Module 6.7 : PCA : Practical Example

- Consider we are given a large number of images of human faces (say, m images)

- Consider we are given a large number of images of human faces (say, m images)
- Each image is 100×100 [10K dimensions]

- Consider we are given a large number of images of human faces (say, m images)
- Each image is 100×100 [10K dimensions]
- We would like to represent and store the images using much fewer dimensions (around 50-200)

- Consider we are given a large number of images of human faces (say, m images)
- Each image is 100×100 [10K dimensions]
- We would like to represent and store the images using much fewer dimensions (around 50-200)
- We construct a matrix $X \in \mathbb{R}^{m \times 10K}$

- Consider we are given a large number of images of human faces (say, m images)
- Each image is 100×100 [10K dimensions]
- We would like to represent and store the images using much fewer dimensions (around 50-200)
- We construct a matrix $X \in \mathbb{R}^{m \times 10K}$
- Each row of the matrix corresponds to 1 image

- Consider we are given a large number of images of human faces (say, m images)
- Each image is 100×100 [10K dimensions]
- We would like to represent and store the images using much fewer dimensions (around 50-200)
- We construct a matrix $X \in \mathbb{R}^{m \times 10K}$
- Each row of the matrix corresponds to 1 image
- Each image is represented using 10K dimensions

- $X \in \mathbb{R}^{m \times 10K}$ (as explained on the previous slide)

- $X \in \mathbb{R}^{m \times 10K}$ (as explained on the previous slide)
- We retain the top 100 dimensions corresponding to the top 100 eigen vectors of $X^T X$

- $X \in \mathbb{R}^{m \times 10K}$ (as explained on the previous slide)
- We retain the top 100 dimensions corresponding to the top 100 eigen vectors of $X^T X$
- Note that $X^T X$ is a $n \times n$ matrix so its eigen vectors will be n dimensional ($n = 10K$ in this case)

- $X \in \mathbb{R}^{m \times 10K}$ (as explained on the previous slide)
- We retain the top 100 dimensions corresponding to the top 100 eigen vectors of $X^T X$
- Note that $X^T X$ is a $n \times n$ matrix so its eigen vectors will be n dimensional ($n = 10K$ in this case)
- We can convert each eigen vector into a 100×100 matrix and treat it as an image

- $X \in \mathbb{R}^{m \times 10K}$ (as explained on the previous slide)
- We retain the top 100 dimensions corresponding to the top 100 eigen vectors of X^TX
- Note that X^TX is a $n \times n$ matrix so its eigen vectors will be n dimensional ($n = 10K$ in this case)
- We can convert each eigen vector into a 100×100 matrix and treat it as an image
- Let's see what we get

- $X \in \mathbb{R}^{m \times 10K}$ (as explained on the previous slide)
- We retain the top 100 dimensions corresponding to the top 100 eigen vectors of $X^T X$
- Note that $X^T X$ is a $n \times n$ matrix so its eigen vectors will be n dimensional ($n = 10K$ in this case)
- We can convert each eigen vector into a 100×100 matrix and treat it as an image
- Let's see what we get
- What we have plotted here are the first 16 eigen vectors of $X^T X$ (basically, treating each 10K dimensional eigen vector as a 100×100 dimensional image)

- These images are called eigenfaces and form a basis for representing any face in our database

- These images are called eigenfaces and form a basis for representing any face in our database
- In other words, we can now represent a given image (face) as a linear combination of these eigen faces

- These images are called eigenfaces and form a basis for representing any face in our database
- In other words, we can now represent a given image (face) as a linear combination of these eigen faces

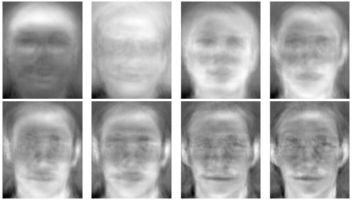
$$\sum_{i=1}^{1} \alpha_{1i} p_i$$

$$\sum_{i=1}^{2} \alpha_{1i} p_i$$

- These images are called eigenfaces and form a basis for representing any face in our database
- In other words, we can now represent a given image (face) as a linear combination of these eigen faces

- These images are called eigenfaces and form a basis for representing any face in our database
- In other words, we can now represent a given image (face) as a linear combination of these eigen faces

$$\sum_{i=1}^{4} \alpha_{1i} p_i$$



- These images are called eigenfaces and form a basis for representing any face in our database
- In other words, we can now represent a given image (face) as a linear combination of these eigen faces

$$\sum_{i=1}^{8} \alpha_{1i} p_i$$

$$\sum_{i=1}^{12} \alpha_{1i} p_i$$

- These images are called eigenfaces and form a basis for representing any face in our database
- In other words, we can now represent a given image (face) as a linear combination of these eigen faces

$$\sum_{i=1}^{16} \alpha_{1i} p_i$$

- These images are called eigenfaces and form a basis for representing any face in our database
- In other words, we can now represent a given image (face) as a linear combination of these eigen faces

$$\sum_{i=1}^{16} \alpha_{1i} p_i$$

- These images are called eigenfaces and form a basis for representing any face in our database
- In other words, we can now represent a given image (face) as a linear combination of these eigen faces
- In practice, we just need to store $p_1, p_2, \cdots, p_k$ (one time storage)

$$\sum_{i=1}^{16} \alpha_{1i} p_i$$

- These images are called eigenfaces and form a basis for representing any face in our database
- In other words, we can now represent a given image (face) as a linear combination of these eigen faces
- In practice, we just need to store $p_1, p_2, \cdots, p_k$ (one time storage)
- Then for each image i we just need to store the scalar values $\alpha_{i1}, \alpha_{i2}, \cdots, \alpha_{ik}$

$$\sum_{i=1}^{16} \alpha_{1i} p_i$$

- These images are called eigenfaces and form a basis for representing any face in our database
- In other words, we can now represent a given image (face) as a linear combination of these eigen faces
- In practice, we just need to store $p_1, p_2, \cdots, p_k$ (one time storage)
- Then for each image i we just need to store the scalar values $\alpha_{i1}, \alpha_{i2}, \cdots, \alpha_{ik}$
- This significantly reduces the storage cost without much loss in image quality