

# MP2: Vision Based Person Identification (Face Recognition)

ECE 417 – Multimedia Signal Processing  
University of Illinois at Urbana-Champaign  
Spring 2014

Instructor: Professor Hasegawa-Johnson

TA's: Daniel Soberal, Po-Sen Huang

Prepared by Po-Sen Huang

# Goal

- Person identification system
- Feature extraction
  - Raw pixel
  - PCA
  - Random projection
- Classification
  - Use K-nearest neighbor for classification

# Data

- 80 different face images
  - 4 different people, 20 images for each person
- Format: [Person][Instance].jpg
- Example:



# Read Images in MATLAB

- Read image files in MATLAB: `imread`
- Work with grayscale images: `rgb2gray`
- Work with `double` type
- Example:  

```
Img=double(rgb2gray(imread('A1.jpg')));
```

# Experiments

- Feature Extraction
  - Raw pixel
  - PCA
  - Random projection
- Classification
  - Nearest neighbor
  - 5-Nearest neighbor

# Feature – Raw Pixel

- Each 90x70 image
- 1. Use “reshape” function to convert it into a 6300x1 vector
  - Ex: `Fea=reshape( Img, [6300, 1] );`
- 2. Resize image to 45x35, 22x17
  - Ex: `Img=imresize( Img, [45, 35] );`
- 3. Pick a pair of N1 and N2 so that N1xN2 is roughly equal to the choice of N in PCA. (the ratio between N1 and N2 is unchanged)

# Feature - PCA

- Implement PCA (not using built-in PCA function)
- Choose  $N$  principal components such that 95% of the total energy is kept
- Project  $6300 \times 1$  image vector onto the PCA subspace, resulting in a  $N \times 1$  vector as a feature

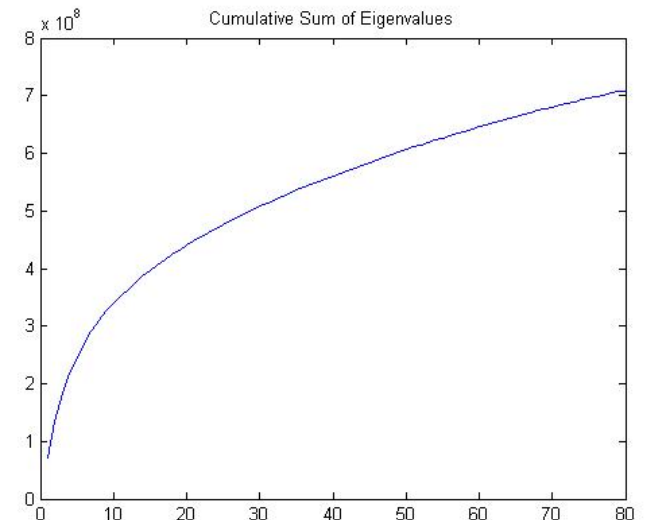
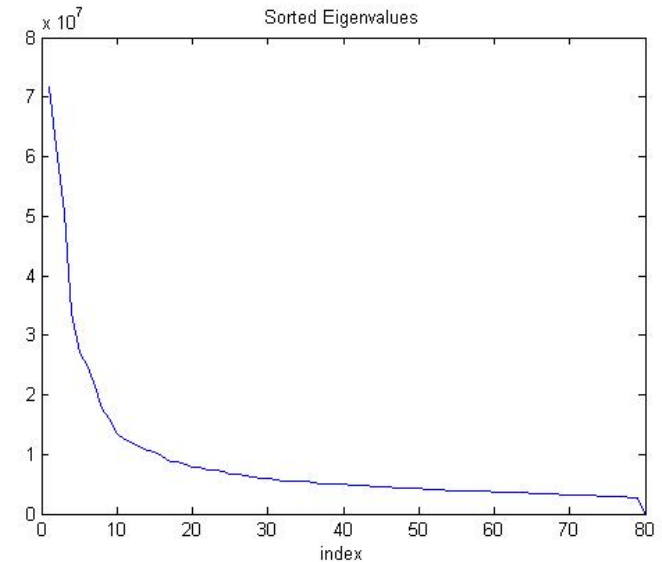
# Implement PCA (I)

- 1. Subtract mean w.r.t. each pixel
  - MATLAB function: `mean, repmat/bsxfun`
- 2. Compute the scatter matrix
  - $S=X' * X$ , where  $X$  the zero-mean 6300x80 matrix,  $S$  is 80x80
- 3. Compute eigenvalue of the scatter matrix
  - MATLAB function: `eig`
  - $[V, D]=\text{eig}(A)$ ,  $V$  is the matrix of eigenvectors,  $\text{diag}(D)$  are eigenvalues



# Implement PCA (II)

- 4. Sort eigenvalues from high to low
  - `[sorted_values, ori_index]=sort(diag(D), 'descend');`
- 5. Find N principal components to keep K% of the total energy
- 6. Projection data on selected eigenvectors



# Feature – Random Projection

- Use `randn` to generate a projection matrix of size  $6300 \times N$
- Project each  $6300 \times 1$  image onto the random subspace, resulting in a  $N \times 1$  vector as a feature
- Select  $N$  to be the same as in PCA case

# Experiment

- Use nearest neighbor, 5-nearest neighbor for classification
- Validation number:
  - Raw features  $6300 \times 1$  + nearest neighbor: 88.75% accuracy
  - PCA (95% energy) + nearest neighbor: 96.25% accuracy