Introduction:

I think this assignment is about creating code that is able to take in and study several different images that identify as a face or background image. The code is supposed to train the face images separately from the background images so that when we test the code later on, it should be able to identify if an image is a face image or not. In order for this to work properly, we must use a Multivariate Gaussian Distribution, which finds $\mu_{y=1}$, $\mu_{y=0}$, $\Sigma_{y=1}$, and $\Sigma_{y=0}$, where y=1 represents face images and y=0 is for background.After having trained the code from each of the training images, we can then use μ and Σ to help us identify whether images given for testing are face images or not by using the method explained in the next step.

Method:

The method used for this assignment was pretty straightforward. I started by training the face images where I made a for loop to go through each of the training images. During each iteration, I converted the 20x15x3 image into a 900x1 vector that I added to $\mu_{y=1}$ (also a 900x1 vector). After this for loop I divided $\mu_{y=1}$ by I (the number of training images), which essentially divides each element in $\mu_{y=1}$ by I. Right after that, I ran another for loop going through each training image where I subtracted the 900x1 vector of the current image from the 900x1 vector of the mean vector ($\mu_{y=1}$), thus subtracting each element in the current vector by its corresponding element in the $\mu_{y=1}$. I then multiplied the subtracted vectors by its transpose, which was then added to the $\Sigma_{y=1}$ (a 900x900 matrix). After the for loop ended, similar to the first for loop, I divided each element in the $\Sigma_{y=1}$ matrix by I to average each of the elements out. This entire process was then repeated for $\mu_{y=0}$ and $\Sigma_{y=0}$ for the background images. Once the training was completely over with, I chose the amount of testing images I wanted to try from both the face and background folders. The results can be seen in the experiment section. When I tested the images, I would take each individual picture and solve for Pr(y=1|x) and Pr(y=0|x). The equation for this was simple in the sense where I would take the log of each element in the diagonal of the covariance matrix ($\Sigma_{y=1}$) and add that up as a sum. I would then subtract each element of the $\mu_{y=1}$ vector from the corresponding element in the current image vector and then square it. I would then take that number and divide it by two times the current element in the diagonal of $\Sigma_{y=1}$. Taking those two numbers, I multiplied the first by -0.5 and then subtracted the second number from that product, creating the variable Pr(y=1|x). The same process was done for the background images (Pr(y=0|x)). I then created an if statement to check if the current image is a face image or not. If Pr(y=1|x) > Pr(y=0|x), then it is considered a face image. Otherwise, it is considered a background image. This was done for each training image and had a counter for correctly identified face and background images.

Experiments:

	Trial 1	Trial 2	Trial 3	Trial 4
# of Face Images Trained	50	100	All (184)	All (184)
# of Background Images Trained	100	200	All (249)	All (249)
# of Face Images Tested	25	25	50	All (564)
# of Background Images Tested	50	50	100	All (232)
Results:	Face Images Correctly Identified: 25 (100%) Background Images Correctly Identified: 49 (98%)	Face Images Correctly Identified: 25 (100%) Background Images Correctly Identified: 40 (80%)	Face Images Correctly Identified: 50 (100%) Background Images Correctly Identified: 75 (75%)	Face Images Correctly Identified: 232 (100%) Background Images Correctly Identified: 421 (74.65%)

Discussion:

As we can see in the table, all of the face images were identified correctly as face images, but the background images do not seem to be as easy to identify from the given algorithm, even though the percentages were still fairly high. The first test worked well when only using 50 face training images and 100 background training images. 100% of the face testing images were correctly identified whereas the correct number of background testing images was 98%, which makes it seem like the algorithm works well. However, when adding lots of different background images, it is sometimes hard to tell whether it is a face image or not. All face image vectors seem to pertain to certain rgb values (skin tones), while the background images can be any color, sometimes including those skin tones. So, there can sometimes be errors where the code will see those colors and think it is closest to a face image. In addition to that, there are some images that were not considered face images from the training folder that did show a mouth and nose, which could be a reason as to why some background images were seen as face images being that it showed more than half of the face. Therefore, in the sense that a high percentage of each type of

image were identified correctly, the code works pretty well, but there can be some changes when it comes to using the rgb vectors of the image.