Contents

SKIN DATASET ANALYSIS

```
clear all

%loading data

filehere= 'Users/Anna/Documents/MATLAB/Skin_NonSkin.txt';
fid=fopen(filehere);
mydata = textscan(fid,'%f%f%f%f','delimiter',',');
fclose(fid);

%naming each feature

Blue = mydata{1};
Green = mydata{2};
Red = mydata{3};
Skin = mydata{4};

X=[Blue Green Red];
labels=Skin;

%Problem: Let's predict the sample to be skin or not skin, based on the B,G,R values
%There are only three features, and in that, all of them have a same scale.

%Plotting data

scatter3(Blue, Green, Red, 30, Skin, 'filled');
title('Skin or not?')
xlabel('Blue')
ylabel('Green')
zlabel('Red')
legend('Not skin pixel')
figure;
%min value of skin colors is around 106.0 and max value of skin colors is 255.0

%Correlation features(color) vs target
r=corr([X labels])
% => So blue has the highest correlation

%mean and standart deviation
mx=mean(X)
stdx=std(X)

% Now I'm gonna visualize the eigenvectors to represent
% my variance
cX =cov(X)
[V,D] = eig(cX)
b=diag(D)
[b idx]=sort(b,'descend'); V=V(:,idx)

plot(V);
```

```
xlabel('x');
ylabel('y');
title('eigen vectors');
set(gca,'XTick',[1 2 3],'XTickLabel',{'Blue', 'Green', 'Red'})
legend ('1. Eigvec ','2. Eigvec ','3. Eigvec')
```

```
r =

    1.0000    0.8553    0.4964    0.0920
    0.8553    1.0000    0.6601   -0.1203
    0.4964    0.6601    1.0000   -0.5700
    0.0920   -0.1203   -0.5700    1.0000


mx =

  125.0654  132.5073  123.1772


stdx =

   62.2557   59.9412   72.5622


cX =

   1.0e+03 *

    3.8758    3.1915    2.2423
    3.1915    3.5929    2.8711
    2.2423    2.8711    5.2653


V =

    0.6031    0.5873    0.5398
   -0.7767    0.2779    0.5653
    0.1819   -0.7602    0.6237


D =

   1.0e+03 *

    0.4421         0         0
         0    2.4834         0
         0         0    9.8084


b =

   1.0e+03 *

    0.4421
    2.4834
    9.8084


V =
    0.5398    0.5873    0.6031
    0.5653    0.2779   -0.7767
    0.6237   -0.7602    0.1819
```

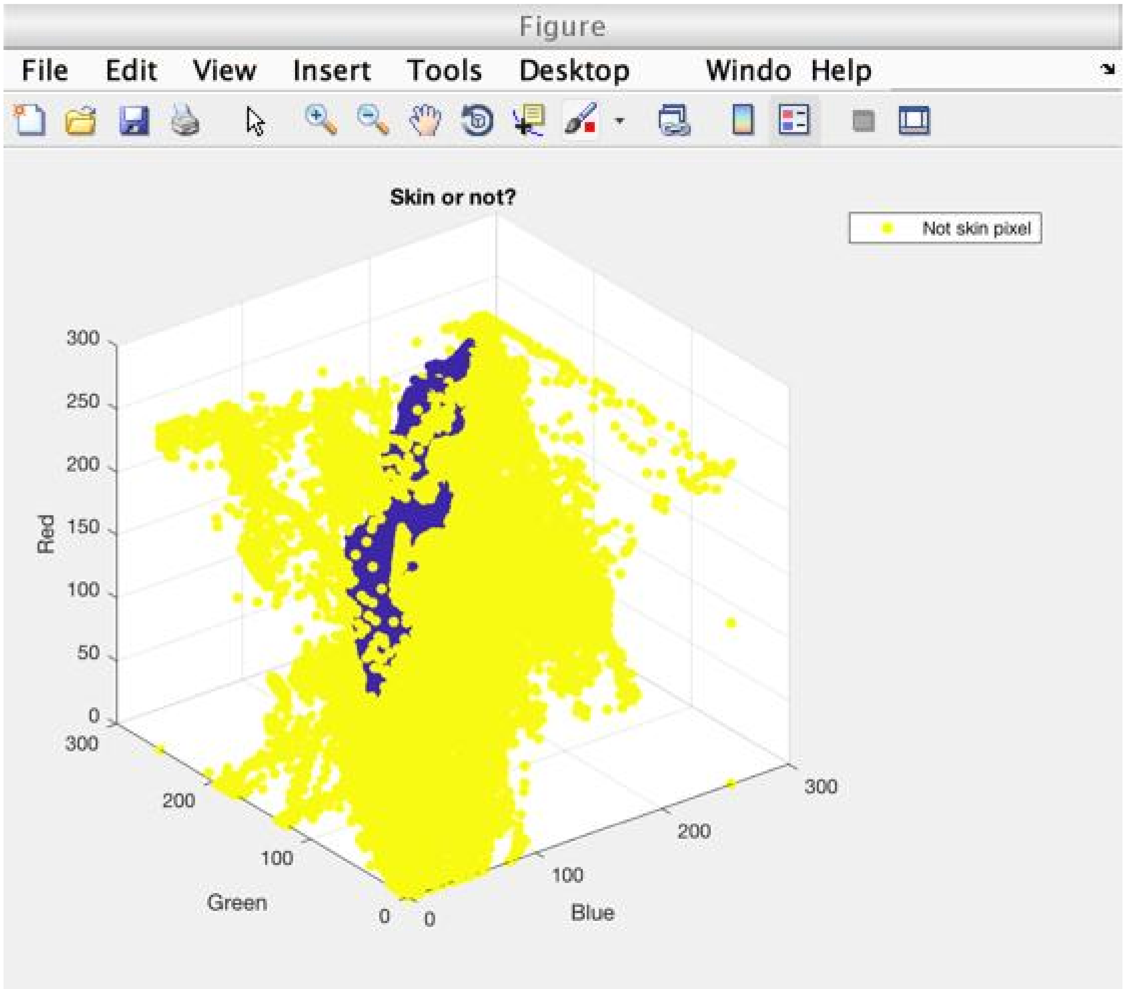


Figure
File Edit View Insert Tools Desktop Windo Help
Skin or not?
Not skin pixel
Red
300
250
200
150
100
50
0
Green
300
200
100
0
Blue
0
100
200
300

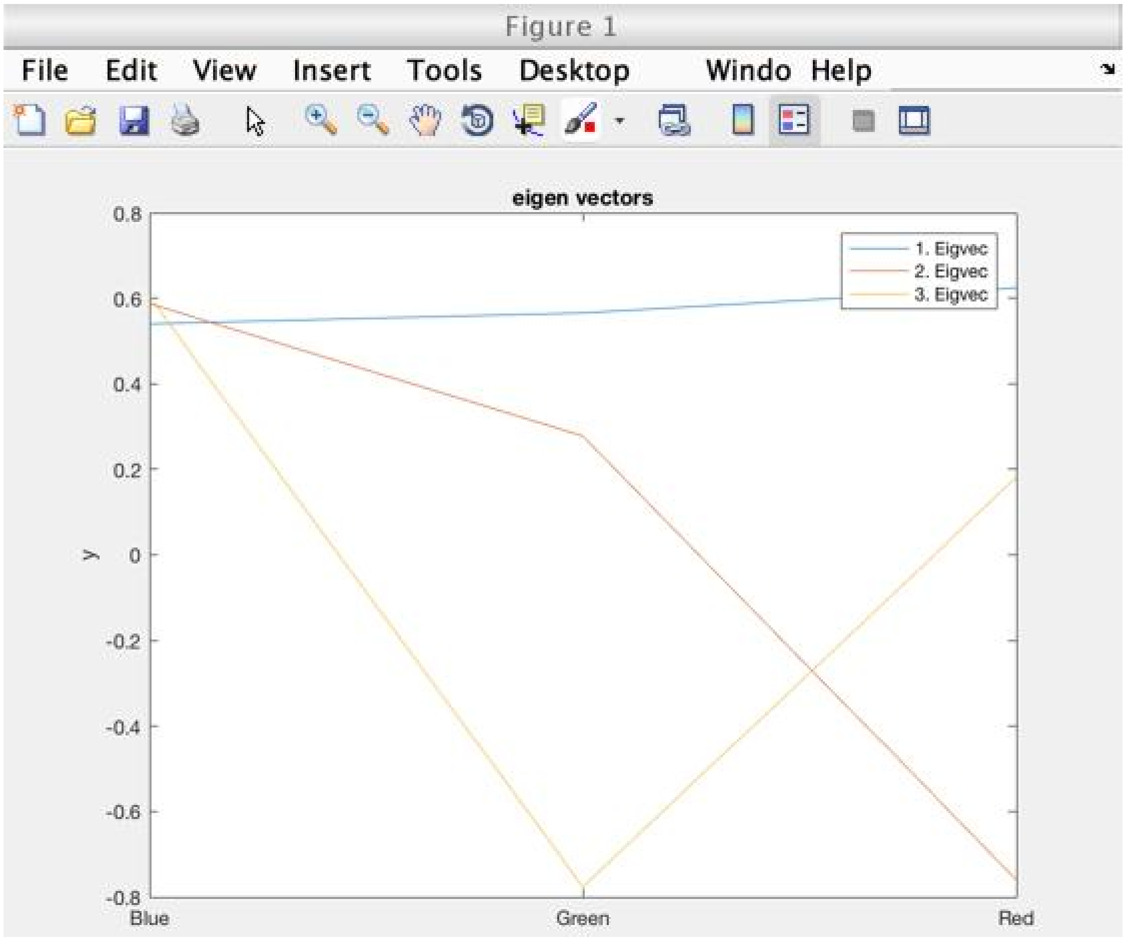


Figure 1
File Edit View Insert Tools Desktop Windo Help
eigen vectors
1. Eigvec
2. Eigvec
3. Eigvec
y
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
Blue
Green
Red

Principal Component Analysis

```
close all
%calculating sample mean
Xmean = mean(X);

%centering data
Xcentered = X - ones(length(X(:,1)),1)*Xmean;

%calculating covariance
Xcov = cov(Xcentered);

%eigenvalue decompositon
[eigenvect,eigenval] = eig(Xcov);

%eigenvalues from diagonal
eigenval = diag(eigenval);
%sorting
[eigenval, idx]=sort(eigenval, 'descend');
%plotting
eigenvect = eigenvect(idx); V = V (: , idx ) ; %(V is my sorted eigenvetors)

%cumulative sum of eigenvalue percentage
cs = cumsum(eigenval)/sum(eigenval)
f = figure
plot(cs)
title ('RGB:Cumulative Eigenvalues ')
set (gca ,'XTick' ,[1 2 3] , 'XTickLabel' ,{'1. ','2. ','3. '});
xlabel (' Eigenvalue ') ;
% First eigenvector explains around 77% of the variance in the data, and
% together with the second 96.5%. That means, the skin dataset would be
% well represented by choosing the first two eigenvalues.


V (: ,1:2)
figure
plot ( V (: ,1:2) )

% projection on eigenvectors with highest eigen values(=reduction to 2
% dimensions)

scores = Xcentered*V(:,1:2);

% conversion of scores to prdataset
addpath(['/Users/Anna/Documents/MATLAB/prtools']);

z = prdataset(scores , labels);

figure
scatterd(z)
xlabel('1st eigenvector')
ylabel('2nd eigenvector')
title('Projections onto eignevectors')

%Reconstruction of dataset based on 2 eigenvectors

Xr = scores(:,[1,2])*V(:,[1,2])'+ones(length(Xcentered(:,1)),1)*Xmean;
```

```
% conversion to PRtools

Rdata = prdataset(Xr,labels,...
'featlab',['Blue ';'Green';'Red  '],'name','Sample Features');

figure
subplot(2,1,1)
scatterd(X(:,[1,2]));
title ('Original ')
xlabel ('Blue')
ylabel ('Green')

subplot(2,1,2)
scatterd(Rdata(:,[1,2]));
xlabel('Blue')
ylabel('Green')
title('Reconstructed based on 2 eigenvectors')
%Those figures don't tell me much
```

```
cs =

    0.7703
    0.9653
    1.0000


f =

  Figure (1) with properties:

      Number: 1
        Name: ''
       Color: [0.9400 0.9400 0.9400]
    Position: [360 278 560 420]
       Units: 'pixels'

  Use GET to show all properties


ans =

    0.6031    0.5873
   -0.7767    0.2779
    0.1819   -0.7602
```

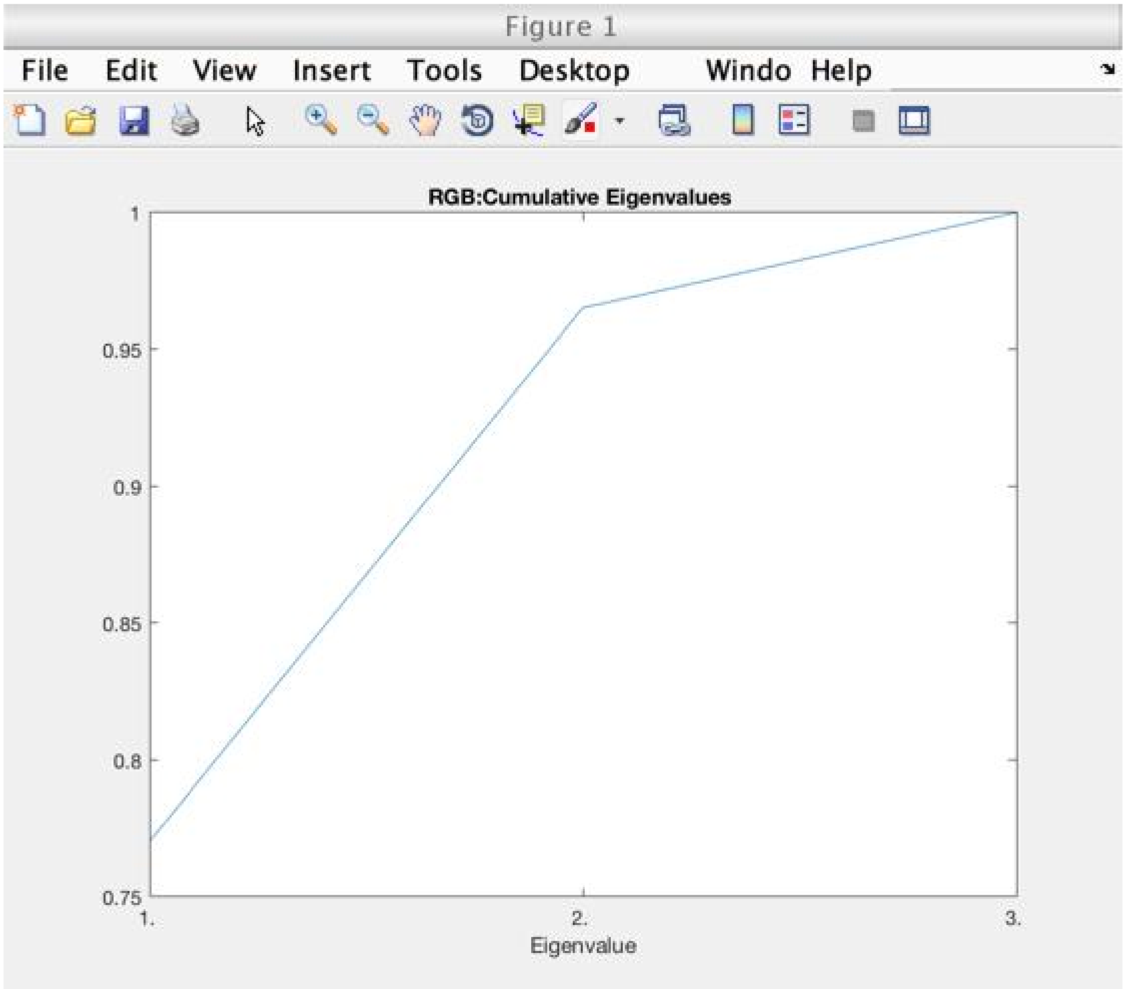


Figure 1
File
Edit
View
Insert
Tools
Desktop
Windo
Help
RGB:Cumulative Eigenvalues
1
0.95
0.9
0.85
0.8
0.75
1.
2.
3.
Eigenvalue

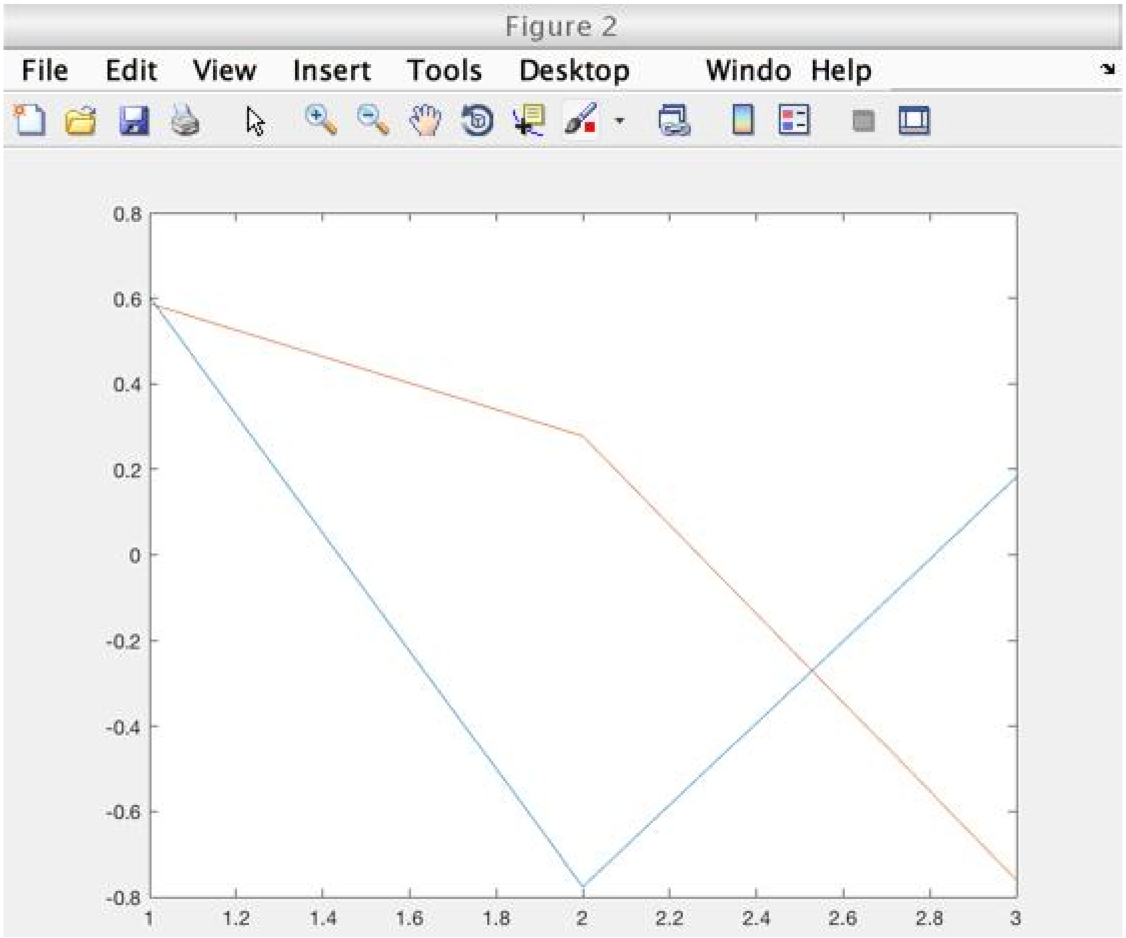


Figure 2
File
Edit
View
Insert
Tools
Desktop
Windo
Help
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3

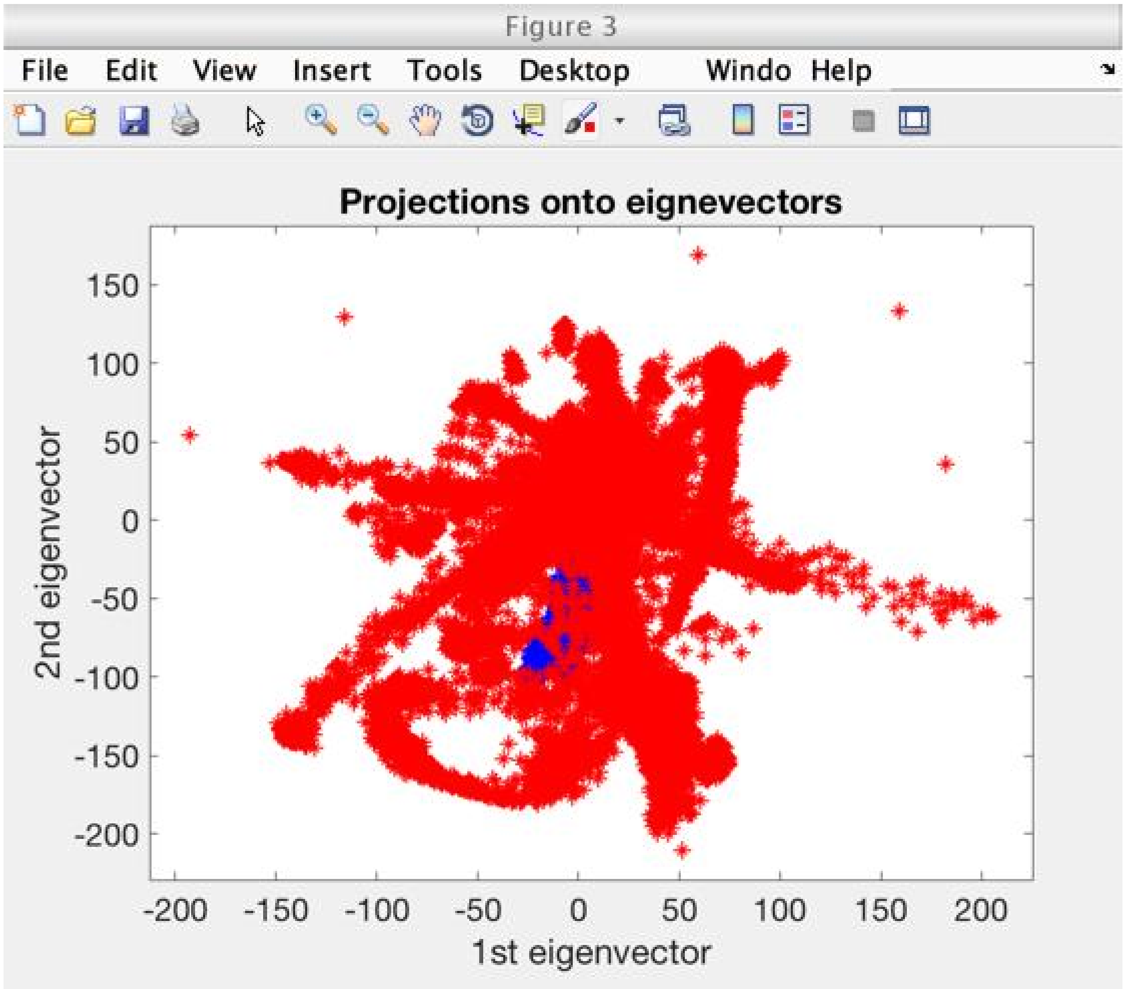
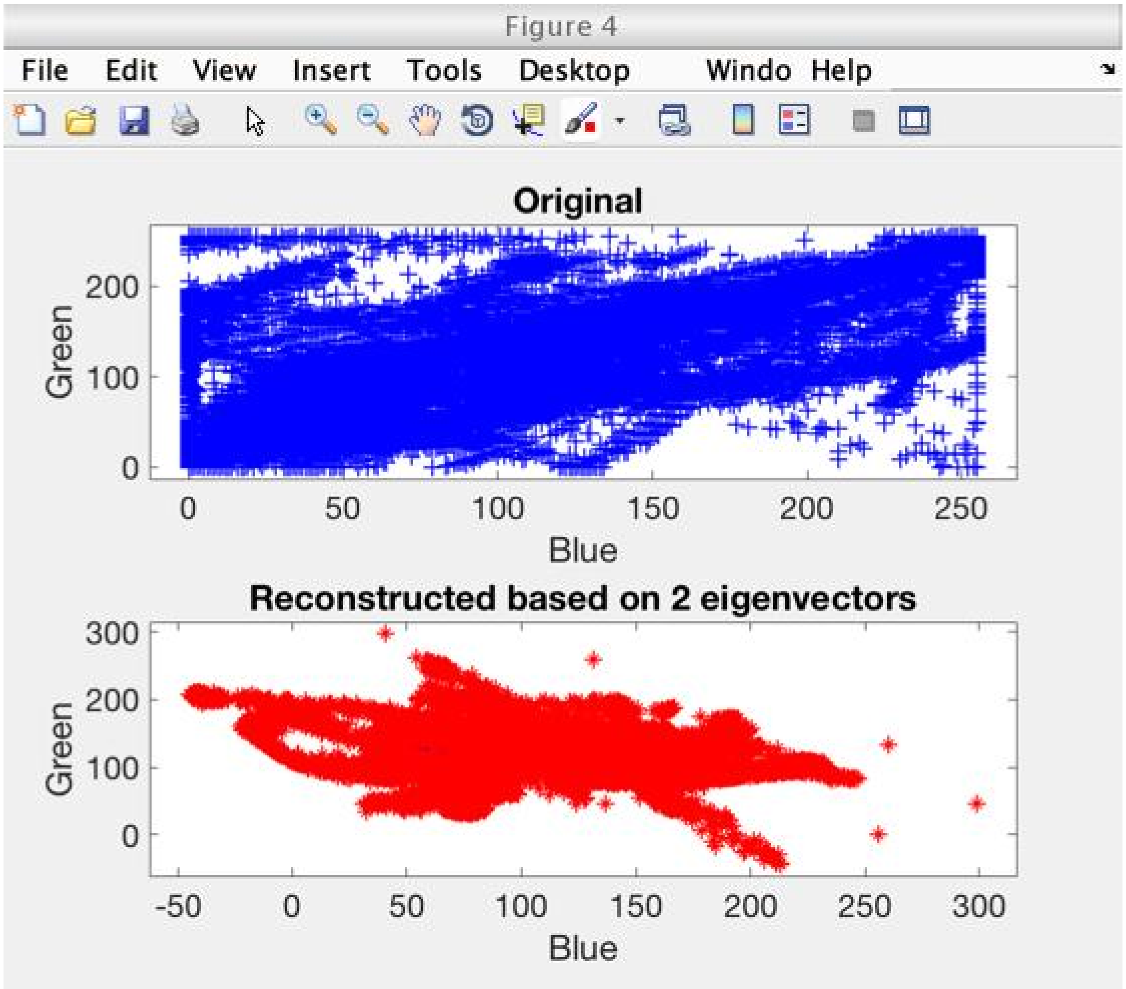


Figure 3
File
Edit
View
Insert
Tools
Desktop
Windo
Help
Projections onto eignevectors
150
100
50
0
-50
-100
-150
-200
2nd eigenvector
-200
-150
-100
-50
0
50
100
150
200
1st eigenvector

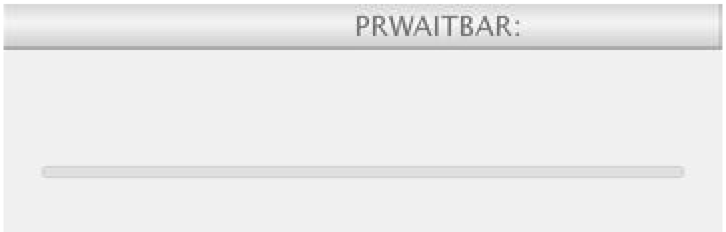


Classification on original data (parametric)

```
clear all

%reloading, naming
filehere= 'Users/Anna/Documents/MATLAB/Skin_NonSkin.txt';
fid=fopen(filehere);
mydata = textscan(fid,'%f%f%f%f','delimiter',',');
fclose(fid);

Blue = mydata{1};
Green = mydata{2};
Red = mydata{3};
Skin = mydata{4};

X=[Blue Green Red];
labels=Skin;

labels (labels==1)=0;
labels(labels==2)=1;
[I,J]=size(X);

addpath(['/Users/Anna/Documents/MATLAB/prtools']);

data_all=prdataset(X,labels);
idx_skin=find(labels==0);
idx_noskin=find (labels==1);
```

```
%randomizing order
id_X = randperm(I)';

%Dividing my dataset into training and testing
tr_idx = id_X(1:220551); % training indices 90% of dataset
tst_idx = id_X(220552:245057); %testing indices 10% of dataset


skin_tr = prdataset ( X ( tr_idx ,:),labels ( tr_idx ) );
skin_tst = prdataset ( X (tst_idx,:),labels ( tst_idx ) );

%---------------------------------------------------------------%
%parametric classifiers

confmat(labels,1*ones(I,1));
c_base=confmat(labels,1*ones(I,1));

tp=c_base(1,1); %true positives
fn=0; %always 0
fp=c_base(2,1); %false positives
tn=0; %always 0
acc_base=(tp+tn)/(tp+fp+fn+tn) %also percentage of one class

cv=10; %10 fold cross validation

for classi=1:3
    switch classi,
        case 1,
            %minimum distance classifier
            w=nmsc([]);
        case 2,
            %linear discriminant analysis
            w=ldc([]);
        case 3,
            %quadratic discriminant analysis
            w=qdc([]);
    end

[e dummy pred_labs]=prcrossval(skin_tr,w,cv);
%show confusion matrix

confmat(labels(tr_idx),pred_labs);
c=confmat(labels(tr_idx),pred_labs);
tp=c(1,1); %t=true, p=positives, f=false, n=negatives
fn=c(1,2);
fp=c(2,1);
tn=c(2,2);

acc1(classi)=(tp+tn)/(tp+fp+fn+tn);
p=tp/(tp+fp); %precision
r=tp/(tp+fn); %recall

%f-measure for all 3 classifiers:
f1(classi)=(2*p*r)/(p+r);

end
f1
acc1

%--------------------------------------------------------%
```

```
  True   | Estimated Labels
  Labels |    1   | Totals
 --------|-------|-------
  0      |50859  |50859
  1      |194198  |194198
 --------|-------|-------
  Totals |245057  |245057


acc_base =

    0.2075

PR_Warning: GETPRIOR: prcrossval: No priors found in dataset, class frequencies are used instead

  True   | Estimated Labels
  Labels |    1      2  | Totals
 --------|--------------|-------
  0      |29611  16061  |45672
  1      |11748  163131  |174879
 --------|--------------|-------
  Totals |41359  179192  |220551

PR_Warning: GETPRIOR: prcrossval: No priors found in dataset, class frequencies are used instead

  True   | Estimated Labels
  Labels |    1      2  | Totals
 --------|--------------|-------
  0      |41312   4360  |45672
  1      |10772  164107  |174879
 --------|--------------|-------
  Totals |52084  168467  |220551

PR_Warning: GETPRIOR: prcrossval: No priors found in dataset, class frequencies are used instead

  True   | Estimated Labels
  Labels |    1      2  | Totals
 --------|--------------|-------
  0      |42278   3394  |45672
  1      |  254  174625  |174879
 --------|--------------|-------
  Totals |42532  178019  |220551


f1 =

    0.6805    0.8452    0.9586


acc1 =

    0.8739    0.9314    0.9835
```

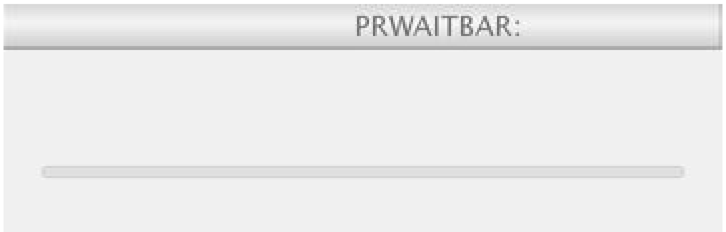


non-parametric knn classifier

```
def_mem=prmemory();
for classi2=1:3,
    switch classi2,
        case 1,
            prmemory(20*def_mem)
            w2=knnc([],1);
        case 2,
            w2=knnc([],2);
        case 3,
            w2=knnc([],3);
    end

 [e dummy pred_labs]=prcrossval(skin_tst,w2,cv);
%show confusion matrix

confmat(labels(tst_idx),pred_labs);
c=confmat(labels(tst_idx),pred_labs);
tp=c(1,1); %t=true, p=positives, f=false, n=negatives
fn=c(1,2);
fp=c(2,1);
tn=c(2,2);

acc1(classi2)=(tp+tn)/(tp+fp+fn+tn);
p=tp/(tp+fp); %precision
r=tp/(tp+fn); %recall

%f-measure for all 3 classifiers:
f1(classi2)=(2*p*r)/(p+r);
end
f1
acc1


%calculating errors
errors=pred_labs~=labels(tst_idx);
sum(errors)
```

```
PR_Warning: GETPRIOR: prcrossval: No priors found in dataset, class frequencies are used instead

  True    | Estimated Labels
  Labels |    1      2   | Totals
 --------|--------------|-------
  0       | 5183       4  | 5187
  1       |   14  19305  |19319
 --------|--------------|-------
  Totals | 5197  19309  |24506

PR_Warning: GETPRIOR: prcrossval: No priors found in dataset, class frequencies are used instead
```

```
  True   | Estimated Labels
  Labels |    1      2   | Totals
 --------|--------------|-------
  0      | 5187      0  | 5187
  1      |   32  19287  |19319
 --------|--------------|-------
  Totals | 5219  19287  |24506

PR_Warning: GETPRIOR: prcrossval: No priors found in dataset, class frequencies are used instead

  True   | Estimated Labels
  Labels |    1      2   | Totals
 --------|--------------|-------
  0      | 5187      0  | 5187
  1      |   24  19295  |19319
 --------|--------------|-------
  Totals | 5211  19295  |24506


f1 =

    0.9983    0.9969    0.9977


acc1 =

    0.9993    0.9987    0.9990


ans =

       24482
```



Published with MATLAB® R2017b