CSE353 - HW2
Yajie Wang

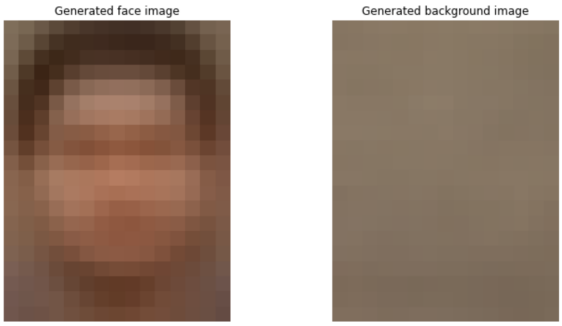
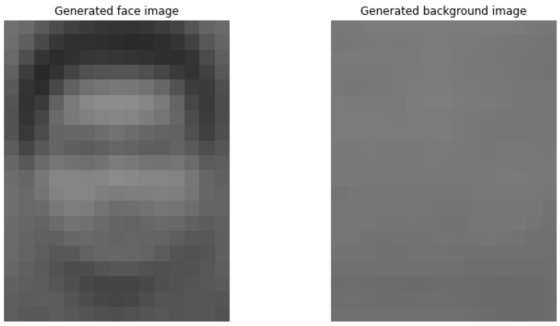
Introduction

In this assignment, we are using a generative model to classify face images from non-face/background images. Our dataset can be divided into two categories, 433 training images and 796 test images. We use 249 background images and 184 face images to generate the detection algorithm. Then we use this algorithm to detect all the test images and get the detected results. Finally, we calculate the accuracy to learn the reliability of our algorithm.

Method

Our algorithm is based on the multivariate normal distribution and Bayes' rule. We divide all the images into two types, face images and background images. We visualize the images by storing all the RGB values into a matrix. In the training process, we calculate the mean and variance of each type and store them into matrices. Then we use these matrices to calculate the likelihood in the testing process. We identify the type of an image by comparing its likelihood of being a face image with its likelihood of being a background image. To facilitate our calculations, we assume the prior is uniform and the covariance matrix is diagonal. We also simplify the normal distribution by taking logarithms to avoid the issues on computation. I mainly use nested for loops to implement my algorithm. (See more details in method.pdf)

Experiments

	
Number of detected background images: 421 Number of actual background images: 564 Accuracy_bg: 0.7464539007092199 Number of detected face images: 191 Number of actual face images: 232 Accuracy_face: 0.8232758620689655 Total precision: 0.5718562874251497 Total recall: 0.8232758620689655 Total fscore: 0.6749116607773853	Number of detected background images: 419 Number of actual background images: 564 Accuracy_bg: 0.7429078014184397 Number of detected face images: 181 Number of actual face images: 232 Accuracy_face: 0.7801724137931034 Total precision: 0.5552147239263804 Total recall: 0.7801724137931034 Total fscore: 0.6487455197132618
with three channels	with grayscale

Number of detected background images: 361 Number of actual background images: 564 Accuracy_bg: 0.6400709219858156 Number of detected face images: 206 Number of actual face images: 232 Accuracy_face: 0.8879310344827587 Total precision: 0.5036674816625917 Total recall: 0.8879310344827587 Total fscore: 0.6427457098283931	Number of detected background images: 401 Number of actual background images: 564 Accuracy_bg: 0.7109929078014184 Number of detected face images: 186 Number of actual face images: 232 Accuracy_face: 0.8017241379310345 Total precision: 0.5329512893982808 Total recall: 0.8017241379310345 Total fscore: 0.6402753872633391	Number of detected background images: 463 Number of actual background images: 564 Accuracy_bg: 0.8209219858156028 Number of detected face images: 157 Number of actual face images: 232 Accuracy_face: 0.6767241379310345 Total precision: 0.6085271317829457 Total recall: 0.6767241379310345 Total fscore: 0.6408163265306122
with a single blue channel	with a single green channel	with a single red channel

Discussions

The result of the test met our expectations. The face image we generated through machine learning looks very similar to a real human face. The accuracies of detecting background images and face images are 75% and 82%, respectively. The accuracy would be slightly lower with grayscale. I also use the single red, green and blue channel for the generative model. The accuracy of detecting face images is highest if I use the single blue channel. The accuracy of detecting background images is highest if I use the single red channel. However, the result of using all three channels at the same time is the most accurate one. I don’t have the chance to try to use the spatial gradient of the images as input. But in theory, the result will be more accurate.