Name: PRATHMESH R. KUMBHARE

CS6001 Assignment 1: Facial region detection

REPORT

A. OUTLINE FOR PROGRAM ALGORITHM:

The algorithm for implementing the program is given below in steps:

1. Matlab function files are created for each model separately. Main program just consists of taking input for model number from the user. The program then executes the desired model & loads all the workspace variables for corresponding model. Images used for training and testing results are displayed to give some visualization. Annotator 'wenjin' is used for this program.

2. Factors which can be varied to get different results for a model are: number of bins & image scaling factor.

3. The program firstly creates 'bwMask' & computes likehood, prior for the training images. To compute likelihood, the program loops through each row and column of 'bwMask' & increases the count of probability 'Pr_x_given_w_equalsTo_1' or 'Pr_x_given_w_equalsTo_0' if the value is 1 or 0 accordingly. Then these values are normalized. Plot for probability distribution of RGB, HSV & YCbCr model is also included.

4. In the testing images loop, the program creates 'detMask' by looping through each row and column of the mask and assigning value 1 or 0 depending on the which value is greater between: (Pr_x_given_w_equalsTo_1) * (Pr_w_equalsTo_1) and (Pr_x_given_w_equalsTo_0) * (Pr_w_equalsTo_0). 'gtMask' (ground truth) is the annotated testing images. To calculate TP, FP & FN the program loop through rows and column of these masks and apply mathematical relations between them.

5. In the last part precision, recall & f_score is calculated from the above values of TP, FP & FN. All the workspace variables are saved in '.mat' file under the model name.

B. RESULTS:

This is the table for all the models implemented using the program with all the important values:

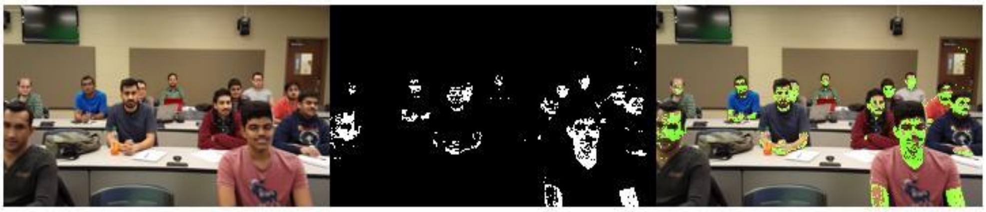
	MODEL	NUM OF BINS	IMAGE SCALING FACTOR	PRECISION (%)	RECALL (%)	F_SCORE
1	RGB	256	1	66.60	65.37	65.98
2	RGB	128	0.5	66.48	64.45	65.45
3	RGB	64	0.25	67.08	66	66.55
4	RGB	32	0.125	67.48	61.96	64.41
5	RGB	16	0.0625	63.09	57.17	59.98
6	RGB	8	0.0313	72.58	12.35	21.12
7	HSV	256	1	66.57	65.41	65.99
8	HSV	64	0.25	66.18	65.18	65.68
9	HSV	32	0.125	66.77	63.73	65.22
10	YCbCR	256	1	63.47	58.52	60.89
11	YCbCR	64	0.25	62.94	59.66	61.26
12	YCbCR	32	0.125	62.34	63.17	62.75
13	RGB + HSV	8	0.0313	63.51	41.83	50.44
14	RGB + HSV	16	0.0625	66.59	62.83	64.66
15	RGB + YCbCr	8	0.0313	61.69	63.61	62.63
16	RGB + YCbCr	16	0.0625	65.18	66.40	65.78
17	RGB + HSV + YCbCr	8	0.0313	65.54	64.13	64.83
18	HSV + GRAY	8	0.0313	63.45	25.06	35.93
19	HSV + GRAY	16	0.0625	65.70	58.87	62.1

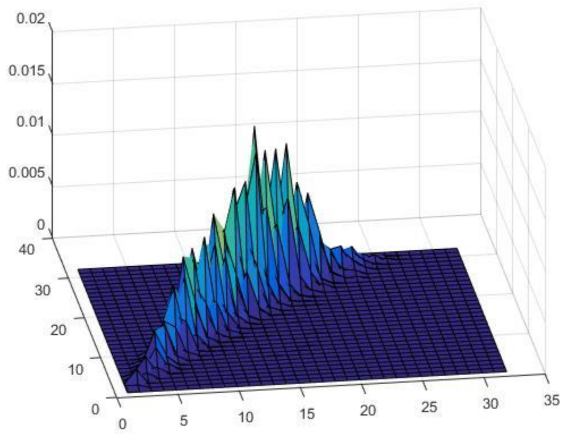
Results for each model are given below separately.

1. RGB Model:

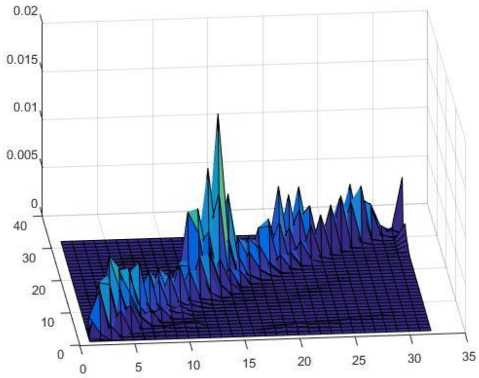
This model uses values of red, green and blue to describe the image.

Training and testing image:





Pr_x_given_w_equalsTo_1 distribution

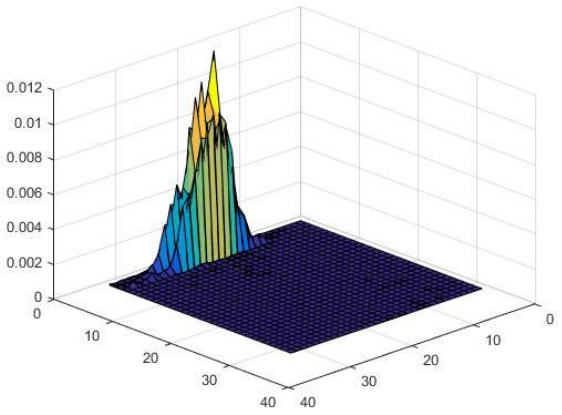


Pr_x_given_w_equalsTo_0 distribution

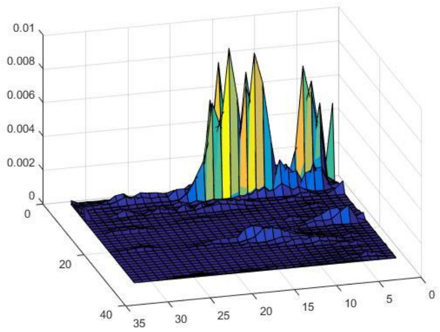
2. HSV model:

This model converts RGB values into Hue, Saturation & Value to describe image.

Training & Testing image:



Pr_x_given_w_equalsTo_1 distribution

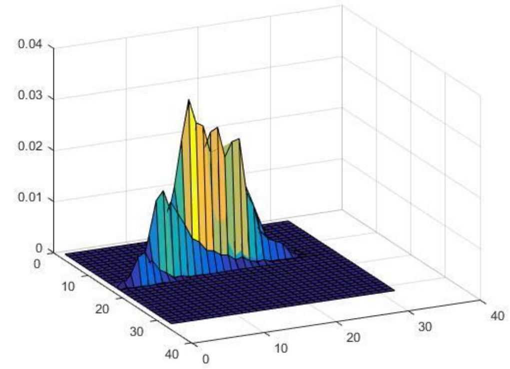


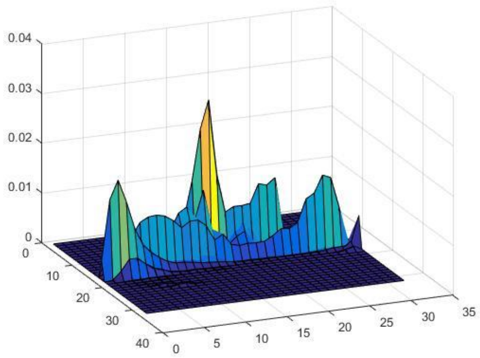
Pr_x_given_w_equalsTo_0 distribution

3. YCbCr model:

This model converts RGB values into Y, Cb and Cr values to describe the image.

Training and testing image:





Pr_x_given_w_equalsTo_1 distribution Pr_x_given_w_equalsTo_0 distribution

4. RGB & HSV Model:

In this model, RGB values of images & their corresponding HSV values are used to plot the probability distribution in 6 dimensions.

Training and Testing Image:

Training image in RGB values & HSV values with bwMask

Testing Image: First image is the skin detection area using model. Second image is HSV image.

5. RGB + YCbCr Model:

In this model, RGB values of images & their corresponding Y, Cb & Cr values are used to plot the probability distribution in 6 dimensions.

Training and Testing Image:

Training image in RGB values & YCbCr values with bwMask

Testing Image: First image is the skin detection area using model. Second image is YCbCr image.

6. HSV & GRAY Model:

In this model, HSV values of images & their corresponding grayscale values are used to plot the probability distribution in 6 dimensions.

Training and Testing Image:

Training image in RGB values & HSV values with bwMask

Testing Image: First image is the test image. Second image is detMask computed using the model. Third image is the skin detected area on test image.

7. RGB, HSV & YCbCr Model:

In this model, RGB values of images with their corresponding HSV & YCbCr values are used to plot the probability distribution in 9 dimensions.

Training & Testing Image:

Training image: First image is the train image in RGB values with bwMask. Second image is the HSV image with bwMask. Third image is the YCbCr image with bwMask.

Testing image: First image is the test image with skin detection area as determined by the model. Second image is the HSV image. Third image is the YCbCr image.

C. General Observation & Comments:

1. Selection of number of bins & image scaling factor is important in this algorithm. As an example, we will consider the case of RGB model for various number of bins & image scaling factor. As we can see that f_score varies for different parameters. Hence, a trade-off between accuracy & computation is there.

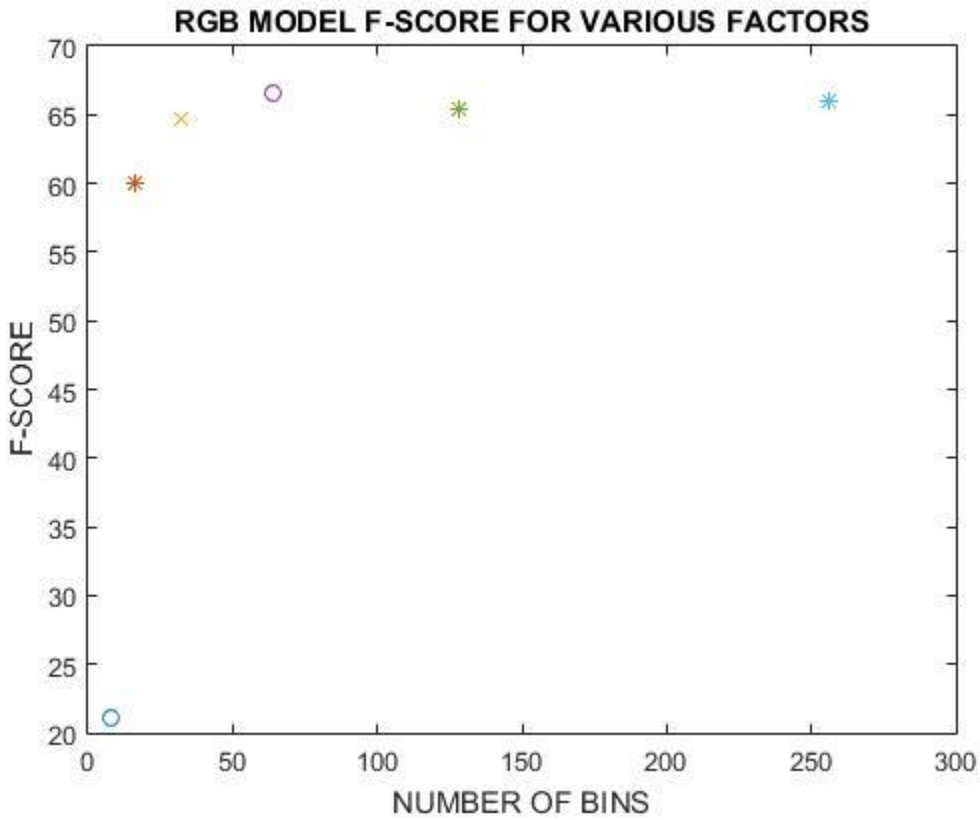


Figure details: ‘o’ (blue) - for 8 bins & (1/32) scaling factor. ‘*’ (red) – for 16 bins & (1/16) scaling factor. ‘x’ (yellow) – for 32 bins & (1/8) scaling factor. ‘o’ (purple) - for 64 bins & (1/4) scaling factor. ‘*’ (green) – for 128 bins & (1/2) scaling factor. ‘*’ (blue) – for 256 bins & 1 scaling factor.

2. Similarly, as the complexity increases the f_score increases. But this also increases computation time.

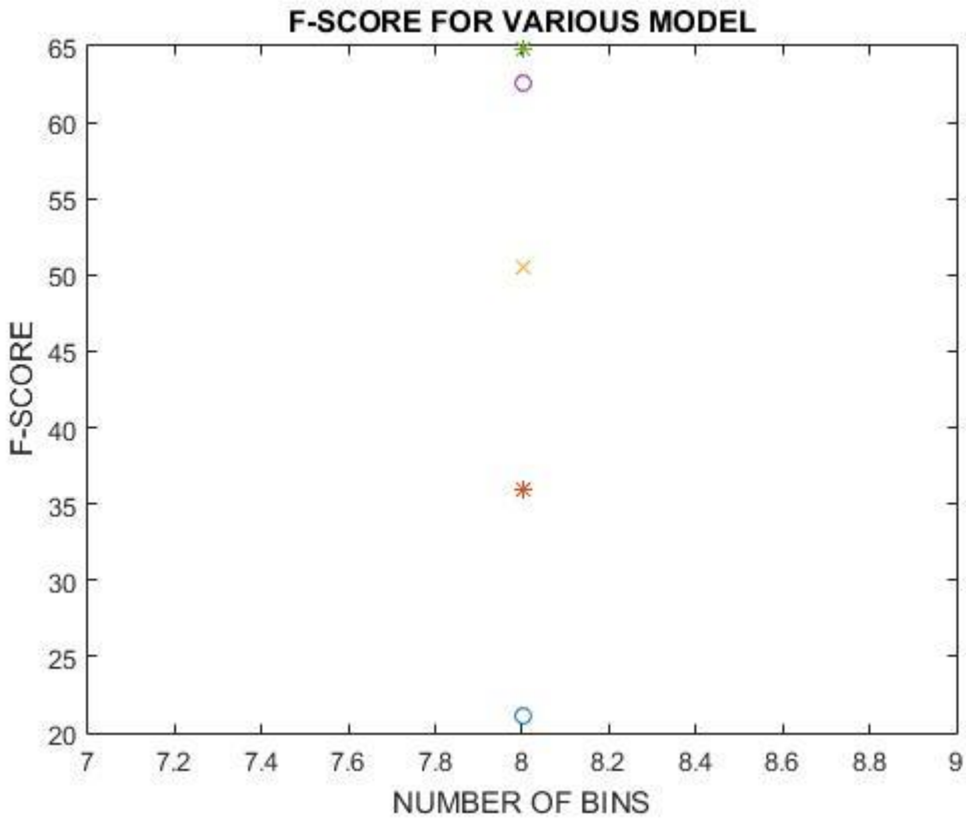


Figure details: ‘o’ (blue) - for 8 bins & f_score for RGB model. ‘*’ (red) – for 8 bins & HSV+GRAY model. ‘x’ (yellow) – for 8 bins & RGB+HSV model. ‘o’ (purple) - for 8 bins & RGB+YCbCr model. ‘*’ (green) – for 8 bins & RGB+HSV+YCbCr model.

D. Summary and Concluding Comments:

1. The aim of this assignment was to detect facial region in the given images. From the results, it is evident that the algorithm also detects skin region of hands in addition to facial region. This makes the algorithm less precise.
2. It could be possible to increase the precision of the algorithm with the addition image filters which ‘clean up’ the image a bit more.
3. Another approach would be to increase the number of training images for the algorithm so that it can ‘learn’ better.