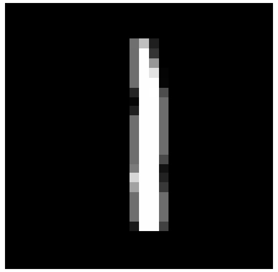
PCA for Eigenface

Represent an image by a vector



This image has 28×28 pixels.

It is a matrix/ 2D array $A \in \mathbb{R}^{28\times 28}$

$$A=\begin{array}{cc} \begin{bmatrix} A_{0,0} & \cdots & A_{0,27} \\ \cdots & \cdots & \cdots \\ A_{27,0} & \cdots & A_{27,27} \end{bmatrix} & \begin{array}{c} \text{row-0} \\ \\ \text{row-27} \end{array} \end{array}$$

$$\boldsymbol{x} = \begin{bmatrix} A_{0,0} \\ A_{0,1} \\ A_{0,2} \\ \cdots \\ \cdots \\ \cdots \\ \cdots \\ \cdots \\ \cdots \\ \cdots \\ A_{27,27} \end{bmatrix}$$

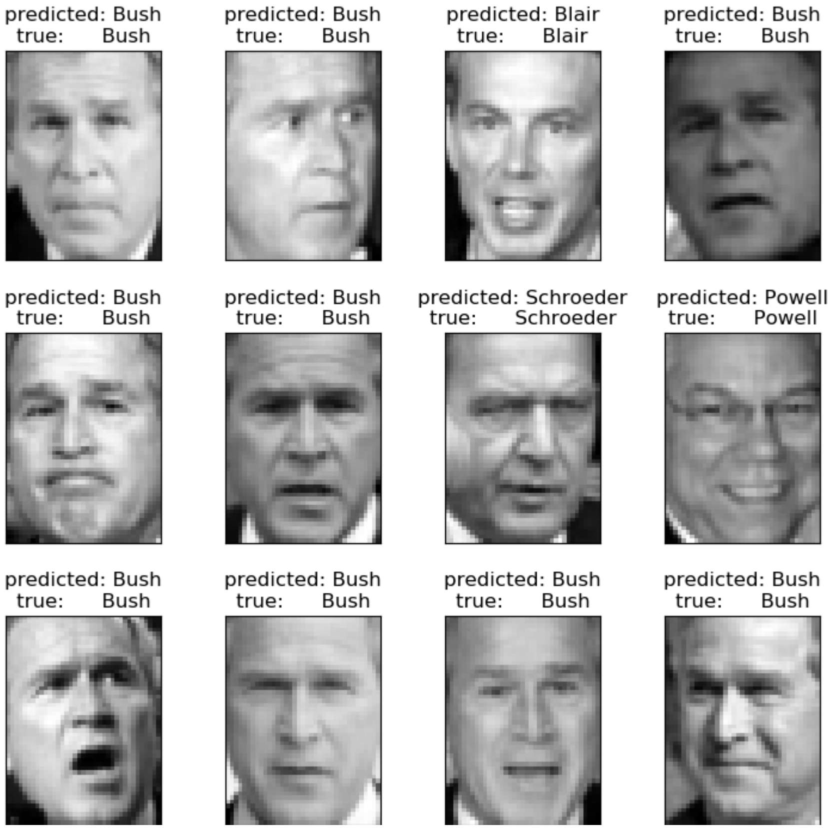
the first row

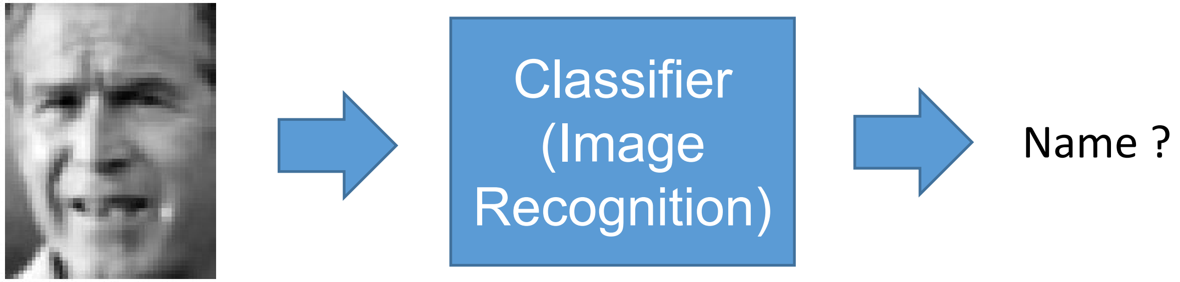
the second row

$\boldsymbol{x} \in \mathbb{R}^{784}$

a vector ~an image ~ a data sample

A Motivating Example - Eigenface

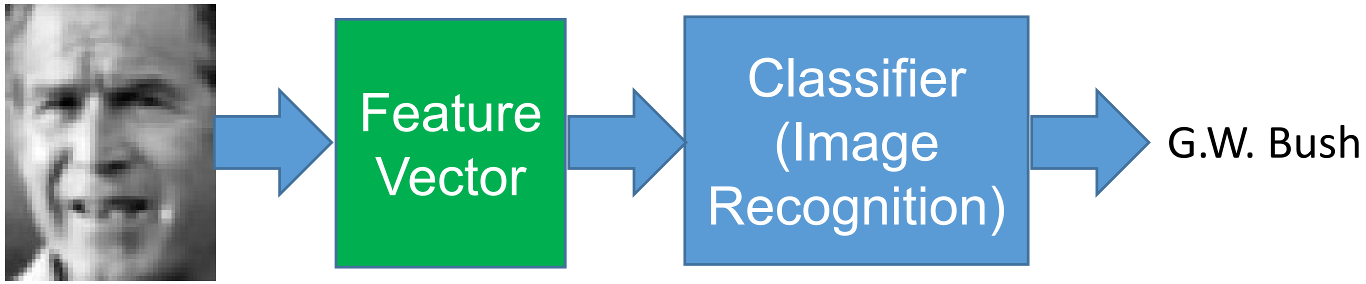




This image has 432x288 pixels

Not every pixel is equally important in classifying faces

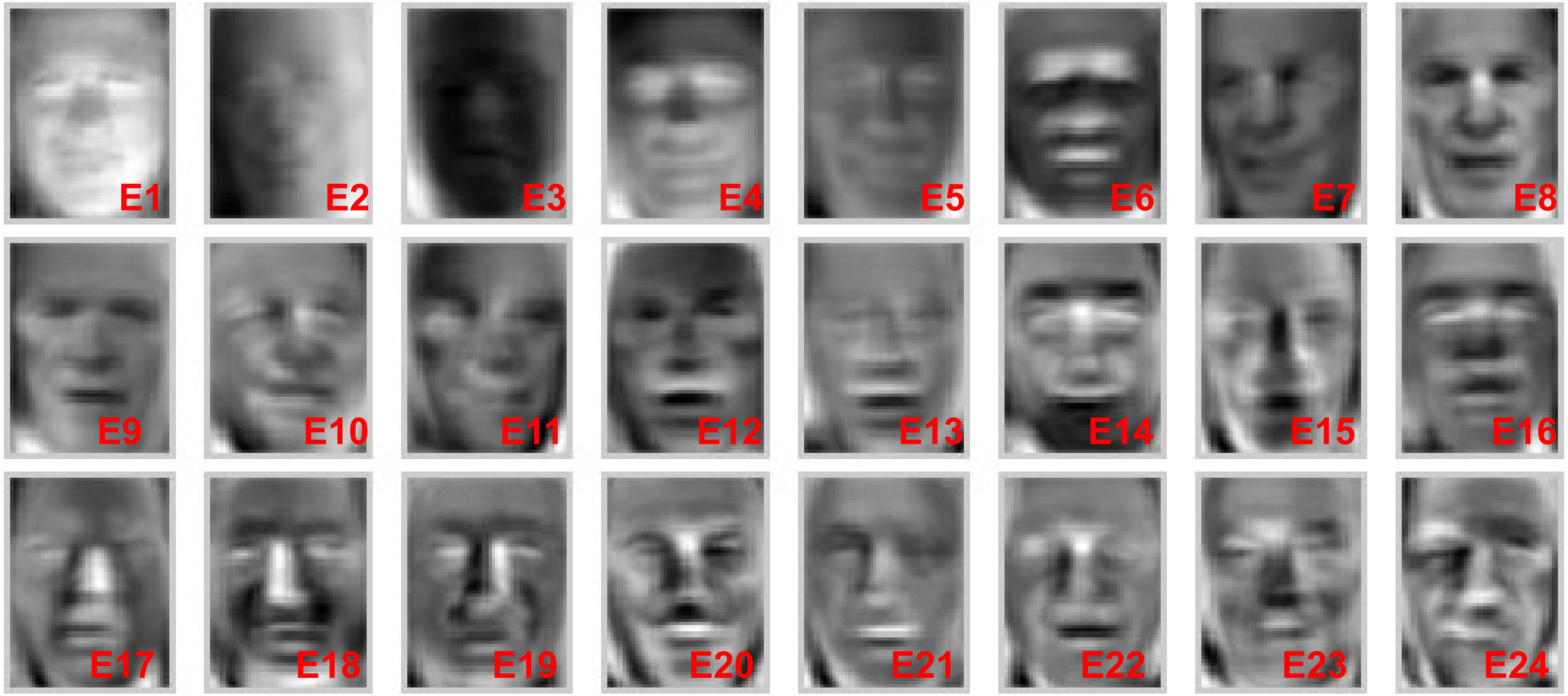
Solution: use ideas from linear algebra to extract features from images - using eigenvectors, eigenvalues



Eigenfaces (using linear algebra)

mean face

E0

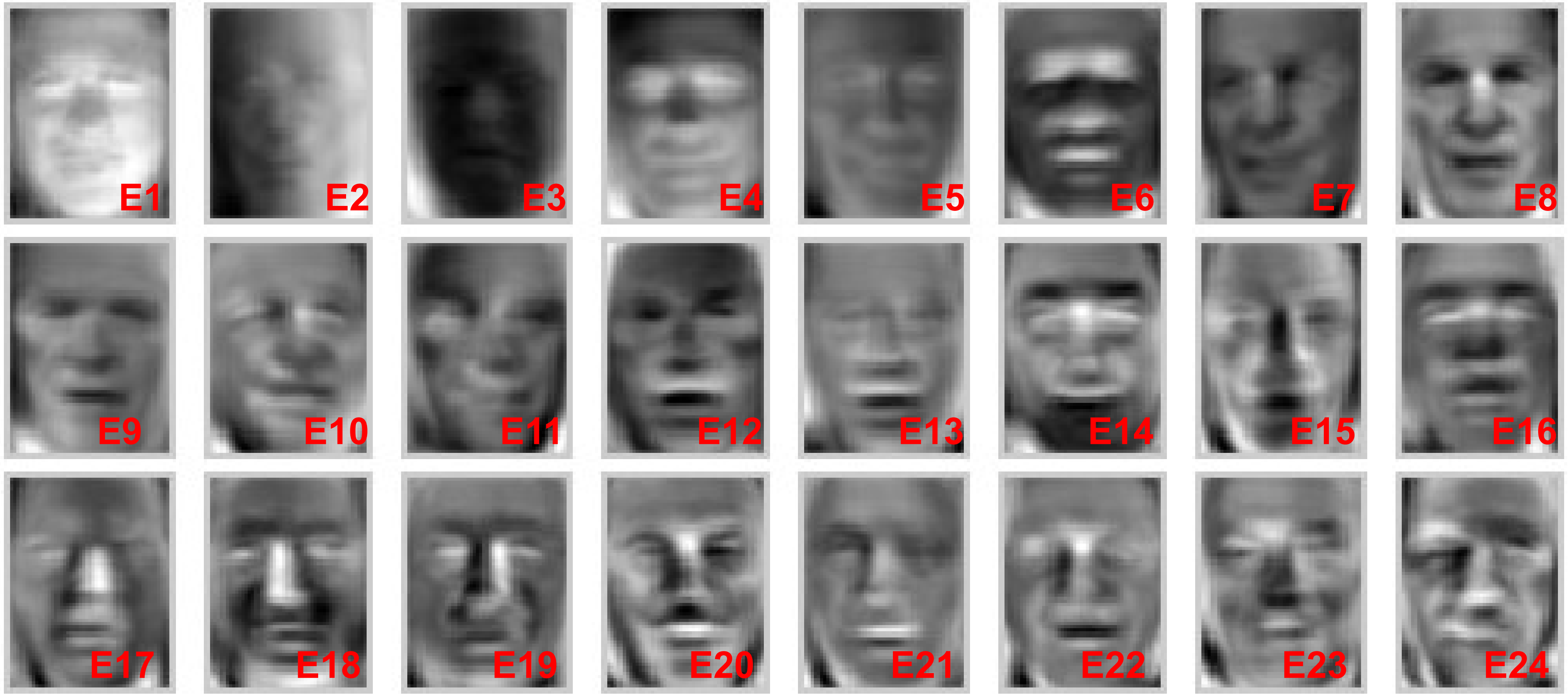


$= \mathbf{E0} + (-0.005)\times\mathbf{E1} + (-0.04)\times\mathbf{E2} + (0.002)\times\mathbf{E3} + \ldots$

Eigenfaces (using linear algebra)

mean face

E0



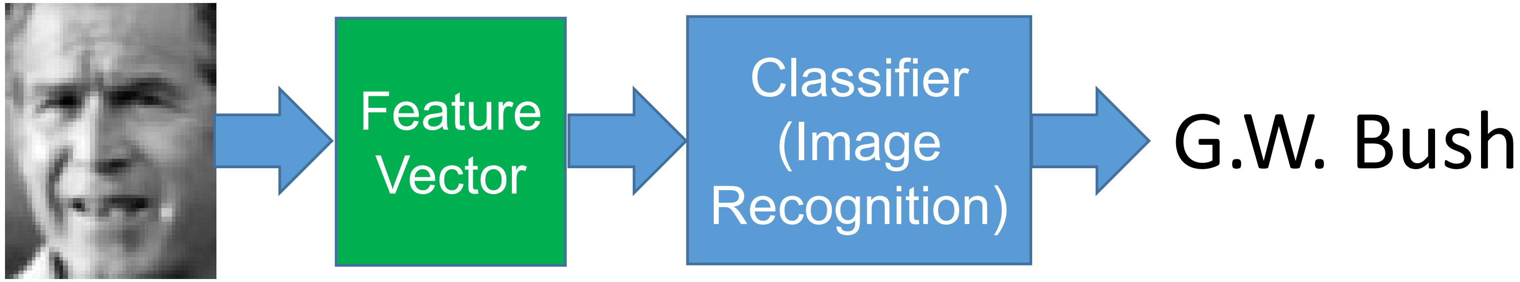
$$= \mathbf{E0} + (-0.015)\times\mathbf{E1} + (0.0037)\times\mathbf{E2} + (-0.01)\times\mathbf{E3} + \ldots$$

$$= E0 + (-0.015)\times E1 + (0.0037)\times E2 + (-0.01)\times E3 + \ldots$$

Feature Vector = [-0.015, 0.0037, -0.01, …]

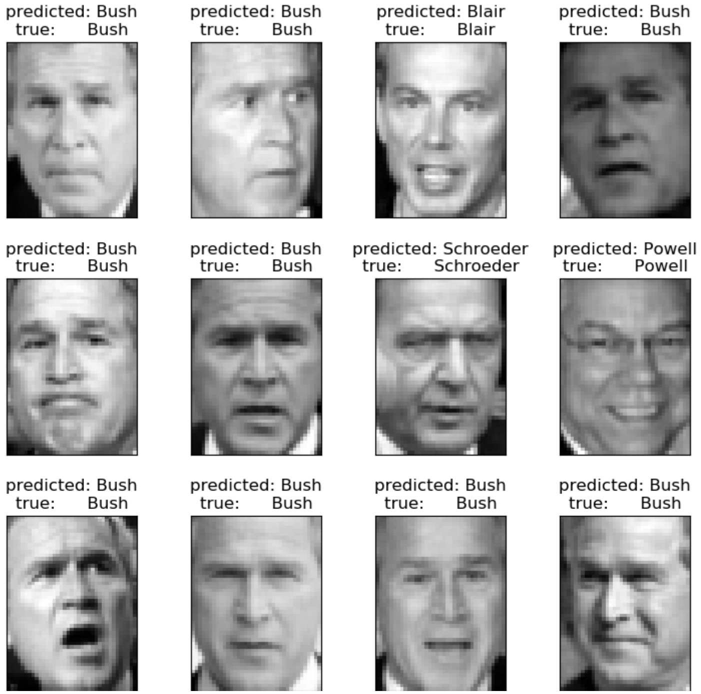
$$= E0 + (-0.005)\times E1 + (-0.04)\times E2 + (0.002)\times E3 + \ldots$$

Feature Vector = [-0.005, -0.04, 0.002, …]



Use PCA to Obtain Eigenfaces

image $x_n \in \mathcal{R}^{2914}$



mean face

μ

The eigenfaces are eigenvectors $\{w_k\}$ of the covariance matrix C of the data

Encoding: $\beta_{n,i} = {w_i}^T(x_n - \mu)$

Decoding: $x_n \approx \beta_{n,1}w_1 + \beta_{n,2}w_2 \ldots + \beta_{n,K}w_K + \mu$

PCA: the inverse transform from y_n or β_n to $\tilde{x}_n$

- x_n can be approximated by a linear combination of the eigenvectors:

$$x_n \approx \tilde{x}_n = \beta_{n,1} w_1 + \beta_{n,2} w_2 \dots + \beta_{n,K} w_K + \mu$$

$$x_n \approx \tilde{x}_n = y_{n,1}\sqrt{\lambda_1} w_1 + y_{n,2}\sqrt{\lambda_2} w_2 \dots + y_{n,K}\sqrt{\lambda_K} w_K + \mu$$

- $\beta_n = [\beta_{n,1} \quad \dots \quad \beta_{n,K}]^T$ where $\beta_{n,k} = w_k^T\,(x_n - \mu)$ is a scalar
- $y_n = [y_{n,1} \quad \dots \quad y_{n,K}]^T$ where $y_{n,k} = \beta_{n,k}/\sqrt{\lambda_k}$
- $y_n\ and\ \beta_n \in \mathcal{R}^K$ are two "new" feature vectors
- y_n is the normalized feature vector $y_n = \sqrt{D^{-1}}\beta_n$ and $\beta_n = \sqrt{D}\, y_n$

$\sqrt{D}$=diag($\{\lambda_1^{1/2}, \dots, \lambda_K^{1/2}\}$) and $\sqrt{D^{-1}}$=diag($\{\lambda_1^{-1/2}, \dots, \lambda_K^{-1/2}\}$)