, face identification, hand gesture recognition, and security surveillance systems. For these complex systems, it is hard to do the required complex computations on each and every pixel. Therefore, in pre-processing step, skin pixels are identified first, and all the remaining computations are done on those specific areas[1].

In a given picture, skin pixel can be separated from the non-skin ones using intelligent color pattern identification systems. The rationale behind these detection techniques is that human skin colors follow a specific color pattern and if an intelligent system is learnt on that pattern, then the system can predict skin pixels in a given picture.

In this work, we have used clustering and decision tree to build a fuzzy inference system (FIS) that can predict a combination of R, G, B colors as skin or non-skin class. To elaborate, pre-processing on data was done to make clusters of data using c-mean algorithm. Then, decision tree was built on the clustered data and rules were extracted from that tree. After that, FIS was built and tested on testing and training data separately. Finally, to verify proposed system performance, another inference system using Adaptive Neuro Fuzzy Inference System toolbox provided in Matlab was built and the results of both these system were compared.

2 Background Knowledge and Literature Review

In the recent time, many techniques have been emerged to recognize skin in an image. The previous work is either predicting a particular pixel to be as skin or non-skin (Pixel based recognition), or the decision is being made based on the region of the image(region based). Region based recognition is mostly being used in face detection algorithm where pixel recognition becomes its pre-processing step.

Selamat et al. (2009)[11] have suggested another strategy to detect skin pixel in a given image. They have used set of fuzzy rules with Fuzzy Mamdani Inference system to differentiate skin pixels from the non-skin ones. Firstly, skin modelling was done using colour histogram method where the skin discrete probability of each colour channel was counted.This was helpful in making the fuzzy rules, and the colour channel with highest skin discrete probability of each colour histogram was used as centre of Gaussian membership function. Moreover, they have used 'Centre of gravity' as the de-fuzzification method. The discussed work was yielding good classification overall, but with pixels values closer to dark areas, the results were poor. In the mentioned work, the authors have given two possible values for every input variable (R,G,B), that is HIGH and LOW. This could be a reason behind low accuracy in dark areas, as the data is more diverse in case of skin detection, so there should be more than two values for each variable to get better results, but this will surely affect the overall efficiency.

Another proposed method by Iraji and Yavari (2009)[4] have used Fuzzy Mamdani Inference system with YCbCr colour space as luminance component (Y) is independent of color. They have figured out two possible values (Low and High) for each of the input variables (Y, Cb, Cr). After identifying the possible values and their ranges, fuzzy rules were designed. Again, due to insufficient number of possible values for variables YCbCr, the system is classifying many non-skin pixels as skin.

Likewise, as a pre-processing step to detect face in a given image, Hmid and Jemaa (2006)[2] have proposed a skin pixel detection solution using Sugeno fuzzy inference system. As they were using YCbCr colour space, the system had two input variables (cb and cr) each of them having three possible values (light, medium and dark) and one output variable. After detecting skin like pixels in the image, other face features (Edge detection and Ellipse detectio) were detected to recognize face images. This system was yielding better accuracy as number of possible values for input variables in this system was highe and the system was using YCbCr colour space which is considered to be a better candidate for skin segmentation problem.

Apart from these techniques, Bayes decision rule for minimum cost could also be used to classify skin color pixels from non-skin[3]. Also, using Fuzzy Clustering and Support vector machine, Juang et al. (2007) [6] have suggested a fuzzy system to detect skin (FS-FCSVM). Firstly, input data was clustered in the proposed system to reduce the number of rules later, and support vector machine was used to efficiently produce the required parameters of the system. The work had used an interesting solution for reducing the number of rules in their. Although, the proposed work have not used fuzzy inference system for later procedure, but the idea of clustering could also be used in inference system where we need to figure out rules mapping possible input data with outputs.

2.1 Fuzzy Inference System and its types

Ross (2009) [10] have explained, Fuzzy Inference system is based on the fuzzy logic that is used to relate entities having soft boundaries. FIS conducts inferencing of IF-THEN rules with fuzzy logic in a computer program. Also, the work describes types (Mamdani, Sugeno, Tsukamoto) of these systems. Out of these systems, Mamdani is being mostly used in different systems that does defuzzification of the fuzzy output to get an answer. But, Sugeno does not require defuzzification method due to its crisp output. This ability of Sugeno Model makes it an efficient option in fuzzy inference systems, but in most of the cases Mamdani system produces better accuracy[10].

2.2 Data Clustering

Clustering is the unsupervised classification of a given data and being widely used in Pattern analysis, classification problems and Machine learning[5]. The process iteratively groups the given data-set in clusters on the basis of certain similarities. Mostly, it is being used as a pre-processing step to analyze data patterns and data distribution. Also, it helps to visualize the overall data by reducing the workspace. This makes it a better pre-processing step for skin segmentation problem where human skin color follows specific combination of Red, Green and Blue colors. Currently, many algorithms exist to cluster the data in different groups. These methods can be categorized as Hard clustering methods and Fuzzy/Soft clustering methods[8]. Distinct clusters are made in Hard clustering where each data element have membership in one cluster only. But, in Fuzzy clustering, data elements can have fuzzy memberships in more than one clusters. In the current clustering algorithms, C-Mean algorithm is a good example of soft/fuzzy clustering and K-mean could be called as hard clustering algorithm[5].

2.3 Decision Tree

A decision tree consists of nodes representing input patterns and leaf nodes that represent the outcome for a particular combination of inputs. Fuzzy Decision tree (FDT) helps to map the inputs with corresponding output, that is, for a specific pattern of input values, what would be the value for output variable. Decision trees are 'pruned' to balance the accuracy and efficiency of their traversals, the process involves removal of irrelevant tree parts[9]. Since FDTs are being used to solve the classification problems[7], and skin segmentation lies in the category of binary classification, therefore skin segmentation could be done with the help of fuzzy decision tree.

3 Main Part

Overall flow of the experiment is shown in figure 1.

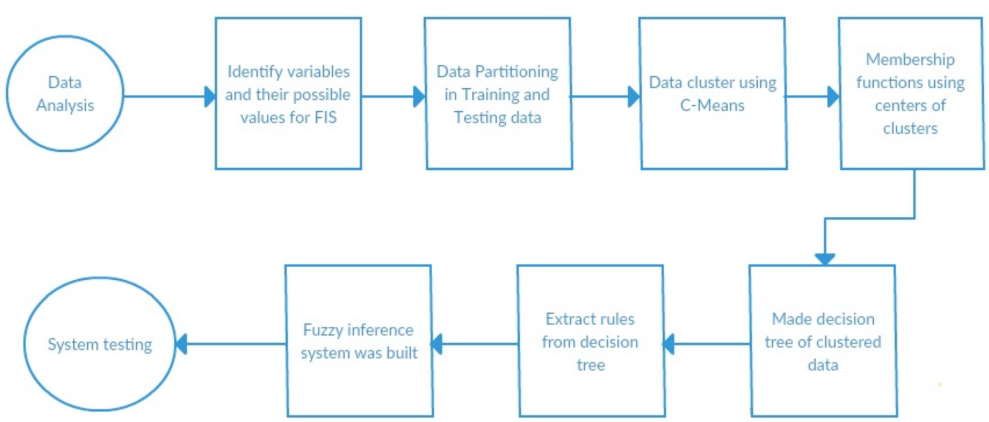


Figure 1: Flow of work

3.1 Data Analysis and Data Partitioning

Firstly, the data was gathered from UCI library where the total data size was 245057; out of which 50859 were the skin samples and 194198 were representing non-skin. Then, skin data was separately analyzed to find maximum and minimum values for R,G,B for skin. Table 1 is showing color ranges for R,G,B colors for a skin pixel.

Table 1: Color Ranges for Skin Pixel

Color	Minimum Value	Maximum
Red	106	255
Green	56	230
Blue	26	225

Also, to visualize the color patterns, skin data was individually plotted. An interesting distribution was seen in Chart 2, skin data is following a particular pattern for R,G,B colors. Specifically, for almost every skin record, value of Red was the greatest and Blue was the smallest, that is values were in the order R > G > B.

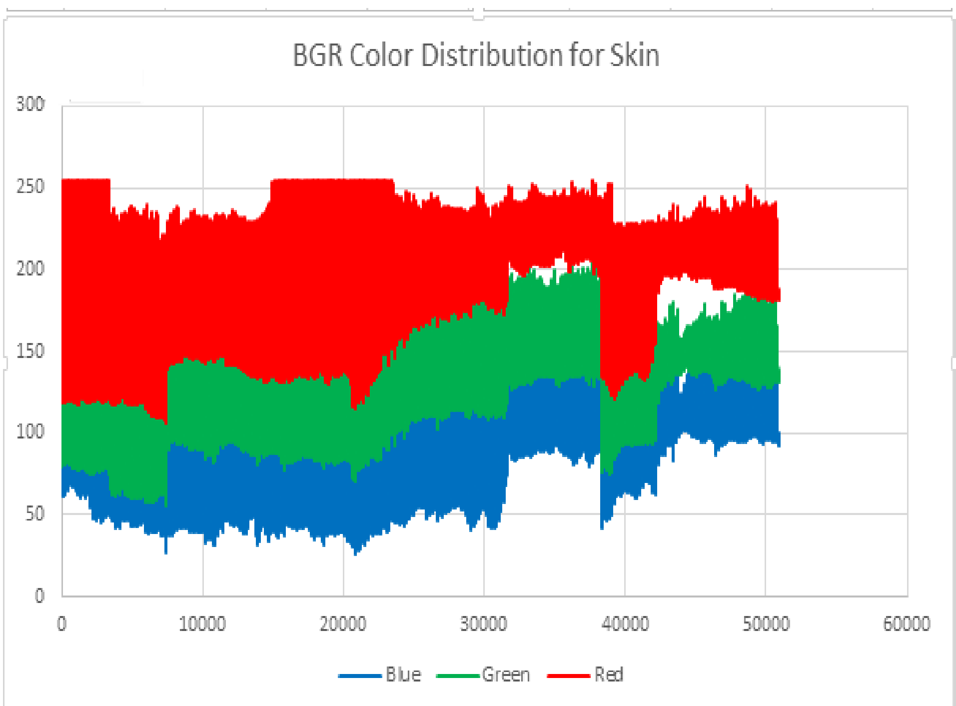


Figure 2: Skin Data Chart

After analyzing the patterns in skin data, training and testing data were separated with the ratio of 80% and 20% respectively.

3.2 Identify variables and their possible values

To relate with the given data, we had identified 'y' as the only output binary variable having 1 and 2 as possible values that represent Skin and Non-Skin respectively.

Similarly, input variables were:

- RED
- GREEN
- BLUE

Each variable was given 5 possible fuzzy values (Very-Light, Light, Medium, Dark, and Very-Dark) representing its intensity on the color spectrum. As, we have to show these possible values in terms of membership functions in the FIS and we were not sure about boundaries of these functions over 0-255 for each color. That is why, we decided to cluster the data in five groups for each color, so that we can use the centers of those clusters in our membership functions later.

3.3 Data Clustering

As mentioned, data clustering was decided to analyze data distributions, so training data was clustered using C-Mean algorithm. This algorithm does minimization of the following objective function[5]:

$$I = \sum_{i=1}^{N} \sum_{j=1}^{K} w_{ij}^{m} |\mathbf{x}_i - \mathbf{c}_j| \tag{1}$$

Where, N is number of data points, K is the number of clusters and m is degree of fuzzification. The use of C-Mean algorithm was because of the fact that it does the fuzzy clustering, and we were also able to use second nearest neighbours for the boundary cases if the results were not according to desire. Number of clusters was kept five, as possible values for each variable were also 5. Using fuzzy c-mean, membership value for every case in every possible cluster was obtained. After that, the value of color (R,G,B) was replaced with the cluster's number in which the value had maximum membership. To elaborate, figure 3 is showing this process in which 3a is showing the first 10 instances for Blue color data, 3c have centers for every cluster made for the respective color and 3b is showing fuzzy membership of first 10 instances for Blue color in 5 clusters. Once, we got fuzzy membership for every case of each color, we replaced the value (0-255) with its belonging cluster, that is, these 10 cases were replaced by 2 (figure 3d is showing the clustered data). To do this data clustering, Matlab code was written that is attached in Appendix A.

	1
1	74
2	73
3	72
4	70
5	70
6	69
7	70
8	70
9	76
10	76

(a) First 10 instances for Blue Color

	1	2	3	4	5
1	0.0263	0.9322	0.0276	0.0089	0.0050
2	0.0222	0.9450	0.0217	0.0071	0.0040
3	0.0182	0.9566	0.0166	0.0055	0.0031
4	0.0109	0.9758	0.0086	0.0030	0.0017
5	0.0109	0.9758	0.0086	0.0030	0.0017
6	0.0077	0.9835	0.0057	0.0020	0.0011
7	0.0109	0.9758	0.0086	0.0030	0.0017
8	0.0109	0.9758	0.0086	0.0030	0.0017
9	0.0347	0.9030	0.0419	0.0131	0.0072
10	0.0347	0.9030	0.0419	0.0131	0.0072

(b) Membership for every Instance in each cluster

	1
1	16.5845
2	64.3517
3	130.0611
4	172.7254
5	206.0650

(c) Center for each cluster

	1
1	2
2	2
3	2
4	2
5	2
6	2
7	2
8	2
9	2
10	2

(d) Data After clustering

Figure 3: Clustering Process

3.4 Decision tree of the clustered data

After clustering the data, decision tree was made for that clustered data using Matlab function 'classregtree'. Also, after trying different levels of pruning, level 1 was decided as tree was looking precise and short. Matlab code for tree and its pruning was:

```
myTree = classregtree( inputs, output, ...
'method','classification','names',{'Blue' 'Green' 'Red'}, ...
'categorical', [1 2 3]);
new_tree = prune(myTree,1);
view(new_tree)
```

The code provided us a decision tree shown in figure 4, this tree was later used to extract rules in clustered data.

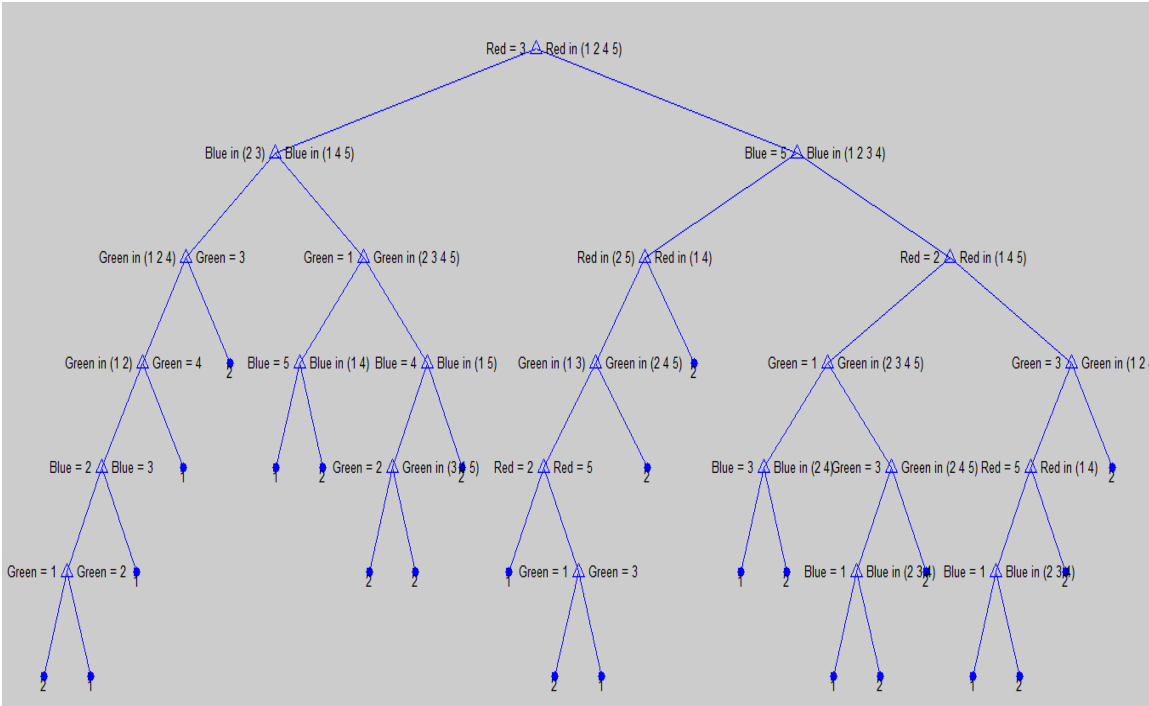


Figure 4: Decision Tree

3.5 Rules Extraction/Enhancement and Membership functions

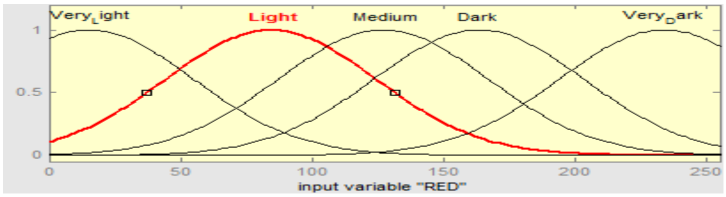
To extract the rules, we started from the leaf nodes and moved towards the root in each branch. Also, if a variable was repeated in a branch, the value nearest to leaf was considered only. After extracting the rules, further optimization was applied to them. The rules at this stage were in raw form, as they were not corresponding to the possible values we have decided earlier. Five of the obtained rules are given below:

- If RED is in 3 AND BLUE is in 2 AND GREEN is in 2 THEN SKIN
- If RED is in 3 AND BLUE is in 2 AND GREEN is in 1 THEN NON-SKIN
- If RED is in 3 AND BLUE is in 3 AND GREEN is in 1, 2 THEN NON-SKIN
- If RED is in 3 AND BLUE is in 2 OR 3 AND GREEN is in 4 THEN SKIN
- If RED is in 3 AND BLUE is in 2 OR 3 AND GREEN is in 3 THEN NON-SKIN

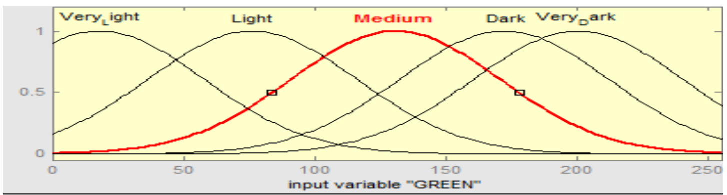
(Here RED is in 3 means value of RED lies in cluster 3)

Then, these rules were converted to synchronize with previously decided possible values for every variable (R,G,B). Centers of clusters during the clustering process were used to do this conversion. That is, the center having the smallest value was given the value 'Very-Light' and the cluster with the largest value was replaced with 'Very-Dark' (see figure 5). Also, centers of clusters were used as centers of membership functions and width of function was kept 40. Moreover, Gaussian type was selected for membership functions due to its smoothness. So, as an example, Blue color cluster with center at 16.58 was converted to membership function like this:

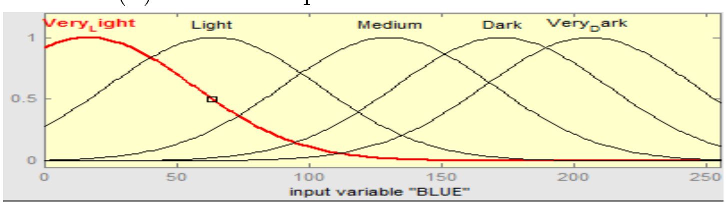
MF1='Very-Light':'gaussmf',[40 16.58]



(a) Membership Functions for RED



(b) Membership Functions for GREEN



(c) Membership Functions for BLUE

Figure 5: Membership Functions

Using this conversion, our rules were easier to infer humans. All the new rules are attached in Appendix B and 5 of them are given below:

- If RED is in Very-Dark AND BLUE is in Dark AND GREEN is in Very-Dark THEN SKIN
- If RED is in Very-Dark AND BLUE is in Dark AND GREEN is in Medium THEN NON-SKIN
- If RED is in Very-Dark AND BLUE is in Medium AND GREEN is in Medium, Very-Dark THEN NON-SKIN
- If RED is in Very-Dark AND BLUE is in Dark OR Medium AND GREEN is in Dark THEN SKIN
- If RED is in Very-Dark AND BLUE is in Dark OR Medium AND GREEN is in Light THEN NON-SKIN

The following rule was added from the data analysis process (Table 1):

- If RED IS VERY-LIGHT THEN NON-SKIN

If a rule had more than one values for one variable, then multiple rules were extended from it.

3.6 Fuzzy Inference System

Using these rules and membership functions, Fuzzy Mamdani Inference System was built, and 'Centroid' was used as a defuzzification method that returns center of area under the curve. Appendix C contains the complete system obtained after the process.

4 Results and Analysis

As the given data-set was already partitioned in 80-20 ratio, so we tested the system on 20% of the data, that was our testing data. Then the proposed system was again tested on the training data. Table 2 shows the results of this testing:

Table 2: Proposed system results

Tested Data	Accuracy	Precision	Recall
20% (Testing Data)	98.9%	95.79%	99.06%
80% (Training Data)	97.46%	93.62%	94.2%

After this, as a bench mark for performance, ANFIS (Adaptive Neuro Fuzzy Inference System) editor provided in Matlab was used to produce another fuzzy inference system using neural networks. Same data portions were given to this system

Table 3: ANFIS editor results

Tested Data	Accuracy	Precision	Recall
20% (Testing Data)	98.15%	91.8%	100%
80% (Training Data)	97.99%	91.25%	99.91%

to compare it with our proposed system. Table 3 is showing the results of system produced from ANFIS editor. FIS testing code is attached in Appendix D.

From the tables 3 and 2, it is cleared that the proposed system is yielding better Precision and same Accuracy, as that of the system build from ANFIS. But, Recall of our system is lower than that of ANFIS.

Recall and Precision of the proposed system are being effected due to less number of clusters for every color. Table 4 is showing the overlapping problem for skin and non-skin data. Here, we have a limited number of clusters and for the same combination of clusters we have different outputs. In such cases, we have decided to declare them as skin and this has affected the precision. Therefore, in order to increase the precision, increase in number of clusters should be considered. Also, different color spaces can be used to test one the same process.

Table 4: Overlapping Clusters

B,G,R	Closer Cluster Blue	Closer Cluster Green	Closer Cluster Red	Y
157,186,231	172.72	172.21	233	Skin
172,173,199	172.72	172.21	233	Non-Skin

5 Conclusion

Skin segmentation is a popular problem amongst researcher due to its wider implications. To classify a given R,G,B colour as skin or non-skin, Fuzzy Inference System was used with the help of clustering. The proposed system had shown good performance by yielding 98.9% accuracy on testing data. Also, it is suggested that enhancing the clustering mechanism and applying the same process on some other colour space could increase the performance.

Bibliography

[1] C. Aiping, P. Lian, T. Yaobin, and N. Ning. Face detection technology based on skin color segmentation and template matching. In *Education Technology and Computer Science (ETCS), 2010 Second International Workshop on*, volume 2, pages 708–711. IEEE, 2010.

[2] M. Ben Hmida and Y. Ben Jemaa. Fuzzy classification, image segmentation and shape analysis for human face detection. In *Electronics, Circuits and Systems, 2006. ICECS'06. 13th IEEE International Conference on*, pages 640–643. IEEE, 2006.

[3] D. Chai and A. Bouzerdoum. A bayesian approach to skin color classification in ycbcr color space. In *TENCON 2000. Proceedings*, volume 2, pages 421–424. IEEE, 2000.

[4] M. S. Iraji and A. Yavari. Skin color segmentation in fuzzy ycbcr color space with the mamdani inference. *American Journal of Scientific Research*, 2011(7):131–137, 2011.

[5] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[6] C.-F. Juang, S.-H. Chiu, and S.-J. Shiu. Fuzzy system learned through fuzzy clustering and support vector machine for human skin color segmentation. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 37(6):1077–1087, 2007.

[7] K. MATIAŠKO, J. BOHÁČIK, V. LEVASHENKO, and Š. KOVALÍK. Learning fuzzy rules from fuzzy decision trees. *Journal of Information, Control and Management Systems*, 4(2), 2006.

[8] S. Naz, H. Majeed, and H. Irshad. Image segmentation using fuzzy clustering: A survey. In *Emerging Technologies (ICET), 2010 6th International Conference on*, pages 181–186. IEEE, 2010.

[9] C. Olaru and L. Wehenkel. A complete fuzzy decision tree technique. *Fuzzy sets and systems*, 138(2):221–254, 2003.

[10] T. J. Ross. *Fuzzy logic with engineering applications.* John Wiley & Sons, 2009.

[11] A. Selamat, M. A. Maarof, and T. Y. Chin. Fuzzy mamdani inference system skin detection. In *Hybrid Intelligent Systems, 2009. HIS'09. Ninth International Conference on*, volume 3, pages 57–62. IEEE, 2009.

6 Appendix A: Clustering Code

```
data = load ('skinTrainingData.txt');

blue = data(:,1);        % Selecting first attribute
green = data(:,2);        % Selecting second attribute
red = data(:,3);        % Selecting third attribute
myclass = data(:,4);     % Output values
n_clusters = 5;          % No of clsuters for c-mean
fid = fopen('DataInClusters.txt','wt');

[center,U,obj_fcn] = fcm(blue, n_clusters);

% Here, fcm is returning a 3D matrix where
% one column contains centers for each cluster
% second has the membership values
% third is the objective_function

[center1,U1,obj_fcn1] = fcm(green, n_clusters);
[center2,U2,obj_fcn2] = fcm(red, n_clusters);

U = U';       % Matrix transpose, U contains memberships
U1 = U1';
U2 = U2';

[rows, columns] = size(U);

output = [];
```

```
% Now we go through membership values for every case
% Here, One row is representing values for one case
% for every row, we obtained the Index having largest value
% Then, that case (0-255) was replaced with largest
% index (1-5).

for i = 1:rows
    u = U(i,:);              % pick membership values for one case in blue
    u = find(u==max(u));     % Find the index of maximum value of membershir
    output = [output u];

    u1 = U1(i,:);        % pick membership values for one case in green
    u1 = find(u1==max(u1));

    u2 = U2(i,:);        % % pick membership values for one case in red
    u2 = find(u2==max(u2));

    temp = myclass(i,1);
                         % Now print the values to file
    c = fprintf (fid,'%i,%i,%i,%i \n',u,u1,u2,temp);
end

fclose(fid);        % close file to see changes
```

7 Appendix B: Fuzzy Rules

```
If RED is Very_Dark AND BLUE is Dark AND . .
GREEN is Very_Dark THEN SKIN
If RED is Very_Dark AND BLUE is Dark AND . .
GREEN is Medium THEN NON-SKIN
If RED is Very_Dark AND BLUE is Medium AND . .
GREEN is Medium, Very_Dark THEN NON-SKIN
If RED is Very_Dark AND BLUE is Dark OR Medium AND . .
GREEN is Dark THEN SKIN
If RED is Very_Dark AND BLUE is Dark OR Medium AND . .
GREEN is Light THEN NON-SKIN
If RED is Very_Dark AND BLUE is Light AND . .
GREEN is Medium THEN SKIN
If RED is Very_Dark AND BLUE is Very_Light, Very_Dark AND . .
GREEN is Medium THEN NON-SKIN
If RED is Very_Dark AND BLUE is Very_Light, Very_Dark, Light AND
. .
GREEN Very_Dark, Light, Dark, Very_Light THEN NON-SKIN
 (Above two rules could be replaced by following two rules)
If RED is Very_Dark AND BLUE is Very_Light, Very_Dark THEN NON-SKIN
If RED is Very_Dark AND BLUE is Light AND . .
GREEN is NOT Medium THEN NON-SKIN
If RED is Dark AND BLUE is Light AND . .
GREEN Medium, Light THEN SKIN
If RED is Medium AND BLUE is Light AND . .
GREEN Medium THEN NON-SKIN
If RED is Medium AND BLUE is Light AND . .
GREEN Light THEN SKIN
If RED is Dark, Medium AND BLUE is Light AND . .
GREEN Very_Dark, Dark, Very_Light THEN NON-SKIN
If RED is Light,Very_Light AND BLUE is Light THEN NON-SKIN
If RED is D AND GREEN is Medium AND BLUE is Medium THEN SKIN
If RED is Dark AND GREEN is Medium AND . .
BLUE is Dark, Very_Dark THEN NON-SKIN
If RED is Dark AND GREEN is Light AND . .
BLUE is Very_Light THEN SKIN
If RED is Dark AND GREEN is Light AND . .
BLUE is Dark, Medium, Very_Dark THEN NON-SKIN
If RED is Dark AND GREEN is Very_Dark, Dark, Very_Light AND . .
BLUE is Very_Light, Dark, Medium, Very_Dark (NOT LIGHT) THEN NON-SKIN
If RED is Medium AND GREEN is Light AND BLUE is Very_Light THEN SKIN
If RED is Medium AND GREEN is Light AND . .
BLUE is Dark, Medium, Very_Dark THEN NON-SKIN
If RED is Light, Very_Light AND GREEN is Light AND . .
```

```
BLUE is Very_Light, Dark, Medium, Very_Dark (NOT LIGHT) THEN NON–SKIN
 If RED is Light, Very_Light, Medium AND  . .
GREEN is (NOT LIGHT) Medium, Very_Dark,Dark, Very_Light AND  . .
BLUE is Very_Light, Dark, Medium, Very_Dark (NOT LIGHT) THEN NON–SKIN
 (Above two rules are optimized to)
 If RED is Light, VeryLIGHT AND BLUE is NOT LIGHT THEN NON–SKIN
 If RED IS VERY_LIGHT THEN NON–SKIN
```

8 Appendix C: Fuzzy Inference System

```
[System]
Name='FirstDraft'
Type='mamdani'
Version=2.0
NumInputs=3
NumOutputs=1
NumRules=44
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

[Input1]
Name='BLUE'
Range=[0 255]
NumMFs=5
MF1='Very_Light':'gaussmf',[40 16.58]
MF2='Very_Dark':'gaussmf',[40 206]
MF3='Light':'gaussmf',[40 64]
MF4='Dark':'gaussmf',[40 172.7]
MF5='Medium':'gaussmf',[40 130]

[Input2]
Name='GREEN'
Range=[0 255]
NumMFs=5
MF1='Medium':'gaussmf',[40 130.8]
MF2='Light':'gaussmf',[40 76.23]
MF3='Very_Light':'gaussmf',[40 18]
MF4='Very_Dark':'gaussmf',[40 200.4]
MF5='Dark':'gaussmf',[40 172.2]

[Input3]
Name='RED'
Range=[0 255]
NumMFs=5
MF1='Light':'gaussmf',[40 84.3]
MF2='Dark':'gaussmf',[40 163.32]
MF3='Medium':'gaussmf',[40 127]
MF4='Very_Light':'gaussmf',[40 15]
MF5='Very_Dark':'gaussmf',[40 233.8]
```

```
[Output1]
Name='Y'
Range=[1 2]
NumMFs=2
MF1='Skin':'trimf',[1.02280423280423 1.24280423280423 1.49280423280423]
MF2='Non-Skin':'trimf',[1.46312169312169 1.71312169312169 1.97312169312169

[Rules]
4 4 5, 1 (1) : 1
4 1 5, 2 (1) : 1
5 1 5, 2 (1) : 1
5 4 5, 2 (1) : 1
4 5 5, 1 (1) : 1
5 5 5, 1 (1) : 1
4 2 5, 2 (1) : 1
5 2 5, 2 (1) : 1
3 1 5, 1 (1) : 1
1 0 5, 2 (1) : 1
2 0 5, 2 (1) : 1
3 -1 5, 2 (1) : 1
3 1 2, 1 (1) : 1
3 1 3, 2 (1) : 1
3 2 3, 1 (1) : 1
3 4 2, 2 (1) : 1
3 5 2, 2 (1) : 1
3 3 2, 2 (1) : 1
3 3 3, 2 (1) : 1
3 5 3, 2 (1) : 1
3 4 3, 2 (1) : 1
3 0 1, 2 (1) : 1
3 0 4, 2 (1) : 1
5 1 2, 1 (1) : 1
4 1 2, 2 (1) : 1
2 1 2, 2 (1) : 1
1 2 2, 1 (1) : 1
4 2 2, 2 (1) : 1
5 2 2, 2 (1) : 1
2 2 2, 2 (1) : 1
-3 5 2, 2 (1) : 1
-3 4 2, 2 (1) : 1
-3 3 2, 2 (1) : 1
-1 2 3, 1 (1) : 1
-3 0 1, 2 (1) : 1
-3 0 4, 2 (1) : 1
0 0 4, 2 (1) : 1
0 5 4, 2 (1) : 1
```

```
0 4 4, 2 (1) : 1
-1 0 4, 2 (1) : 1
0 -3 4, 2 (1) : 1
5 -1 1, 2 (1) : 1
5 -5 1, 2 (1) : 1
4 -5 1, 2 (1) : 1
```

9 Appendix D: FIS testing code

```matlab
predictedOutput = [];
data = load ('skinTrainingData.txt');
inputs = data(:,1:3);     % get input values
out = data(:,4);

[rows, cols] = size(inputs);
myfis = readfis('FirstDraft'); % read fis file

for i = 1:rows
    output = evalfis(inputs(i,:),myfis);

    % evalfis forwards the given input to fis
    % and returns the result

    if output > 1.5           % 1.5 is the threshold
        predictedOutput = [predictedOutput 2];
    elseif output <= 1.5
        predictedOutput = [predictedOutput 1];
    end
end

predictedOutput = predictedOutput';

% Following are the variables used for confusion matrix
a = 0;
b = 0;
c = 0;
d = 0;

for i = 1:rows
    actual = out (i,:);
    predicted = predictedOutput (i,:);
    if (actual == 1 && predicted == 1)       % Both are skin
        a = a + 1;
    elseif (actual == 2 && predicted == 2)   % Both are Non-Skin
        d = d + 1;
     elseif (actual == 2 && predicted == 1)
        c = c + 1;                           % Actual is non-skin and predicted is skin
     elseif (actual == 1 && predicted == 2)
        b = b + 1;                           % Actual is skin and predicted is non-skin
    end
end
%counter
accuracy = ((a+d)/(a+b+c+d))*100
precision = ((a)/(a+c))*100
recall = ((a)/(a+b))*100
```