ECE 763 Computer Vision Project1

Sreeraj Rajendran ID: 200210462 UID: srajend2 email: srajend2@ncsu.edu

Objective:

Face image classification using Gaussian model, Mixture of Gaussian model, t-distribution, Mixture of t-distribution, Factor Analysis and Mixture of Factor Analyzer

Data Preparation:

AWFL dataset , Grayscale, Resized dimension: (20,20)

AWFL dataset is used to train the face. Grayscale is used as it obtained smaller array size and saved execution time. Initial testing was performed using a size of 60x60. But the same took extremely long periods to execute and python overflow errors were encountered. Hence grayscale images of dimension 20,20 is used for testing. User can change the size as required in the program call.

Method:

1. Learn and visualize parameters:

Two world states of face and non face are considered. Initially, parameters required by desired models are obtained from training images of known world state. These parameters are used to find the likelihood of test data Pr (x|w). The estimated mean and covariance matrices for face and non-face are displayed.

2. Test and evaluate learned model with threshold=0.5 for posterior

The likelihood equations and corresponding parameters are implemented based for each model. These likelihoods are used to find world state(face or non-face) from the data(test image) provided using the below equation.

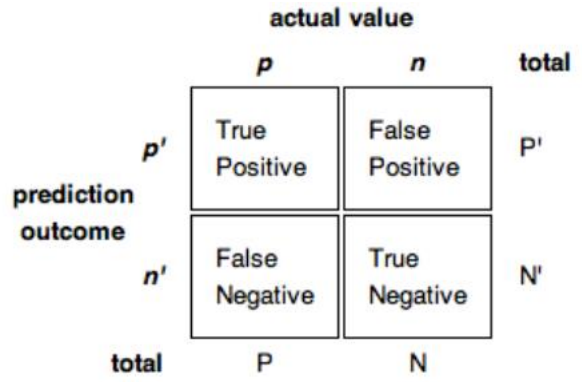
$$Pr(w=1|\mathbf{x}) = \frac{Pr(\mathbf{x}|w=1)Pr(w=1)}{\sum_{k=0}^{1} Pr(\mathbf{x}|w=k)Pr(w=k)}.$$

Pr(w=face) and Pr(w=nonface) are considered as 0.5 and hence the same reduces to:

Pr(face|x) = Pr(x|face) / (Pr(x|face) + Pr(x|non-face))

The tested model is evaluated by comparing the predictions with actual state. Following classifications are made:

a. True Positives: Predicted ‘Face’ when actual data is Face.
b. False Positives: Predicted ‘Face’ when actual data is not Face
c. False Negatives: Predicted ‘Non-Face’ when actual data is Face
d. True Negatives: Predicted ‘Non-Face’ when actual data is Not Face



This data is used to compute the following:

- false positive rate: negatives being classified as faces / total negatives i.e. FP/(FP+TN)
- false negative rate: positives being classified as non-face / total positives i.e. FN/(FN+TP)
- misclassification rate: (false positives + false-negative)/total testing images

3. Report and depict result analysis - Plot ROC curve:

False positive rate vs True Positive rate is plotted . The corresponding data is obtained by varying the threshold for posterior. When threshold is very high, most faces might be classified as non face as the test image must match the parameters of trained image to a very high degree. As the threshold is decreased, the number of detections will increase.

Models:

Model 1: Gaussian model

The data is considered as normal distribution.

Parameters: Mean and Covariance

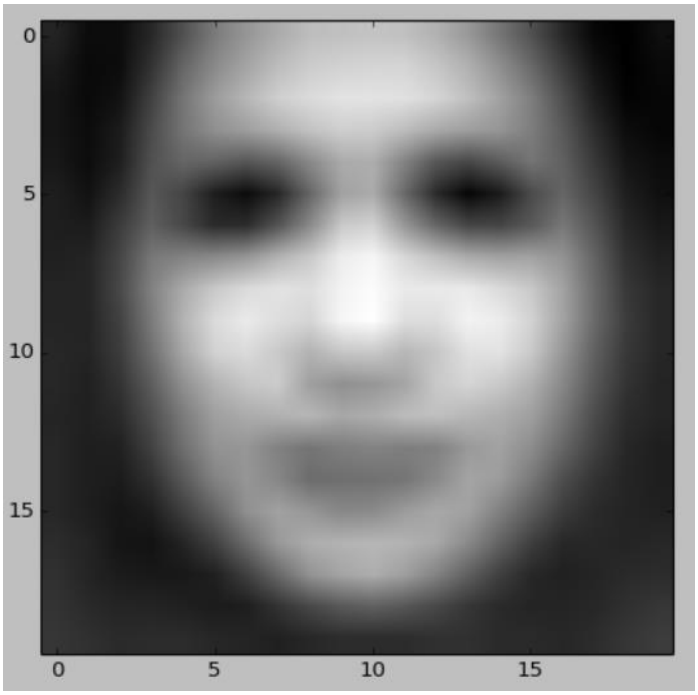
The likelihood is obtained using following equation:

$$Pr(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left[-0.5(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]$$

$$\begin{aligned} Pr(\mathbf{x}|w=0) &= \text{Norm}_{\mathbf{x}}[\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0] \\ Pr(\mathbf{x}|w=1) &= \text{Norm}_{\mathbf{x}}[\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1]. \end{aligned}$$

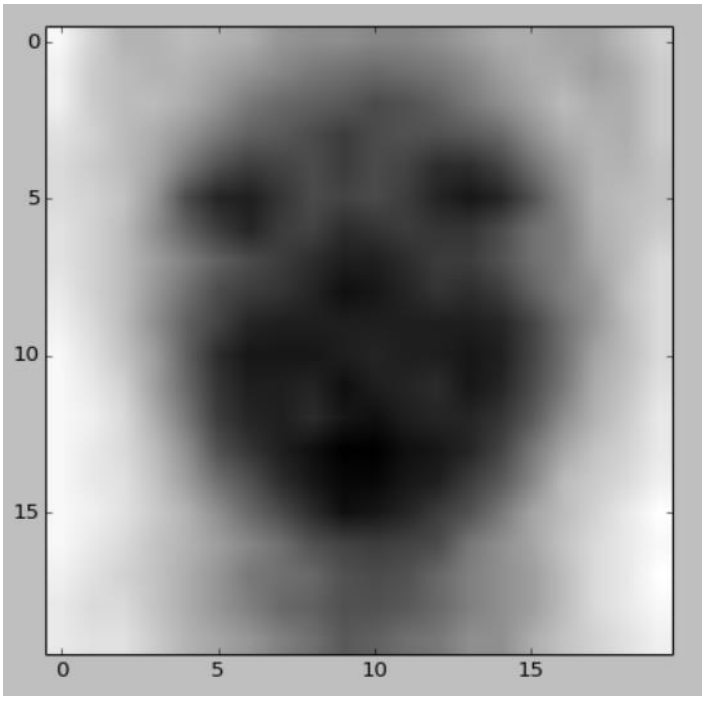
[Mu0, sigma0]= non -face and [mu1,sima1] for face

Mean Face for Gaussian model:

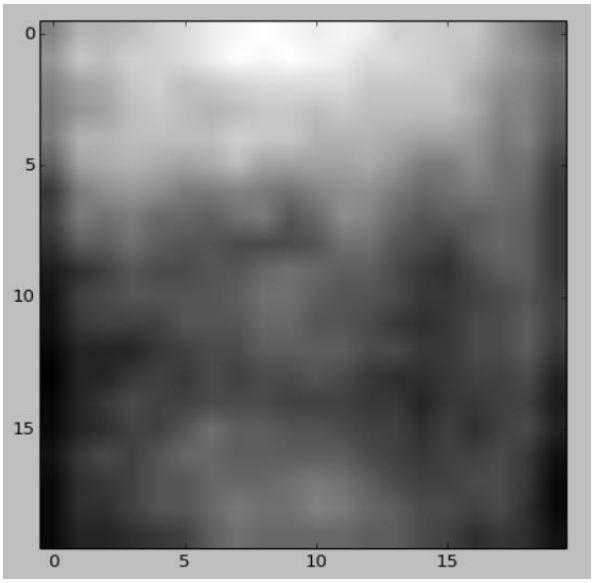


Covariance for Gaussian model:

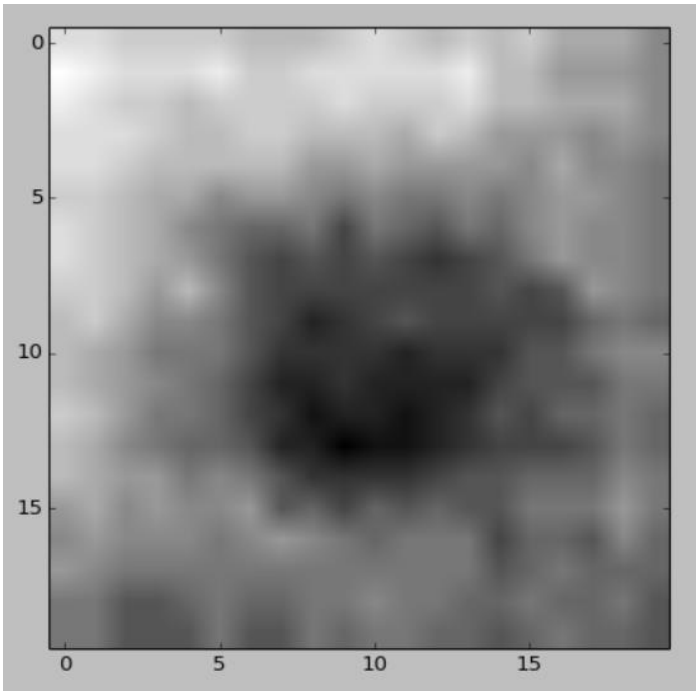
The square root of diagonal elements are considered for displaying covariance



Mean Background (Gaussian):



Covariance of Background Data (Gaussian):



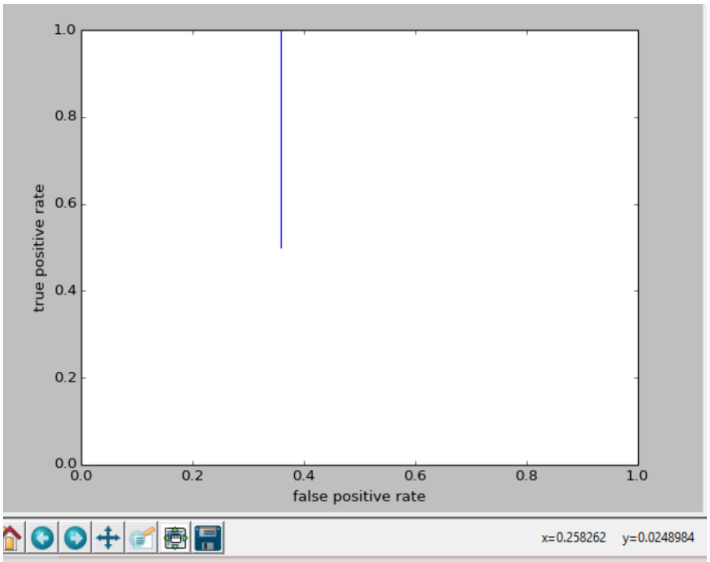
False Positive Rate: 0.36

False Negative Rate: 0.50

Misclassification rate: 0.43

```
Enter model number:
 1 for Gaussian
 2 for Mixture of Gaussian
 3 for t distribution
 4 for mixture of t
 5 for factor analyzer
 6 for mixture of t factor analyzer1
('face mean shape:', (400,))
('covariance shape:', (400, 400))
('Non-face mean shape:', (400,))
('covariance shape:', (400, 400))
C:\Python27\lib\site-packages\ipykernel_launcher.py:12: RuntimeWarning: overflow encountered in exp
  if sys.path[0] == '':
C:\Python27\lib\site-packages\ipykernel_launcher.py:9: RuntimeWarning: overflow encountered in exp
  if __name__ == '__main__':
C:\Python27\lib\site-packages\ipykernel_launcher.py:16: RuntimeWarning: invalid value encountered in double_scalars
  app.launch_new_instance()
0
('number of faces detected in face(gaussian):', 50)
('number of non-faces detected in face data(gaussian):', 50)
('number of faces detected in non-face data(gaussian):', 36)
('number of non-faces detected in non-face data(gaussian):', 64)
('false_positive_rate', 0.36)
('false negative rate:', 0.5)
('missclassification rate', 0.43)
```

ROC Plot:

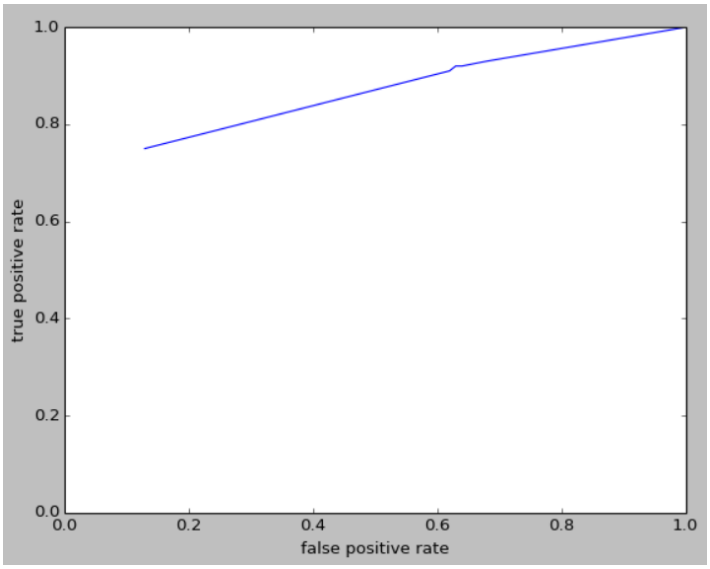



```
[ 0.36  0.36  0.36  0.36  0.36  0.36  0.36  0.36  0.36  0.36  0.38]
[ 0.5  1.   1.   1.   1.   1.   1.   1.   1.   1.   1. ]
```

```
[ 0.36  0.36  0.36  0.36  0.36  0.36  0.36  0.36  0.36  0.36  0.38]
[ 0.5  1.   1.   1.   1.   1.   1.   1.   1.   1.   1. ]
('number of faces detected in face(gaussian):', 100)
('number of non-faces detected in face data(gaussian):', 0)
('number of faces detected in non-face data(gaussian):', 38)
('number of non-faces detected in non-face data(gaussian):', 62)
('false_positive_rate', 0.38)
('false negative rate:', 0.0)
('missclassification rate', 0.19)
```

Alternate result on multiplying sigma with high value:

```
Gaussian in progress...
('face mean shape:', (400,))
('covariance shape:', (400, 400))
('Non-face mean shape:', (400,))
('covariance shape:', (400, 400))
('number of faces detected in face(gaussian):', 92)
('number of non-faces detected in face data(gaussian):', 8)
('number of faces detected in non-face data(gaussian):', 63)
('number of non-faces detected in non-face data(gaussian):', 37)
('false_positive_rate', 0.63)
('false negative rate:', 0.08)
('missclassification rate', 0.355)
[ 1.    0.68  0.64  0.64  0.63  0.63  0.63  0.63  0.63  0.62  0.13]
[ 1.    0.93  0.92  0.92  0.92  0.92  0.92  0.92  0.92  0.91  0.75]
```

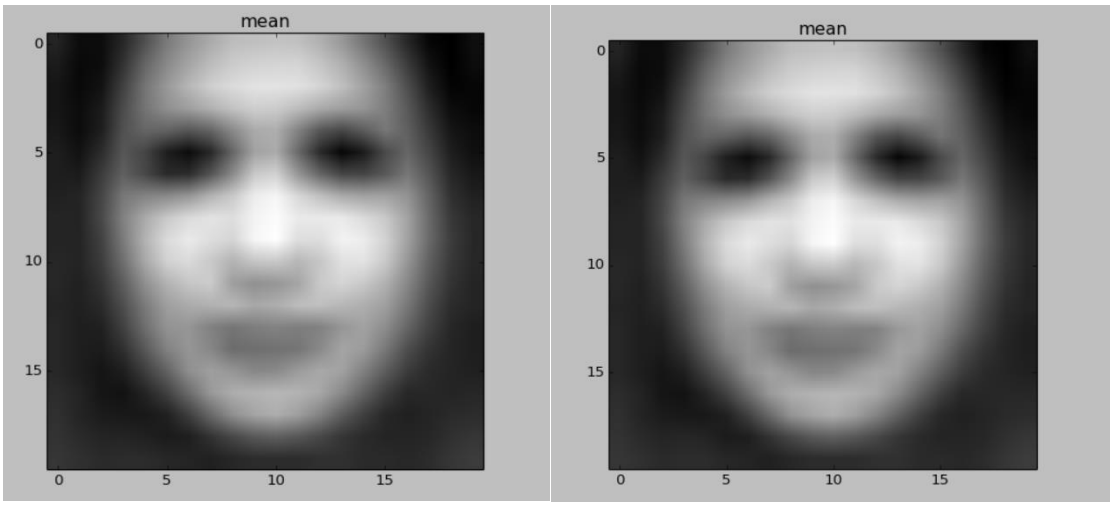


Model 2: Mixture of Gaussian

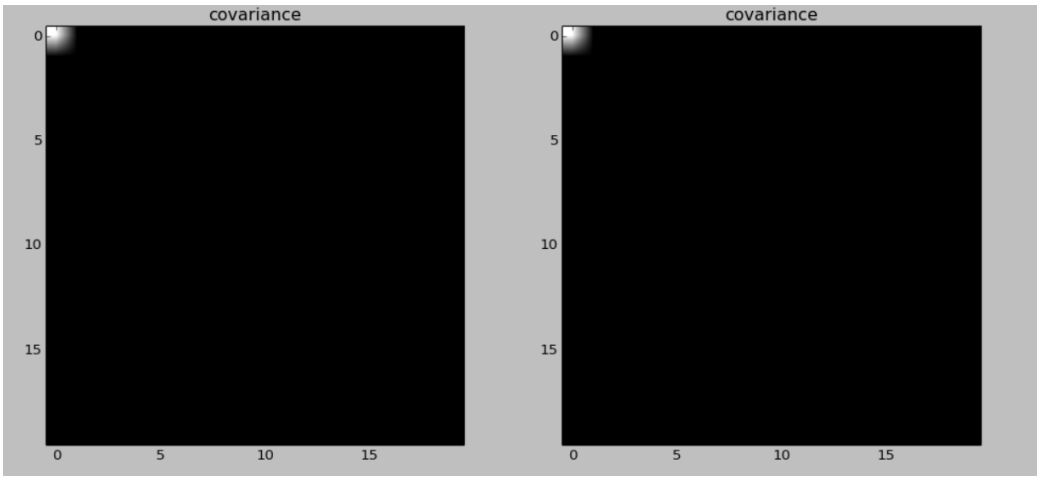
$$Pr(\mathbf{x}|\boldsymbol{\theta}) = \sum_{k=1}^{K} \lambda_k \mathrm{Norm}_{\mathbf{x}}[\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k]$$

For number of mixtures, K=2

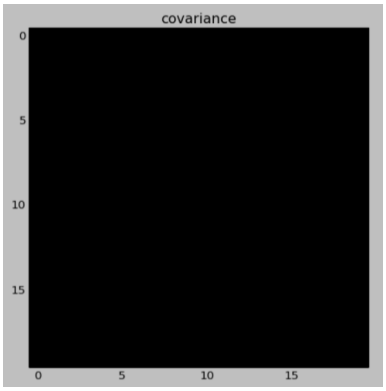
Mean Face:

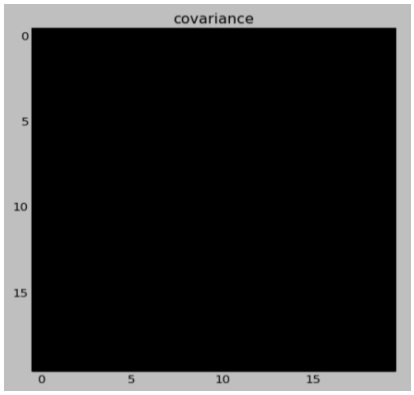


Covariance (without taking square root of diagonal elements)

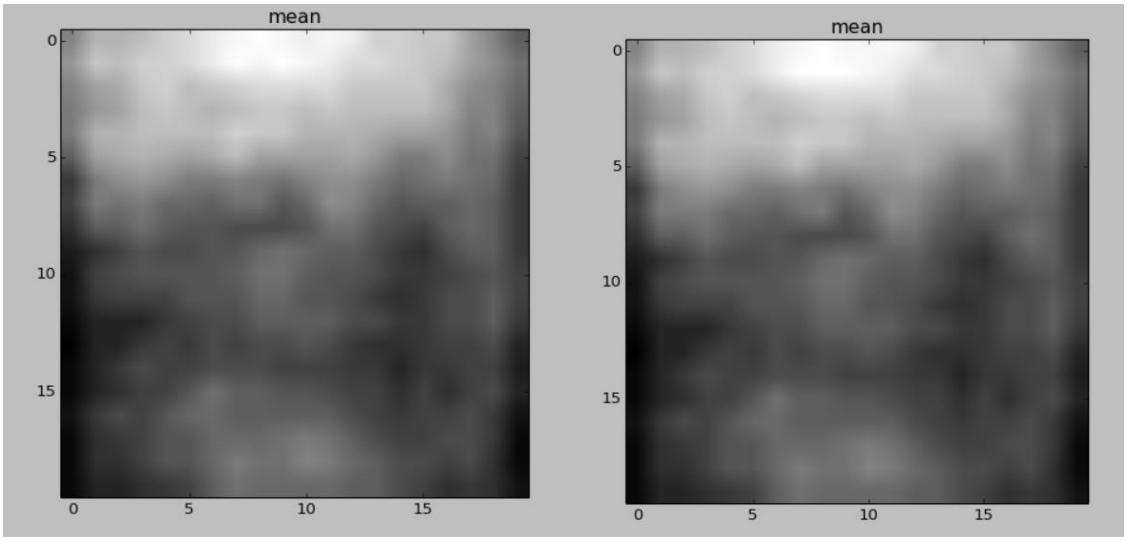


Covariance when Square root (diag(covariance)) is used

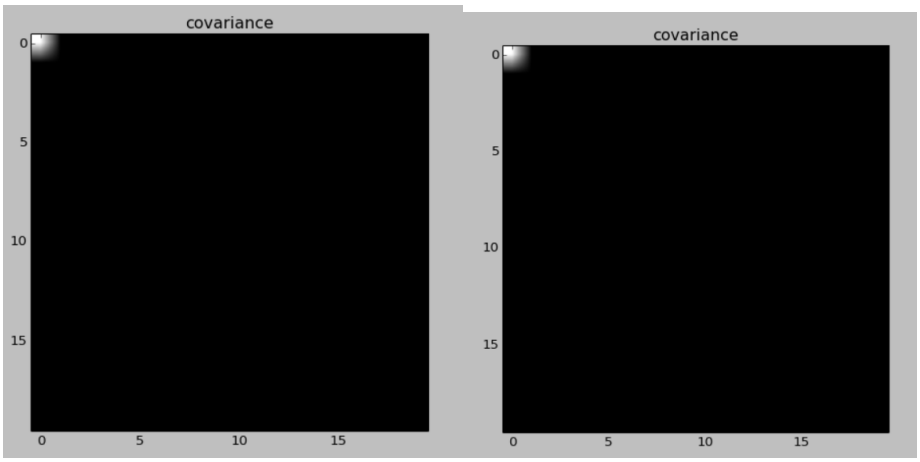




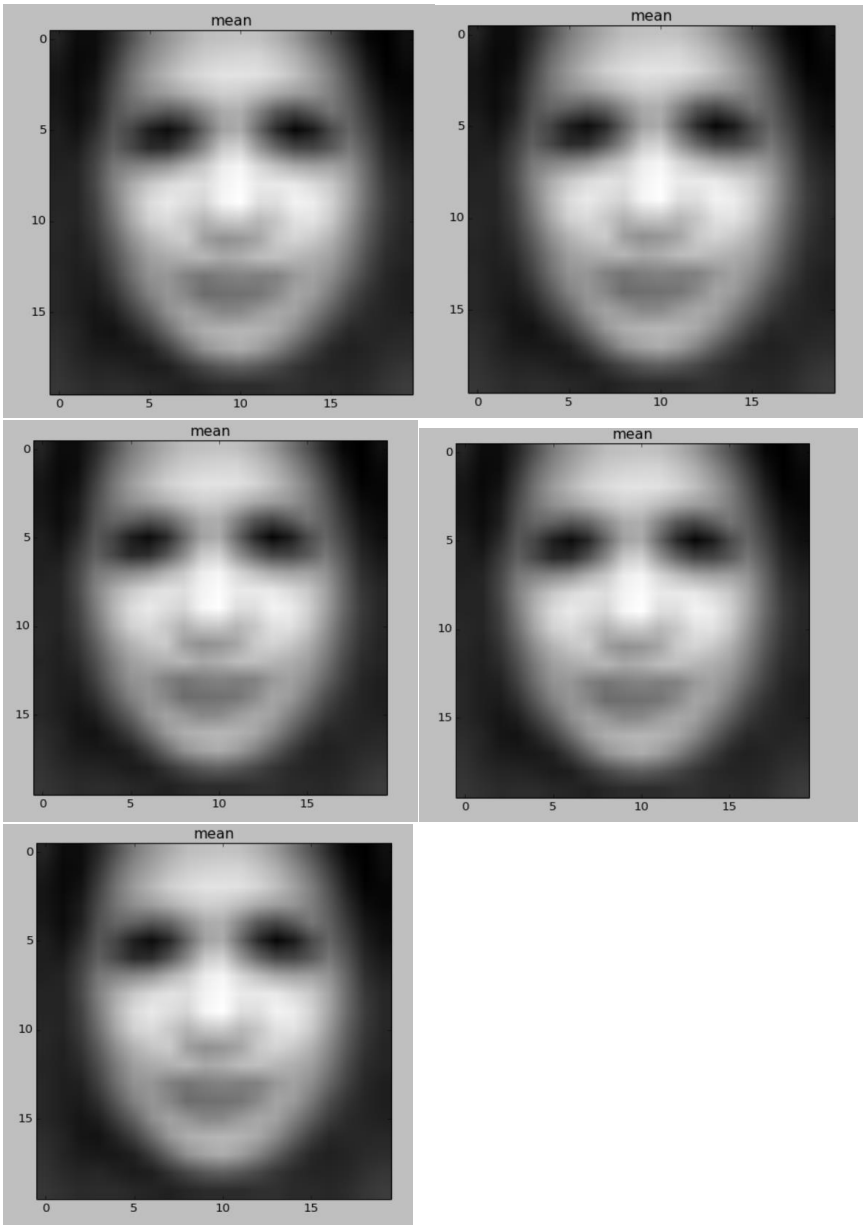
Mean Non-Face:



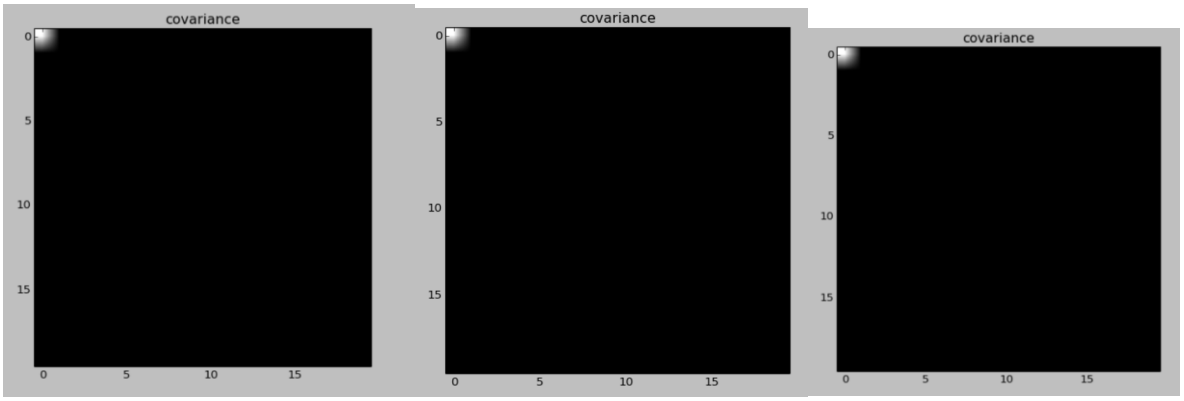
Covariance Non-Face:

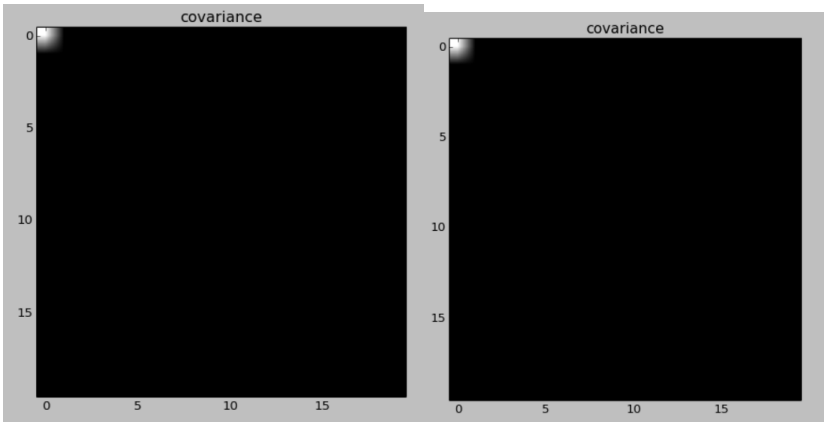


MoG : k=5



Covariance for K=5



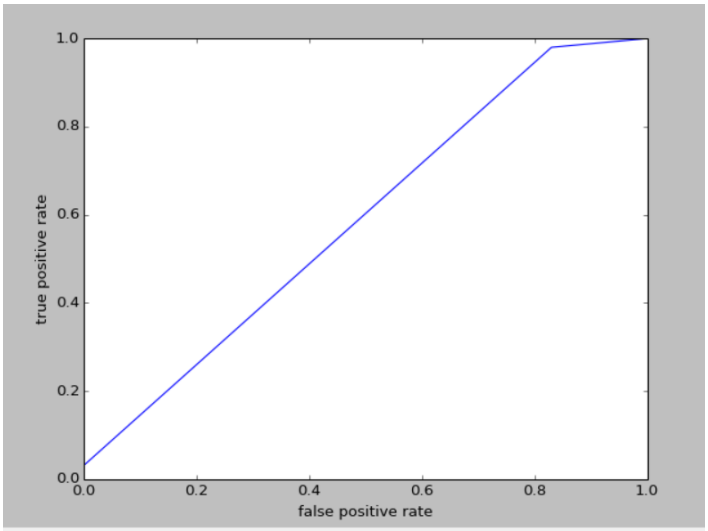


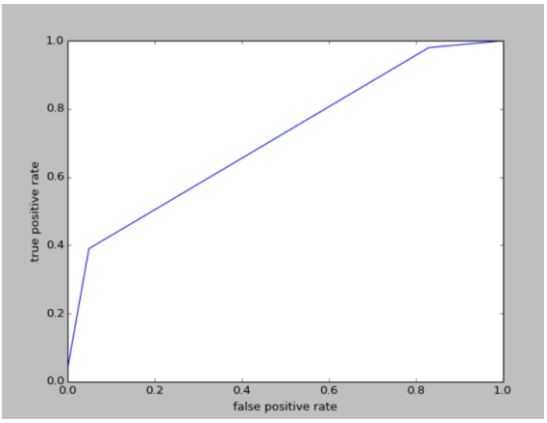

```
('lamda', array([[ 0.19999512],
       [ 0.2000106 ],
       [ 0.19998787],
       [ 0.19999602],
       [ 0.20001039]]))
('mog_mean', array([[102,  94,  94, ..., 105, 108, 110],
       [102,  94,  94, ..., 105, 108, 110],
       [102,  94,  94, ..., 105, 108, 110],
       [102,  94,  94, ..., 105, 108, 110],
       [102,  94,  94, ..., 105, 108, 110]]))
('mog_covariance', 5)
```

Results:

By using different initializations and by considering either diagonal elements of covariance matrix or complete matrix itself in calculation of pdf, different results were obtained:

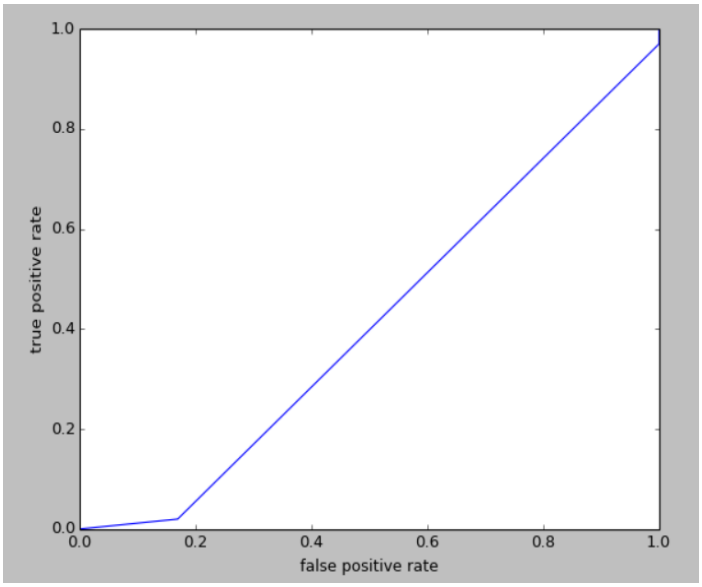
```
2
number of mixtures: 2
('K:', 2)
('false positive rate', array([ 1.  ,  1.  ,  1.  ,  1.  ,  1.  ,  0.83,  0.  ,  0.  ,  0.  ,
        0.  ,  0.  ]))
('true positive rate', array([ 1.  ,  1.  ,  1.  ,  1.  ,  1.  ,  0.98,  0.03,  0.  ,  0.  ,
        0.  ,  0.  ]))
('number of faces detected in face(mog):', 0)
('number of non-faces detected in face data(mog):', 100)
('number of faces detected in non-face data (mog):', 0)
('number of non-faces detected in non-face data(mog):', 100)
('false_positive_rate', 0.0)
('false negative rate:', 1.0)
('missclassification rate', 0.5)
```






```
[ 1.   1.   1.   1.   1.   1.   1.   1.   1.   1.   0.83 0.05
  0.   0.   0.   0.   0.   0.   0.   0.  ]
[ 1.   1.   1.   1.   1.   1.   1.   1.   1.   1.   0.98 0.39
  0.03 0.   0.   0.   0.   0.   0.   0.  ]
```

```
number of mixtures: 2
('K:', 2)
[ 0.    0.    0.    0.    0.    0.17  1.    1.    1.    1.    1.  ]
[ 0.    0.    0.    0.    0.    0.02  0.97  1.    1.    1.    1.  ]
('number of faces detected in face(mog):', 100)
('number of non-faces detected in face data(mog):', 0)
('number of faces detected in non-face data (mog):', 100)
('number of non-faces detected in non-face data(mog):', 0)
('false_positive_rate', 1.0)
('false negative rate:', 0.0)
('missclassification rate', 0.5)
```



Model 3 : T-distribution

Parameters: mean (mu), covariance(sigma) and degrees of freedom (nu)

EM algorithm is used and parameters are updated using following equations:

E[hi] is the expectation

$$\mathrm{E}[h_i] = \frac{(\nu + D)}{\nu + (\mathbf{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_i - \boldsymbol{\mu})}$$

$$\mathrm{E}[\log[h_i]] = \Psi\left[\frac{\nu + D}{2}\right] - \log\left[\frac{\nu + (\mathbf{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_i - \boldsymbol{\mu})}{2}\right]$$

$$\boldsymbol{\mu}^{[t+1]} = \frac{\sum_{i=1}^{I} \mathrm{E}[h_i] \mathbf{x}_i}{\sum_{i=1}^{I} \mathrm{E}[h_i]}$$

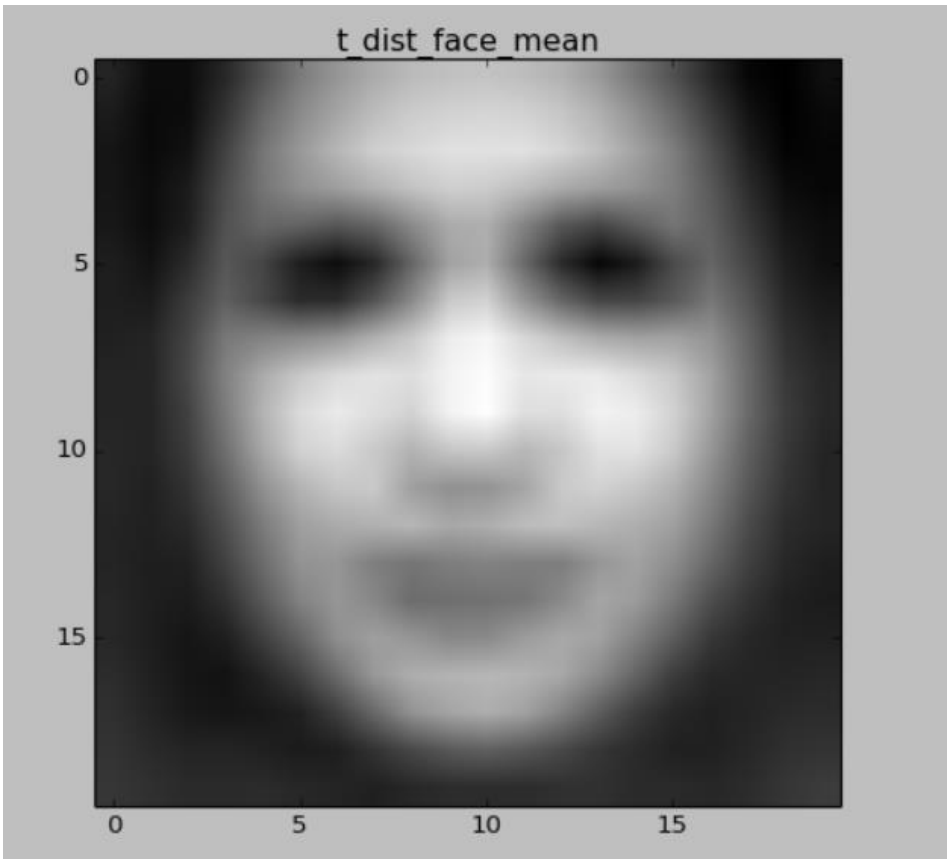
$$\boldsymbol{\Sigma}^{[t+1]} = \frac{\sum_{i=1}^{I} \mathrm{E}[h_i] (\mathbf{x}_i - \boldsymbol{\mu}^{[t+1]})(\mathbf{x}_i - \boldsymbol{\mu}^{[t+1]})^T}{\sum_{i=1}^{I} \mathrm{E}[h_i]}.$$

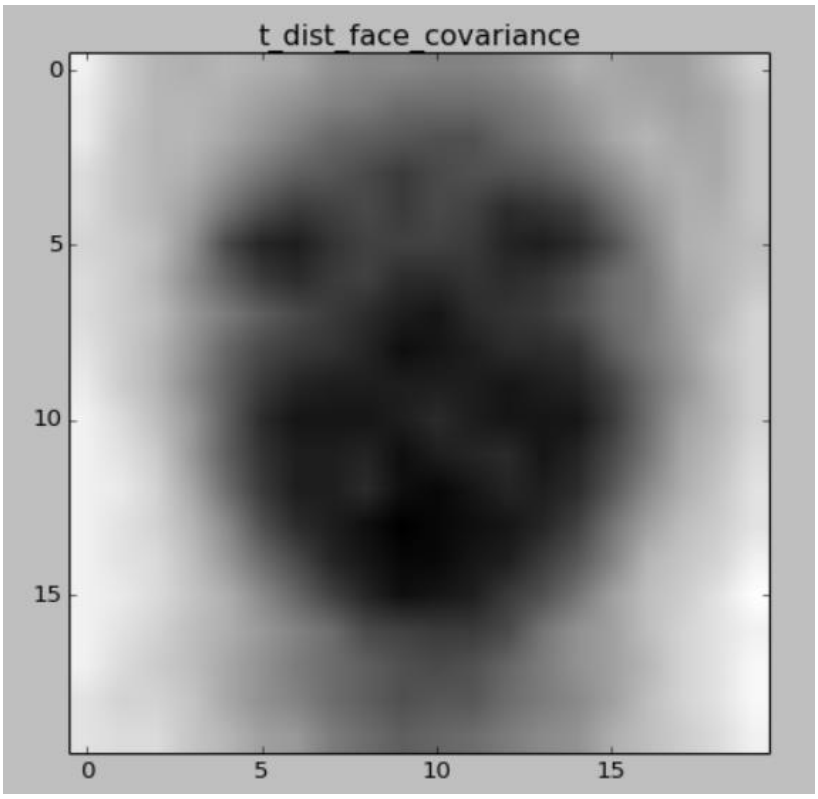
Nu is updated such that cost is minimized:

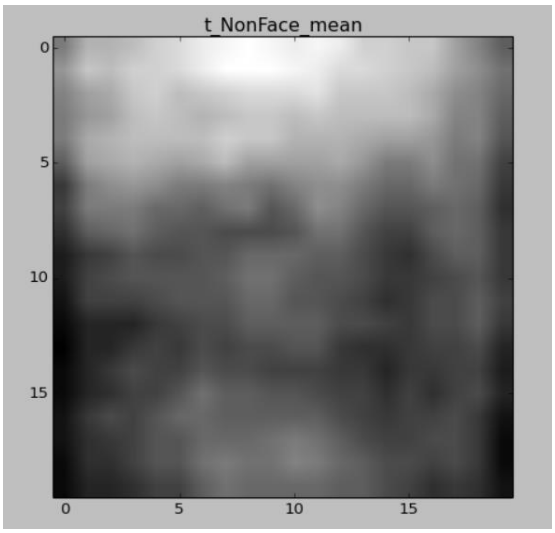
$$\mathrm{tCost}\left[\nu, \{E[h_i], E[\log[h_i]]\}_{i=1}^{I}\right] = -\sum_{i=1}^{I} \frac{\nu}{2} \log\left[\frac{\nu}{2}\right] + \log\left[\Gamma\left[\frac{\nu}{2}\right]\right] - \left(\frac{\nu}{2} - 1\right) E[\log[h_i]] + \frac{\nu}{2} E[h_i].$$

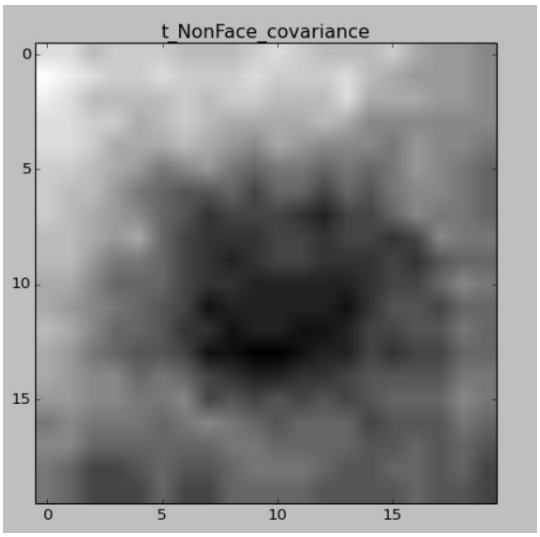
Likelihood is obtained using the pdf equation:

$$Pr(\mathbf{x}) = \frac{\Gamma\left(\frac{\nu+D}{2}\right)}{(\nu\pi)^{D/2} |\boldsymbol{\Sigma}|^{1/2} \Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})}{\nu}\right)^{-(\nu+D)/2}$$


t_dist_face_mean
0
5
10
15
0
5
10
15


t_dist_face_covariance
0
5
10
15
0
5
10
15

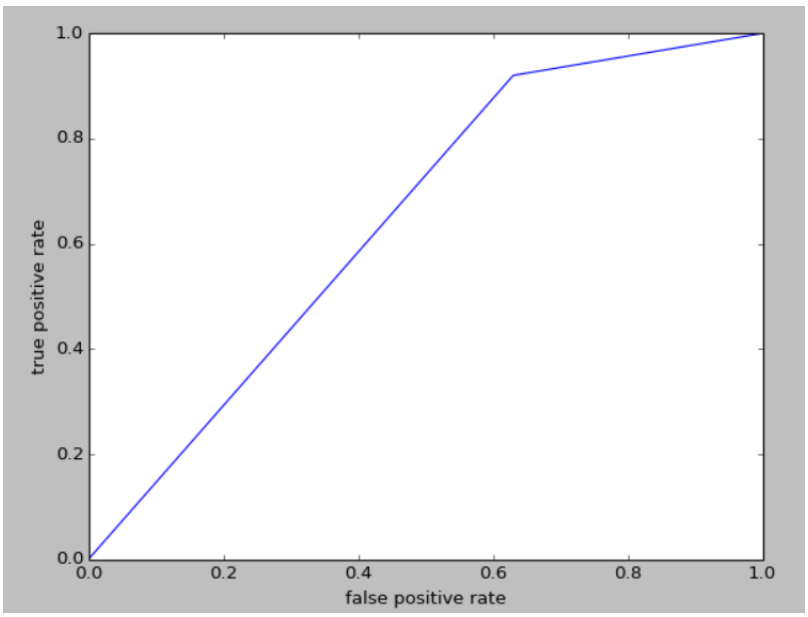




Result:

```
('df', 4)
('number of faces detected in face:', 92)
('number of non-faces detected in face data:', 8)
('number of faces detected in non-face data :', 63)
('number of non-faces detected in non-face data:', 37)
('false_positive_rate', 0.63)
('false negative rate :', 0.08)
('missclassification rate ', 0.355)
('false positive rate', array([ 1.  ,  1.  ,  1.  ,  1.  ,  1.  ,  0.63,  0.  ,  0.  ,  0.  ,
        0.  ,  0.  ]))
('true positive rate', array([ 1.  ,  1.  ,  1.  ,  1.  ,  1.  ,  0.92,  0.  ,  0.  ,  0.  ,
        0.  ,  0.  ]))
```

EOC Curve:

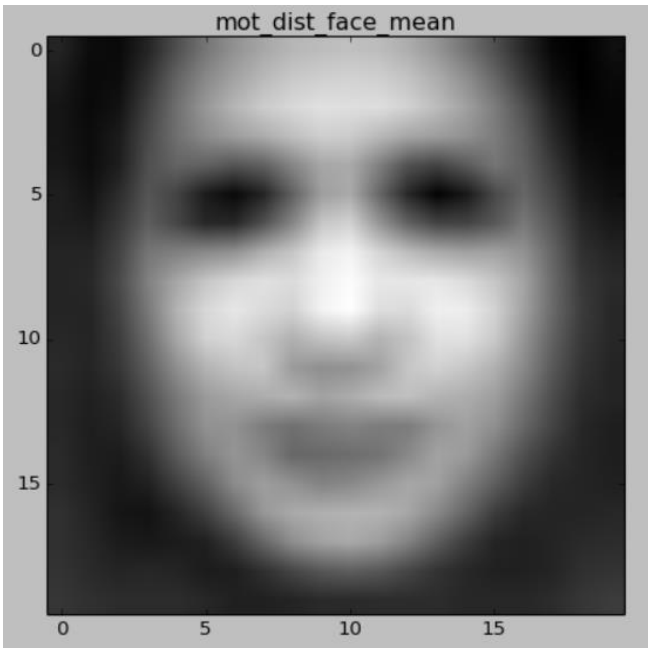


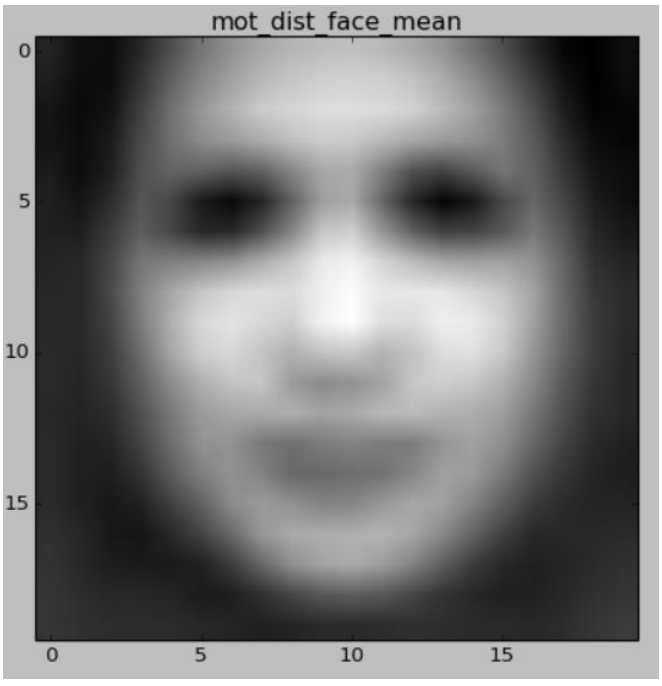
Model 4: Mixture of T distribution

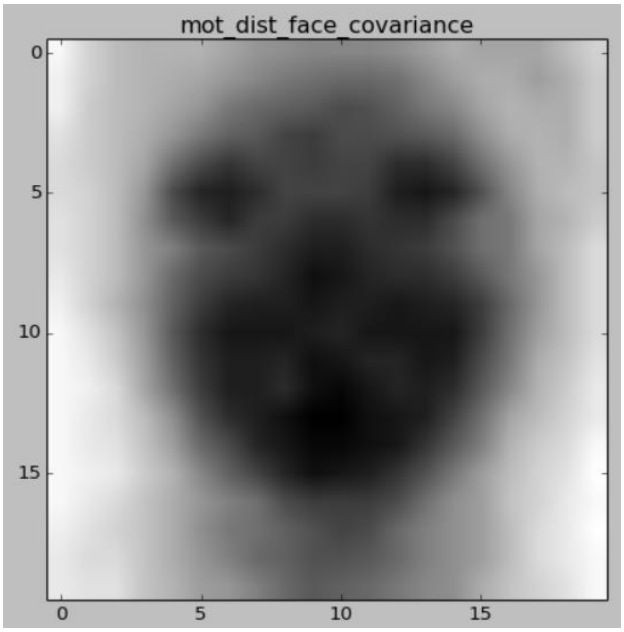
$$f(\boldsymbol{y};\ \boldsymbol{\Psi}) = \sum_{i=1}^{g} \pi_i\, f(\boldsymbol{y};\ \boldsymbol{\mu}_i,\ \boldsymbol{\Sigma}_i,\ \nu_i)$$

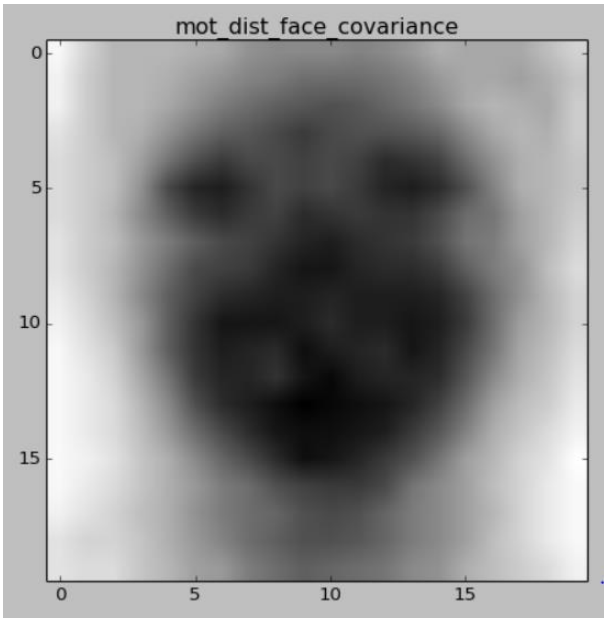
Pdf is calculated using the equation :

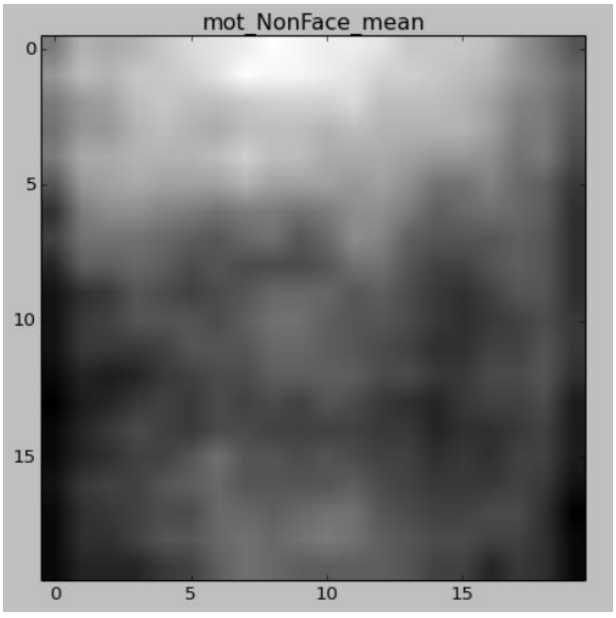
Where g is the number of mixtures.

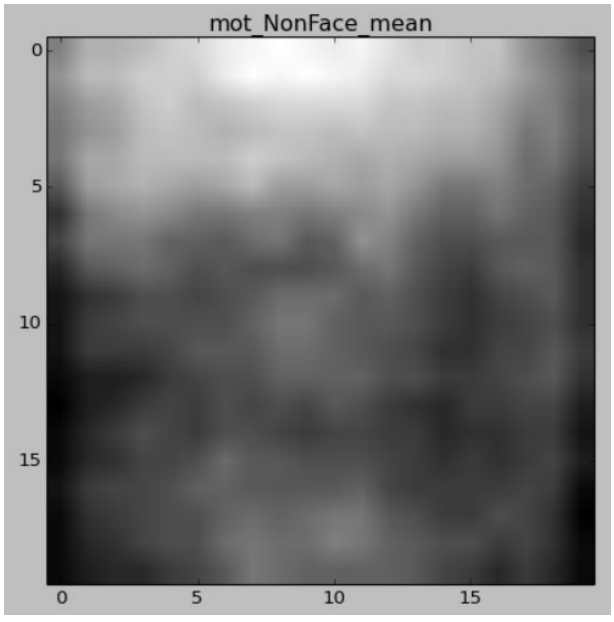


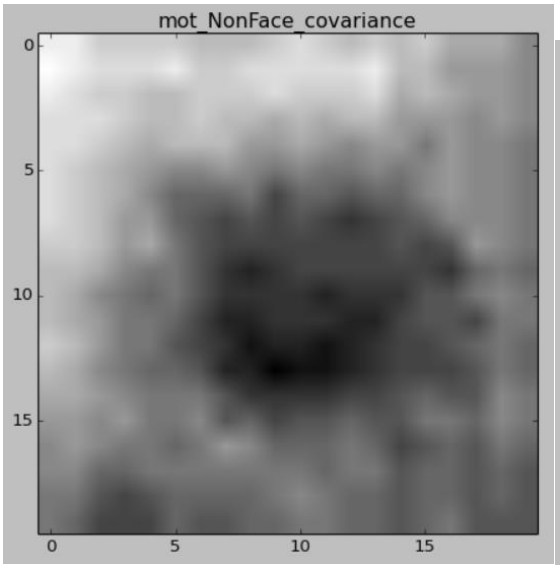


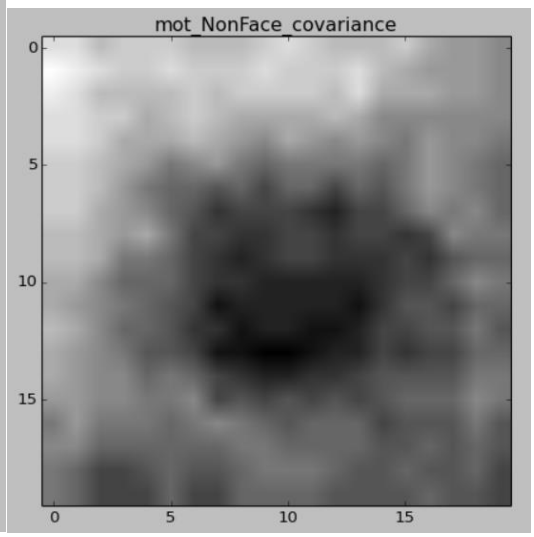

mot_dist_face_covariance
0
5
10
15
0
5
10
15


mot_dist_face_covariance
0
5
10
15
0
5
10
15


mot_NonFace_mean
0
5
10
15
0
5
10
15


mot_NonFace_mean
0
5
10
15
0
5
10
15

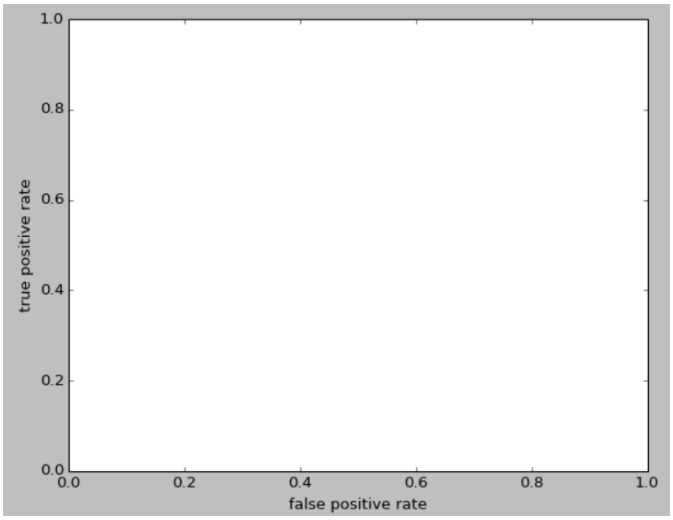




Result:

```
('number of faces detected in face:', 100)
('number of non-faces detected in face data:', 0)
('number of faces detected in non-face data :', 100)
('number of non-faces detected in non-face data:', 0)
('false_positive_rate', 1.0)
('false negative rate :', 0.0)
('missclassification rate ', 0.5)
('false positive rate', array([ 1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.]))
('true positive rate', array([ 1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.]))
```

Unsatisfactory



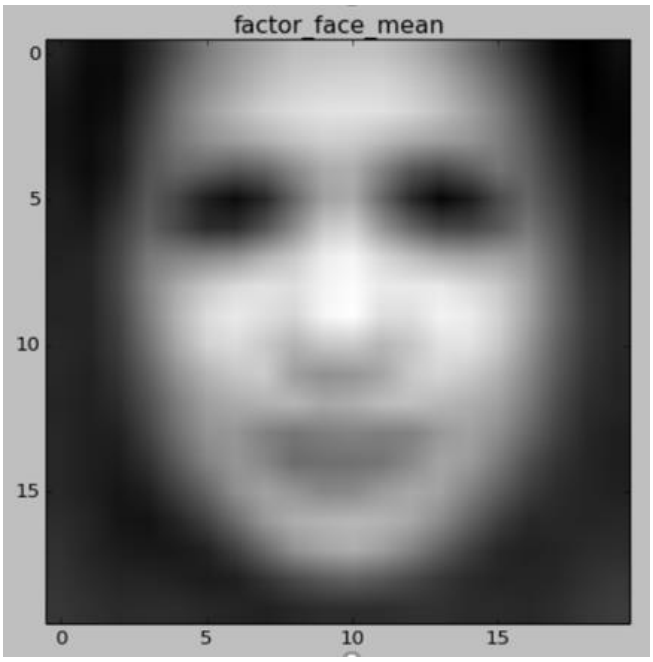
Model 5 : Factor Analyzer

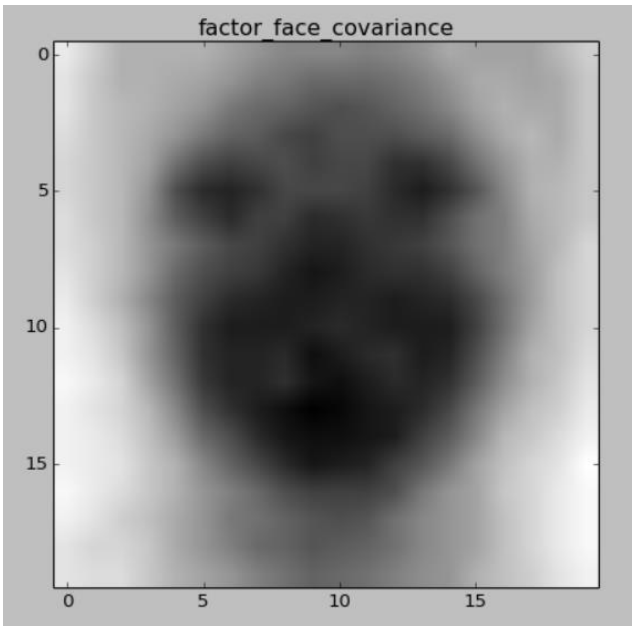
$$Pr(\mathbf{x}|\boldsymbol{\theta}) = \text{Norm}_{\mathbf{x}}[\boldsymbol{\mu}, \boldsymbol{\Phi}\boldsymbol{\Phi}^T + \boldsymbol{\Sigma}]$$

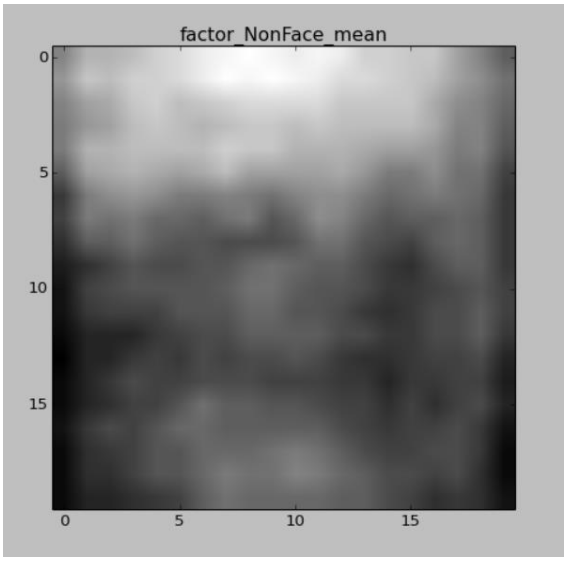
Pdf is obtained using:

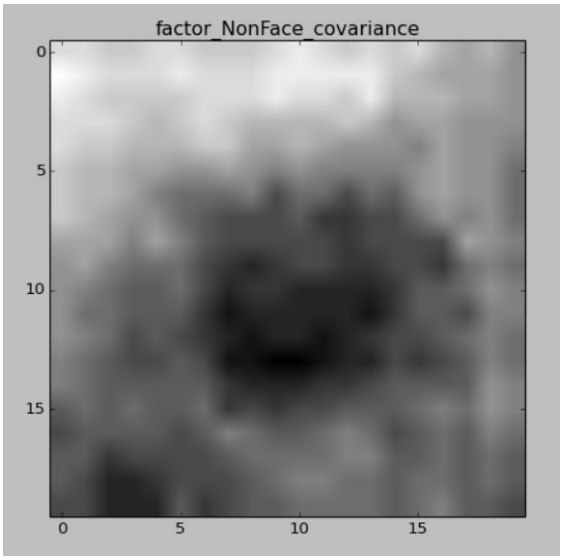
Parameters: mu -mean vector , phi – contains the factor element (DxK matrix) and sigma diagonal matrix DxD corresponding to noise.

Results:









Result:

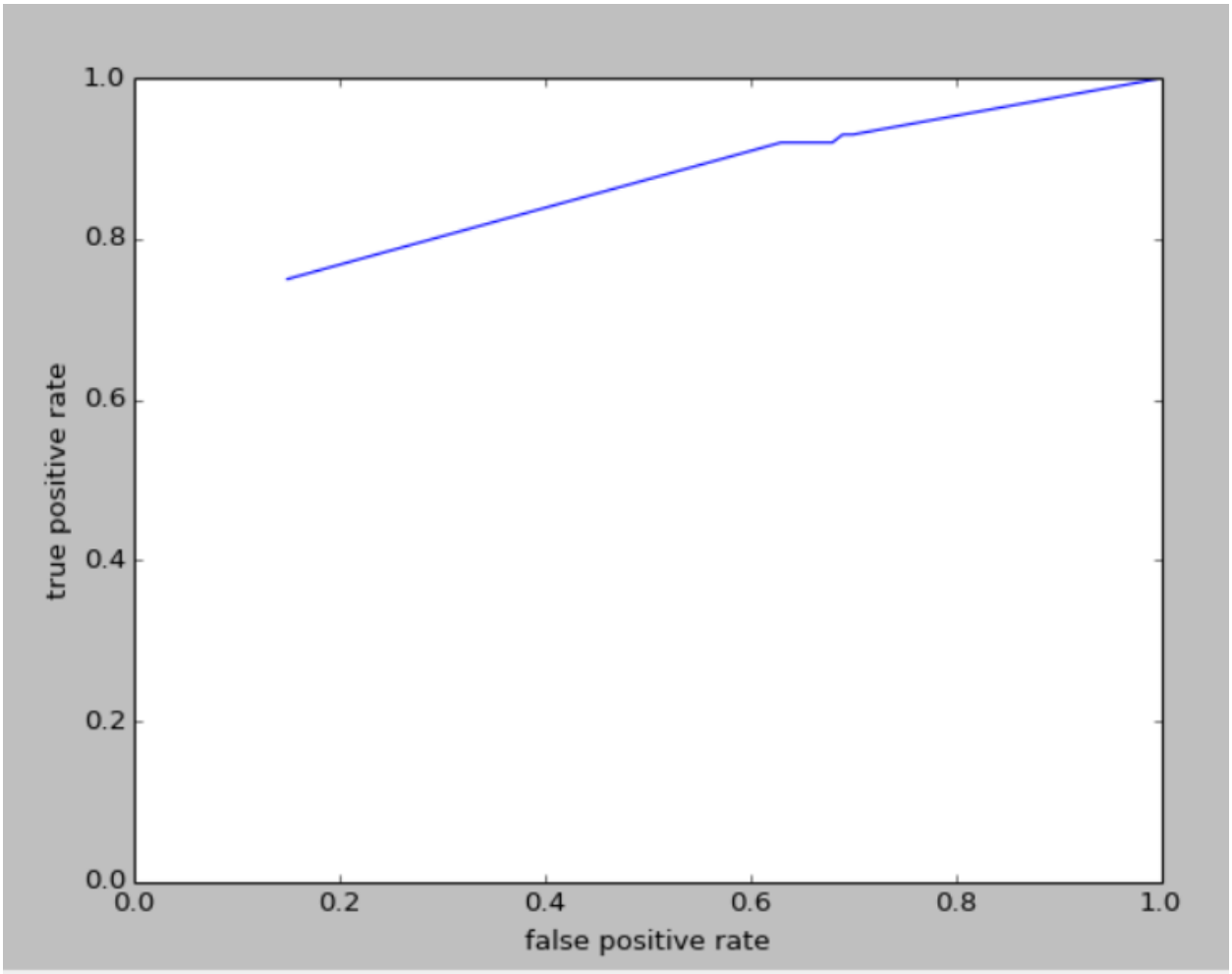
False Positive rate = 0.65

False Negative Rate = 0.08

Misclassification rate = 0.365

```
number of factors: 2
('K:', 2)
('number of faces detected in face(factor):', 92)
('number of non-faces detected in face data(factor):', 8)
('number of faces detected in non-face data (factor):', 65)
('number of non-faces detected in non-face data(factor):', 35)
('false_positive_rate(factor)', 0.65)
('false negative rate (factor):', 0.08)
('missclassification rate (factor)', 0.365)
('false positive rate', array([ 1.  ,  0.7 ,  0.69,  0.68,  0.66,  0.65,  0.63,  0.63,  0.63,
        0.63,  0.15]))
('true positive rate', array([ 1.  ,  0.93,  0.93,  0.92,  0.92,  0.92,  0.92,  0.92,  0.92,
        0.92,  0.75]))
```

Factor EOC plot:



Usage:

1. Open the srajend2_project1.ipynb python notebook file
2. Change directory to image folders and desired output path for resized images

```
face_data="C:/Users/sreer/Desktop/Computer Vision ECE 763/srajend2_project1/0"
face_data_resized= "C:/Users/sreer/Desktop/Computer Vision ECE 763/srajend2_project1/resized_images"
back_data= "C:/Users/sreer/Desktop/Computer Vision ECE 763/srajend2_project1/background"
back_data_resized= "C:/Users/sreer/Desktop/Computer Vision ECE 763/srajend2_project1/background_resized"
test_data_resized = "C:/Users/sreer/Desktop/Computer Vision ECE 763/srajend2_project1/test_data_resized"
testface_resized= "C:/Users/sreer/Desktop/Computer Vision ECE 763/srajend2_project1/testface_resized"
testnonface_resized="C:/Users/sreer/Desktop/Computer Vision ECE 763/srajend2_project1/testnonface_resized"
```

3. Run the main routine.
4. Select desired model or run all by pressing 0 to the input query.

```
Enter model number:
 1 for Gaussian
 2 for Mixture of Gaussian
 3 for t distribution
 4 for mixture of t
 5 for factor analyzer
 6 for mixture of t factor analyzer
 0 for all
```

5. Enter number of mixtures or factors if requested.
6. Close the pop up windows displaying output images to proceed further.

Execution Result for all models:

```
Enter model number:
 1 for Gaussian
 2 for Mixture of Gaussian
 3 for t distribution
 4 for mixture of t
 5 for factor analyzer
 6 for mixture of t factor analyzer
 0 for all
0
Gaussian in progress...
('face mean shape:', (400,))
('covariance shape:', (400, 400))
('Non-face mean shape:', (400,))
('covariance shape:', (400, 400))
('number of faces detected in face(gaussian):', 92)
('number of non-faces detected in face data(gaussian):', 8)
('number of faces detected in non-face data(gaussian):', 63)
('number of non-faces detected in non-face data(gaussian):', 37)
('false_positive_rate', 0.63)
('false negative rate:', 0.08)
('missclassification rate', 0.355)
[ 1.    0.68  0.64  0.64  0.63  0.63  0.63  0.63  0.63  0.62  0.13]
[ 1.    0.93  0.92  0.92  0.92  0.92  0.92  0.92  0.92  0.91  0.75]
Mixture of Gaussian in progress...
number of mixtures: 3
('K:', 3)
('false positive rate', array([ 1.  ,  1.  ,  1.  ,  1.  ,  1.  ,  0.83,  0.  ,  0.  ,  0.  ,
        0.  ,  0.  ]))
('true positive rate', array([ 1.  ,  1.  ,  1.  ,  1.  ,  1.  ,  0.98,  0.03,  0.  ,  0.  ,
        0.  ,  0.  ]))
('number of faces detected in face(mog):', 0)
('number of non-faces detected in face data(mog):', 100)
('number of faces detected in non-face data (mog):', 0)
('number of non-faces detected in non-face data(mog):', 100)
('false_positive_rate', 0.0)
('false negative rate:', 1.0)
('missclassification rate', 0.5)
t-distribution in progress...

[[ 6014.59251117  4796.76503027  3330.36661164 ...,   930.38686334
   1025.08505961  1095.46931524]
 [ 4796.76503027  5320.30636799  4109.66746148 ...,   856.63222006
```

```
    887.12886302   949.32717049]
 [ 3330.36661164  4109.66746148  4761.47336538 ...,   964.70115773
    982.61038165  1063.38090537]
 ...,
 [  930.38686334   856.63222006   964.70115773 ...,  5291.15574292
   4882.70399737  4319.96166849]
 [ 1025.08505961   887.12886302   982.61038165 ...,  4882.70399737
   5597.57010997  5321.45344097]
 [ 1095.46931524   949.32717049  1063.38090537 ...,  4319.96166849
   5321.45344097  6006.87959436]]
('var', array([[ 6014.59251117,     0.        ,     0.        , ...,
           0.        ,     0.        ,     0.        ],
       [    0.        ,  5320.30636799,     0.        , ...,
           0.        ,     0.        ,     0.        ],
       [    0.        ,     0.        ,  4761.47336538, ...,
           0.        ,     0.        ,     0.        ],
       ...,
       [    0.        ,     0.        ,     0.        , ...,
         5291.15574292,     0.        ,     0.        ],
       [    0.        ,     0.        ,     0.        , ...,
           0.        ,  5597.57010997,     0.        ],
       [    0.        ,     0.        ,     0.        , ...,
           0.        ,     0.        ,  6006.87959436]]))
('df', 4)
C:\Python27\lib\site-packages\ipykernel_launcher.py:5: RuntimeWarning: divide by zero encountered in log
  """
C:\Python27\lib\site-packages\ipykernel_launcher.py:5: RuntimeWarning: invalid value encountered in double_scal
ars
  """

[[ 3945.2715918   3463.75593502  2834.12087612 ...,   410.58387618
    411.53987021   343.28546201]
 [ 3463.75593502  3942.82845631  3363.35173155 ...,   381.24973984
    356.38909801   285.06819671]
 [ 2834.12087612  3363.35173155  3704.49515153 ...,   457.3953155
    357.51637974   304.28766022]
 ...,
 [  410.58387618   381.24973984   457.3953155  ...,  2947.74247152
   2629.28258779  2355.90475397]
 [  411.53987021   356.38909801   357.51637974 ...,  2629.28258779
   2954.18358559  2702.04445498]
 [  343.28546201   285.06819671   304.28766022 ...,  2355.90475397
   2702.04445498  2876.39848653]]
('var', array([[ 3945.2715918 ,     0.        ,     0.        , ...,
```

```
            0.        ,     0.        ,     0.        ],
       [    0.        ,  3942.82845631,     0.        , ...,
            0.        ,     0.        ,     0.        ],
       [    0.        ,     0.        ,  3704.49515153, ...,
            0.        ,     0.        ,     0.        ],
       ...,
       [    0.        ,     0.        ,     0.        , ...,
         2947.74247152,     0.        ,     0.        ],
       [    0.        ,     0.        ,     0.        , ...,
            0.        ,  2954.18358559,     0.        ],
       [    0.        ,     0.        ,     0.        , ...,
            0.        ,     0.        ,  2876.39848653]]))
('df', 4)
('number of faces detected in face:', 92)
('number of non-faces detected in face data:', 8)
('number of faces detected in non-face data :', 63)
('number of non-faces detected in non-face data:', 37)
('false_positive_rate', 0.63)
('false negative rate :', 0.08)
('missclassification rate ', 0.355)
('false positive rate', array([ 1.  ,  1.  ,  1.  ,  1.  ,  1.  ,  0.63,  0.  ,  0.  ,  0.  ,
        0.  ,  0.  ]))
('true positive rate', array([ 1.  ,  1.  ,  1.  ,  1.  ,  1.  ,  0.92,  0.  ,  0.  ,  0.  ,
        0.  ,  0.  ]))
mixture of t...
K(default=2)2


('number of faces detected in face:', 100)
('number of non-faces detected in face data:', 0)
('number of faces detected in non-face data :', 100)
('number of non-faces detected in non-face data:', 0)
('false_positive_rate', 1.0)
('false negative rate :', 0.0)
('missclassification rate ', 0.5)
('false positive rate', array([ 1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.]))
('true positive rate', array([ 1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.]))
factor analyzer...
number of factors: 5
('K:', 5)
('number of faces detected in face(factor):', 92)
('number of non-faces detected in face data(factor):', 8)
('number of faces detected in non-face data (factor):', 63)
('number of non-faces detected in non-face data(factor):', 37)
```

```
('false_positive_rate(factor)', 0.63)
('false negative rate (factor):', 0.08)
('missclassification rate (factor)', 0.355)
('false positive rate', array([ 1.  ,  0.68,  0.64,  0.64,  0.63,  0.63,  0.63,  0.63,  0.62,
        0.61,  0.13]))
('true positive rate', array([ 1.  ,  0.93,  0.92,  0.92,  0.92,  0.92,  0.92,  0.92,  0.92,
        0.91,  0.73]))
```

In [ ]: