



**University Of Technical Education
Ho Chi Minh City**

Nhận dạng đối tượng dùng PCA và Neural Network

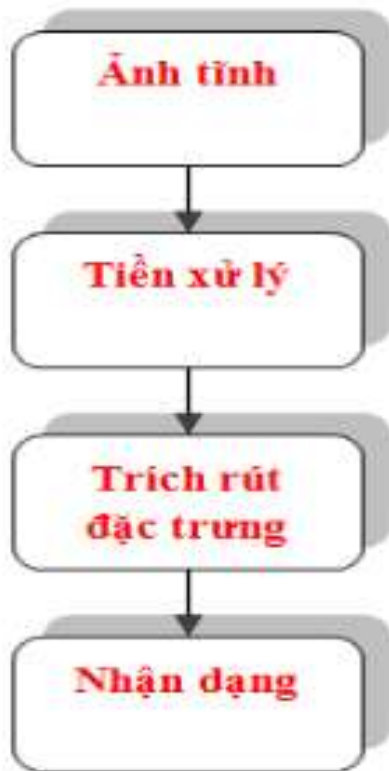


1. Tổng quan về nhận dạng đối tượng

- Nhận dạng đối tượng là được ứng dụng rộng rãi trong đời sống hằng ngày như các hệ thống giám sát, quản lý, tìm kiếm thông tin, điều khiển tự động.
- Có nhiều phương pháp nhận dạng:
 - + Nhận dạng dựa trên đặc trưng của các đối tượng: SIFT, SURF
 - + Mạng Neural, SVM
 - + Nhận dạng dựa vào phương pháp phân tích thành phần chính PCA, LDA, LF.



2. Xây dựng hệ thống nhận dạng

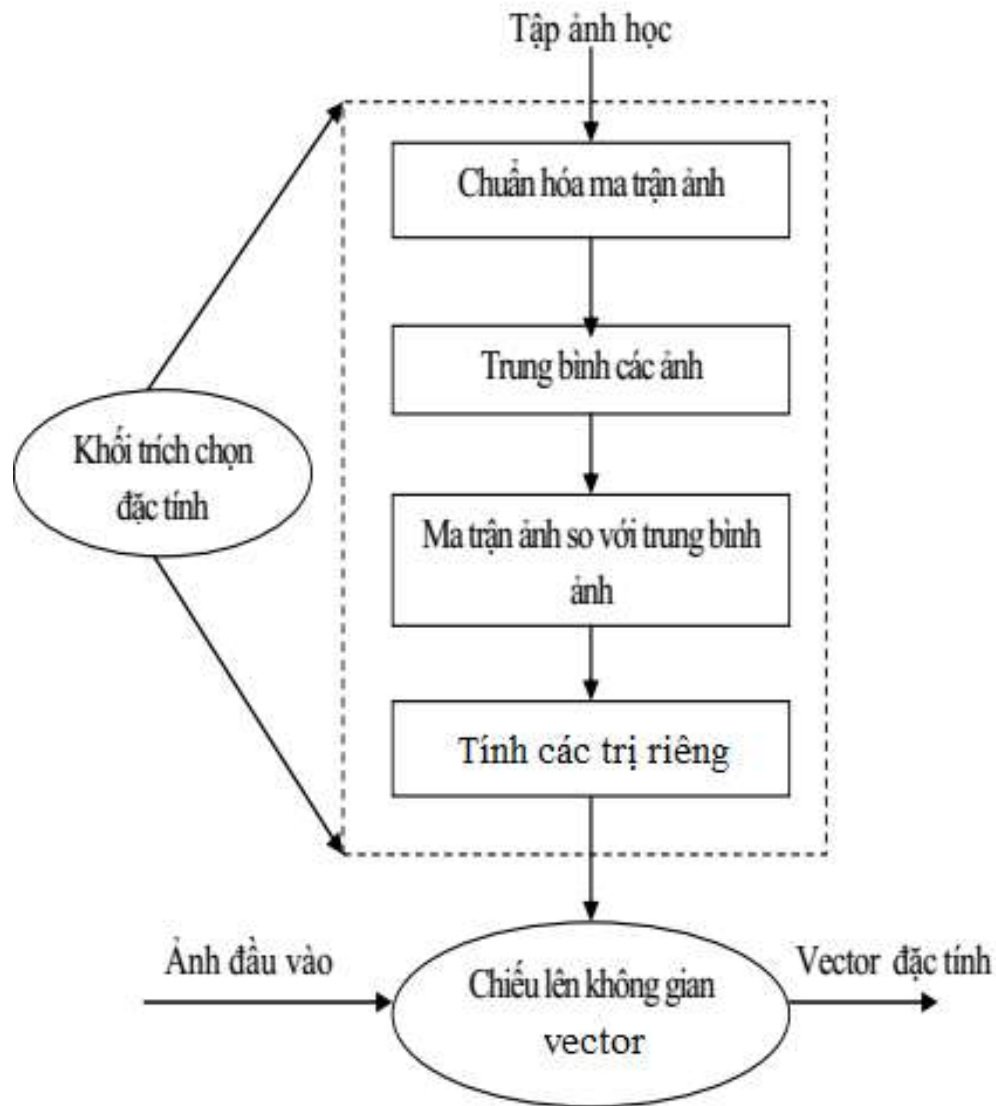


Hình 1. Hệ thống nhận dạng khuôn mặt cơ bản

- **Tiền xử lý:** nhằm nâng cao chất lượng ảnh, chuẩn hóa dữ liệu, kích thước ảnh.
- **Trích rút đặc trưng:** sử dụng các thuật toán để lấy ra những thông tin mang những đặc điểm riêng biệt của đối tượng.
- **Nhận dạng:** phân loại đối tượng dùng khoảng cách Euclides, Mahalanobis, dùng mạng nơron



3. Trích đặc trưng dùng PCA



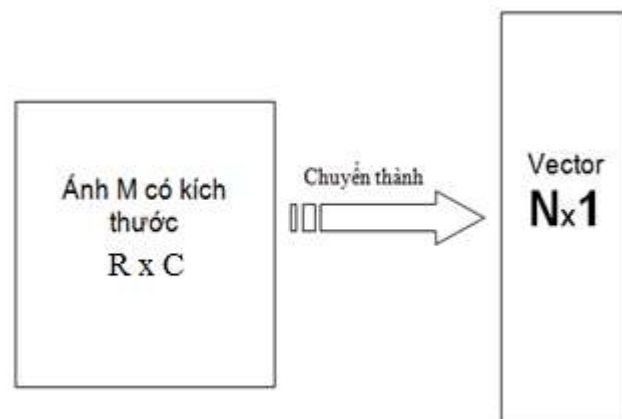
Hình 2. Quá trình trích rút đặc trưng dùng PCA



3. Trích đặc trưng dùng PCA

❖ *Tạo một tập ảnh học*

- Tạo một tập **S** gồm m ảnh kích thước $R \times C$.
- Mỗi ảnh được chuyển thành một vector ($N = R \times C$) chiều.



$$\mathbf{S}_i = [a_1 a_2 \dots a_n]^T, \quad i = \overline{1, m}$$

$$\mathbf{S} = \{\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_m\}$$

Hình 3 Biểu diễn ảnh thành vector



3. Trích đặc trưng dùng PCA

❖ *Tính ảnh trung bình*

$$\mathbf{M} = \frac{1}{m} \sum_{i=0}^m x_i \quad i = \overline{1, m}$$

Hay:

$$\mathbf{M} = \frac{1}{m} \begin{pmatrix} a_{11} + a_{12} + a_{13} + \dots + a_{1m} \\ a_{21} + a_{22} + a_{23} + \dots + a_{2m} \\ a_{31} + a_{32} + a_{33} + \dots + a_{3m} \\ \dots \\ a_{n1} + a_{n2} + a_{n3} + \dots + a_{nm} \end{pmatrix} = \begin{pmatrix} m_1 \\ m_2 \\ m_2 \\ \dots \\ m_n \end{pmatrix}$$



3. Trích đặc trưng dùng PCA

❖ **Tính sai lệch của các ảnh đầu vào so với trung bình**

- Nhằm tạo ra sự giãn tương đối giá trị pixel của các ảnh

$$\mathbf{n}_1 = \begin{pmatrix} a_{11} - m_1 \\ a_{21} - m_2 \\ a_{31} - m_3 \\ \dots \\ a_{n1} - m_n \end{pmatrix}, \quad \mathbf{n}_2 = \begin{pmatrix} a_{12} - m_1 \\ a_{22} - m_2 \\ a_{32} - m_3 \\ \dots \\ a_{n2} - m_n \end{pmatrix}, \dots, \quad \mathbf{n}_m = \begin{pmatrix} a_{1m} - m_1 \\ a_{2m} - m_2 \\ a_{3m} - m_3 \\ \dots \\ a_{nm} - m_n \end{pmatrix}$$

- Xây dựng ma trận từ các vector tìm được

$$\mathbf{A} = \{\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_m\}$$



3. Trích đặc trưng dùng PCA

❖ *Tính ma trận hiệp phương sai*

$$\mathbf{C} = \mathbf{A} \cdot \mathbf{A}^T$$

- Vì ma trận \mathbf{C} có kích thước quá lớn ($N \times N$) nên để tìm eigenvector của \mathbf{C} ta tìm eigenvector và eigenvalue của ma trận \mathbf{L} :

$$\mathbf{L} = \mathbf{A}^T \mathbf{A}$$

- Trong Matlab dùng lệnh *$\text{eig}(\mathbf{L})$* để tìm eigenvalue của \mathbf{L} .
- Trị riêng (eigenvalue λ_i), và vector riêng (eigenvector x_i) của ma trận hợp phương sai này chính là đặc trưng thành phần thiết yếu của ảnh.



3. Trích đặc trưng dùng PCA

❖ Ví dụ:

$$X1 = \begin{bmatrix} 1 \\ 2 \\ 5 \\ 4 \\ 7 \\ 8 \\ 9 \\ 4 \\ 11 \end{bmatrix}$$

$$X2 = \begin{bmatrix} 12 \\ 1 \\ 13 \\ 1 \\ 5 \\ 2 \\ 17 \\ 7 \\ 21 \end{bmatrix}$$

$$X3 = \begin{bmatrix} 17 \\ 21 \\ 3 \\ 16 \\ 24 \\ 32 \\ 10 \\ 1 \\ 1 \end{bmatrix}$$

$$Y = \frac{1}{M} \sum_{i=1}^M X_i \quad Y = \begin{bmatrix} 10 \\ 8 \\ 7 \\ 7 \\ 12 \\ 14 \\ 12 \\ 4 \\ 11 \end{bmatrix}$$



3. Trích đặc trưng dùng PCA

❖ Ví dụ: $\theta_i = X_i - Y$

$$\theta_1 = \begin{bmatrix} -9 \\ -6 \\ -2 \\ -3 \\ -5 \\ -6 \\ -3 \\ 0 \\ 0 \end{bmatrix}$$

$$\theta_2 = \begin{bmatrix} 2 \\ -7 \\ 6 \\ -6 \\ -7 \\ -12 \\ 5 \\ 3 \\ 10 \end{bmatrix}$$

$$\theta_3 = \begin{bmatrix} 7 \\ 13 \\ -4 \\ 9 \\ 12 \\ 18 \\ -2 \\ -3 \\ 10 \end{bmatrix}$$

$$C = \frac{1}{M-1} AA^T \quad A = [\theta_1, \theta_2, \theta_3]$$

$$A^T = \begin{bmatrix} -9 & -6 & -2 & -3 & -5 & -6 & -3 & 0 & 0 \\ 2 & -7 & 6 & -6 & -7 & -12 & 5 & 3 & 10 \\ 7 & 13 & -4 & 9 & 12 & 18 & -2 & -3 & 10 \end{bmatrix}$$

$$A = \begin{bmatrix} -9 & 2 & 7 \\ -6 & -7 & 13 \\ -2 & 6 & -4 \\ -3 & -6 & 9 \\ -5 & -7 & 12 \\ -6 & -12 & 18 \\ -3 & 5 & -2 \\ 0 & 3 & -3 \\ 0 & 10 & 10 \end{bmatrix}$$

$$A^T A = \begin{bmatrix} 200 & 122 & -322 \\ 122 & 452 & -374 \\ -322 & -374 & 896 \end{bmatrix}$$



3. Trích đặc trưng dùng PCA

❖ **Ví dụ:** eigenvectors of square matrix $A^T A$.

$$\det(A - \lambda I) = 0 \text{ with } I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \Leftrightarrow \det \left(\begin{bmatrix} 200 & 122 & -322 \\ 122 & 452 & -374 \\ -322 & -374 & 896 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) = 0$$

$$\Leftrightarrow \begin{bmatrix} 200 - \lambda & 122 & -322 \\ 122 & 452 - \lambda & -374 \\ -322 & -374 & 896 - \lambda \end{bmatrix} = 0$$

$$D = (200 - \lambda) \times [(452 - \lambda) \times (896 - \lambda) - (-374) \times (-374)] - 122 \times [122 \times (896 - \lambda) - (-374) \times (-322)] + (-322) \times [122 \times (-374) - (-322) \times (452 - \lambda)]$$

The eigenvalues (λ) are roots of equation D and the eigenvectors are calculated by:

$$\begin{cases} (200 - \lambda)x_1 + 122x_2 - 322x_3 = 0 \\ 122x_1 + (452 - \lambda)x_2 - 374x_3 = 0 \\ -322x_1 - 374x_2 + (896 - \lambda)x_3 = 0 \end{cases} \quad (*)$$

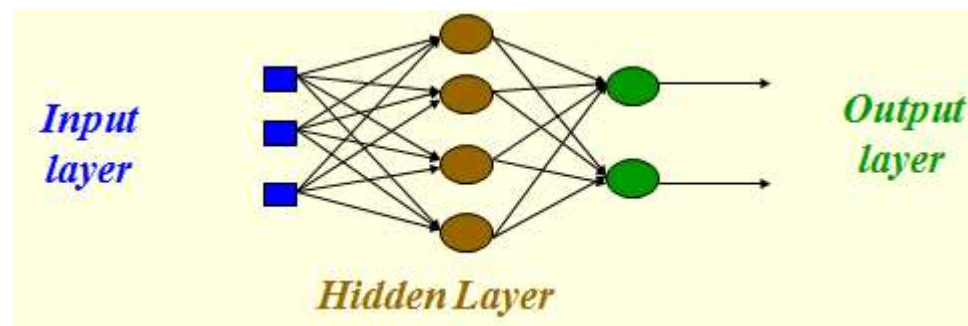
With x_1, x_2, x_3 are eigenvectors of each value of eigenvalue (λ).

```
>> lambda = eig(A)
lambda =
    1.0e+03 *
    0.0715
    0.2541
    1.2224
```



4. Nhận dạng đối tượng bằng mạng nơron

- Mạng nơron bao gồm các phần tử đơn giản (còn gọi là nơron) hoạt động song song được nối với nhau bằng các liên kết có trọng số để kích thích hoặc ức chế giữa các nơron.
- Có nhiều cấu trúc mạng nơron khác nhau như mạng hồi qui (feedback), mạng tự tổ chức (self-organizing), mạng truyền thẳng (feedforward), mạng truyền thẳng đa lớp với thuật toán lan truyền ngược (back propagation)



Hình 4. Mạng truyền thẳng 3 ngõ vào, 4 nút ẩn, 2 ngõ ra



4. Nhận dạng đối tượng bằng mạng nơron

❖ Huấn luyện mạng

- Quá trình huấn luyện mạng chính là quá trình huấn luyện các mẫu học $\mathbf{S} = \{\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_m\}$ để giá trị ra cuối cùng $\mathbf{T} = \{\mathbf{T}_1, \mathbf{T}_2, \dots, \mathbf{T}_n\}$ như ta mong muốn.
- Để chọn số nút ẩn cho mạng ta làm các bước sau:
 - + Bước 1 : Chọn số nút ẩn đủ để mạng hội tụ.
 - + Bước 2: Tính sai số trên tập kiểm tra. Nếu sai số trên tập kiểm tra tăng theo số lần luyện thì mạng chưa đủ nút ẩn để quá khớp. Tăng nút ẩn cho đến khi sai số giảm đến giá trị mong muốn.



4. Nhận dạng đối tượng bằng mạng nơ-ron

❖ Huấn luyện mạng

- Khởi tạo mạng cần có:

net : loại mạng nơ-ron

P : vector ngõ vào

T : vector ngõ ra

Ví dụ:

- `net = newff(minmax(P),[S1 S2],{'tansig' 'logsig'},'traingdx')`

Hàm kích hoạt cho ngõ ra của lớp vào là *tansig* và lớp ra là *logsig*. Thuật toán học theo phương pháp thích nghi, số nút lớp ẩn là S1, số ngõ ra là S2



4. Nhận dạng đối tượng bằng mạng nơron

❖ Huấn luyện mạng

- Các thông số chính mạng nơron:

`net.performFcn = 'sse';` : Sum-Squared Error performance function

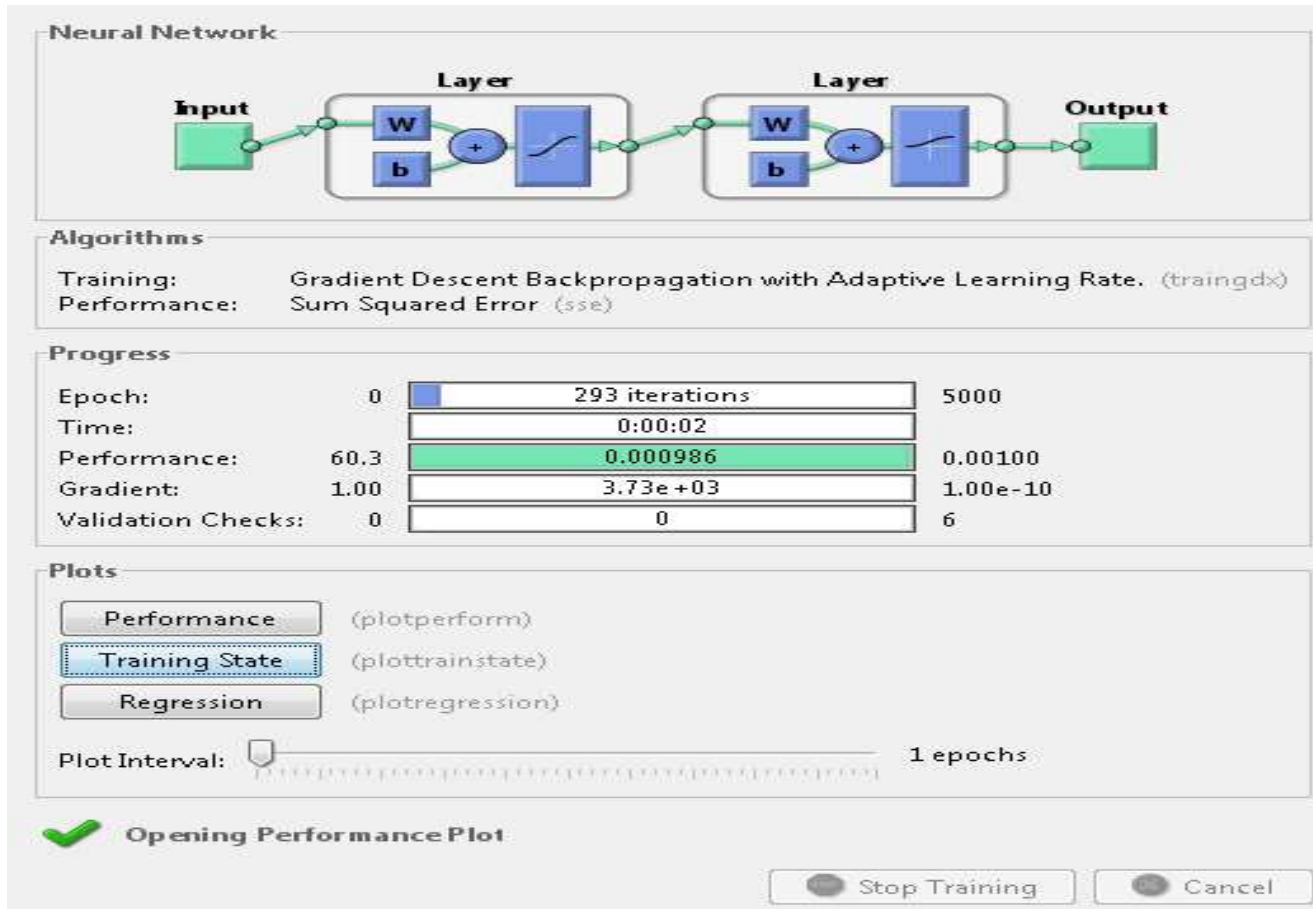
`net.trainParam.goal = 0.001;` : Sum-squared error goal.

`net.trainParam.epochs = 2000;` : Số lần học

- Khi huấn luyện mạng đạt được sai số mong muốn thì dừng lại để kiểm tra.



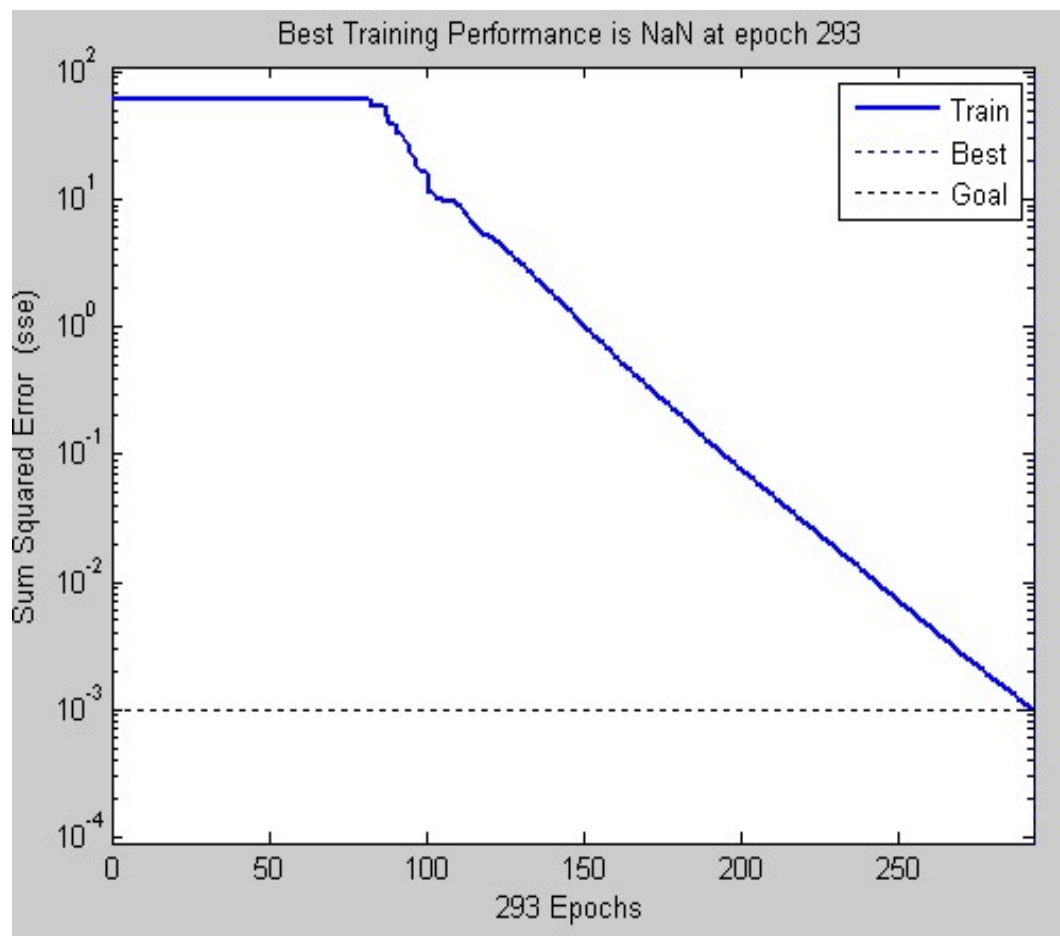
4. Nhận dạng đối tượng bằng mạng nơ-ron



Hình 6. Tiến trình huấn luyện mạng nơ-ron



4. Nhận dạng đối tượng bằng mạng nơ-ron



Hình 7. Tổng sai số bình phương trong quá trình luyện đạt đến giá trị mong muốn

THE END