MP2 Walkthrough & Q/A

- MP2 goals
  - Practice applying Discrete Cosine Transform (DCT) and Principal Component Analysis (PCA) techniques on image analysis
  - Apply K-nearest-neighbor (KNN) techniques on image classification

- **Problem 2.0**
  - Flatten all given images from 2D array to 1D array in row-major (‘C’) order
  - Stack them together
- **Problem 2.1**
  - Calculate the mean of each pixel across all the images and put it in a vector
- **Problem 2.2**
  - Calculate centered matrix by subtracting the mean vector from each image vector

- **Problem 2.3**
  - Calculate transformed matrix for DCT and PCA
  - DCT
    - $basis[k1, n1, nrows] = \frac{D}{\sqrt{nrows}} \cos(\pi(n1 + 0.5) * \frac{k1}{nrows})$; $D = 1\ if\ k1 = 0\ and\ D = \sqrt{2}\ otherwise$
    - $transform[ktot, ntot] = basis[k1, n1, nrows] * basis[k2, n2, ncols]$
    - $ktot = k1 + k2$ (k1 and k2 should be scanned in L2R diagonal order, first increase k1 until $k1 + k2 = \sqrt{nfeats}$, then increase k2)
    - $ntot = n1 * ncols + n2$

- **Problem 2.3**
  - PCA
    - First method (Recommended)
      - Take SVD of the centered matrix using np.linalg.svd ($X = USV^T$)
      - Take first nfeats of eigenvectors of V^T
    - Second method
      - Do eigenanalysis on gram matrix (XX^T) using np.linalg.eig and obtain U and S ($XX^T = US^2U^T$)
      - Calculate the eigenvectors from U ($V = X^TUS^{-1}$)
      - Take first nfeats eigenvectors of V

- **Problem 2.3**
  - Eigenvectors generated from np.svd are put in rows and sort by eigenvalues in descending order
  - Eigenvectors generated from np.eig are put in columns and not sorted in any order
  - Consult the numpy documents for more information
  - Since the signs of eigenvectors calculated from numpy library are arbitrary, remember to take the opposite sign of eigenvector if the first element of eigenvector is negative
    - $if\ transform[k,0] < 0$

      $transform[k,:] = -transform[k,:]$

- **Problem 2.3**

  Q: Can I calculate PCA transform matrix by computing eigenvectors directly from covariance matrix of the centered matrix?

  A: Theoretically you can, but it most likely will time out autograder. So it’s not recommended.

- **Problem 2.4**
  - Apply transform matrix on centered matrix to obtain features matrix
- **Problem 2.5**
  - Calculate energy spectrum
  - First calculate total energy of the centered matrix ($E_{tot} = \sum_i ||x(i)||^2$)
  - Then calculate the fraction of the energy of each feature vector y
    - $\mathrm{E}_0 = \sum_i \left||y_0(i)|\right|^2 / E_{tot}$
    - $\mathrm{E}_1 = (\sum_i \left||y_0(i)|\right|^2 + \sum_i \left||y_1(i)|\right|^2) / E_{tot}$
  - Sum up the fraction of all feature vectors along the feature dimension up to nfeats

- **Problem 2.6**
  - Apply KNN techniques on each feature vectors
  - Calculate the closest K neighbors for each feature vector using squared distance metric excluding feature vector itself
  - Store the indexes of these neighbors
- **Problem 2.7**
  - Set hypothesis for each feature vector for majority vote of labels of its closest K neighbors
  - For K>1, go back to the case of K=1 when tie occurs

- **Problem 2.8**
  - Compute confusion matrix of classification results
  - Confusion[r,h]=# of images of person r that were classified as person h
- **Problem 2.9**
  - Calculate accuracy, recall, and precision of KNN results, which recall and precision are the average rates across all people
  - Accuracy = # of people that are correctly predicted/# of total people that are predicted
  - Recall for person p = # of person p who are correctly predicted/ # of person p are present in dataset
  - Precision for person p = # of person p who are correctly predicted/ # of person p are predicted

More questions?

- Ask me now
- Office hour:
  - Thursday(9/19) 4-6pm ECEB 2036
- Post questions on Piazza