

Assignment-1 : LARP

Sankaran Vaidyanathan and Karthik Thiagarajan (CS16S027)

October 5, 2017

Contents

1	Question-2	2
1.1	Part-a	2
1.2	Part-b	4

1 Question-2

1.1 Part-a

Observations



Figure 1: Faces 2, 5 and 6



Figure 2: Faces 8, 9 and 10

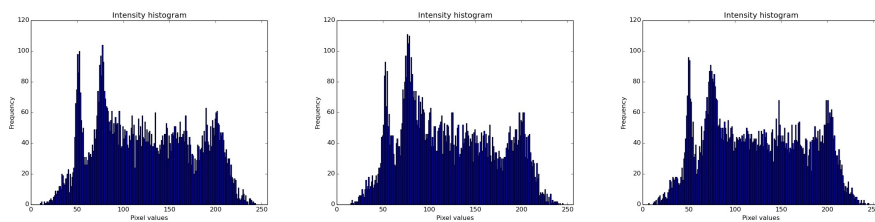


Figure 3: Faces 2, 5 and 6

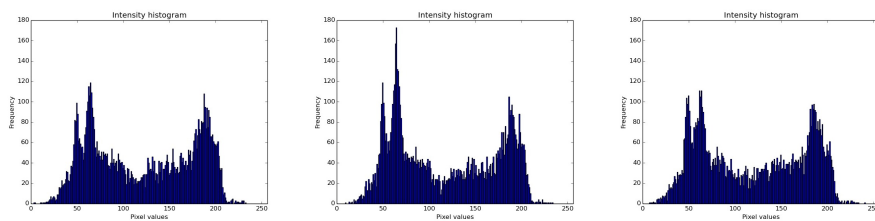


Figure 4: Faces 8, 9 and 10

Inference

All ten images look very similar and the intensity histograms reflect this similarity. In general, the intensity distribution has three peaks, one centred at 200, the other two occurring near the intensity value of 50. There is an interesting observation concerning faces with glasses. The distribution is quite uniform between the left and right peaks. Contrast this with the distribution of the faces without glasses - there is a clear valley between the peaks. Though the global shape of the intensity distribution is the same across all ten images, there are subtle local variations. These variations are identical for images that are similar to each other. “Wearing glasses” is one notion of similarity.

1.2 Part-b

Flesh, Blood and Bone

Before looking at the basis vectors, it is essential to understand the process used to generate them. The reason will become clear as we go along. Let us assume that we are working with a collection of N face images. Each image is a 92×92 matrix, and can be converted to an 8464 dimensional vector. To begin with, we assume that the vectorization proceeds row-wise.

$$\begin{aligned}\mu &= \frac{1}{N} \sum_{i=1}^N f_i \\ \Sigma &= \frac{1}{N} \sum_{i=1}^N (f_i - \mu)(f_i - \mu)^T \\ \Sigma &= Q\Lambda Q^T\end{aligned}$$

, where f_i is the i^{th} face-vector, μ is the estimated mean and Σ is the estimated covariance matrix. Σ is an 8464×8464 symmetric matrix. By the spectral theorem, we can find a basis for \mathbb{R}^{8464} consisting of orthonormal eigenvectors of Σ . Q is the matrix whose columns are these eigenvectors. We have been given this matrix to work with.

When an eigenvector is extracted from Q and reshaped to 92×92 , we observe that the resulting image - called an eigenface in computer vision literature - is rotated 90 degrees anti-clockwise. Clearly, the order of reshaping is different. MATLAB uses the Fortran ordering to reshape matrices. NumPy uses the C ordering. By explicitly specifying the Fortran ordering, we get the correct eigenfaces.



Figure 5: Flesh : Eigenfaces 1 to 5



Figure 6: Bone : Eigenfaces 101 to 105

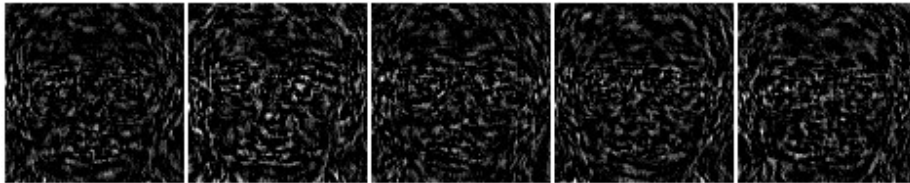


Figure 7: Blood : Eigenfaces 301 to 305

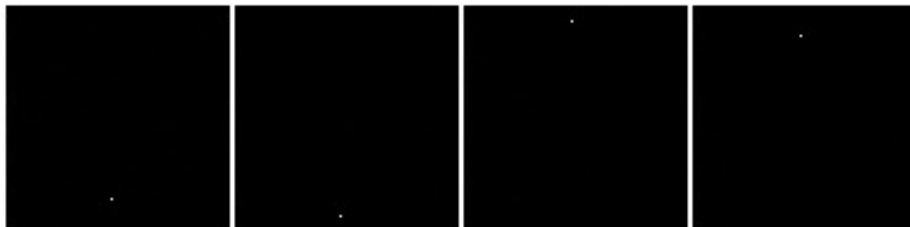


Figure 8: Pixel-faces : Eigenfaces beyond 400