Computer Vision Project 1

Experimentation with Face Image Classification

- Harish Pullagurla

Objectives:

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

Database Creation :

Annotated Faces in the Wild - AFLW dataset was used for this purpose.
AFLW, consists of 22,000 full resolutions colour images with around 26000 labeled faces. Database labeling is available as a .sqlite format file, with rectangular coordinates for each face in the image labeled. X pos, Y pos , width and height are provided for each of the faces.

Face images were cropped from the full resolution images and saved. Non face regions are selected randomly from the images to represent background pixel information. The rectangle regions are of the same size as that of the original face cropped, and are chosen such that a intersection over union of less than 0.3 is present with the face regions. 3,000 images each of face and non face were cropped from the full database for further experimentation in this project.

Manual data cleaning was done on these cropped regions, to ignore images with not good face attributes , such as high occlusion, sunglasses , shadows etc.
Random numbered picking of images was done to select 1000, face and non face images for training and 200, of each for testing purpose.
Appropriate resizing was before the image was used in the model for fitting the parameters, based on computational limitation of that particular model .

Code Available on GitHub : https://github.ncsu.edu/hpullag/FaceModeling

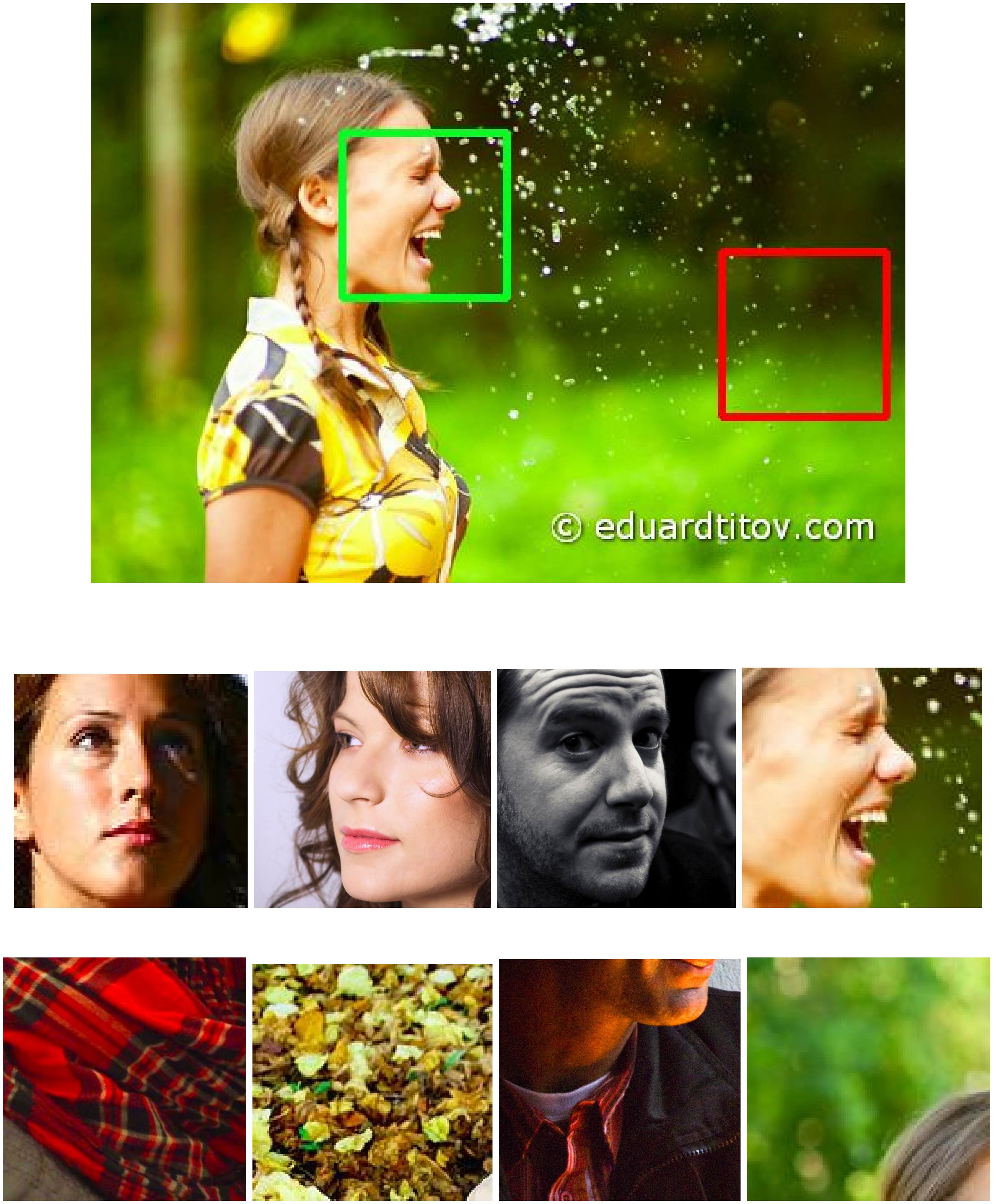


Figure :- A. Illustration of Data Set from which images were cropped. B:- Positive Face images C: Negative Face Images

Code Available on GitHub : https://github.ncsu.edu/hpullag/FaceModeling

Model 1 :- Single Gaussian Model

Fitting a single Gaussian model to the set of images, that is positive and negative images.
For this model, each image is resized and loaded as a [30,30,3] sized object .
1000 images each of face and non face category for training were loaded. For testing purpose , 200 of each category were use.

As a input, each image is flattened into a 1D array, here in this case it becomes a 30x30x3 = 2700 sized single dimensional array .
In this model, 2 single gaussian models are learnt, one for each category such that

$$\Pr(x|w = 0) = Norm_x[\mu_0\ ,\ \Sigma_0]$$
$$\Pr(x|w = 1) = Norm_x[\mu_1\ ,\ \Sigma_1]$$

A maximum likelihood based approach is followed to get parameters involved in this modeling

$$\begin{aligned} \hat{\boldsymbol{\mu}}_0, \hat{\boldsymbol{\Sigma}}_0 &= \underset{\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0}{\operatorname{argmax}} \left[\prod_{i \in \mathcal{S}_0} Pr(\mathbf{x}_i | \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) \right] \\ &= \underset{\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0}{\operatorname{argmax}} \left[\prod_{i \in \mathcal{S}_0} \operatorname{Norm}_{\mathbf{x}_i}[\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0] \right]. \end{aligned}$$

Taking the derivative and equating it zero we obtain closed form equations for the mean and covariance matrices

$$\hat{\mu} = \frac{1}{N} \sum_{n=1}^{N} x_n \qquad \hat{\Sigma} = \frac{1}{N} \sum_{n=1}^{N} (x_n - \hat{\mu})(x_n - \hat{\mu})^T$$

The obtained 1D array of shape 2700x1 is reshaped into initial image dimensions and is displayed.
For the standard deviation image, square root of the diagonal elements of the covariance matrix was taken. The diagonal matrix will be of the shape 2700x1. These diagonal elements were rescaled to be in the range of 0 - 255 and were shown as images,

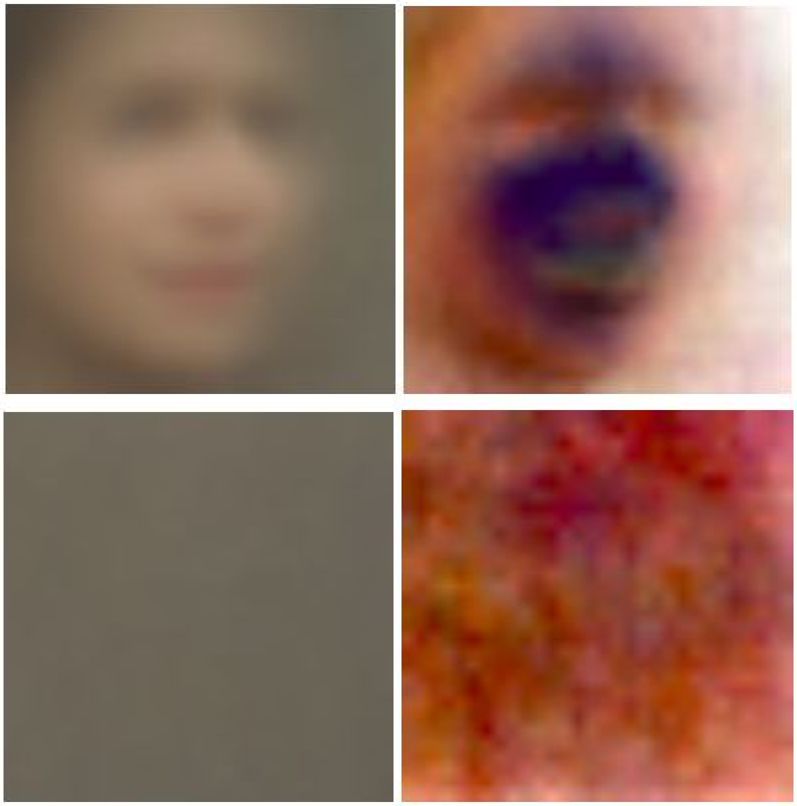


Figure :- A. Mean Positive Face image B. Standard deviation positive face image C. Negative Face image mean D. Standard deviation of negative face images

Model Evaluation :-

Evaluation of the model was done on a [9,9] gray scale image.

Dimensions for this images are 81x1. This step of calculating at the lower dimensions was because of the numerical issues faced in calculating the inverse of covariance matrix, which when goes to higher dimensions leads to overflow issues . This happens because large multiplication terms in the cov matrix.
These problems were encountered in the process of calculating the multivariate normal distributions probability density functions, given the parameters.

For simple comparison of values log probability estimates can be used, which enables us to go to a higher range of values which in turn enables us to run the matrix at [30,30] grayscale dimensions. But in order to do further experimentations such as obtaining the ROC curve it was needed for the probabilities to be in normal dimensions instead of logarithmic dimensions.

Probabilities were obtained using the formale of, where the values were got from Norm of multinomial distribution .

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

Evaluating the model on a threshold of 0.5 on the posterior probability, the following observed for a [9,9] grayscale image

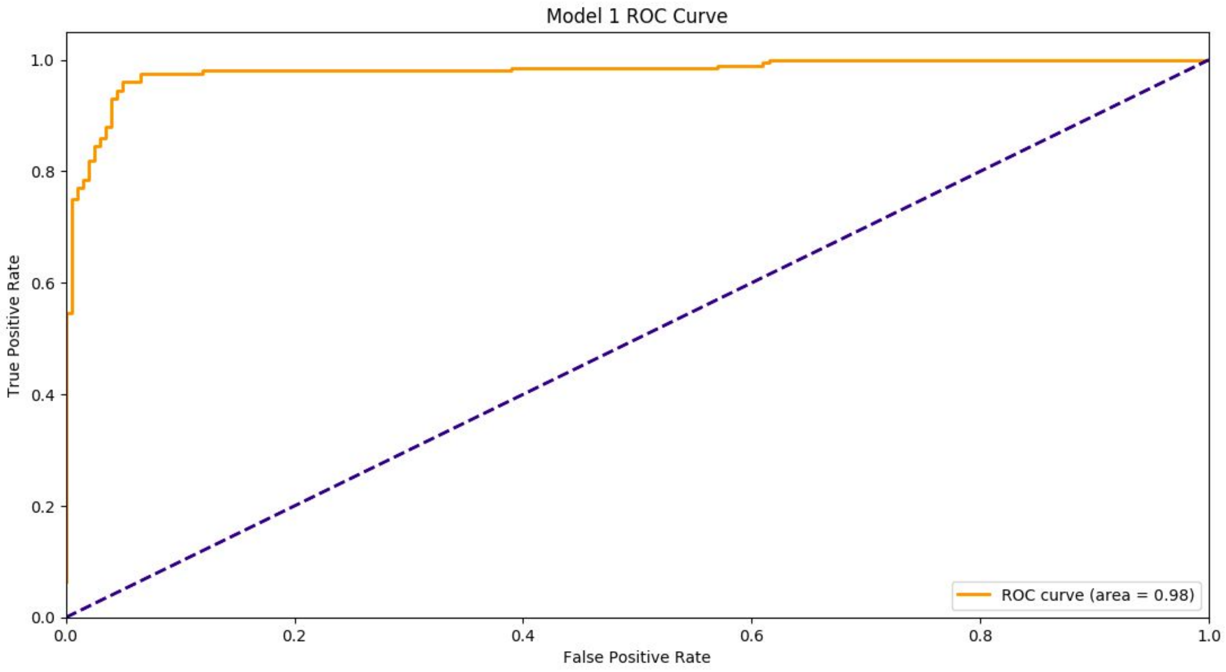
	Positive Face Image Test Set	Negative Face Image Set
Count Positive	184	11
Count Negative	16	189
False Negatives (Positive faces being classified as non faces)	0.08	-
False Positives (Negative faces being classified as faces)	-	0.055
% correctly predicted	92 %	94.5 %

Misclassification Rate = $\frac{Flase\ Positives + False\ Negatives}{Total\ Testing\ Images}$ = 0.0676

Experiments were done on a 30,30 grayscale images and it was seen that correct prediction rate of around 40-50 % was observed in comparison to 90% + in this dimension. In this case pdf were estimated with log probability values and higher of the two values were given the better score.

A possible reasoning for this observation is that lesser number of parameters are being learnt and they were able to split the data well, based on means at this ranges.

Roc curve which is a plot between true positive rate and false positive rate is shown in figure below.


Model 1 ROC Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (area = 0.98)

Model 2 : Mixture of Gaussian model

Evaluation of the model was done on a [6,6] gray scale image.

Mixture of Gaussian tries to represent each pixel on the image as a weighted summation of values coming from multiple, multivariate gaussian distributions.
This is done with a view that data cannot be generalised as coming from a single gaussian. Assuming them as a set of multiple gaussians lets the data to be modelted in a more flexible way. A k gaussian mixture is represented as :-

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

Here each model has k (num of gaussian) values of

Lambda - scaling factor for each GM
Mean vector - 1D vector of image dime
Covariance Matrix - 2D vector of image_dim X image_dim

Each of which has to be learnt.

The model cannot be solved in a direct closed form way , and is hence looked at from a hidden variable perspective, which is sampled from a categorical distribution.

$$\begin{aligned} Pr(\mathbf{x}|h, \boldsymbol{\theta}) &= \text{Norm}_{\mathbf{x}}[\boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h] \\ Pr(h|\boldsymbol{\theta}) &= \text{Cat}_h[\boldsymbol{\lambda}], \end{aligned} \qquad Pr(\mathbf{x}|\boldsymbol{\theta}) = \sum_{k=1}^{K} Pr(\mathbf{x}, h = k|\boldsymbol{\theta})$$

Expectation step for fitting a mixture of gaussian model is given by

$$Pr(h_i = k|\mathbf{x}_i, \boldsymbol{\theta}^{[t]}) = \frac{\lambda_k \text{Norm}_{\mathbf{x}_i}[\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k]}{\sum_{j=1}^{K} \lambda_j \text{Norm}_{\mathbf{x}_i}[\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j]}$$

we compute the probability Pr such that the kth normal distribution was responsible for the ith data point.
The Maximization step for the model is given by

$$\lambda_k^{[t+1]} = \frac{\sum_{i=1}^{I} r_{ik}}{\sum_{j=1}^{K} \sum_{i=1}^{I} r_{ij}} \qquad \boldsymbol{\mu}_k^{[t+1]} = \frac{\sum_{i=1}^{I} r_{ik} \mathbf{x}_i}{\sum_{i=1}^{I} r_{ik}}$$

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

Number of Gaussians Considered in this experimentation - 5

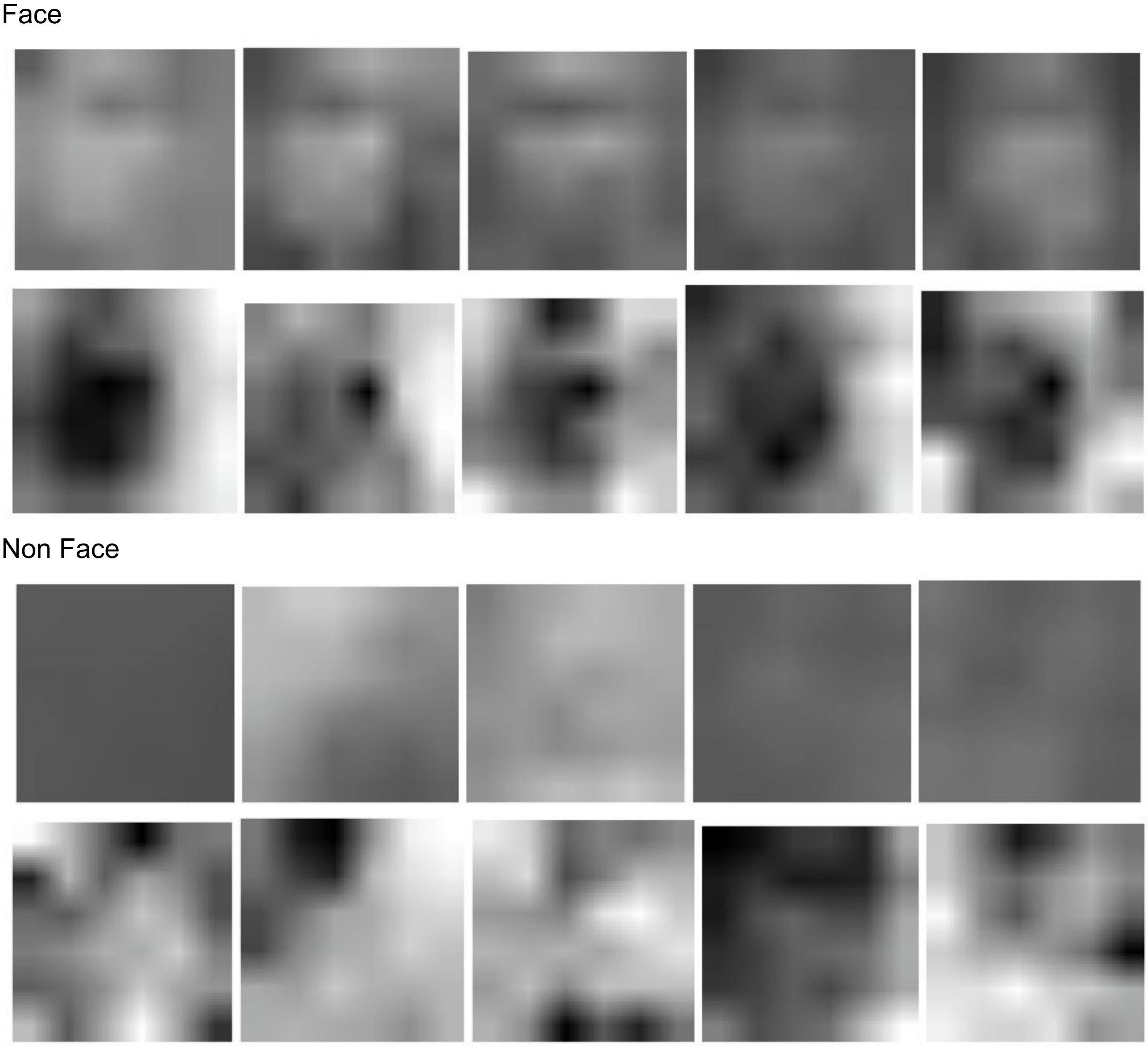


Figure :- Mean and Standard Deviation image plots from different gaussian models learnt using the Expectation Maximization algorithm for both Face Image dataset and Non Face Image data set. Dimensions - [6,6]

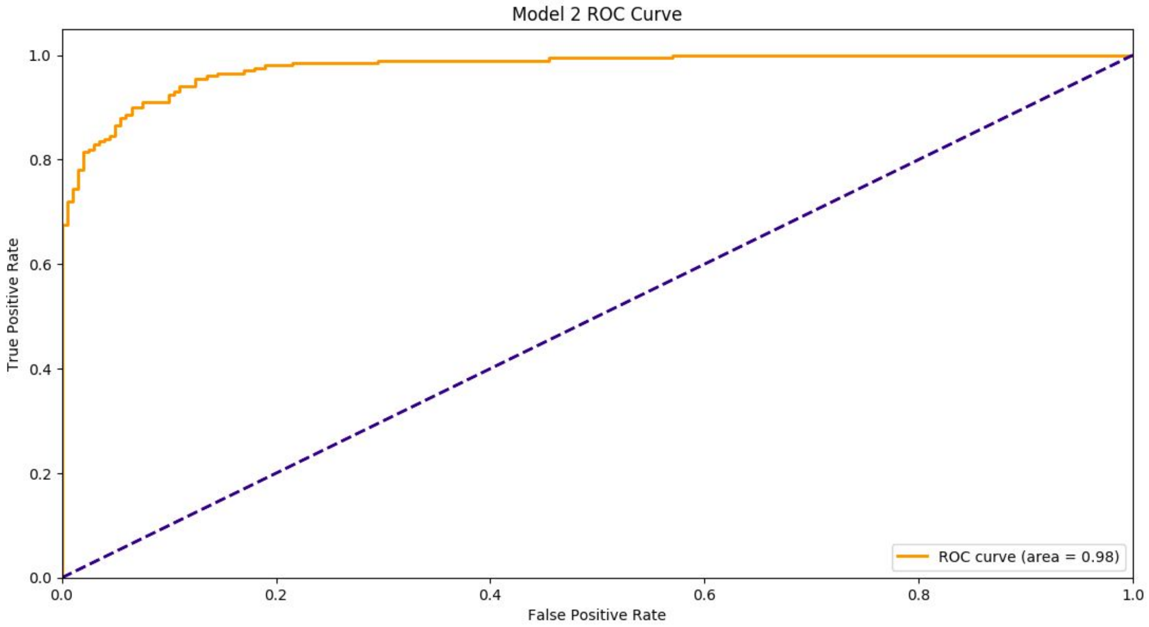
Initialization of the mean vector was done using the random values in the range of 0 - 255. Covariance matrix was initialized using a diagonal matrix with elements in the range of 0 - 4000 , as higher covariance would mean a not so low probability value, restricting upon hitting the under flow error in initial stages of the learning process.
Hence it was important in setting the initializations correctly, and not completely randomly.
Also setting all the k distributions in the same way will not let the model learn robustly .

Model Evaluation :-

Evaluating the model on a threshold of 0.5 on the posterior probability, the following observed for a [6,6] grayscale image

	Positive Face Image Test Set	Negative Face Image Set
Count Positive	188	25
Count Negative	12	175
False Negatives (Positive faces being classified as non faces)	0.06	-
False Positives (Negative faces being classified as faces)	-	0.125
% correctly predicted	94 %	87.5 %

Misclassification rate :- 0.0925



Model 3 :- t-distribution model

The problem with using the normal distribution to describe visual data is that it is not robust: the height of the normal pdf falls off very rapidly as we move into the tails. The effect of this is that outliers (unusually extreme observations) drastically affect the estimated parameters. The t-distribution is a closely related distribution in which the length of the tails is parameterized.

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

Here nu is the degrees of freedom for the data set which describes the fit of std values. D is the dimensionality of data.
Stud’s t distribution looks at the model in a different way .

$$\begin{aligned} Pr(\mathbf{x}|h) &= \text{Norm}_x[\boldsymbol{\mu}, \boldsymbol{\Sigma}/h] \\ Pr(h) &= \text{Gam}_h[\nu/2, \nu/2], \end{aligned} \qquad Pr(\mathbf{x}) \quad \int \text{Norm}_x[\boldsymbol{\mu}, \boldsymbol{\Sigma}/h]\text{Gam}_h[\nu/2, \nu/2]dh$$

This formulation provides a method to generate data from the t-distribution; h is first generated from the gamma distribution and then x from the associated normal distribution Pr(x|h). The hidden variable tells, which one of the continuous family of underlying normal distributions was responsible for this data point.

Expectation Maximization method is used to learn the parameters from this distribution,
E stem, we compute

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

Maximization step involves

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

Nu for data - faces was learned as 0.04 , non faces as 0.0007.
Plotting of image data was done at [30,30] dimensions. But evaluations were performed at [9,9] grayscale dimensions because of Pdf computations that were to be done.

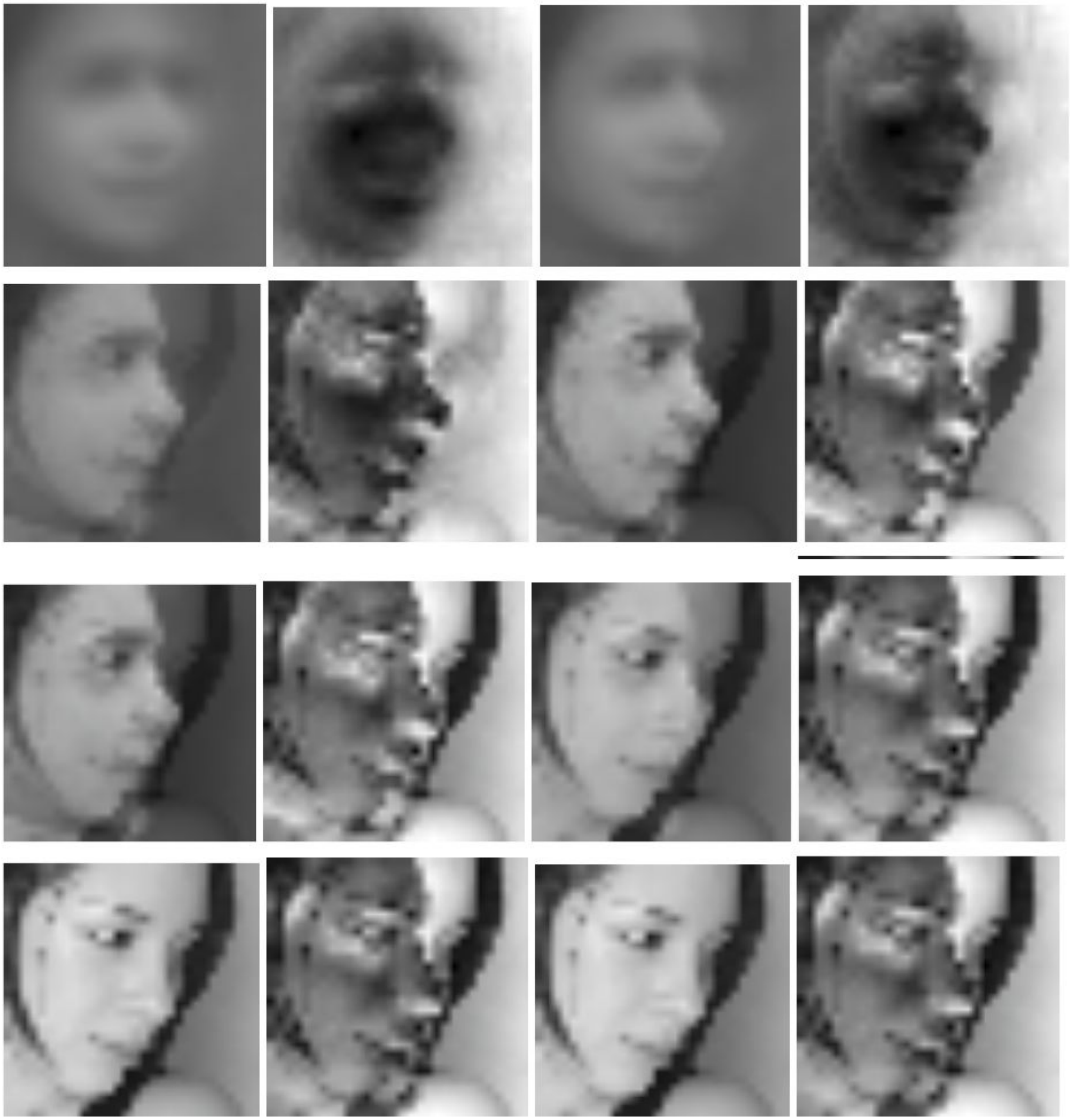


Figure :- Figure shows the plots of Mean face and corresponding standard deviation matrix, learnt over multiple iterations. Plots are at an interval after 5 iterations of EM algorithm. Dim [30,30]

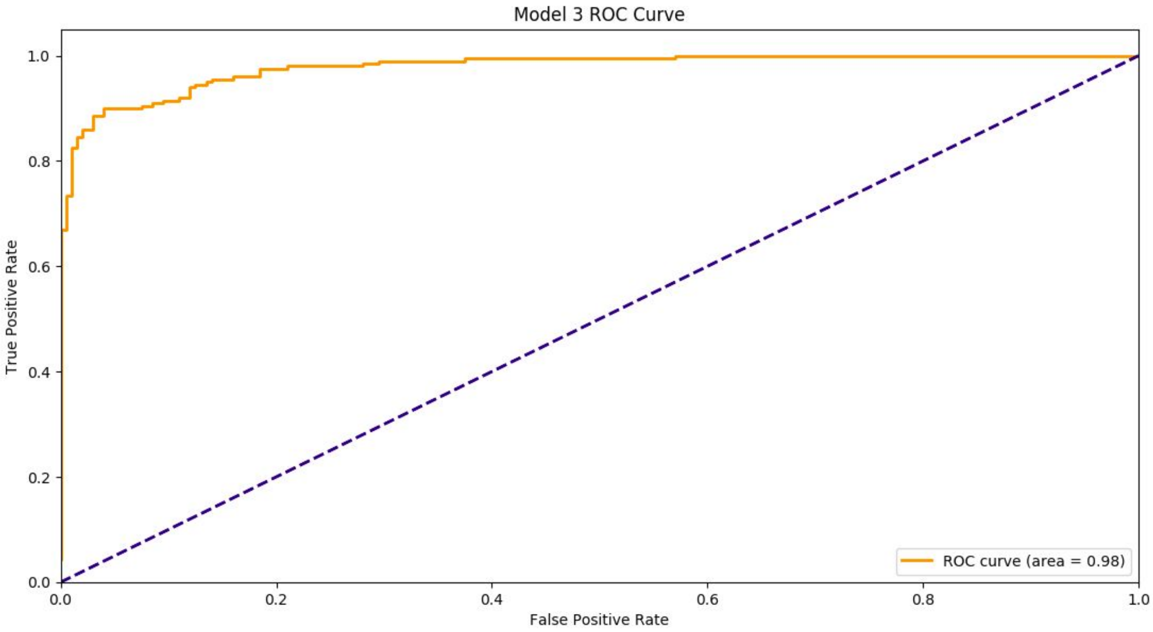
It is observed from the iterations that over time, the model has learn where the bulk of the images in the dataset looks like, and the corresponding std plot shows its modifications trying to if in all the images in the dataset.

Model Evaluation :

Evaluating the model on a threshold of 0.5 on the posterior probability, the following observed for a [9,9] grayscale image

	Positive Face Image Test Set	Negative Face Image Set
Count Positive	195	40
Count Negative	5	160
False Negatives (Positive faces being classified as non faces)	0.025	-
False Positives (Negative faces being classified as faces)	-	0.2
% correctly predicted	97.25 %	80 %

Misclassification rate :- 0.1125



The above model would also run at higher dimensions than plotted if the pdf's are looked at in the logarithmic scale

Code Available on GitHub : https://github.ncsu.edu/hpullag/FaceModeling

Model 4 - Mixture of t - distributions

Mixture of t was implemented as a combination of model 2 and 3.
Each element in the dataset is considered to be obtained from multiple t distributions, like in model 2.

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

Each element is considered as a mixture of ‘g’ t distributions, each contributing to a weight of pi/lambda. Each of the t distributions, has a mean , covariance matrix , degree of freedom parameter.
EM algorithm was used to learn the parameters in the model.
In the E step we compute :-

$$\tau_{ij}^{(k)} = \frac{\pi_i^{(k)} f\left(\boldsymbol{y}_j;\, \boldsymbol{\mu}_i^{(k+1)},\, \boldsymbol{\Sigma}_i^{(k+1)},\, \nu_i^{(k+1)}\right)}{f\left(\boldsymbol{y}_j;\, \boldsymbol{\Psi}^{(k+1)}\right)}$$

$$u_{ij}^{(k)} = \frac{\nu_i^k + p}{\nu_i^k + \delta\left(\boldsymbol{y}_j,\, \boldsymbol{\mu}_i^{(k)};\, \boldsymbol{\Sigma}_i^{(k)}\right)}.$$

$$E_{\Psi^{(k)}}(\log U_j \mid \boldsymbol{y}_j,\, z_{ij} = 1) = \log u_{ij}^{(k)} + \left\{\psi\left(\frac{\nu_i^{(k)} + p}{2}\right) - \log\left(\frac{\nu_i^{(k)} + p}{2}\right)\right\}$$

In the M step, we update the parameters for each of the g t - distributions

$$\pi_i^{(k+1)} = \sum_{j=1}^{n} \tau_{ij}^{(k)} / n$$

$$\boldsymbol{\mu}_i^{(k+1)} = \sum_{j=1}^{n} \tau_{ij}^{(k)} u_{ij}^{(k)} \boldsymbol{y}_j \Big/ \sum_{j=1}^{n} \tau_{ij}^{(k)} u_{ij}^{(k)}$$

$$\boldsymbol{\Sigma}_i^{(k+1)} = \frac{\sum_{j=1}^{n} \tau_{ij}^{(k)} u_{ij}^{(k)} \left(\boldsymbol{y}_j - \boldsymbol{\mu}_i^{(k+1)}\right)\left(\boldsymbol{y}_j - \boldsymbol{\mu}_i^{(k+1)}\right)^T}{\sum_{j=1}^{n} \tau_{ij}^{(k)}}$$

The parameter update equations where very much similar to models 2 and 3.

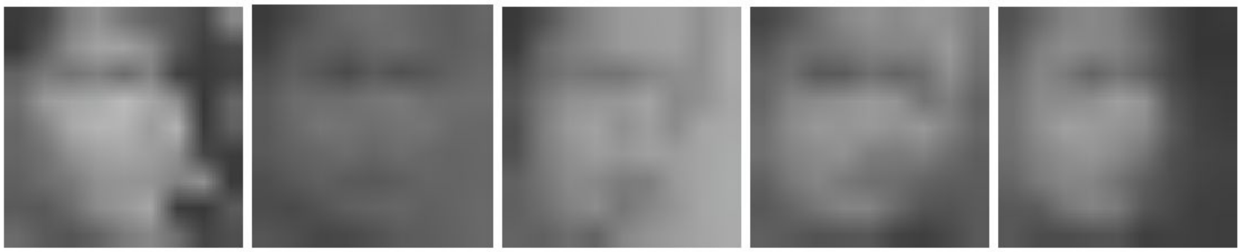


Figure:- Mean Positive face images size [9,9] for learned model of 5 - t distribution models .

The dof obtained over multiple iterations is given by

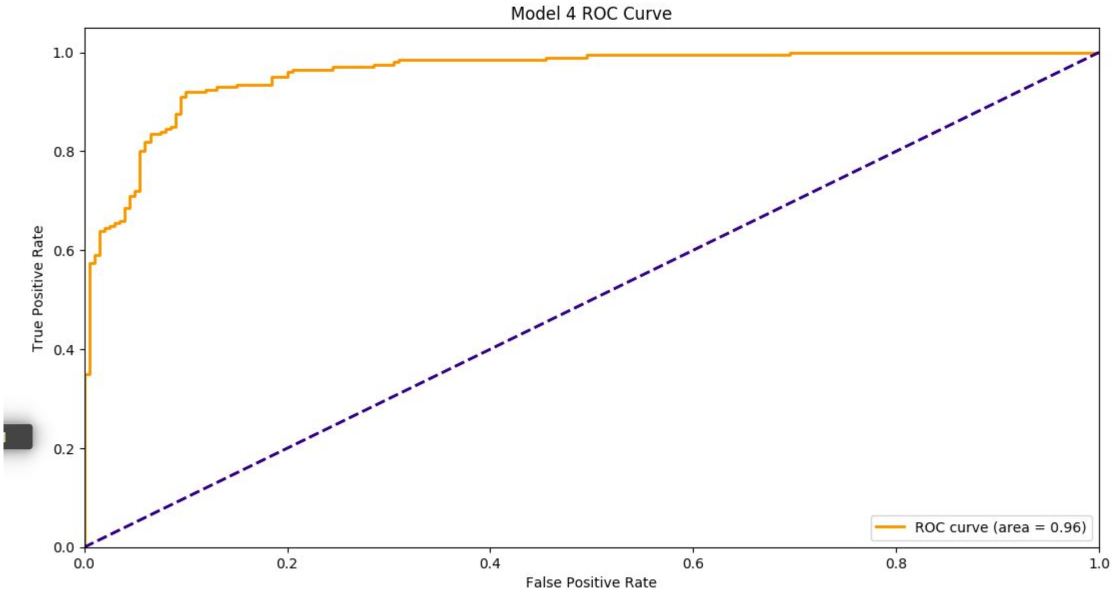
Iteration	T dist 1	T dist 2	T dist 3	T dist 4	T dist - 5
1	15.3	27.76	15.4	32.27	19.02
2	6.44	16.6	4.0	21.62	10.45
3	3.85	9.49	1.29	14.32	5.36
4	3.05	6.50	0.89	10.82	3.58
5	2.68	5.27	0.70	9.30	2.88
6	2.47	4.77	0.633	8.70	2.24
7	2.28	4.57	0.60	8.51	2.00

We could see from the values different models converging to a different data point / distribution

Evaluating the model on a threshold of 0.5 on the posterior probability, the following observed for a [5,5] grayscale image.
Here a 5,5 was used because of the need to compute pdf values in a normal scale which was leading to underflow issue

	Positive Face Image Test Set	Negative Face Image Set
Count Positive	183	20
Count Negative	17	180
False Negatives (Positive faces being classified as non faces)	0.085	-
False Positives (Negative faces being classified as faces)	-	0.1
% correctly predicted	91.5 %	90 %

Misclassification rate :- 0.095



Model 5 - Factor Analysis

Factors Analysis tries to decrease the model complexity by changing the number of parameters learnt in the covariance matrix. Here an attempt is made modify the matrix, by not learning all the DxD parameters required. Only the main diagonal elements are taken, modifying all the other elements to zeros. The remaining parameters are tried to be represented in a subspace model which is learnt.

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

For the EM algorithm, the model is expressed as

$$\begin{aligned} Pr(\mathbf{x}|\mathbf{h}) &= \text{Norm}_{\mathbf{x}}[\boldsymbol{\mu} + \boldsymbol{\Phi}\mathbf{h}, \boldsymbol{\Sigma}] \\ Pr(\mathbf{h}) &= \text{Norm}_{\mathbf{h}}[\mathbf{0}, \mathbf{I}], \end{aligned}$$

$$Pr(\mathbf{x}) = \int \text{Norm}_{\mathbf{x}}[\boldsymbol{\mu} + \boldsymbol{\Phi}\mathbf{h}, \boldsymbol{\Sigma}]\text{Norm}_{\mathbf{h}}[\mathbf{0}, \mathbf{I}]\, d\mathbf{h}$$

Expressing factor analysis as a marginalization reveals a simple method to draw samples from the distribution. We first draw a hidden variable h from the normal prior. We then draw the sample x from a normal distribution with mean μ + Φh and diagonal covariance Σ

Expectation Step is given by

$$\begin{aligned} &= \frac{\text{Norm}_{\mathbf{x}_i}[\boldsymbol{\mu} + \boldsymbol{\Phi}\mathbf{h}_i, \boldsymbol{\Sigma}]\text{Norm}_{\mathbf{h}_i}[\mathbf{0}, \mathbf{I}]}{Pr(\mathbf{x}_i|\boldsymbol{\theta}^{[t]})} \\ Pr(\mathbf{x}) &= \text{Norm}_{\mathbf{h}_i}[(\boldsymbol{\Phi}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\Phi} + \mathbf{I})^{-1}\boldsymbol{\Phi}^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu}), (\boldsymbol{\Phi}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\Phi} + \mathbf{I})^{-1}] \end{aligned}$$

Maximization Step

$$\begin{aligned} \hat{\boldsymbol{\mu}} &= \frac{\sum_{i=1}^{I} \mathbf{x}_i}{I} \\ \hat{\boldsymbol{\Phi}} &= \left(\sum_{i=1}^{I} (\mathbf{x}_i - \hat{\boldsymbol{\mu}})\text{E}[\mathbf{h}_i]^T\right)\left(\sum_{i=1}^{I} \text{E}[\mathbf{h}_i\mathbf{h}_i^T]\right)^{-1} \\ \hat{\boldsymbol{\Sigma}} &= \frac{1}{I}\sum_{i=1}^{I} \text{diag}\left[(\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^T - \hat{\boldsymbol{\Phi}}\text{E}[\mathbf{h}_i](\mathbf{x}^T - \boldsymbol{\mu})\right] \end{aligned}$$

Plotting of images and learning was done at 30,30,3 dimensions.

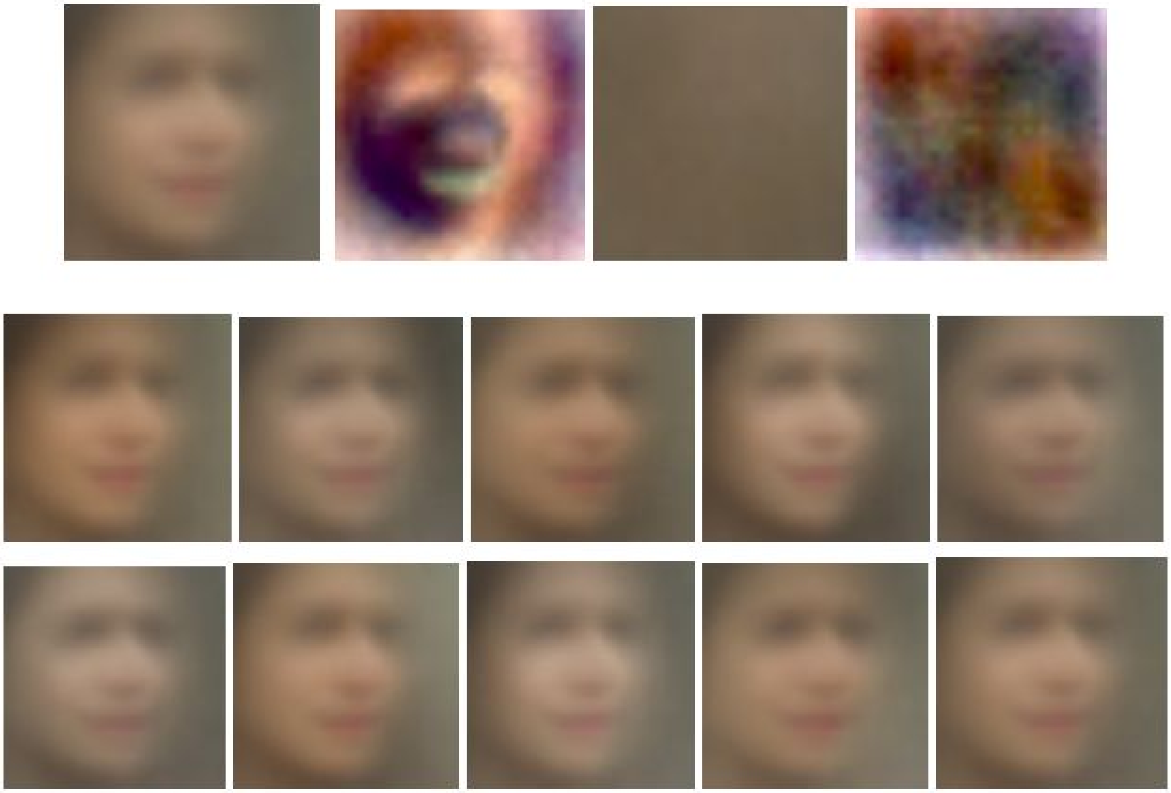


Figure :- First row of images represent the mean and standard deviation plots of both face and non face images . The subsequent rows represent images of mean +/- 2* each factor . column represents the +ve and -ve difference of the factor with respect to the original mean face

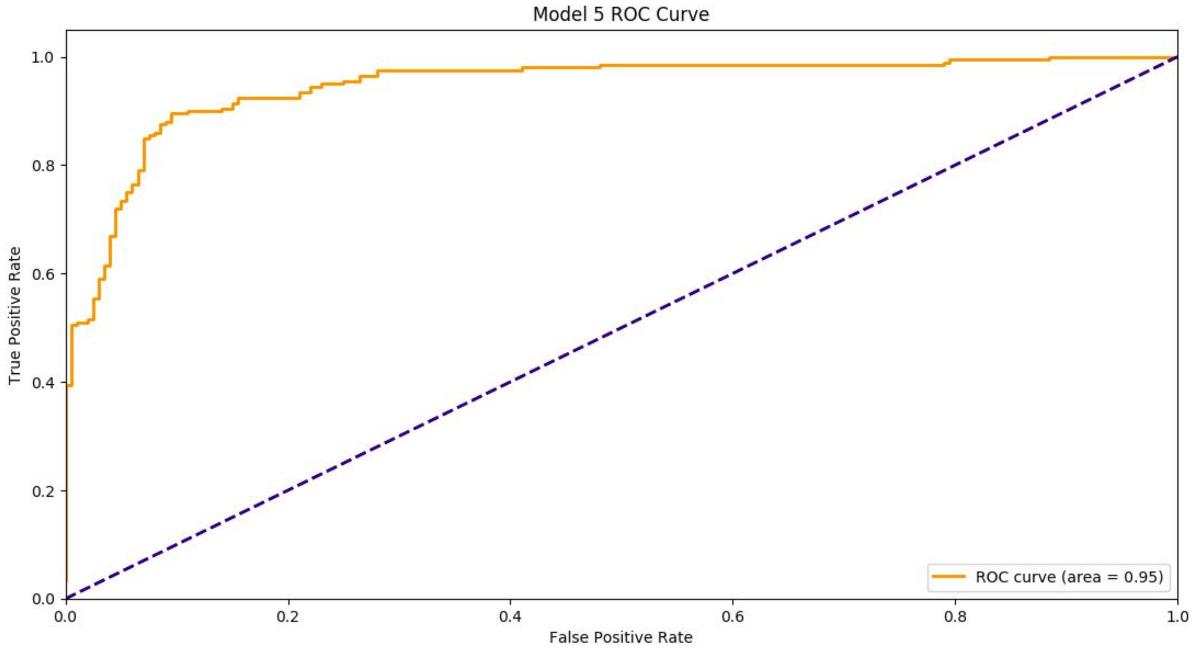
This learning was done using logpdf hence the higher dimensions was possible . The method could be run in higher dimensions if the posteriors were to be observed in the logarithmic scale.

Model Evaluation :-

Evaluating the model on a threshold of 0.5 on the posterior probability, the following observed for a [9,9] grayscale image, because , computation of multinomial pdf is to be done for evaluations similar to the earlier cases.

	Positive Face Image Test Set	Negative Face Image Set
Count Positive	177	19
Count Negative	23	181
False Negatives (Positive faces being classified as non faces)	0.115	-
False Positives (Negative faces being classified as faces)	-	0.095
% correctly predicted	88.5 %	90. %

Misclassification rate - 0.105


Model 5 ROC Curve
True Positive Rate
False Positive Rate
ROC curve (area = 0.95)