

# IDENTIFICAÇÃO BIOMÉTRICA ATRAVÉS DA IMPRESSÃO DIGITAL USANDO REDES NEURAS ARTIFICIAIS

## Renan Corio Mazi

Instituto Tecnológico da Aeronáutica – ITA  
Rua: João Batista Soares Queiroz Júnior, 2296, apto. 54  
Bairro: Jardim das Indústrias  
Bolsista PIBIC-CNPq  
r\_mazzi@hotmail.com

## Arnaldo Dal Pino Júnior

Instituto Tecnológico da Aeronáutica – ITA  
dalpino@ita.br

### Resumo

*O propósito deste sistema se resume na aplicação de redes neurais artificiais para a identificação e reconhecimento biométrico de impressões digitais. O ambiente de trabalho Matlab 7.0 foi utilizado para a elaboração dos algoritmos. O sistema baseia-se na aquisição da imagem através de um leitor biométrico da Microsoft, no tratamento da imagem, na captura das minúcias, na eliminação das falsas minúcias, no treinamento da rede neural e na comparação dos resultados através de validações.*

*Palavras chave: biometria, impressão digital, reconhecimento biométrico, redes neurais artificiais.*

## 1. Introdução

A biometria sempre foi utilizada para o reconhecimento pessoal. Consiste em uma alternativa de identificação por apresentar as seguintes características: universalidade, unicidade, imutabilidade, classificabilidade, facilidade de coleta e aceitação pública. A utilização das Impressões Digitais para reconhecimento biométrico oferece segurança e eficácia, podendo substituir os cartões e senhas que utilizamos no dia a dia.

Existem diferentes métodos para a verificação e identificação de uma impressão digital. Neste sistema foi utilizado o método clássico, que se resume em aquisição da imagem, tratamento da imagem, afinamento das linhas, captura das minúcias, eliminação de falsas minúcias, armazenamento e comparação. Esta última etapa foi realizada através de redes neurais artificiais, buscando obter agilidade e precisão na verificação e identificação da impressão digital.

## 2. Impressão Digital

As Impressões Digitais são formadas por dobras da pele encontradas nos dedos das mãos e dos pés. Associada ao acrônimo de ID, são datadas desde o período pré-histórico, mas foi em 1667, no período científico, que iniciou-se estudos que provaram as características das IDs. O resultado desses estudos foi chamado de Dactiloscopia.

A pele humana é constituída de duas camadas, a derme, que é a camada mais profunda e apresenta elevações chamadas de papilas dérmicas, que só são perceptíveis através da epiderme, uma camada superficial e transparente que recobre a derme, realçando essas elevações chamadas de papilas.

As papilas, também chamadas de linhas ou cristas, são separadas por sulcos ou vales, caracterizando assim uma ID.

Os diferentes desenhos das papilas contribuem para a importância das características, que são:

- Unicidade - não existem dois indivíduos diferentes com impressões idênticas;
- Imutabilidade - a característica da impressão se mantém igual desde o nascimento até a decomposição após a morte.
- Praticabilidade - os desenhos papilares podem ser obtidos com grande facilidade e rapidez;
- Classificabilidade - os desenhos papilares, apesar de sua infinita variedade nas minúcias, atendem a um limitado número de tipos fundamentais, tornando possível classificá-los.

Existem também, as glândulas sudoríparas e sebáceas. Essas glândulas estão mergulhadas na derme e servem, respectivamente, para drenar o suor e secretar substâncias gordurosas que lubrificam a pele. É através dessa secreção que podemos ver nossas impressões digitais estampadas em objetos.

Ao examinar as papilas, notam-se algumas variações das cristas que podem ajudar na identificação da pessoa. Entre essas variações podem estar alguns acidentes de cristas interrompidas, bifurcadas, entre outros. Esses acidentes são chamados de minúcias ou pontos característicos, e são utilizados na fase da identificação da ID.

As minúcias são cristas descontínuas, que se interrompem ou bifurcam abruptamente. Essas minúcias definem a unicidade da pessoa. Alguns tipos de minúcias estão apresentados na figura 2.

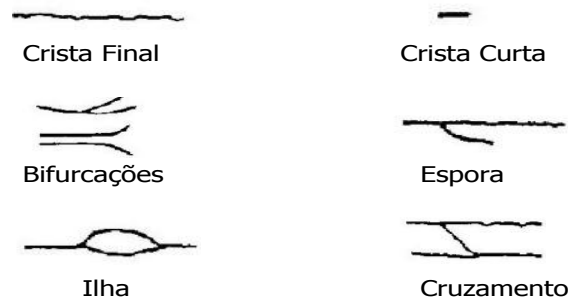


Figura 2 - Alguns tipos de minúcias encontradas nas IDs.

Mas não só esses tipos de minúcias classificam uma ID. Existem também mais dois tipos: o núcleo, que é a volta mais interna do conjunto das cristas; e o delta, que corresponde ao maior ângulo ou a um triângulo isósceles entre as cristas (Figura 2.1). Estes podem ser encontrados em uma ID em diferentes tamanhos e quantidades, podem possuir um núcleo e um delta, ou então um núcleo e dois deltas, ou ainda por cima, podem não existir.

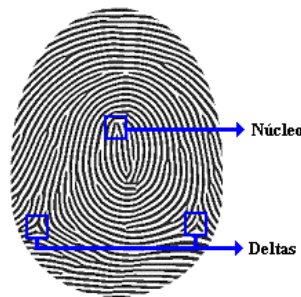


Figura 2.1- Em destaque, estão o núcleo e os deltas.

As minúcias são os pontos característicos que possibilitam o reconhecimento pessoal. Dentre os diversos tipos de IDs, foram utilizados neste sistema apenas terminações e bifurcações para o reconhecimento da ID.

Na classificação das IDs, encontramos quatro tipos, cuja denominação varia de acordo com a quantidade e posições dos deltas. Arco, provavelmente sem delta; Presilha Interna, delta a direita do observador; Presilha Externa, delta a esquerda do observador; e Verticilo, possui dois deltas, um a direita e o outro a esquerda do observador. Em (Kehdy, 1968, p.42) apresenta-se uma lista, onde arco plano e arco angular são subtipos da classe arco, diferente do sistema de Henry (1901) que os classifica como classes distintas.



Figura 2.2 - Classificação das IDs.

### 3. Tratamento da Imagem

A imagem da impressão digital foi capturada a partir do *Microsoft Fingerprint Reader*, um leitor biométrico específico equipado com um sensor capaz de capturar as linhas das impressões digitais. O software *Giaule SDK* foi utilizado para adquirir essas imagens, possibilitando a criação de um banco de dados.

O tratamento da imagem é a etapa mais importante deste sistema. Ela visa melhorar a qualidade da imagem fornecida pelo leitor biométrico para que o programa identifique as minúcias com maior precisão. Uma imagem de má qualidade acaba sendo prejudicial ao sistema, gerando muitos erros, como distorções na imagem e minúcias espúrias.

No processo de obtenção da imagem, as diferentes pressões que as pessoas exerciam sobre o leitor geravam imagens com tons de cinzas mais escuros ou mais claros em relação às outras, figura 3-a. Dessa maneira, o pré-processamento da imagem nem sempre produzia resultados satisfatórios. Assim, decidiu-se introduzir o processo de equalização das imagens. Foi utilizado um comando da área de trabalho *Matlab 7.0* que utiliza um histograma para equalizar as imagens, o que resultou na figura 3-b.

A partir da imagem equalizada, o filtro gaussiano entra em ação, suavizando os *pixels* da imagem e eliminando alguns pontos indesejáveis que se localizam nas bordas das cristas podendo ser os poros ou até mesmo sujeira. A figura 3-c mostra a ação do filtro gaussiano.

Alguns desses poros são suficientemente grandes que dependendo de onde estão localizados podem causar a interpretação errada do programa, gerando minúcias falsas. Dessa maneira, o filtro *motion blur* foi utilizado, causando uma leve distorção nos quadros da imagem. A olho nu é quase imperceptível, pois o filtro foi configurado para analisar a imagem em uma janela  $2 \times 2$  *pixels* distorcendo muito levemente essa região. Este filtro realça as linhas facilitando o processo de binarização da imagem. A figura 3-d apresenta o resultado da ação do filtro *motion blur* a partir da figura 3-c.

A imagem agora é submetida a um processo de binarização, que consiste em transformar os tons de cinza da imagem em preto e branco, ou seja, transformar os tons de cinza em uma matriz de zeros e uns. Esta é uma etapa muito importante do tratamento da imagem, pois uma binarização mal executada acaba gerando muita perda de informação.

Os *pixels* da imagem possuem tons de cinza variando entre 0 e 256. As linhas da imagem possuem valores elevados em relação ao restante da imagem, a parte vazia. Para binarizar uma imagem, a linha cinza escura deveria ser transformada em preto e o espaço vazio da imagem em branco. Para isso, toda a imagem foi analisada em quadros de  $8 \times 8$  *pixels* e obtido um *threshold* para cada uma das regiões através da média geométrica dos *pixels* daquela região. Assim, todos os *pixels* que estivesse acima desse *threshold* seria transformado no valor 1 e o que estivesse abaixo em 0. A figura 3-e representa a imagem binarizada com o *threshold* adaptativo.

O próximo passo é o afinamento das linhas. O algoritmo de afinamento analisa a imagem da ID e remove os *pixels* redundantes das linhas que formam as cristas. Esse processo é repetido até que não se tenha mais *pixels* redundantes, resultando no esqueleto da imagem, figura 3-f. A varredura da imagem é feita linha a linha, examinando a vizinhança e verificando se o *pixel* pode ou não ser apagado. Quando um *pixel* é apagado, seu valor muda de 1 para 0, e a imagem é dita afinada.



Figura 3 – a - imagem obtida do leitor biométrico; b – imagem equalizada; c – imagem sob ação do filtro gaussiano; d – imagem sob ação do filtro *motion blur*; e – imagem binarizada com *threshold* adaptativo; f – imagem afinada com um *pixel* de largura.

#### 4. Extração das Minúcias

Nesta etapa, um processo chamado de *crossing number* analisa os *pixels* da imagem em uma matriz 3 x 3 à procura de alguns pontos característicos. Quando o *pixel* central contiver um único vizinho, ele é dito terminação, e quando ele contiver três vizinhos, ele é dito bifurcação.

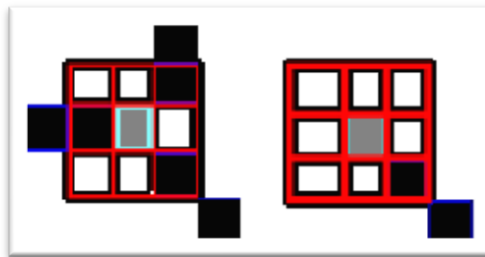


Figura 4 – À esquerda, a representação de uma bifurcação e à direita, a representação de uma terminação.

Porém são obtidas várias minúcias espúrias, pois no processo de afinamento da imagem, a linha pode apresentar algumas interrupções repentinas e logo em seguida a sua continuação, gerando para o algoritmo duas terminações. Neste caso, a qualidade da imagem e o tratamento da mesma naquela região não foram suficientes para gerar um bom afinamento das linhas.

Como isso depende muito da qualidade da imagem e do seu tratamento, foi utilizado um algoritmo capaz de eliminar essas falsas minúcias através do cálculo da distância euclidiana entre elas. Assim, quaisquer minúcias que distam em até nove *pixels* entre si foram eliminadas. A figura 4.1 compara as minúcias encontradas inicialmente e logo após o processo de eliminação. Os pontos vermelhos representam as terminações e os pontos azuis representam as bifurcações.

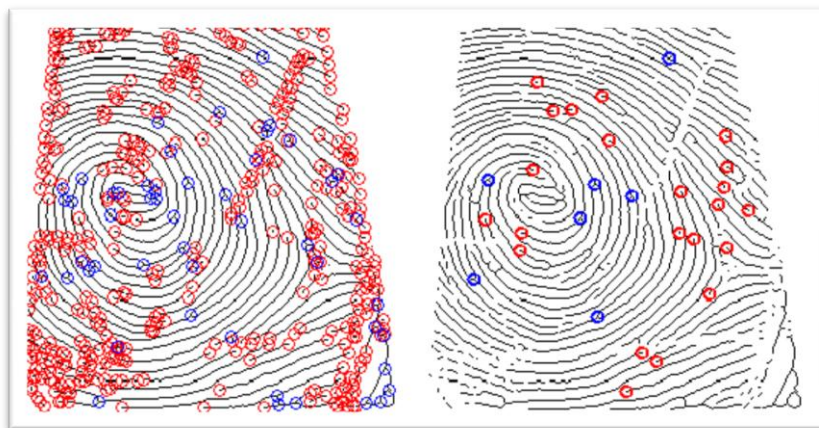


Figura 4.1 – À esquerda, todas as minúcias encontradas e à direita, apenas as minúcias verdadeiras.

## 5. Redes Neurais Artificiais

O cérebro humano é o dispositivo mais complexo conhecido pelo Homem. Sua capacidade de pensar, memorizar e resolver problemas tem conduzido muitos cientistas a tentar modelar sua operação. Diversos pesquisadores têm buscado criar modelos computacionais que representam a funcionalidade do cérebro. Um destes modelos resultou na criação das Redes Neurais Artificiais (RNAs).

As RNAs são sistemas computacionais de implementação em *software* ou *hardware*, que imitam as habilidades dos neurônios biológicos, usando para isto um grande número de neurônios artificiais interconectados.

Existem vários tipos de RNAs, onde os mais conhecidos são *ADALINE/MADALINE*, *Backpropagation Perceptron* (ou *feedforward*), *Time-Delay*, *Recurrent*, *BAM* (Memória Associativa Bidirecional), *Hofield* e *Kohonen* (Loesch, 1996, p. 46).

Para que as RNAs possam resolver os problemas para o qual foram projetadas, as mesmas devem ser treinadas. Todo o desempenho da RNA está ligado ao processo de treino que deve adaptar seus pesos e sinais de saída de acordo com os objetivos finais. O processo de treino envolve ajustar os pesos das conexões entre os neurônios (ajuste sináptico) e configurar os valores de saída para cada caso verificando a quantidade de erro sobre o valor de saída do neurônio (saída desejada).

São conhecidos dois métodos de treinamentos: o supervisionado e o não-supervisionado. O primeiro consiste em um aprendizado auxiliado por um treinador. É o método mais comum onde o treinador apresenta os dados de entrada para a RNA e os valores de saída desejados para aquela entrada. Depois de uma iteração, os valores de saída são comparados e os pesos são ajustados. Esse processo se repete até que o erro seja minimizado. No treinamento não-supervisionado, a RNA deve aprender sozinha, o treinador apresenta para a rede apenas uma amostra dos dados de entrada e ela deve ajustar os pesos e os valores de saída automaticamente pela classificação dos dados de entrada pelos neurônios.

Para confirmar a capacidade de aprendizado da rede, utiliza-se um processo de validação, o qual emprega o uso de diferentes conjuntos de entrada dos que foram ensinados, ou conjuntos de validação. Assim verifica-se a capacidade de generalização da rede, se esta não atingir um desempenho ótimo, treina-se a rede novamente com o conjunto de treino adicionado de algumas amostras do conjunto de validação e/ou reinicializando seus pesos (Masters, 1993, p.9-11).

### 5.1 Feedforward Backpropagation

A RNA *feedforward*, também conhecida como MLP (*Multi Layer Perceptron*), é uma das mais especificada em livros. Também é a mais utilizada, cerca de 90% entre todas as outras RNAs utilizadas. Além de grande capacidade de abstração e generalização que permite a ela classificar um padrão complexo que não pertenceu ao conjunto de treino, possui robustez que a faz imune a pequenas falhas relativas a ruídos e distorções da entrada (Loesch, 1996, p.67).

Uma rede *feedforward* multicamada consegue aprender uma função ensinada. Se esta apresentar problemas não é devido ao modelo em si, é por causa do treinamento insuficiente, número pequeno de neurônios na camada oculta ou tentativa de aprender uma função que não é determinística (Masters, 1993, p.87).

A figura 4.2 representa uma rede *feedforward* com três camadas:

- uma de entrada, onde é apresentado os dados para a rede;
- uma intermediária ou oculta, onde é feita a maior parte do processamento, através das conexões ponderadas; podem ser consideradas extratoras de características;
- e uma de saída, onde o resultado final é concluído e apresentado.

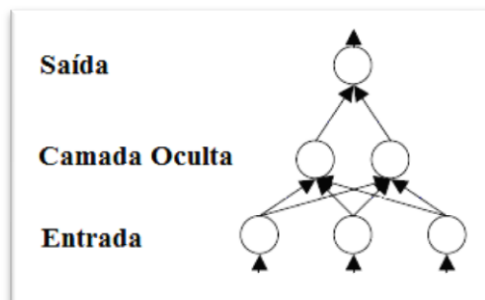


Figura 5.1 – Representação da rede neural *feedforward*.

A rede pode apresentar mais do que uma camada oculta, mas é conhecido que uma única camada oculta consegue resolver problemas determinísticos (Masters, 1993, p.79).

A quantidade de neurônios na camada de entrada e de saída depende muito do problema a ser resolvido, mas a quantidade de neurônios ocultos é variável. Ela pode ser determinada através da equação (1), onde  $m$  é a quantidade de neurônios de saída e  $n$  a quantidade de neurônios de entrada e  $O$  o número de neurônios na camada oculta.

$$O = \sqrt{m * n} \quad (1)$$

A rede é chamada de *feedforward*, pois seu sinal de entrada sempre se propaga na rede em um fluxo para frente, passando de camada em camada. A camada de entrada serve apenas para passar os sinais para frente, nela não é feito nenhum cálculo. A entrada de cada camada vem da saída da camada anterior. Cada camada é plenamente conectada a camada seguinte, onde cada neurônio é conectado com todos os neurônios da camada posterior. Cada camada possui uma função de ativação que é utilizada para todos os neurônios daquela camada. Além disso, podem-se utilizar funções de ativações diferentes para cada camada, dependendo do tipo de saída que se deseja obter.

A função de ativação de um neurônio é uma função não-linear que quando aplicada a soma ponderada determina a saída deste neurônio, ela é usada para limitar a amplitude destes valores. As faixas de valores de algumas funções são limitadas de 0 a +1 enquanto outras de -1 a +1. A figura 4.2.1 representa as três diferentes funções de ativação utilizadas para as redes neurais *feedforward*.

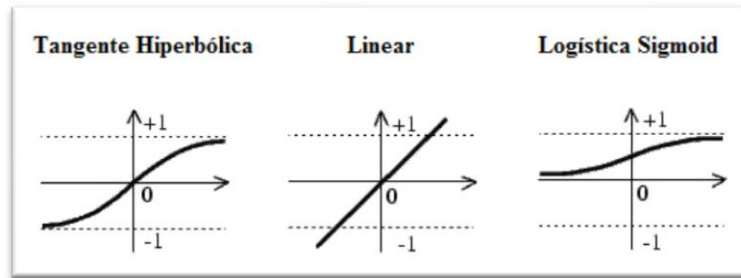


Figura 5.2 – Funções de Ativação

O *backpropagation* é o algoritmo de treinamento da rede. Com ele, o treinamento da rede opera em uma seqüência de dois passos: o primeiro é o passo *feedforward*, onde os dados de entrada fluem através da rede, camada por camada, até que a resposta seja produzida pela camada de saída; no segundo passo, o resultado obtido é comparado com o resultado desejado para este padrão particular. Se esta não estiver correta, o erro é calculado. O erro é propagado a partir da camada de saída até a camada de entrada, atualizando os pesos das conexões dos neurônios das camadas internas conforme o erro é retro propagado.

A medida mais comum do erro é dada pelo *erro quadrado médio* ou EQM, de acordo com a equação (2).

$$E_p = 1/n * \sum_{j=1}^n (t_{pj} - O_{pj})^2 \quad (2)$$

Onde:  $n$  é o número de neurônios na camada de saída;

$j$  é o neurônio da camada de saída;

$t_{pj}$  é a correta ativação alvo do neurônio  $j$ ;

$O_{pj}$  é a ativação observada.

Todo esse processo de apresentação dos dados de entrada à rede, calcular as ativações das camadas, calcular os gradientes e o erro, e reajustar os pesos, é chamado de *epoch* ou época. Cada época é uma iteração que procura diminuir o erro encontrado até que este seja suficientemente pequeno. O número de iterações pode ser determinado com o intuito de barrar o treinamento para que este não se prolongue por tanto tempo.

Após o treinamento alcançar o erro máximo permitido dentro do limite das épocas determinado, a rede é dita treinada.



## 6. Aplicação e Treinamento

Para o treinamento da rede neural *feedforward backpropagation* os dados de entrada e de saída tiveram que ser determinados e preparados. Foram criadas matrizes quadradas de zeros de dois tamanhos, onde as coordenadas das minúcias da imagem original foram divididas e inceridas proporcionalmente nessas matrizes. Assim, a matriz resultante corresponde a uma imagem de cor preta contendo pontos brancos que indicam as respectivas minúcias encontradas na ID. Somente matrizes de mesmo tamanho podem ser usadas no treinamento.

Foram usadas matrizes de tamanhos 10x10 e 15x15 para agilizar o treinamento, pois usar a imagem inteira como dado de entrada em uma rede *feedforward backpropagation* seria inviável pelo seu longo tempo de treinamento. A figura 6 representa as minúcias encontradas na imagem da direita da figura 3.1.



Figura 6 – Matriz 15x15 representando a Figura 4.1.

É notável que a existência de menos pontos na matriz da figura anterior do que na Figura 3.1. Isso se dá ao fato de que a compressão da matriz 300x300 para a matriz 15x15 faça com que alguns pontos próximos se localizem em cima de outros, devido à divisão das coordenadas e ao seu arredondamento. Entretanto, pontos coincidentes ou não foram deixados com valor 1 para minúcia e 0 para qualquer outro ponto.

Além disso, essas matrizes quadradas foram transformadas em vetores de uma única coluna, onde cada linha está ligada a um neurônio da rede.

Como esta rede neural, para este caso, deve reconhecer o certo do errado, foram disponibilizados para a rede quais seriam os dados certos e quais seriam os errados. Antes de o treinamento começar, foram carregados os dados da seguinte maneira: oito IDs do mesmo dedo da mesma pessoa a ser cadastrada e dezesseis IDs aleatórias do banco de dados. Os dados de saída foram dispostos de acordo com as IDs. Para as IDs que seriam cadastradas foram adotados o valor 100 e para as IDs que teriam que ser rejeitadas o valor zero.

Diversos testes foram realizados com diversos métodos de treinamentos. Foram observados que apenas dois métodos obtiveram melhores resultados e com isso, os testes foram baseados nesses dois métodos. Utilizou-se a técnica do gradiente descendente e o método de Levenberg-Marquardt (LM).

Foram feitos sete tipos de treinamentos levando em conta o erro máximo que a rede poderia atingir e o número de neurônios na camada oculta (NO). Os resultados obtidos estão dispostos no próximo item.

## 7. Resultados

- Treinamento com matriz 10x10 utilizando gradiente descendente.

### Percentual de acertos X Neurônios na camada oculta

NO	Erro=0.01	Erro=0.001	Erro=0.0001
05	25%	25%	16,60%
10	33,33%	8,33%	16,60%
15	25%	25%	17%
20	25%	16,60%	8,33%
25	16,66%	16,60%	17%
50	0%	8,33%	8,33%
335	0%	0%	0%

### Percentual de Falsas Aceições X Neurônios na camada oculta

NO	Erro=0.01	Erro=0.001	Erro=0.0001
05	3%	0%	4,76%
10	1,58%	3,17%	1,58%
15	1,58%	1,58%	1,58%
20	1,58%	1,58%	0%
25	0%	1,58%	3%
50	0%	0%	0%
335	0%	0%	0%

- Treinamento com matriz 10x10 utilizando LM.

**Percentual de acertos X Neurônios na camada oculta**

NO	Erro=0.01	Erro=0.001	Erro=0.0001
05	50%	50%	41,60%
10	33,33%	33,33%	41,60%
15	16,60%	50%	42%
20	25%	25%	8,33%
25	16,60%	50%	50%
50			
335			

**Percentual de Falsas Aceitações X Neurônios na camada oculta**

NO	Erro=0.01	Erro=0.001	Erro=0.0001
05	14,28%	15,87%	4,70%
10	7,80%	11,10%	9,52%
15	4,70%	4,70%	9,52%
20	0%	3,17%	3,17%
25	4,70%	6,50%	4,70%
50			
335			

Os treinamentos com 50 e 335 neurônios, que estão ressaltados em cinza, não puderam ser completados por insuficiência de memória RAM do computador utilizado.

- Treinamento com matriz 15x15 utilizando gradiente descendente.

**Percentual de acertos X Neurônios na camada oculta**

NO	Erro=0.01	Erro=0.001	Erro=0.0001
05	33%	50%	58,30%
10	33,33%	41,60%	33,33%
15	33,33%	16,60%	25%
20	16,60%	16,60%	16,60%
25	8,33%	8,33%	0%
50	0%	8,33%	8,33%
335	0%	0%	0%

**Percentual de Falsas Aceitações X Neurônios na camada oculta**

NO	Erro=0.01	Erro=0.001	Erro=0.0001
05	8%	7,90%	12,70%
10	4,70%	3,17%	4,70%
15	3,17%	4,70%	5%
20	3,17%	3,17%	0%
25	3,17%	1,58%	0%
50	0%	0%	0%
335	0%	0%	0%

- Treinamento com matriz 15x15 utilizando LM.

**Percentual de acertos X Neurônios na camada oculta**

NO	Erro=0.01	Erro=0.001	Erro=0.0001
05	33%	83,30%	50%
10	16,60%	33%	25%
15	25%	42%	42%
20	25%	50%	25%
25	25%	25%	33%
50			
335			

**Percentual de Falsas Aceitações X Neurônios na camada oculta**

NO	Erro=0.01	Erro=0.001	Erro=0.0001
05	16%	22,22%	17%
10	3,17%	17,46%	19%
15	6,30%	0%	14%
20	1,58%	12,70%	7,90%
25	7,90%	7,90%	10%
50			
335			

Os treinamentos com 50 e 335 neurônios, que estão ressaltados em cinza, não puderam ser completados por insuficiência de memória RAM do computador utilizado.



## 8. Conclusão

O refinamento nos processos de tratamento da imagem contribuiu para a eliminação de alguns ruídos da imagem original. A etapa de binarização com *threshold* adaptativo resultou em uma melhora significativa para os processos de afinamento da imagem e extração das minúcias.

A utilização da rede neural *feedforward backpropagation* para o reconhecimento das IDs não foi de absoluto sucesso. O método de treinamento utilizando matrizes de tamanhos menores que a imagem original e a disponibilização dos dados de saída da rede se mostraram inconveniente para este sistema. Esperava-se obter uma taxa de acertos superior a 90% e uma taxa de falsas aceitações menor que 5%, porém o melhor resultado obtido, combinando os acertos e as falsas aceitações, está descrito na última tabela, com 42% de acertos e 0% de falsas aceitações.

## 9. Agradecimentos

Agradeço primeiramente ao CNPq, por ter me proporcionado a bolsa de estudos de iniciação científica. Fico grato também ao ITA pela infra-estrutura disponibilizada, ao professor Marim e ao meu orientador Arnaldo por terem me auxiliado e compartilhado seus conhecimentos para que eu pudesse desenvolver este projeto.

Agradeço também a todos que colaboraram para a criação do banco de dados de impressões digitais, sem os quais esse estudo seria incompleto.

Por fim, agradeço à minha família, que me deu o suporte necessário para que eu chegasse até aqui, e vivenciasse essa grande oportunidade.

## 10. Referências Bibliográficas

- Araokar, Shashank. Visual Character Recognition using Artificial Neural Networks. MGM's College of Engineering and Technology, University of Mumbai, Índia.
- Brown, Eric W. Applying Neural Networks to Character Recognition. Disponível em: <<http://www.ccs.neu.edu/home/feneric/charrecnn.html>>. Acesso em: 13 Nov. 2008.
- FVC. Fingerprint verification competition: first international competition for fingerprint verification algorithms, Cesena, 2000. Disponível em: <<http://bias.csr.unibo.it/fvc2000/download.asp>>. Acesso em: Out. 2008. Biometric System Lab, Department of Computer Science, University of Bologna, Cesena, Italy.
- Gumz, Rafael Araújo. Protótipo de um sistema de identificação de minúcias em impressões digitais utilizando redes neurais artificiais multicamada. 2002. 134 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.
- Haykin, Simon. Neural networks: a comprehensive foundation. Englewood Cliffs: Macmillan, 1994.
- Henry, Edward Richard. Classification and Uses of Finger Prints (1901). London: HM Stationery Office.
- Kehdy, Carlos. Elementos de criminalística. 3.ed. São Paulo: Sugestões Literárias, 1968.
- Loesch, Cláudio; SARI, Solange T. Redes neurais artificiais: fundamentos e modelos. Blumenau: Ed. da FURB, 1996.
- Maio, Dario; MALTONI, Davide. Direct Gray-Scale Minutiae Detection In Fingerprints. 1997, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19.
- Masters, Timothy. Practical neural network recipes in C++. San Diego: Academic Press, 1993.
- Masters, Timothy. Signal and image processing with neural networks: a C++ sourcebook. New York: Academic Press, 1994.
- MATHWORKS, the. MatLab official site. Disponível em: <[http://www.mathworks.com/products/tech\\_computing/](http://www.mathworks.com/products/tech_computing/)>. Acesso em: Agosto, 2008.
- Rashid, Md. Mamunur; HOSSAIN, Akatar. Fingerprint Verification System Using Artificial Neural Network. 2006, Asian Network for Scientific Information.
- Silva, Alex Sandro da. Protótipo de software para classificação de impressão digital. 1999. 70 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.
- Tafner, Maucon; XEREZ, Marcos de; RODRIGUES FILHO, Ison W. Redes neurais artificiais. Blumenau: Ed. da FURB, 1995.
- Tavares Jr., Gilberto da Silva. A papiloscopia nos locais de crime. São Paulo: Ícone, 1991.