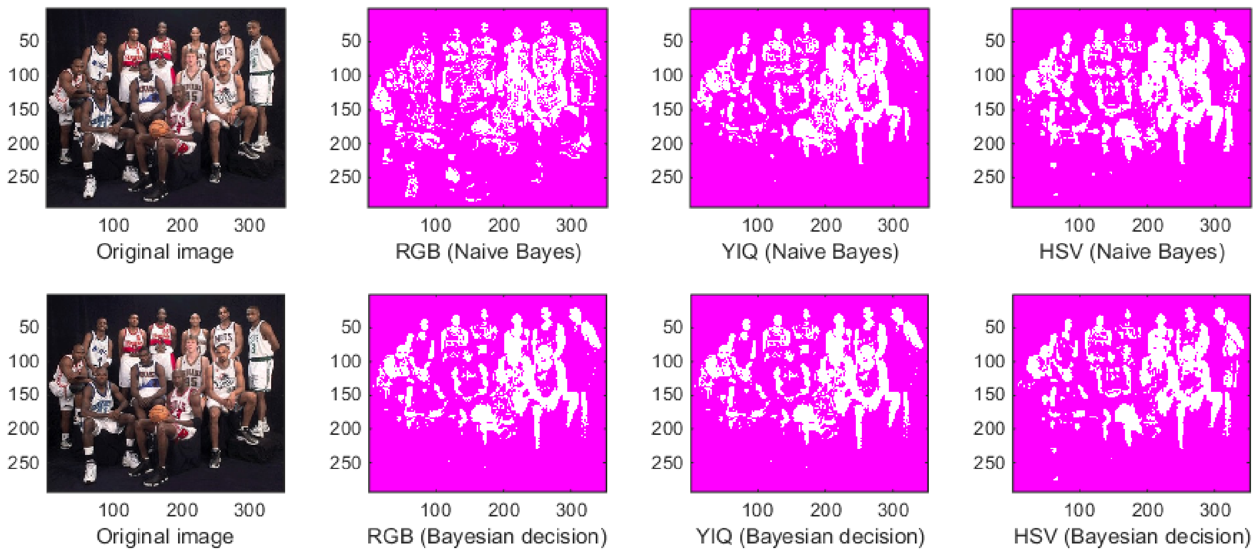
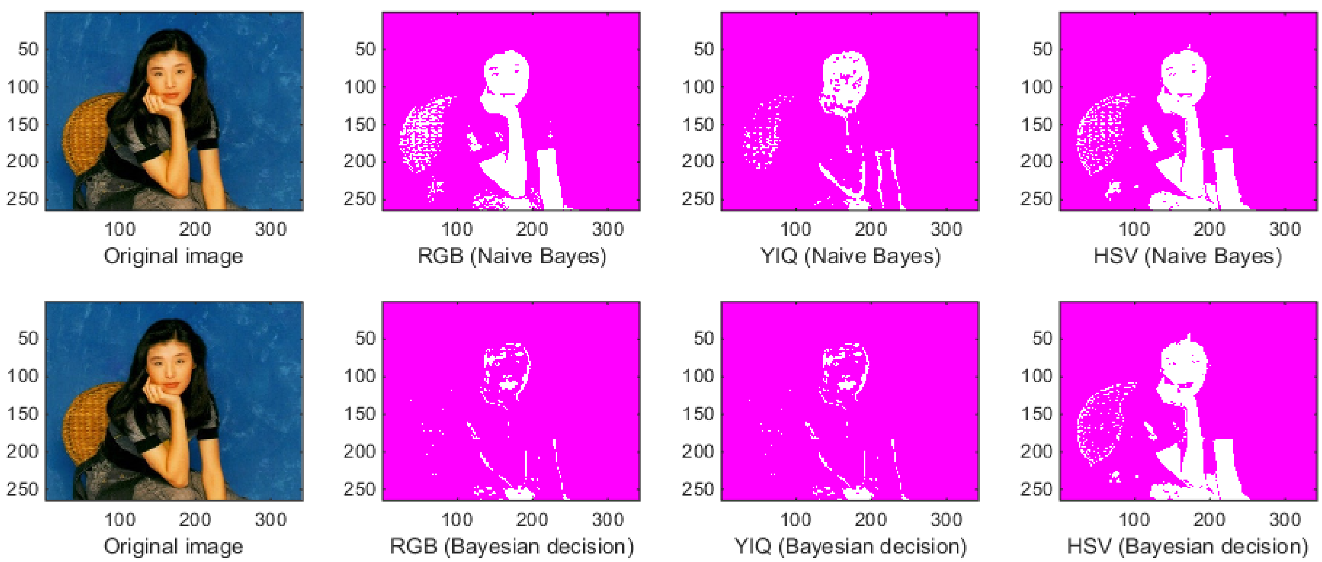
Skin Detection Report

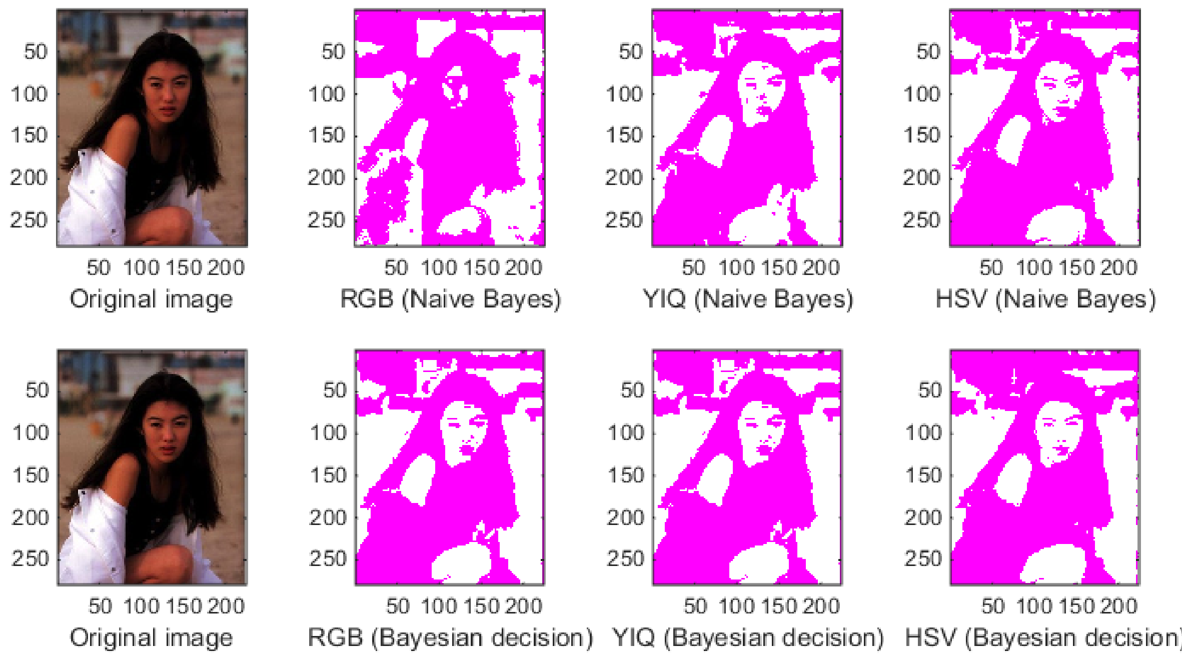
Dong Han

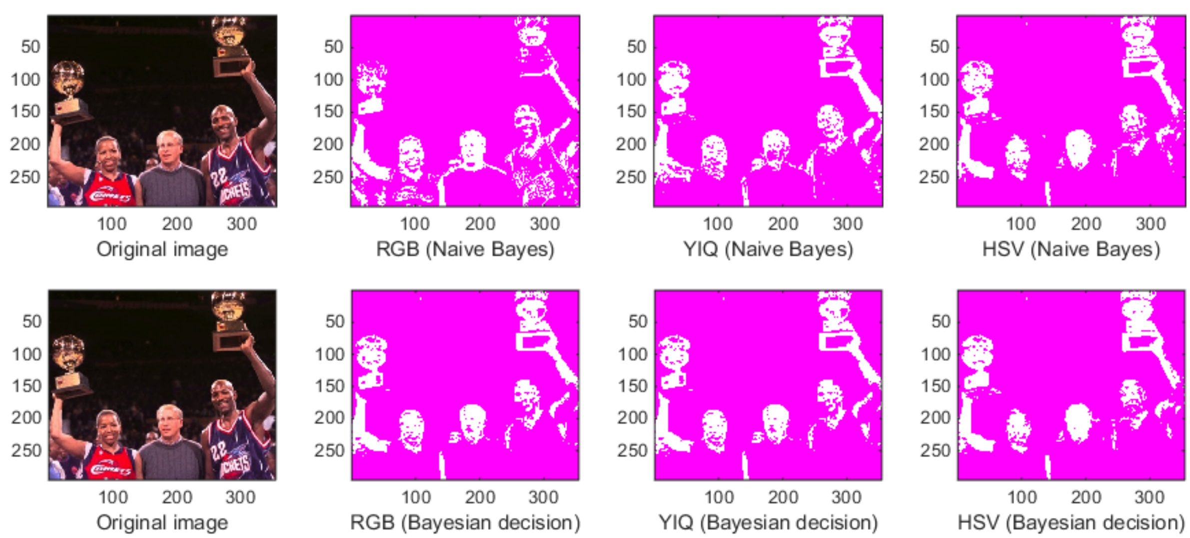
CSE 616

Preview: Classification results based on the given pictures.






Original image
RGB (Naive Bayes)
YIQ (Naive Bayes)
HSV (Naive Bayes)
Original image
RGB (Bayesian decision)
YIQ (Bayesian decision)
HSV (Bayesian decision)


Original image
RGB (Naive Bayes)
YIQ (Naive Bayes)
HSV (Naive Bayes)
Original image
RGB (Bayesian decision)
YIQ (Bayesian decision)
HSV (Bayesian decision)

1. Introduction

This is a report for skin detection. The objective of the project is to build a classification system that can discriminate skin pixels from non-skin pixels in a picture. The system can help to detect faces and other skin areas of human bodies of pictures. We plan to build the system by following these steps.

Firstly, we begin a dataset collection procedure. In this step, we select different kinds of pictures from Internet. These pictures shows skins of different colors with various backgrounds. The pictures with a pure background color are excluded from collection, because in real environment, the background has various colors. In particular, we choose pictures that contains skin colors including white, yellow, and black. The reason to choose different color is that different race of human have different skin colors. We want our project builds a general skin detection system that can recognize various skins. In this step, we get 21 pictures that contain some typical skin colors of human and typical backgrounds of different scenarios. After we collect enough pictures in this step, the next step is to create training set and testing set.

In the step of training set and testing set building. We focus on the pixel representation. A pixel is represented by 3 well-known color models, which are RGB, YIQ and HSV. In each of these models, each pixel can be present as a vector with 3 elements, each vector has a specific physical meaning. For example, in RGB color model, the 3 elements denote red, green, and blue color, respectively. In 21 collected pictures, we choose 13 out of 21 pictures as training set, then the rest 8 pictures are used for testing. For each picture in training set or testing set, we randomly pick round 100 pixels of the picture, specifically, we choose around half of 100 pixels from skin area, the other half from background area. When we pick a pixel from picture, we also label it as skin pixel or non-skin pixel manually. To make the labeling procedure more convenient, a picture annotation application is created. By using the application, we can use mouse to choose any pixels in the picture. At the same time, we can define the selected pixel is from skin or non-skin area. The system records 3 color models of all selected pixels to a CSV file for future analysis. After we build the training and test set, we do some statistics to show the data set. Then we begin classifiers' training step.

In this step, we choose two classifiers: naïve Bayes and Bayesian decision. For each type of classifier, we choose three different color models for training. For example, for the RGB model, we give 3 features as red, green and blue to train a classifier. Then we do the same training procedure for the other two color models. Then we get 3 classifiers for three color models under naïve Bayes method and Bayesian decision method, respectively. After training these classifiers, we use test set for performance evaluation. We also verify the effectiveness of our methods by using some other pictures. To show the result of classification, we use two distinct colors to display skin area and non-skin area. Then we can intuitively verify the two color picture manually.

The result of the report is organized as follows. In section two, the pixel selection method is introduced, the design and implementation of a pixel picker is shown in the section. We will discuss how to show a picture then process different color models and the format of selected pixels saved in storage. In section three, we first introduce a few theory backgrounds of naïve Bayes and Bayesian decision methods. Then we show how to apply the two methods to train skin detection systems. In section four, we do some experiment to show some statistical result of pixels and some performance evaluations based on the trained classifiers. In section 5, we summarize the project and draw possible future work that can further improve the system.

2. Pixel selection

In the section, the system for picture showing and pixel selection is discussed. To build a prototype efficiently, we choose MATLAB platform to implement the project. All our works, including pixel selection and classifiers training, are based on the platform. In MATLAB there are quite a few of tools and middle wares can help to manipulate pictures. However, not all of them have functions that we need in the project. In our project, we need a tool that a user can select a picture from storage, then show the picture in computer screen. Then the user can pick some pixels from the picture by mouse cursor. The user can label the pixel as skin or non-skin. After chosen, three color model: RGB, YIQ and HSV of the pixel and the labeling result are recorded by the tool. When use decides to finish pixel selection, the recorded pixel information can be saved on disk as CSV format. After doing some studies of MATLAB existing picture processing method, we decide to implement our tools based on an existed tool, Imtool. Though the tool lacks some key feature functions we need, it has provided some basic functions such as open and show a picture, select a pixel, as shown in Figure 1. Then we add some codes patching the tool to implement our functions. The result of patched Imtool is shown in Figure 2.

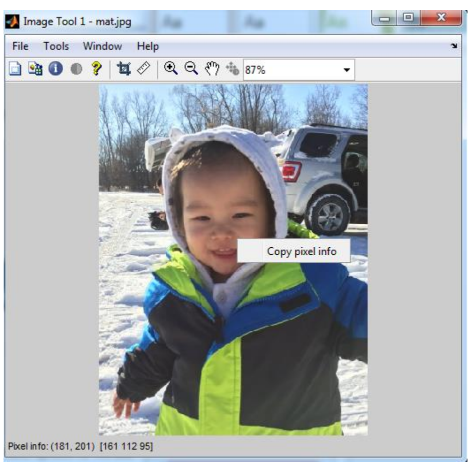


Figure 1. Original Imtool

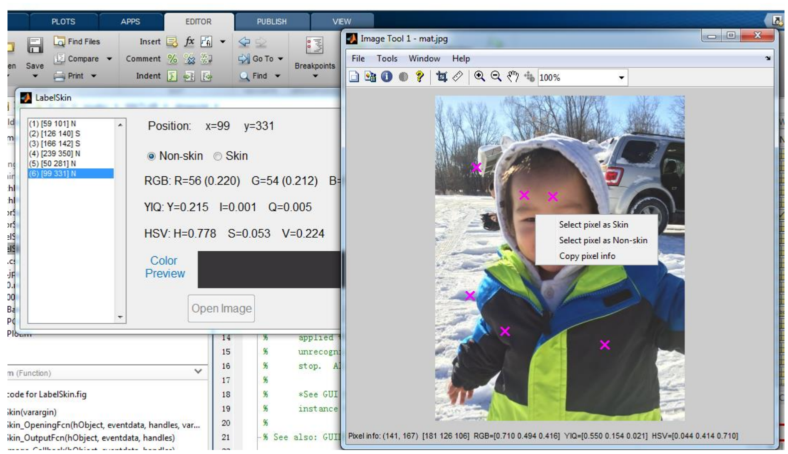


Figure 2. Patched Imtool with new features

In original Imtool, it can copy a pixel information, however, the pixel information has limited information. It provides (x, y) position of pixel and only RGB color model. Also, after picking a pixel, there is no marker to notify the user which pixel has been selected, which is an important feature for building a training set. We want to select pixels uniformly from different positions of a picture. If a user cannot see the selected pixels, it is difficult to select pixels evenly in the picture. Therefore, we implement these functions in the patched one. When user wants to select a picture, the user clicks right mouse on the pixel, a context menu is popped. We added two more menu entries in the popup menu to help user label the selected pixel. For the two menu entries, the user can decide the selected pixel is a skin pixel or a non-skin pixel.

Therefore, during the pixel selection, the user can finish the pixel label at the same time. As shown at the bottom side of the tool, we not only give position and RGB of a pixel, in the patched tool, the YIQ and HSV information are also given. Moreover, we add a control panel as another window of the patched tool. All selected pixels are recorded in the left side of the control panel window as a list. At the right side of the control panel window, a user can preview any selected pixel in a larger color block. The detailed information of the pixel is given at the right side of the window. If a user made a wrong choice in labeling a skin pixel to non-skin pixel or vice versa, the user can update the label information by selecting the pixel, then change the label at the right side of control panel. When user collects enough pixels, the save button can direct user to save selected pixels' record.

The menu entries adding and pixel information structure are implemented as the following code in "impixelinfoval2.m" file

```
%~updated
menuItemHandle2 = uimenu(cmenu,...
    'Label','Select pixel as Skin', ...
    'Callback',@selectPixToListSkin,...
    'Tag','Select pixel from image as skin');
menuItemHandle3 = uimenu(cmenu,...
    'Label','Select pixel as Non-skin', ...
    'Callback',@selectPixToListNonskin,...
    'Tag','Select pixel from image as non-skin');
%~end

menuItemHandle = uimenu(cmenu,...
    'Label',getString(message('images:impixelinfovalUIString:copyMenuLabel')), ...
    'Callback',@copyPixInfoToClipboard,...
    'Tag','Copy pixel info menu item');

%~updated
selectedColor = struct('index',-1,'skin',-1,'x',-1,'y',-1,'rgbi',[-1,-1,-1],'rgb',[-1,-1,-1],'yiq',[-1,-1,-1],'hsv',[-1,-1,-1]);
%~end
```

Then we implement a function named "selectPixToList" that implements marker rendering and adding information of selected pixel to the list of control panel.

```
function selectPixToList()
    hold on;
    plot(selectedColor.x,selectedColor.y,'x','linewidth',2.5, ...
      'MarkerSize',13,'MarkerEdgeColor','m',...
      'MarkerFaceColor',[0.5,0.5,0.5]);
    hold off;
    %add to listbox
    imgHandle = guidata(parent);
    % Modify the value of your counter
    listContent= get(imgHandle.labelSkinHandles.labelListBox,'String');
    itemNum = length(listContent);

    skinStr= 'S';
    if selectedColor.skin == 0
        skinStr = 'N';
    end
```

Similarly, we add codes at different places like above to implement our functions in the source code of Imtool. We get our own pixel selection and labeling tool, named as Imtool2.

The recorded pixels are saved in a CSV file, the format of the CSV file is easy to be parsed, as shown in Figure 3. Each pixel information is recorded in one row. The column 1 and 2 are x and y position of the pixel. The column 3 to 5, 6 to 8, and 9 to 11 are 3 elements of color models for RGB, YIQ and HSV, respectively. The last column is labeling information, value 1 denotes a skin pixel, value 0 denotes a non-skin pixel. Based on the file, in classification process, we can get labeling information of a label from the last column.

565	324	0.854902	0.619608	0.580392	0.685474	0.152833	0.037552	0.02381	0.321101	0.854902	1
509	351	0.776471	0.584314	0.529412	0.635496	0.132169	0.023543	0.037037	0.318182	0.776471	1
472	331	0.913725	0.694118	0.643137	0.753954	0.147267	0.030571	0.031401	0.296137	0.913725	1
45	225	0.847059	0.847059	0.894118	0.852425	-0.01513	0.014655	0.666667	0.052632	0.894118	0
55	277	0.431373	0.262745	0.129412	0.297951	0.143367	-0.00586	0.073593	0.7	0.431373	0
113	244	0.482353	0.305882	0.231373	0.35014	0.129126	0.01412	0.049479	0.520325	0.482353	0

Figure 3. An example of pixels records in CSV file

3. Classification process

Two classification methods, naïve Bayes and Bayesian decision [1], are used in the project. In naïve Bayes method, we assume that all features are independent. Then we can get posterior probability from prior probability. For the probability of each feature, we assume it follows Gaussian distribution. We estimate parameters of Gaussian distribution based on samples in training set. For the likelihood distribution, we also simplified it as a Gaussian distribution. Then we can get posterior probability as follows:

$$P(w_i|x_j) = \frac{P(x_j|w_i)P(w_i)}{P(x_j)}$$

In our case, the w_i has two possible values: skin or non-skin. The x_j is an element of a specific color model. In RGB color model, x_1 is red (R), x_2 is green (G), x_3 is blue (B). In YIQ color model, x_1 is luma (Y), x_2 is orange-blue (I), x_3 is purple-green (Q). In HSV, x_1 is hue (H), x_2 is saturation (S), x_3 is value (S). Though these color models use different specifications for a pixel. They can be converted between each other without losing any color information. In our pixel selection tool, we get a pixel color from its RGB color model, then convert the RGB color model to the other two color models. In naïve Bayes method, since features are independent, to get the posterior probability of $P(w_i|X)$, in which X denote $[x_1, x_2, x_3]$ for a color model. We can use the formula:

$$P(w_i|X) = P(w_i|x_1)P(w_i|x_2)P(w_i|x_3)$$

In a real naïve Bayes implementation, we can convert the above formula to an equivalent expression as:

$$P(w_i|X) \sim P(x_1|w_i)P(x_2|w_i)P(x_3|w_i)$$

Then we compare $P(w_1|X)$ and $P(w_2|X)$ for a given color model pattern X, and classify X to the one with larger value.

The Bayesian decision classifier [1] is more complicated than the naïve Bayesian method by considering means and covariance of features in each class. In the classification model, we have 2 classes, each class has its own mean μ_i and covariance Σ_i. We use the formula as discrimination function:

$$g_i(X) = -\frac{1}{2}(X-\mu_i)^T\Sigma_i^{-1}(X-\mu_i) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma_i| + \ln P(w_i)$$

Because we have two classes, we calculate $g_1(X)$ and $g_2(X)$ for the two classes, respectively. Then classifies X to the one with larger value. In MATLAB, the skeleton of training and testing are shown as following code.

```
%% Train Naive Bayes for three color models
NBModel_RGB = fitNaiveBayes(X_RGB,Y);
NBModel_YIQ = fitNaiveBayes(X_YIQ,Y);
NBModel_HSV = fitNaiveBayes(X_HSV,Y);

%% Test Naive Bayes
predictNB_RGB = predict(NBModel_RGB,tX_RGB);
predictNB_YIQ = predict(NBModel_YIQ,tX_YIQ);
predictNB_HSV = predict(NBModel_HSV,tX_HSV);

%% Train Covariate Bayes for three spaces
COVModel_RGB = fitCovBayes(X_RGB,Y);
COVModel_YIQ = fitCovBayes(X_YIQ,Y);
COVModel_HSV = fitCovBayes(X_HSV,Y);

%% Test Covariate Bayes
predictCOV_RGB = predictCovBayes(COVModel_RGB,tX_RGB);
predictCOV_YIQ = predictCovBayes(COVModel_YIQ,tX_YIQ);
predictCOV_HSV = predictCovBayes(COVModel_HSV,tX_HSV);
```

For the training procedure, we load previous pixels file from storage, then separate the features to three groups for three different models, and ground truth to an individual set. For each training, we use three features as three elements from one color model. After training, we evaluate the classifier model by using pixels files from testing set. A pixels file in testing set has the same format as the file in training set. We also use our Imtool2 to randomly label some pixels of pictures from testing set. Then we can verify if a pixel in test set is correctly classified to not. Furthermore, we use some pictures from third part source to evaluate these classifiers as shown in the preview section.

4. Experiment

The experiment part is divided into two subsections. In the first subsection, we discuss some statistics from collected pixels. In the discussion, we figure out some key observation from some statistical plots. These observations can direct us in feature selection and classification model selection. In the second section, we mainly show performance evaluation of each classifiers. Then we will discuss which classifier and features are better than others. From the evaluation result, we can get some idea about which features are crucial in the skin detection, and know which problems are difficult in current system.

The basic setup of the experiment is that we get 21 pictures from Internet. These are typical pictures shows different colors of skins, and different colors of background. Some pictures have more than one people. Some pictures have a group of peoples, even in one picture, these peoples are from different races and shows different skin colors. Some basic statistics are shown in Figure 4 and Figure 5.

These pictures are showing different scenarios, for some scenarios, the backgrounds have very similar colors of foreground portraits. For example, the last picture in test set, the ground of background is yellow floor. The foreground is a group of members from a basketball team. In addition, these members have different skin colors causes by their different races and perspective of lighting. This brings more difficulties to skin detection. If the color spectrum of background is include in the foreground skin, we cannot distinguish skin and non-skin in this case. However, we use three color models, in different models we can get different results.

Skin pixels	Non-skin pixels	Total
1052	1218	2270
46.34%	53.66%	100%

Figure 4. Training set

Skin pixels	Non-skin pixels	Total
404	568	972
41.56%	58.44%	100%

Figure 5. Testing set

4.1 Observations from statistics

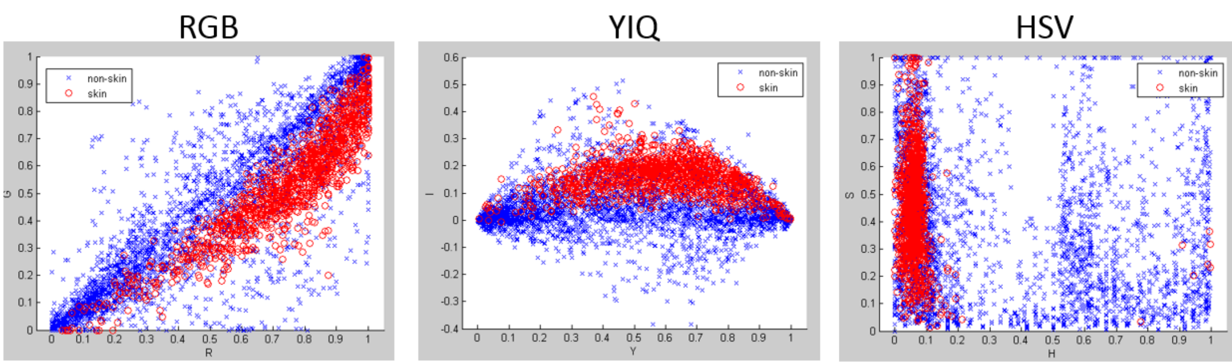


Figure 6. The comparisons of first two features in each color model. For the any two features' comparisons, please refer the appendix of the report.

Based on the collected dataset, including training samples and testing samples, we draw a scatter plot to show the pixels of skin and non-skin in a two dimensions plot. In Figure 6, we just show the first two features from each color model as 2 dimension of a scatter plot. The Figure 6 (RGB) plot shows the red dots from skin, the blue dots from non-skin. The x axis is value of red feature, the y axis is value of green feature. For the comparison of any two features in each color model, please refer the appendix of the report. Form the appendix, we can get that other two features' comparisons in a same color model is similar to the plots here. From the features, we can find out the skin pixels occupies partial part in the non-skin area. However, it is difficult to discriminate skin from non-skin. However, we can still draw some important observations. In RGB space, the skins' pixels have higher red and green values. In YIQ space, the skin pixels is at the upper part of the scatter plot, means that skin pixels' have large values in I component. In HSV space, the skin pixels have lower H component. From the comparisons, we know that based on the basic color model, it is difficult to fully figure out skin from non-skin. However, in our classification model, we are using probability model for classification, different from linear discrimination model. The probability model discriminate different patterns based on distribution of patterns in each class. To draw the distribution of samples in each class, we do some statistics based on histogram plots. We plot all histograms of each component in each color model. There are 9 histograms are gotten. The full histograms is attached in appendix. Based on the discussion in features comparison, we get that skin pixels are scattered more in higher R, higher I and lower H component. Therefore we plot histogram as shown in Figure 7. By just reviewing Figure 7, we cannot get clear idea of distribution of skin and non-skin. However, we can get some ideas of color distribution in pictures. The red component in RGB color space does not show any special distribution. It looks like uniform distribution with a little bit higher probability when R is near 1. For the phase component in YIQ space, it shows a shape likes Gaussian distribution, in which most of samples are near I=0. For the component H in HIV space, the distribution of H shows a long tail shape distribution.

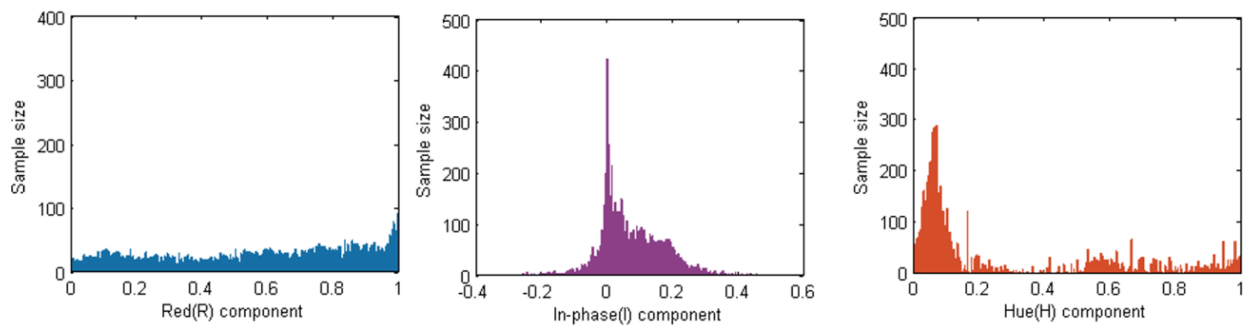


Figure 7. The histogram of pixels, the histogram for all 3 components in each color model is attached in appendix.

Since, we do not get any significant observations from Figure 7. We can plot skin and non-skin at two different plots. Then observe the different distributions of these two classes.

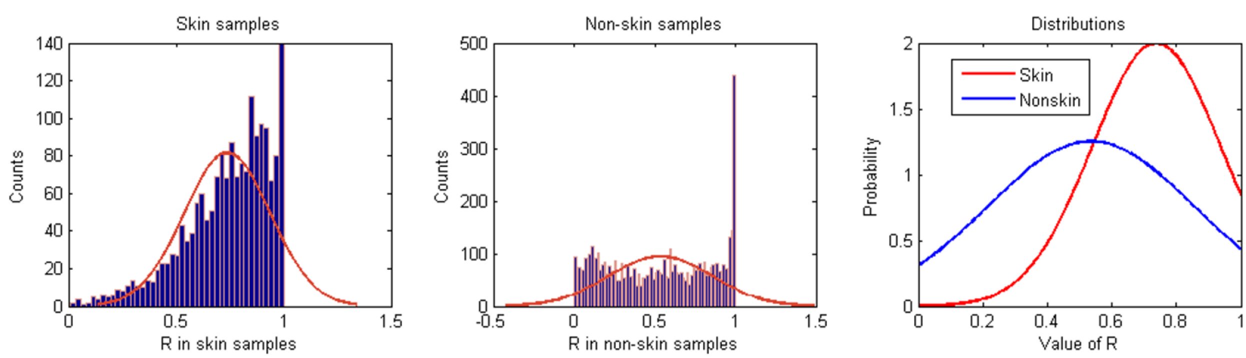


Figure 8. The histogram of skin and non-skin in R component of RGB space

In the RGB space, we show the histogram of skin and non-skin of R component as Figure 8. In the figure, we fit the two different histogram to two Gaussian distributions shown as plot of the right side in Figure 8. From plot, it is clear that the skin and non-skin have different distribution. The pixels from skin has higher mean and smaller variation. However, from the middle plot of Figure 8, the distribution of non-skin cannot simply be fitted to a Gaussian distribution. Therefore, by assuming it is a Gaussian distribution, we may lose some precision in training classification. In the project, to simple the case, we assume all distributions follows Gaussian distribution, then we can train the proposed two classifiers. The Figure 9 gives histograms and fitted distributions of I component in YIQ space. In this figure, we can see skin and non-skin follow Gaussian distribution well. In the distribution comparisons, we can see two significant different distribution. Therefore, we can infer that the skin and non-skin are differentiable on the component.

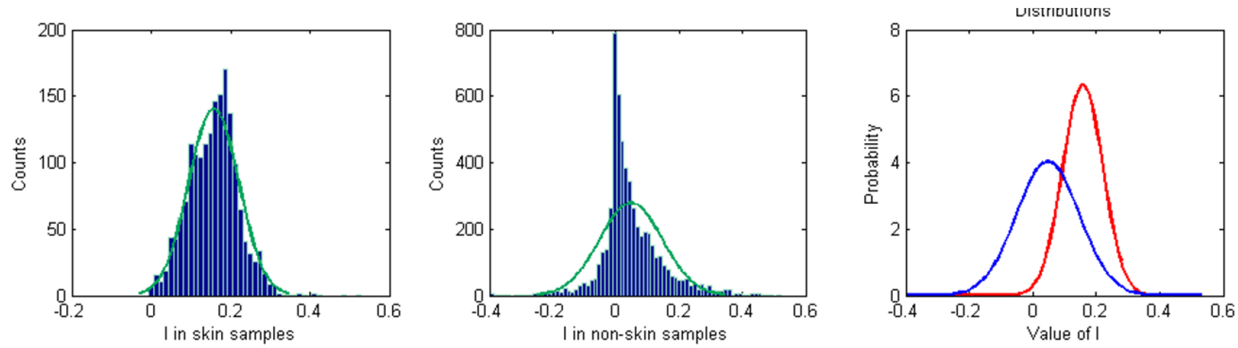


Figure 9. The histogram of skin and non-skin in I component of YIQ space

The histograms from Figure 10 shows H component in HSV space, it is difficult to say that non-skin follows a Gaussian distribution, because there are three peaks in the middle plot. It looks more like addition of multiple different Gaussian distribution. To simply the analysis here, we just fit it to one Gaussian distribution. The distribution shows significant difference between skin and non-skin. However, we have over simplified the non-skin distribution, the fitted distribution does not accurately present the actual distribution of non-skin pixels.

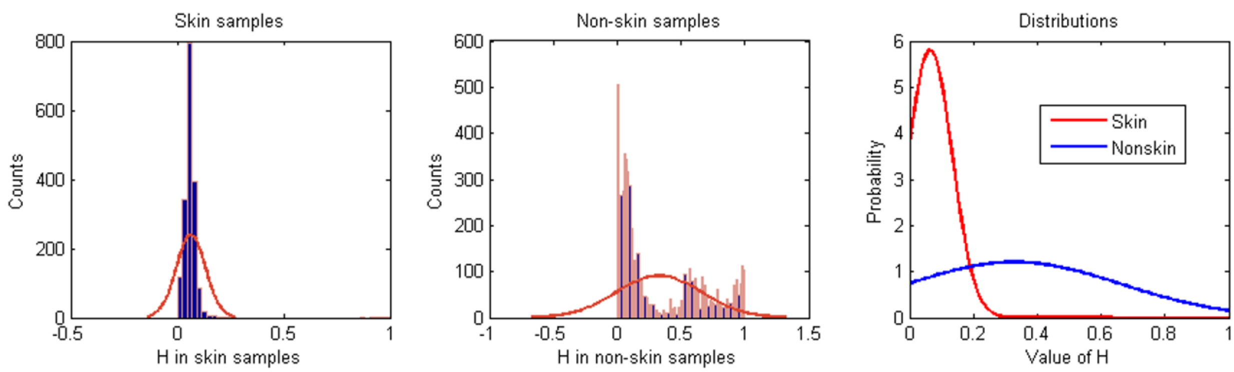


Figure 10. The histogram of skin and non-skin in H component of HSV space

Furthermore, to analyze these 9 features in three color models, we put all these features in one set and do PCA analysis. Based on the PCA analysis, we may extract composition of features and reduce dimensions. Based on the top 3 largest eigenvalues, we get corresponding eigenvectors. Then we use these eigenvectors as coefficient to combining all 9 features. Then we get 3 principal dimensions from 9 dimensions, as shown in Figure 11. For the full 9 dimensions after PCA process can be found in appendix. The result of the PCA analysis shows that values in dimension 2 and 3 are limited in a short interval. To get a clearer showing of distribution for skin and non-skin pixels, we compare each 2 pairs of dimensions in Figure 12. The observer from the right plot of Figure 12 shows a clear distinction between skin and non-skin pixels distribution. It also shows some features that are linear discriminable. In the left and middle plots, the pixels from two classes are intermingled. From these statistics, we can draw some preliminary conclusion. If we detect skin only based on the color of single pixel, some background pixels that have similar color can be miss classified as skin. However, based on some features from different color models, we can improve performance in skin classifications.

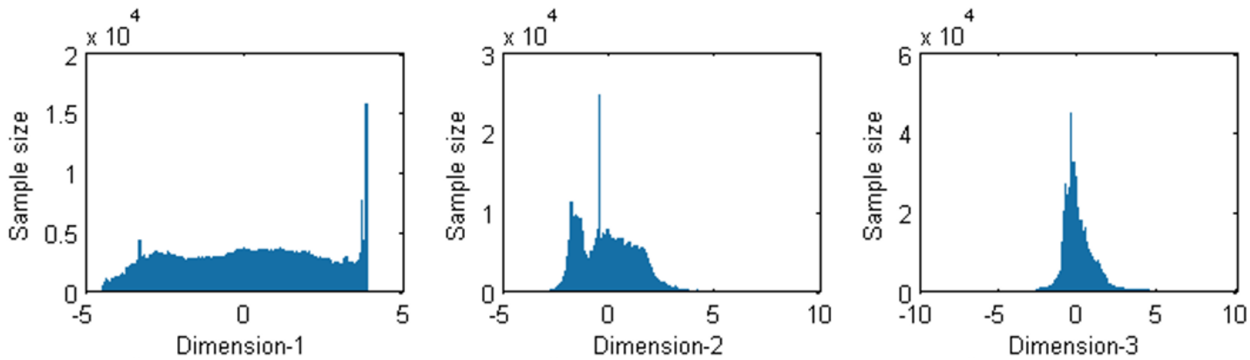


Figure 11. The 3 important dimensions in PCA analysis

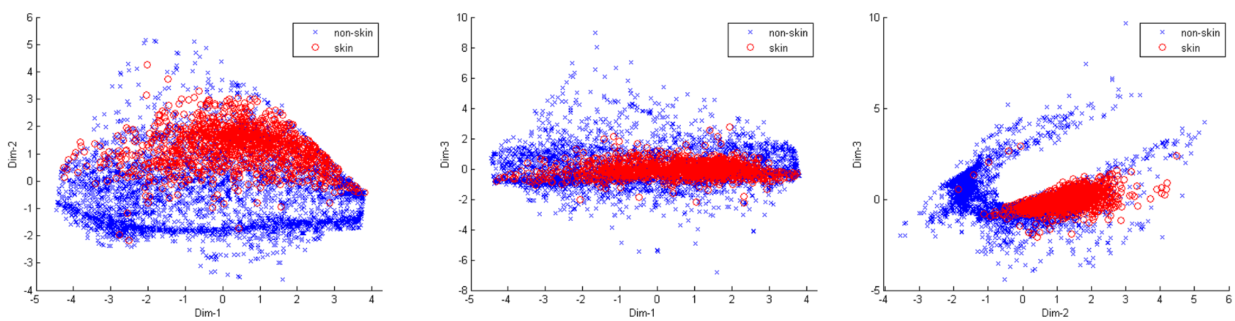
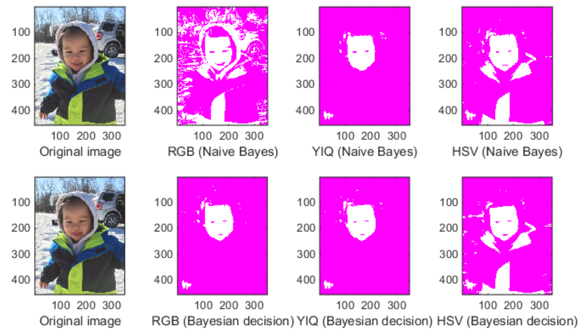


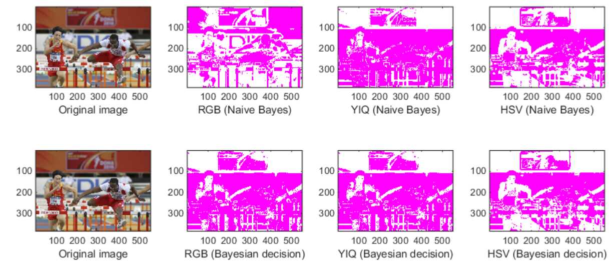
Figure 12. Comparison of any 2 dimensions from 3 principle dimensions

4.2 Performance evaluation

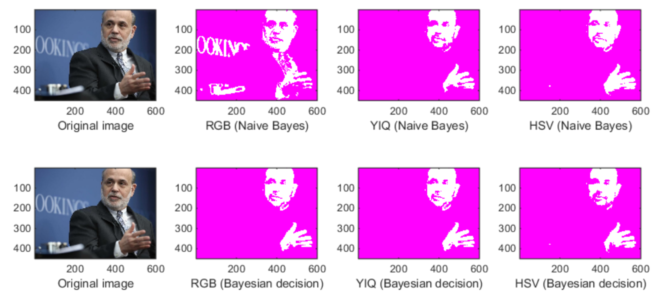
Based on some statistical analysis, we training 6 classifiers. 3 out of 6 classifiers are using naïve Bayes method, the rest 3 are using Bayesian decision method. In each method, one classifier is trained by 3 features. The 3 features is from RGB, or YIQ, or HSV. The testing set is used to estimate the performance. We evaluate the performance based on the precision of classification result. The detection results of 8 pictures from testing set are listed as follows. In each block, there are two rows to show pictures. The original pictures are shown at the left side, the pictures processed based on Naïve Bayes classification are list at the first row for RGB space, YIQ space and HSV space from left to right. The second row shows the pictures processed based on Bayesian decision method. From left to right, the pictures are processed based on RGB, YIQ and HSV. The skin area is white color, the non-skin area is purple color. From our intuition for these pictures, we can see that the YIQ color model gives the best result. In general, the Bayesian decision method is better than Naïve Bayes method. For example, in picture 5 in the following table, the naïve Bayes method miss classifies a large part of yellow background as skin. While the Bayesian decision method just miss classifies a smaller part of yellow background. There are some similar missed classification in picture 7 and picture 8, which have some skin colors like background. Another observation is in YIQ color model, the Naïve Bayes method and Bayesian decision method get similar result, at least for all the 8 pictures in testing set.



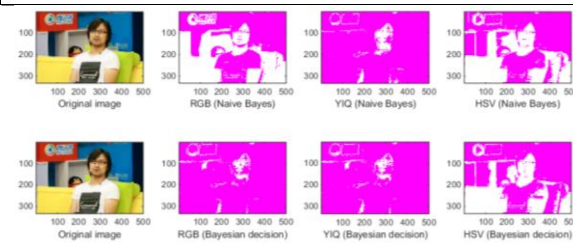
Picture 1



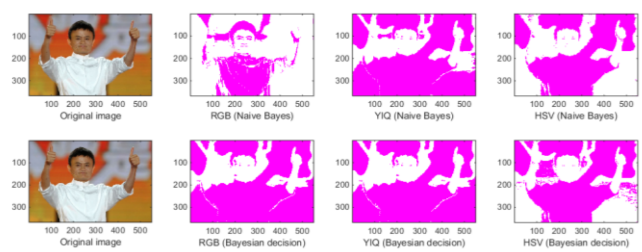
Picture 2



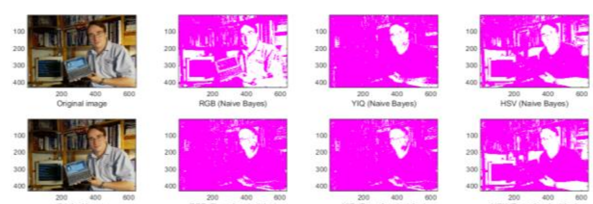
Picture 3



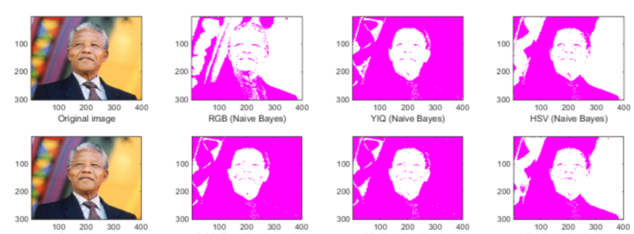
Picture 4



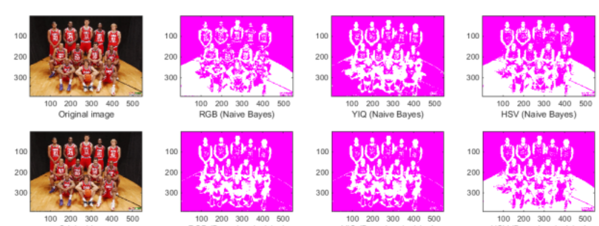
Picture 5



Picture 6



Picture 7



Picture 8

The YIQ color model shows a kind of stable under two different classification methods. Because the classification results do not show significant difference for the two methods. Based on the pixels we picked for testing dataset, we can get the precision for each method under each color model. In naïve Bayes method, the color model shows the best discrimination is YIQ, as shown in Figure 13. We also train a classifier based on 3 dimensions transformed and reduced by PCA, it shows some improvement for RGB and HSV color space. Nevertheless, it is not better than YIQ color space.

Figure 13. Confusion matrix of Naive Bayes method

For the Bayesian decision method, as shown in Figure 14, we can figure out that the method performs well in both RGB and YIQ color space. The precision is 86.5%. The HSV method gives the worst result.

By comparing naïve Bayes method and Bayesian decision method, we can draw the fact that, for each color space, the Bayesian decision is better than naïve Bayes. The Bayesian decision method based on YIQ color space gives the best result in these 8 classifiers. The reason is that the Bayesian decision method considers the covariance of features, which is not considered in naïve Bayes method. For the PCA method, it is not better than YIQ method, the reason is that RGB, YIQ and HSV are linear combination of each other, the reduced dimensions based on PCA do not provide more information in discrimination. However, in a general case, if we do not know which color model is the best one, we can apply the PCA method, because it gives a convincing result, though it is not the optimal one. In addition, as we have mentioned before, the YIQ color space performs very similar result in both two classification methods. Since naïve Bayes uses less resources for training and testing because it is simpler than Bayesian decision method. This gives us a benefits that on resource limited devices, we can use naïve Bayes method to get comparable result as Bayesian decision.

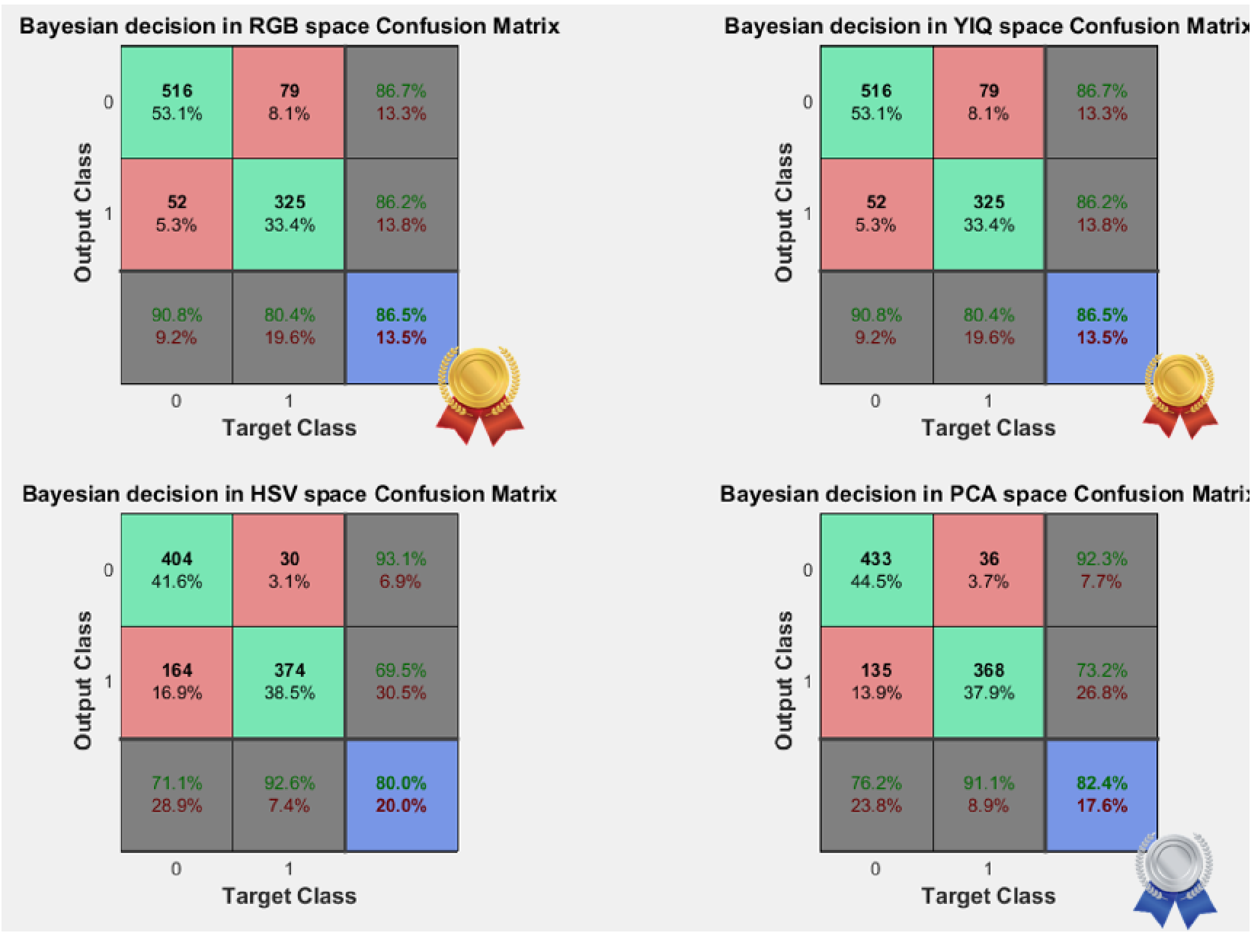


Figure 14. Confusion matrix of Bayesian decision method

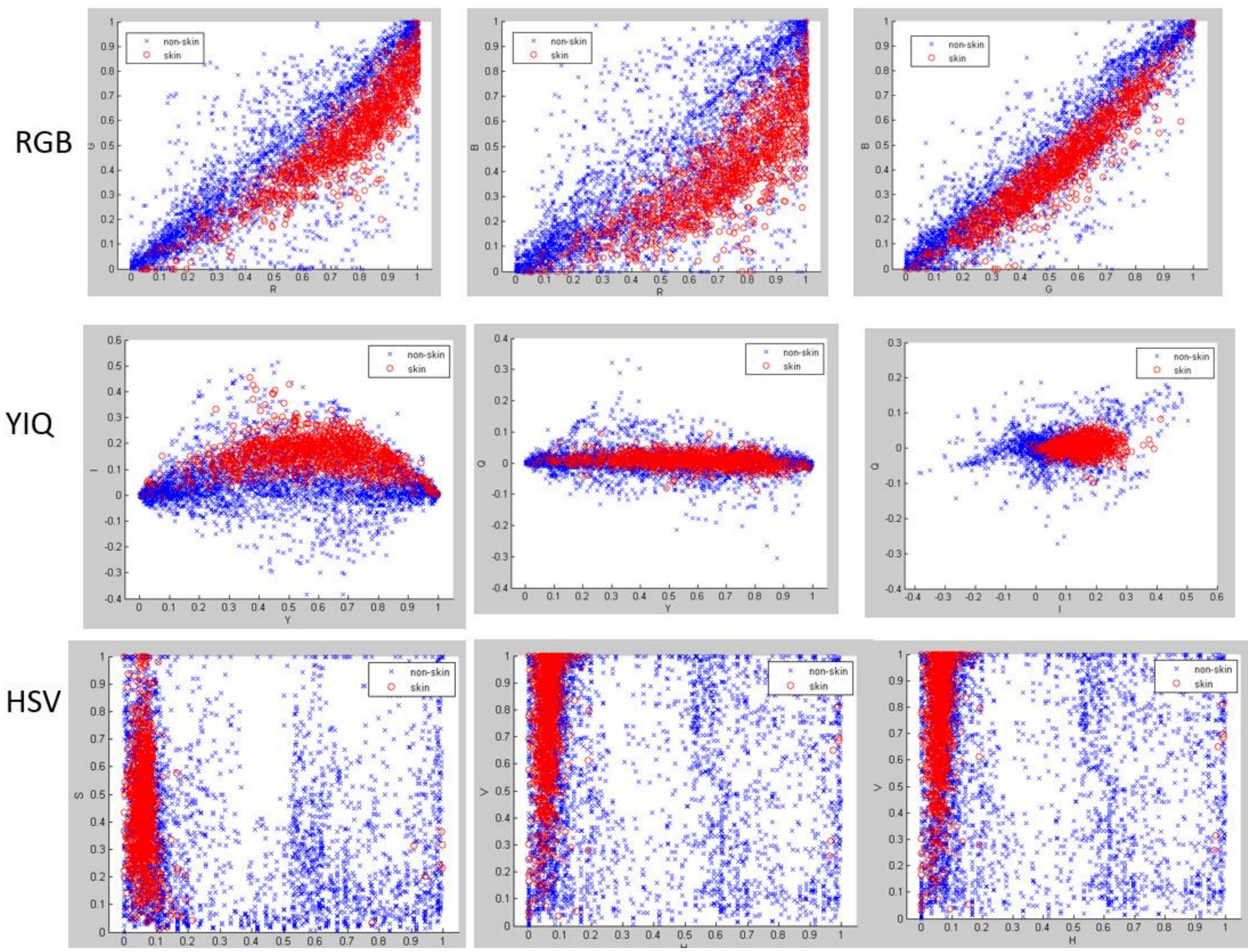
5. Conclusion

From the experiment result, we present that our two methods are both effective for the skin detection. In particular, the Bayesian decision by considering different means and covariance in different classes gives the best result in YIQ color space. In addition, the naïve Bayes method in YIQ color space also gives a result better than other 2 two color spaces. For the future work, we can still improve the skin detection by considering some other features. In the current work, if background color is similar to the human skin color, the proposed method here cannot distinguish them. Therefore, we can consider some other features to improve the current work.

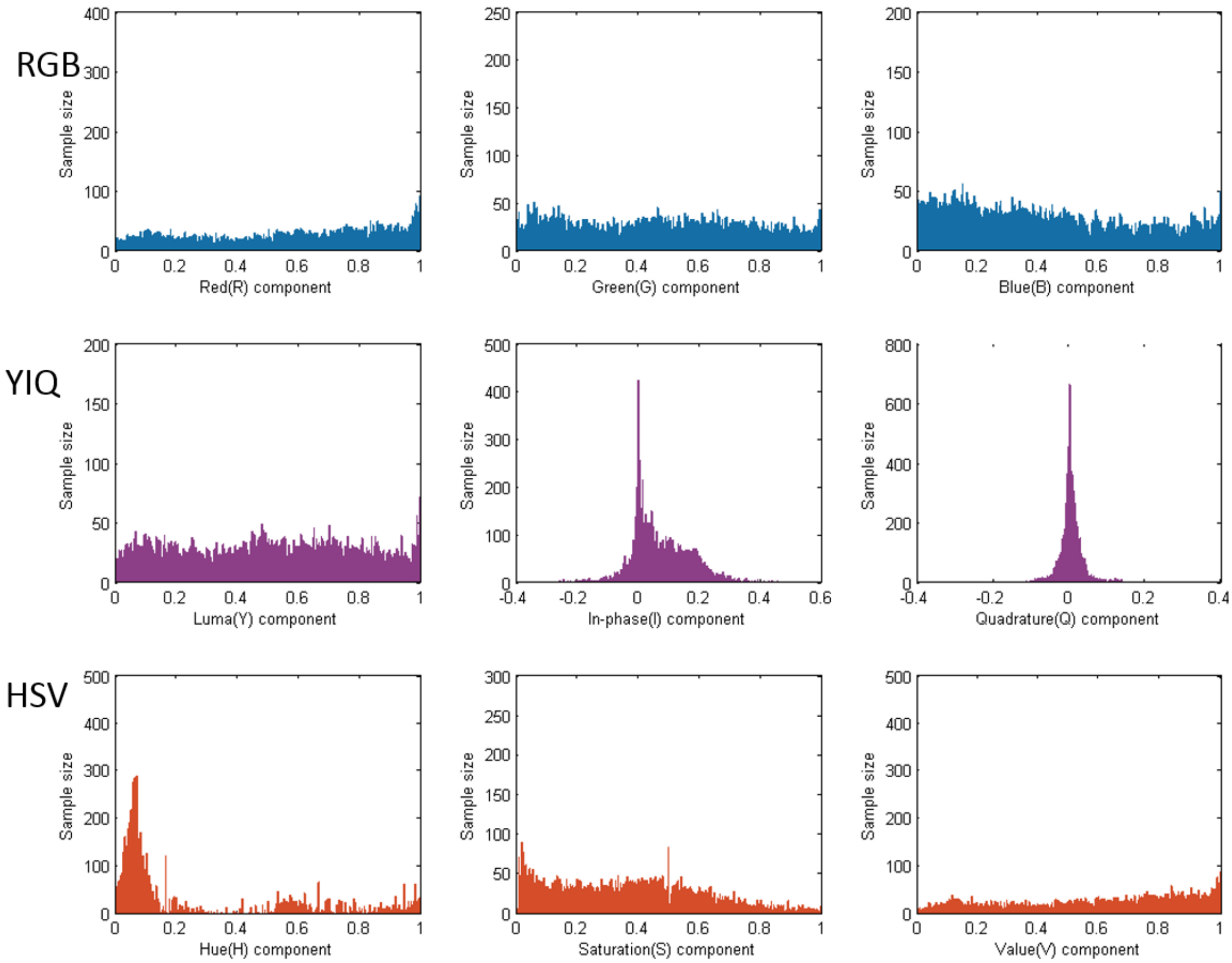
References

[1] Duda, Richard O., Peter E. Hart, and David G. Stork. Pattern classification. John Wiley & Sons, 2012.

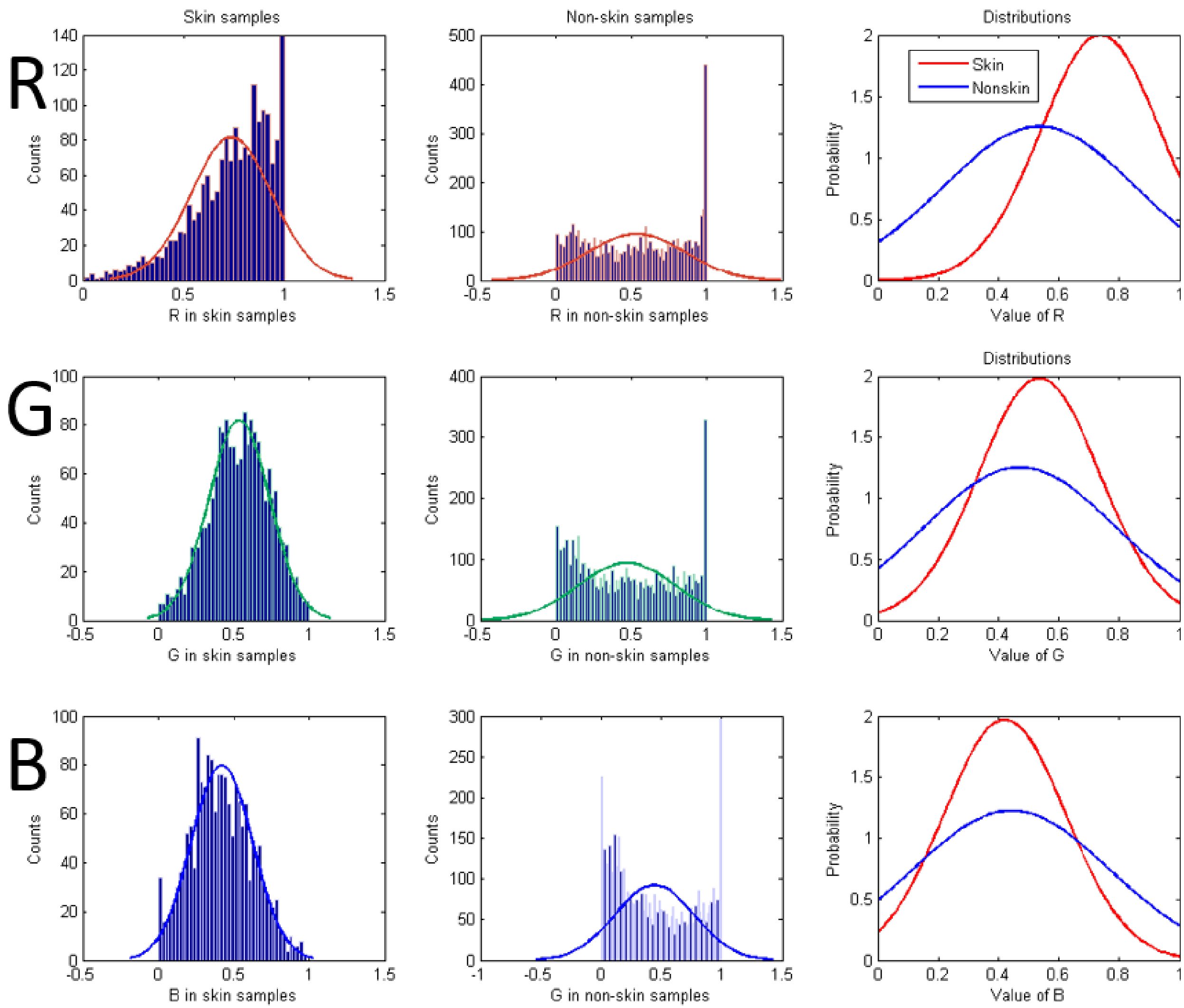
Appendix



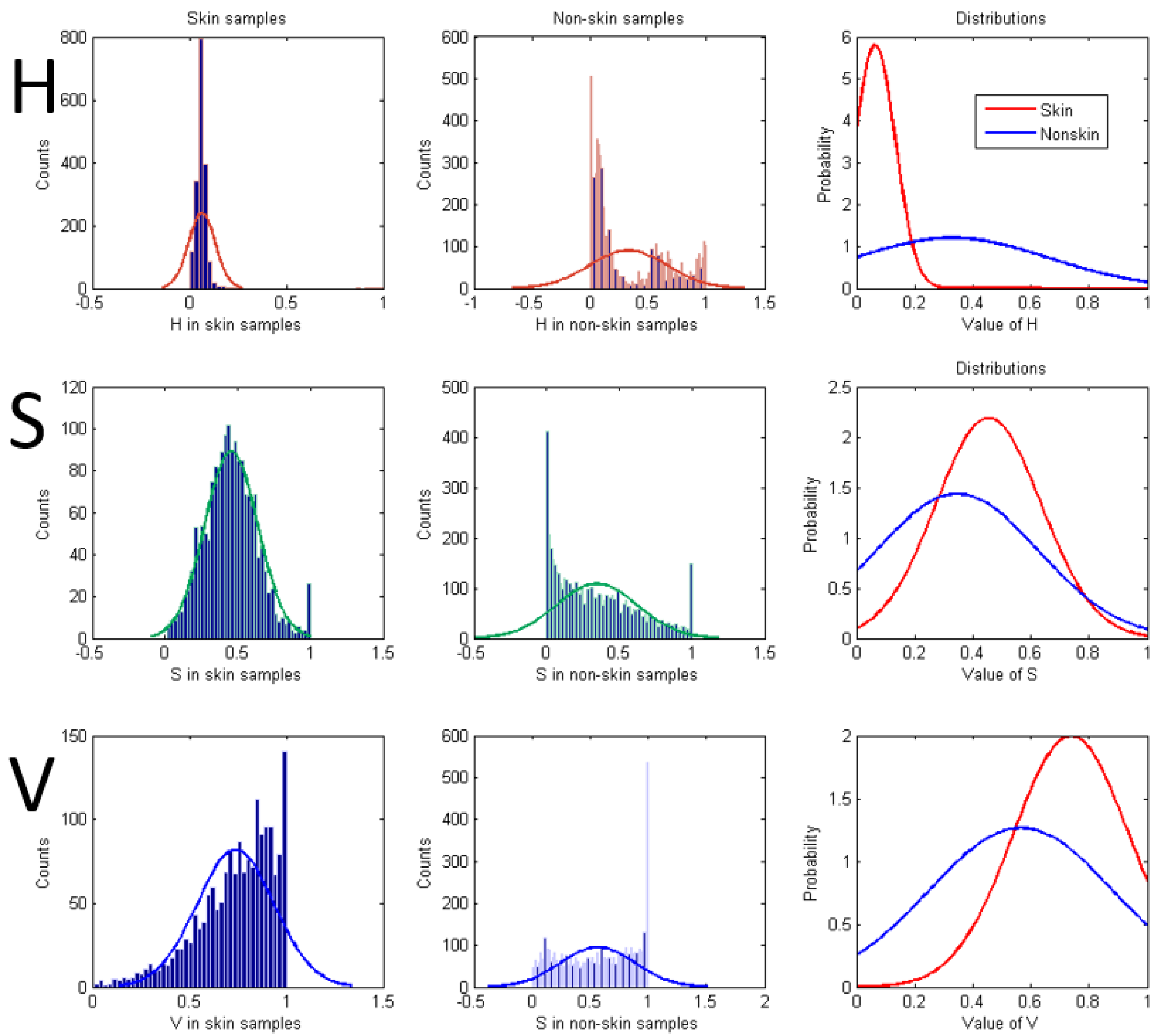
The scatter plots based on any two elements from vectors in three color models.



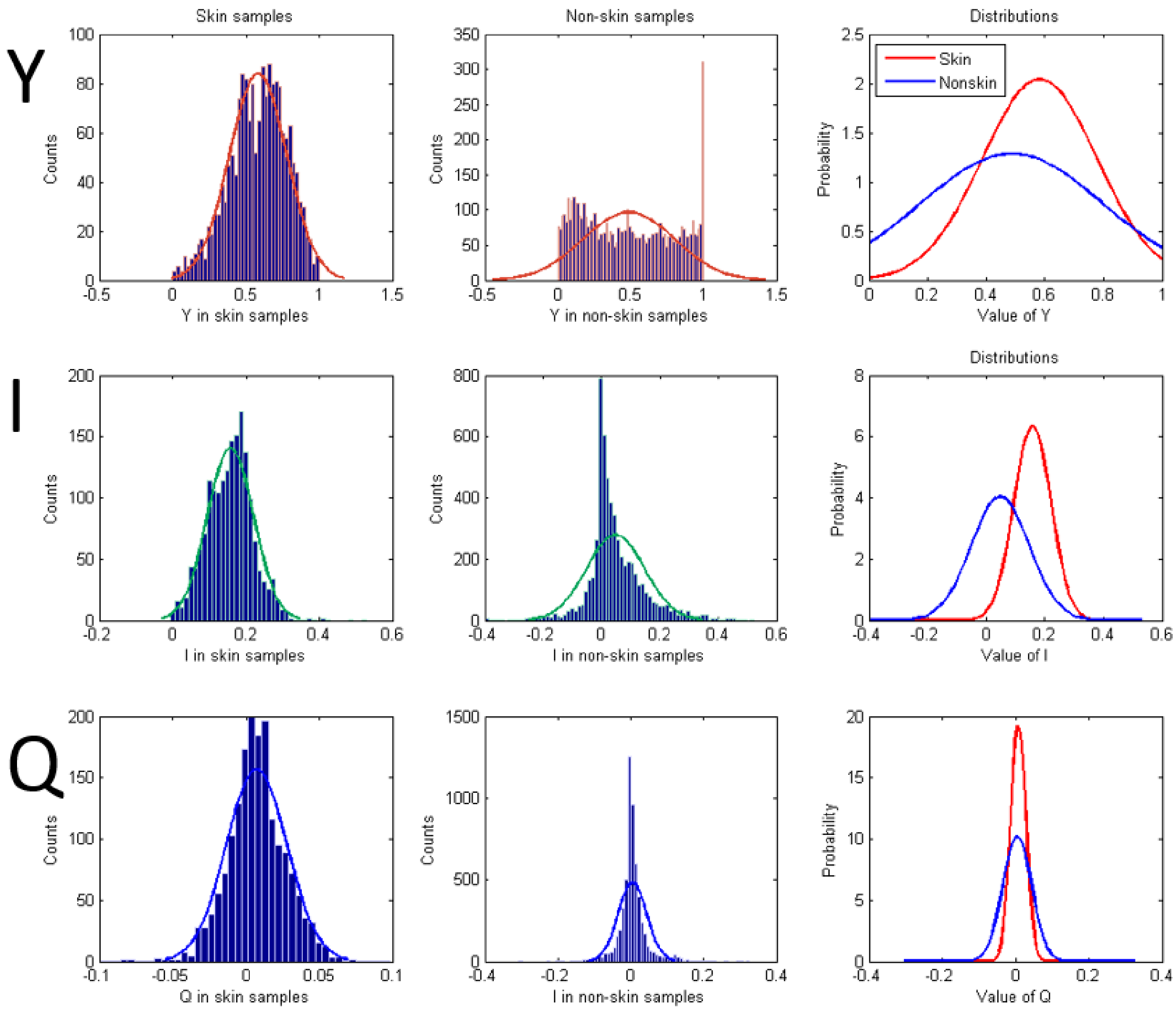
The collected pixels' histograms in each color models.



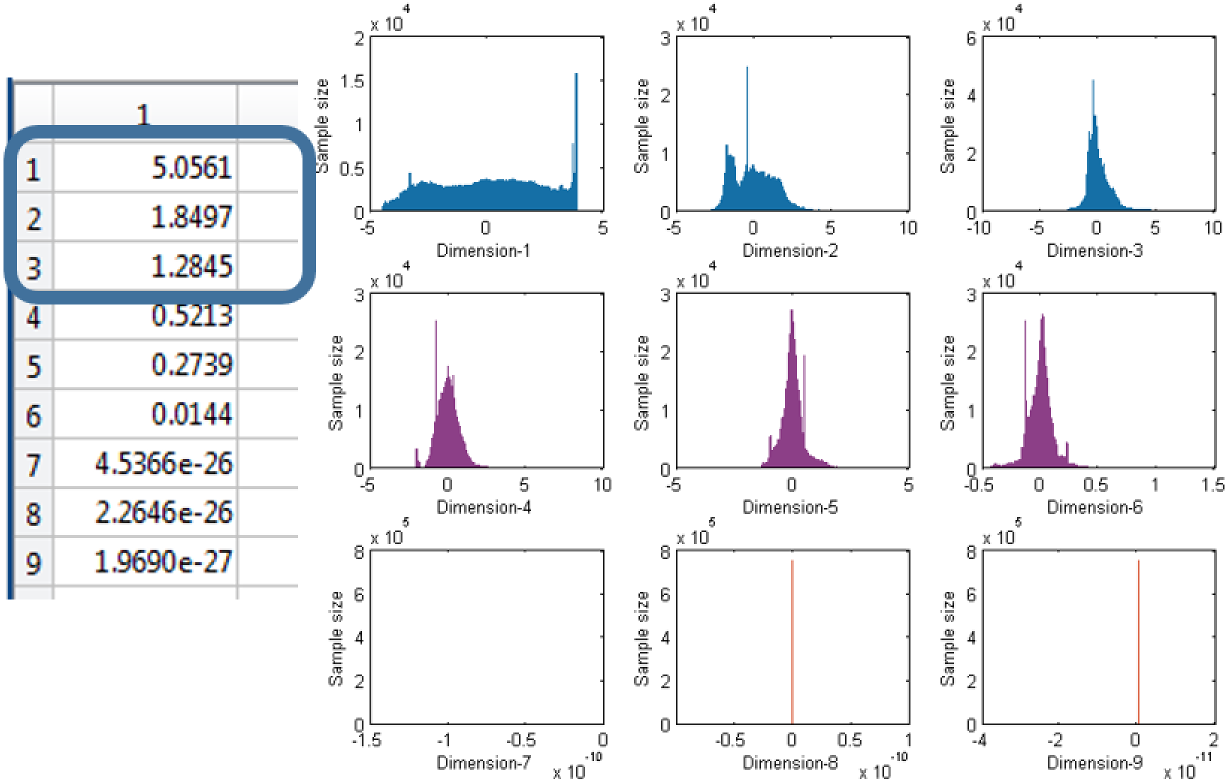
Fitting samples in RGB color model to Gaussian distributions.



Fitting samples in HSV color model to Gaussian distributions.



Fitting samples in YIQ color model to Gaussian distributions.



PCA method for 3 color models.