One Face, Many Vectors

Ranil Weerackoon
Joseph Swabeck
Jason Pillon
**Introduction**

- **Purpose:** to develop a scheme for facial recognition and get important info on a face image
- **Goal:** to encode info from a group of faces to distinguish them from one another
- **Result:** transforms face images into eigenfaces \( \Rightarrow \) the training set of face images
- **Overview:** eigenfaces encode and decode face images that gives info on the content of face features
The Math behind the Eigenface

- Eigenvectors of the covariance matrix form main components of the face distributions
  - Basically, it characterizes the variation in face images
- Eigenvectors are created by each image location
- Eigenvectors together form the eigenface
- Face images in the training set are represented in a linear combination of the eigenfaces
- Number of possible eigenfaces equals number of face images in training set
  - Use fewer eigenfaces for computation efficiency
    => Want to compute cheaply and quickly!
Checking for Eigenfaces

- Eigenfaces span an m-dimensional subspace of the image space from the m largest eigenvalues => creates the face space.

- For new faces/images:
  - Calculate the weights of the new image and the amount of eigenfaces you already have and project the new image onto each eigenface.
  - Determine if the image is a face and check to see if it is the face space.
  - If found to be a face, you can classify the weight pattern to see if it matches that of a face you already have.
This matrix $y$ represents a face of $nxn$ pixels.

The concatenation process takes the full face matrix and condenses it into a single vector in $\mathbb{R}^{nn}$. 

$$y_1 = \begin{bmatrix} y_{11} & \cdots & y_{1n} \\ \vdots & \ddots & \vdots \\ y_{n1} & \cdots & y_{nn} \end{bmatrix}$$

a face in $n \times n$ space

$$y_1 \xrightarrow{\text{concatenation}} f_1 = \begin{bmatrix} y_{11} \\ \vdots \\ y_{n1} \\ \vdots \\ y_{nn} \end{bmatrix} \in \mathbb{R}^{nn}$$
\[ x = [f_1 \ f_2 \ \ldots \ f_p] \]

\[ z = x - m \]

\[ m = \text{mean}(x) \]

- \(x\) is the “Training Set,” which consists of \(p\) face matrices \(f\).
- This represents removing the mean face from the training set, which sets them to a common origin.
By taking $x^T x$, a matrix is formed that contains the inner products for all $f$.

The resulting matrix shows a difference of each matrix from the others. If two matrices are the same, their entry will be 0.
The eigenvalues and eigenvectors of the $x^T x$ matrix are found and ordered. The amount of vectors and values are usually truncated to a reasonable number.

$$eig(x^T x) = \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$$

$$v_1 \quad v_2 \quad \ldots \quad v_n$$
The face space is the space that is spanned by the eigenvectors derived from the face set, or eigenfaces, since these vectors represent faces.

\[ \text{span} \{ v_1, \ldots, v_n \} = \text{face space} \]
Using these ideas, any new face could be interpreted in a similar way to the previous faces to find if it matches a known face.
By projecting the new face into face space, the difference in the projections can be calculated and compared with known faces. The closest match (smallest value) is probably the matching face.

\[
f_{\text{new}} \in \text{face space}
\]

\[
\text{proj}_{\text{face space}}f_{\text{new}} = [p_{\text{new}}]
\]

\[
[p_{\text{new}}] - [p_i] = [\text{value}]
\]

where \([p_i] = \text{proj}_{\text{face space}}f_i\)
Results

- Each person has a collection of pictures which has an optimal basis for the eigenvectors of the face.

- A photo database of people is developed and if a picture is taken its eigenvectors can be compared to the bases in the database and the deviation can be used to establish a close or exact match of subject.
Summary/Conclusion

- Face recognition is based on a set image features that approximate known face images the best
  - Eigenfaces present the best solution to determining face recognition
  - Reasons why: it’s fast, simple, and works well
  - Eigenfaces can be used also for face detection and image compression (to save space)
Acknowledgements

Mean Image
Deviations From Mean Image