

# Reconhecimento de face utilizando banco de imagens monocromáticas e coloridas através dos métodos da análise do componente principal (PCA) e da Rede Neural Artificial (RNA)

[Recognition to face using the monochromatic archive or the color archive by the methods of principal components analysis (PCA) and the Artificial Neural Network (ANN)]

Benedito Alencar de Arruda<sup>1</sup>; Gilberto Arantes Carrijo<sup>2</sup>

<sup>1</sup>Universidade Federal de Mato Grosso, Cuiabá/MT  
alencar@ufmt.br

<sup>2</sup>Universidade Federal de Uberlândia, Uberlândia/MG  
gilberto@ufu.br

**Abstract.** *This research aims at implementing a system of face identification using the algorithm of Principal Components Analysis and the algorithm of the Artificial Neural Networks, comparing the performance between the two methods.*

**Keywords:** *face recognition; artificial neural network; principal component analysis.*

**Resumo.** *Este trabalho objetiva implementar um sistema de reconhecimento de face usando o algoritmo da Análise do Componente Principal (PCA) e o algoritmo da Rede Neural Artificial (RNA), comparando a performance entre os dois métodos na utilização de banco de imagens monocromáticas e coloridas*

**Palavras-chave:** *reconhecimento de face; rede neural artificial; análise do componente principal.*

## 1. Introdução

O aumento vertiginoso do processo produtivo e de serviços com o respectivo aumento de pessoas envolvidas obriga as instituições e/ou empresas a darem atenção especial ao controle de pessoas que adentram em seus espaços. Com o controle feito por seres humanos tornando-se cada vez mais inviável, entra em cena a biometria que é usada para a identificação de pessoas através das características únicas de cada indivíduo, que neste trabalho foi pela a face.

## 2. Método da PCA [1]

### 2.1. Quanto ao Treinamento

Na fase de treinamento opera-se com arquivo composto de vetores de imagens de várias pessoas  $P$  (arquivo-treino), aqui chamadas de classes, sendo que cada classe pode ter várias diferentes imagens ou poses.

### 2.2. Quanto ao Reconhecimento

Na fase de reconhecimento ou teste, a idéia é dar uma ou mais imagem de  $j$  pessoas para serem encontradas comparando suas características com aquelas do arquivo-treinamento.

### 2.3. Cálculo pela PCA

Supondo a experiência com um arquivo-treino composto por  $p$  imagens. Cada imagem terá seu vetor  $P_i$ ,  $i=1,2,\dots,p$  convertido por concatenação em vetor pixel simples  $V_i$  na forma de vetor linha sendo  $V_i \in R^k$  onde  $k = M \times N$ . Aplicando a PCA para o conjunto de imagem a ser treinada obtém-se a matriz  $A$  que contém  $p$  linhas, sendo cada linha formada pelo vetor  $V_i$ . A dimensão de  $A$  é  $p \times k$ . Seguindo, calcula-se a matriz  $C_A$  através da covariância de  $A$ . Na seqüência calculam-se os autovalores e seus autovetores da matriz  $C_A$ . Obtém-se  $k$  autovalor e  $k$  autovetor, ou seja, para cada autovalor tem-se o autovetor correspondente  $e_i$  com dimensão  $k$ . Depois se ordenam todos os autovetores com seus autovalores em ordem decrescente de autovalor. Os autovalores e autovetores além de ordenados devem estar também emparelhados de tal maneira que o  $i$  autovalor corresponde ao  $i$  autovetor. Selecionam-se os maiores  $d$  autovalores e autovetores emparelhados. Cria-se a matriz transformação  $S$  ao dispor em coluna os autovetores selecionados. O que se pretende é calcular os vetores características  $W_i$  sendo  $W_i \in R^d$  com  $d \ll k$ . Através da matriz  $S$  acha-se  $W_i$ 's de  $V_i$ . Este cálculo é feito pela equação:

$$W_i = S^T V_i^T \quad (1)$$

Onde:

$S^T$  - matrizes transpostas de  $S$

$V_i^T$  - matrizes transpostas de  $V_i$

Cada coluna de  $S$  é um autovetor de comprimento  $k$ . Esta matriz é de comprimento igual a  $M \times N$ , ou seja, é a resolução da imagem de entrada. Assim pode-se converter cada autovetor em imagem revertendo a operação de concatenação. Estes autovetores imagens convertidos são chamados de autofaces desde que evidentemente sejam resultados de processamento de faces humanas.

Assim que se obter  $W_i$ 's pode-se reconstruir a imagem da pessoa  $i$ . Ao usar todos os autovetores  $k$  em vez de  $d$  para formar  $S$ , a imagem reconstruída será  $P_i$ . Entretanto neste caso pretende-se usar a resolução simplificada com  $d \ll k$ . Quanto mais autovetores serem usados para na formação de  $S$  mais próximo da imagem original ficará a imagem reconstruída.

### 3. Método da RNA

#### 3.1. Quanto ao Treinamento

A Rede Neural Artificial (RNA) é um processador inspirado na inteligência humana que adquire conhecimento através de experiências. Pela RNA o conhecimento é adquirido pelo processo de aprendizagem e forças de conexão entre neurônios, conhecidas como pesos sinápticos, que armazenam o conhecimento [2].

Na fase de treinamento possui dois passos distintos: um passo de propagação e outro passo de retro propagação para minimizar o erro entre a saída gerada pela rede e a saída desejada.

#### 3.2. Quanto ao Reconhecimento

Na fase de reconhecimento ou teste, dá-se uma ou mais imagem de  $j$  pessoas, para serem encontradas comparando suas características com aquelas das imagens contidas no arquivo-treinamento que foi na fase inicial construído com vetores de todas as imagens (poses) de todas as classes.

#### 3.3. Cálculo pela RNA

Pelo algoritmo da retro propagação é possível saber como ajustar os pesos sinápticos de modo a reduzir o valor da função custo [2]. Na propagação para frente, dado o sinal de entrada, calcula-se a saída da rede para a iteração  $n$ . A saída do neurônio  $j$  é dada pela equação apresentada a seguir:

$$y_j(n) = \phi(v_j(n)) \quad (2)$$

Onde:

$v_j(n)$  - campo local induzido (soma ponderada de todas as entradas sinápticas acrescida do bias) do neurônio  $j$  na iteração  $n$ , dado por:

$$v_j(n) = \sum_{i=0}^m w_{ji}(n) \cdot x_i(n) \quad (3)$$

Onde:

$m$  - o número total de entradas da rede.

$w_{ji}(n)$  - o peso sináptico que conecta o neurônio  $i$  ao neurônio  $j$ .

$x_i(n)$  - o sinal de entrada do neurônio  $j$  ou, equivalentemente, o sinal de saída do neurônio  $i$ .

O sinal de saída é então comparado com a resposta desejada presente no conjunto de treinamento, de modo a se obter o sinal de erro para o  $j$ -ésimo neurônio de saída, de acordo com a equação abaixo:

$$e_j(n) = d_j(n) - y_j(n) \quad (4)$$

Onde:

$d_j(n)$  – resposta desejada na saída.

$y_j(n)$  – sinal na saída.

Na etapa de propagação para trás (retro propagação), o erro é propagado para trás em direção às entradas da RNA, atualizando os seus parâmetros livres de acordo com uma regra de correção do erro, de modo que a saída da rede aproxime-se da saída desejada. A forma como os parâmetros livres da rede são atualizados é dada pela *regra delta* [2].

$$\Delta w_{ji}(n) = \eta \cdot \delta_j(n) \cdot y_i(n) \quad (5)$$

Onde:

$n$  - índice da iteração.

$\Delta w_{ji}(n)$ - correção de peso aplicada na conexão entre o neurônio  $i$  e o neurônio  $j$ .

$\eta$  - parâmetro taxa de aprendizagem.

$\delta_j(n)$  - gradiente local do neurônio  $j$ .

$y_i(n)$  - sinal de entrada do  $i$ -ésimo neurônio ou o sinal de saída do  $i$ -ésimo neurônio.

As equações utilizadas no cálculo do gradiente local podem ser encontradas com detalhes em Haykin [2]. Resumidamente tem-se: O gradiente local de um neurônio localizado na camada de saída é dado por:

$$\delta_j(n) = e_j(n) \cdot \phi_j'(v_j(n)) \quad (6)$$

O gradiente local de um neurônio localizado na camada intermediária  $j$  é dado por:

$$\delta_j(n) = \phi_j'(v_j(n)) \cdot \sum_k \delta_k(n) \times w_{kj}(n) \quad (7)$$

Onde:

$k$  - índice do(s) neurônio(s) à direita do neurônio  $j$ .

$\phi_j'(v_j(n))$  - derivada da função de ativação.

A *Função Tangente Hiperbólica*, assume valores entre 1 e -1, para  $a$  e  $b$  constantes [3][4] e é representada por:

$$\phi(v) = a \tanh(b \times v) \quad (8)$$

Onde:

$a$  - parâmetro de inclinação da curva, limite superior.

$b$ -parâmetro de inclinação da curva, limites inferior.

$v$  - valor da ativação.

#### 4. A Experiência do trabalho

Quanto ao hardware usou-se o notebook Dell, CPU com processador 1.40 .GHz, memória 504 MB; sistema operacional Microsoft Windows XP e linguagem de programa Matlab 7.0.1. Foram usados dois bancos de imagens. O arquivo Olivetti Research Laboratory da Universidade Britânica de Cambridge (ORL) [5] e o arquivo Face Recognition Data,

University of Essex, UK [6]. O primeiro contém fotos monocromáticas de 40 pessoas cada uma com 10 diferentes poses, num total de 400 fotos de dimensões 112 por 92 pixels no formato pgm. O segundo contém fotos coloridas de 20 pessoas cada uma com 20 poses diferentes, totalizando 400 fotos de dimensões 180 por 200 pixels no formato jpg. Em ambas as situações utilizaram-se um total de 400 imagens, sendo para a ORL 40 pessoas com 10 poses diferentes e para a UK 20 pessoas com 20 poses.

## 5. Comparação entre a experiência utilizando a PCA e a RNA

A tabela e as figuras estão no anexo. A experiência mostrou que o reconhecimento pela PCA é mais lento porém com maior taxa de acerto, já pela RNA o processamento é mais veloz no entanto a taxa de erro é um pouco maior.

Com relação ao arquivo monocromático a citada tabela apresenta os resultados comparando o de tempo de reconhecimento e a taxa de acerto entre a PCA e a RNA e a figura 1 mostra a curva de aprendizagem para a RNA.

Com relação ao arquivo de imagens coloridas a tabela do anexo apresenta os resultados comparando o de tempo de reconhecimento e a taxa de acerto entre a PCA e a RNA e a figura 2 mostra a curva de aprendizagem para a RNA.

## 6. Considerações Finais

Este trabalho mostrou uma das formas de se utilizar o método da PCA e da RNA. Basicamente tratou-se de reconhecer por dois métodos, uma dada imagem verificando se esta imagem estava contida em um arquivo dado. Neste caso trata-se dos arquivos de imagens de pessoas ORL preto e branco e o UK coloridas. O trabalho também mostrou que se pode facilmente incrementar os arquivos de dados e que embora havendo incremento de tempo de processamento, isto não torna inviável a sua aplicação, muito pelo contrário, se vislumbra um campo muito amplo para o exercício prático destes modelos. A ressalva a ser feita é com relação a PCA que aumenta muito o tempo de reconhecimento com o aumento do número de classes do arquivo. Portanto para o uso da PCA não se pode desejar respostas rápidas. Quanto a RNA, apesar de ligeira queda na taxa de acerto é um algoritmo muito rápido nas respostas. Assim, a cada um destes métodos pode-se ter um aplicação adequada às suas características. Tanto a PCA quanto a RNA são ferramentas robustas, compatível com o MATLAB, necessitando, para melhor uso, de uma configuração de hardware favorável quanto a memória e velocidade de processamento.

## 7. Referências

- [1] Kim, Kyungnam. Face Recognition using Principal Components Analysis.
- [2] Haykin, Simon. Redes Neurais - Princípios e Prática, Bookman, 2001
- [3] LeCun, Y., 1993. Efficient Learning and second-order Methods, A Tutorial at NIPS 93, Denver.
- [4] LeCun, Y., 1989. "Generalization and network design strategies,

” Technical Report CRG-TR-89-4, Department of computer Science, University of Toronto, Canada.

[5] Pagina eletrônica da Universidade Britânica de Cambridge. Disponível em julho de 2010.

[6] Pagina eletrônica da Universidade de Essex, United Kingston. Disponível em 30 julho de 2010.

## ANEXO

**Tabela: Resultado do processamento para o arquivo monocromático e para o arquivo UK**

Método Variáveis	PCA		RNA	
	* Arq ORL	**Arq UK	* Arq ORL	**Arq UK
Nº de imagem treino por étodo	400	400	400	400
Nº de imagem teste	120	60	120	60
Tempo de Reconhecimento	1 h e 54 min.	3 h e 30 min	30 min.	23 seg
Taxa de Acerto	100%	100%	95%	100%

\* arquivo monocromático

\*\* arquivo colorido

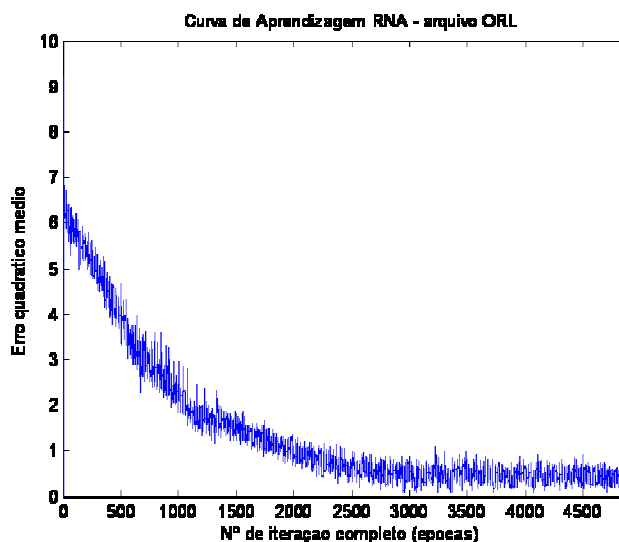


Figura 1- Curva de aprendizagem da RNA considerando o arquivo de imagens monocromáticas

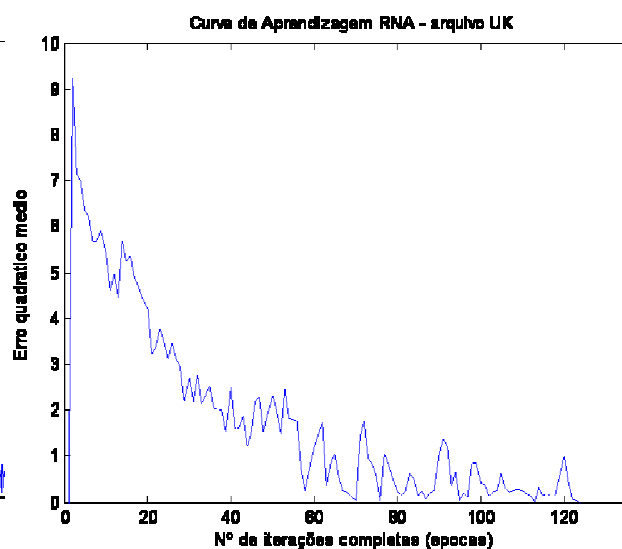


Figura 2- Curva de aprendizagem da RNA considerando o arquivo com imagens coloridas