

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO

**PROTÓTIPO DE UM SISTEMA DE IDENTIFICAÇÃO DO(S)
DELTA(S) E NÚCLEO EM IMPRESSÕES DIGITAIS
UTILIZANDO REDES NEURAS ARTIFICIAIS.**

CAIO RAFAEL SILVA MATIAS

BLUMENAU
2004

2004/2-09

CAIO RAFAEL SILVA MATIAS

**PROTÓTIPO DE UM SISTEMA DE IDENTIFICAÇÃO DO(S)
DELTA(S) E NÚCLEO EM IMPRESSÕES DIGITAIS
UTILIZANDO REDES NEURAS ARTIFICIAIS.**

Trabalho de Conclusão de Curso submetido à Universidade Regional de Blumenau para a obtenção dos créditos na disciplina. Trabalho de Conclusão de Curso II do curso de Ciências da Computação — Bacharelado.

Prof. Jomi Fred Hübner - Orientador

**BLUMENAU
2004**

2004/2-09

**PROTÓTIPO DE UM SISTEMA DE IDENTIFICAÇÃO DO(S)
DELTA(S) E NÚCLEO EM IMPRESSÕES DIGITAIS
UTILIZANDO REDES NEURAS ARTIFICIAIS.**

Por

CAIO RAFAEL SILVA MATIAS

Trabalho aprovado para obtenção dos créditos
na disciplina de Trabalho de Conclusão de
Curso II, pela banca examinadora formada
por:

Presidente: _____
Prof., Jomi Fred Hübner – Orientador, FURB

Membro: _____
Prof. Mauro Marcelo Mattos, FURB

Membro: _____
Prof. Paulo César Rodacki Gomes, FURB

Blumenau, 09 de Fevereiro de 2005.

Dedico este trabalho a toda a minha família em especial à minha noiva, pessoas estas que me apoiaram durante toda a minha vida.

“A mente que se abre a uma nova idéia jamais volta ao seu tamanho original”.

Albert Einstein

AGRADECIMENTOS

A Deus, pela oportunidade da existência.

A meus pais, Jair Matias e Roseli Silva Matias, por toda a educação recebida ao longo destes meus 22 anos, educação esta que foi fundamental para a formação de meu caráter.

À minha noiva Gabriela Porto pela sua maturidade e companheirismo mostrados nos momentos de minha ausência em que me dediquei a este trabalho.

Ao meu orientador, Jomi Fred Hübner pelo seu bom humor, e conhecimentos técnicos, bem como pelo seu apoio e incentivo prestados a mim durante todo o período de desenvolvimento.

Aos meus companheiros Alex Sandro Silva e Maurício Edgar Stivanello, com os quais troquei muitas idéias durante todo o período de desenvolvimento deste trabalho.

E por fim a todos aqueles amigos e companheiros, que já passaram ou estão presentes em minha vida, e que já torceram ou torcem pelo meu sucesso enquanto ser humano.

RESUMO

Este trabalho apresenta o desenvolvimento de um protótipo para reconhecimento de delta(s) e núcleo em impressões digitais. Para este reconhecimento utilizou-se uma Rede Neural Artificial *feedforward* multicamada com treinamento *backpropagation*. Como entradas para esta rede são apresentados fragmentos de imagens, e como resposta espera-se que a rede classifique esta entrada em três padrões, os quais indicarão a qual região este fragmento de imagem pertence. Ainda são mostrados neste trabalho a construção de um sistema de identificação das impressões digitais através da comparação de uma impressão digital com outras de usuários previamente cadastrados, bem como as técnicas, dificuldades e resultados referentes as etapas de estudo e desenvolvimento.

Palavras chaves: Inteligência Artificial; Redes Neurais Artificiais; Impressões Digitais;

ABSTRACT

This work describes the development of prototype to recognize delta(s) and kernel in fingerprints. It was applied a feedforward multi-layer artificial neural network with backpropagation algorithm to classify the fingerprints. Some regions of the fingerprints image are used as training samples for this network and it answer by classify theses inputs as the either delta or kernel. This work shows the implementation of a fingerprints recognizing system by comparing fingerprints with another one previously provided. Some techniques, difficulties and results identified while this working was being done are show.

Palavras chaves: Artificial Inteligence; Artificial Neural Network; Fingerprints;

LISTA DE ILUSTRAÇÕES

Figura 1 – Regiões dos sistemas de linhas.	21
Figura 2 – Tipos fundamentais de IDs no sistema de Juan Vucetich.	23
Figura 3 – Em destaque núcleo e deltas de uma ID.	24
Figura 4 – Representação macro de um sistema de identificação biométrica.	26
Figura 5 – Representação de um neurônio biológico.	32
Figura 6 – Representação do funcionamento de um neurônio artificial.	35
Figura 7 – Representação dos Fluxos das RNA.	36
Figura 8 – Função logística de ativação sigmóide binária.	38
Figura 9 – Exemplo de regiões da IID passadas como entrada para a RNA.	44
Figura 10 – Definição da divisão de quadrantes numa IID.	44
Figura 11 – Exemplo de distâncias calculadas para os 3 tipos fundamentais de ID.	45
Figura 12 – Diagrama de caso de uso primário do ator “Especialista”.	46
Figura 13 – Diagrama de caso de uso primário do ator “Usuário”.	46
Figura 14 – Diagrama de caso de uso secundário do ator “Usuário”.	47
Figura 15 – Diagrama de Classes do protótipo.	47
Quadro 1 – Rotina principal de treinamento da classe RedeNeural.	53
Quadro 2 – Rotina de ativação dos neurônios das camadas ocultas e de saída.	54
Quadro 3 – Rotina de cálculo do delta dos neurônios da camada de saída da RNA.	54
Quadro 4 – Rotina de cálculo do delta dos neurônios da camada oculta RNA.	55
Quadro 5 – Rotina de ajuste dos pesos de todas as camadas da RNA.	55
Quadro 6 – Rotina de cálculo do Erro Quadrado Médio EQM.	56
Quadro 7 – Rotinas da Função Sigmóide Binária e sua derivada.	56
Quadro 8 – Rotina de conversão de imagens 20x20 para os valores de entrada da RNA.	56
Quadro 9 – Rotina da heurística do cálculo da distância.	57
Figura 16 – Exemplo de IIDs capturadas com tinta comum e tinta tipográfica.	59
Figura 17 – IID capturada através de um leitor biométrico.	59
Figura 18 – Exemplos de IIDs capturadas na Internet.	60
Figura 19 – Exemplos de IIDs capturadas na Internet.	60
Figura 20 – Exemplos de configuração do Aplicativo Sfinge.	61
Figura 21 – Exemplos de configuração do Aplicativo Sfinge.	61
Figura 22 – Exemplo de IIDs geradas pelo aplicativo Sfinge.	62
Figura 23 – Amostras de treinamento de regiões dos deltas ampliadas em 5 vezes.	65
Figura 24 – Amostras de treinamento de regiões dos núcleos ampliadas em 5 vezes.	65
Figura 25 – Amostras de treinamento de regiões quaisquer ampliadas em 5 vezes.	65
Figura 26 – Tela principal do protótipo.	66
Figura 27 – Pasta Treinamento Automático da tela de Treinamento da RNA.	67
Figura 28 – Pasta Treinamento Manual da tela de Treinamento da RNA.	68
Figura 29 – Pasta de Pesos da tela de Treinamento da RNA.	69
Figura 30 – Pasta de Validação Automática da tela de Validação da RNA.	70
Figura 31 – Pasta de Validação Manual da tela de Validação da RNA.	70
Figura 32 – Tela de cadastramento de usuários.	71
Figura 33 – Tela de identificação das IIDs através da comparação das distâncias.	72
Figura 34 – Gráfico de testes com diferentes valores em parâmetros da RNA.	73

LISTA DE TABELAS

Tabela 1 – Quantidade de amostras por Grupo para cada tipo de região	64
Tabela 2 – Resultado da taxa de acerto da RNA com vários NO na camada Oculta	73
Tabela 3 – Resultados por grupos de Amostras	74

LISTA DE SIGLAS

AFIS – *Automatic Fingerprint Identification System*

BD – Banco de Dados

BMP – *BitMaP* ou mapa de bits

DPI – *Dots Per Inch* ou pontos por polegada

EQM – Erro Quadrado Médio

ID – Impressão Digital

IID – Imagem de Impressão Digital

NO – Neurônio Oculto

Pixel – *Picture Element* ou elemento da imagem

RGB – *Red+Green+Blue* ou Vermelho+Verde+Azul

RNA – Rede Neural Artificial

RNA-FM – Rede Neural Artificial *Feedforward* Multicamada

SGBD – Sistema Gerenciador de Banco de Dados

UML – *Unified Modeling Language* ou Linguagem de Modelagem Unificada

XOR – *eXclusive OR* conhecido como OU exclusivo

SUMÁRIO

1 INTRODUÇÃO.....	13
1.1 ORIGEM DO TRABALHO.....	13
1.2 ÁREAS.....	14
1.3 PROBLEMA.....	15
1.4 JUSTIFICATIVAS.....	15
1.5 OBJETIVOS.....	16
1.6 ESTRUTURA DO TRABALHO.....	16
2 FUNDAMENTAÇÃO TEÓRICA.....	18
2.1 IMPRESSÕES DIGITAIS.....	18
2.1.1 Datiloscopia.....	18
2.1.2 Estudo da Pele.....	19
2.1.3 Sistema Datiloscópico de Juan Vucetich.....	21
2.1.3.1 Minúcias.....	23
2.1.3.2 Núcleo e Deltas.....	24
2.2 BIOMETRIA.....	25
2.2.1 Sistemas de Identificação Biométrica.....	25
2.2.2 Sistemas de Identificação Biométrica através das Impressões Digitais.....	26
2.3 PROCESSAMENTO DE IMAGENS.....	28
2.3.1 Imagem Digital.....	28
2.3.1.1 Definição de Cores.....	29
2.3.1.2 Formato de Imagens.....	30
2.3.2 Técnicas de Processamento de Imagens.....	30
2.4 REDES NEURAIS.....	31
2.4.1 Sistema Nervoso e Neurônio Biológico.....	31
2.4.2 Redes Neurais Artificiais.....	33
2.4.2.1 Neurônio Artificial.....	34
2.4.2.2 Rede Neural Artificial <i>Feedforward</i> Multicamada.....	36
2.4.2.3 Funções de Ativação.....	37
2.4.2.4 Treinamento e Aprendizado.....	38
2.4.2.4.1 Aprendizado por Retropropagação (<i>backpropagation</i>).....	39
2.4.2.5 Escolha da Estrutura das Redes.....	41
2.5 CONTEXTO ATUAL DO TRABALHO.....	41

2.5.1 Trabalhos Correlatos	41
3 DESENVOLVIMENTO DO PROTÓTIPO	43
3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO.....	43
3.2 VISÃO GERAL.....	43
3.3 ESPECIFICAÇÃO	45
3.3.1 Casos de Uso	46
3.3.2 Diagramas de Classes.....	47
3.3.3 Integração.....	48
3.3.4 Classe da Rede Neural	48
3.3.5 Estrutura da Rede	50
3.3.6 Parâmetros de Treino	51
3.4 IMPLEMENTAÇÃO	52
3.4.1 Técnicas e Ferramentas Utilizadas.....	52
3.4.2 Imagens de Impressões Digitais.....	58
3.4.3 Pré-Processamento em Imagens de Impressões Digitais	62
3.4.4 Treinamento da Rede Neural Artificial.....	63
3.4.5 Operacionalidade da Implementação	66
3.5 RESULTADOS E DISCUSSÃO	72
4 CONCLUSÕES.....	76
4.1 OBJETIVOS DO TRABALHO	76
4.2 FERRAMENTAS.....	76
4.3 PROTÓTIPO	76
4.4 REDE NEURAL ARTIFICIAL <i>FEEDFORWARD</i> MULTICAMADA.....	77
4.5 CONCLUSÕES FINAIS	78
4.6 EXTENSÕES	78
REFERÊNCIAS BIBLIOGRÁFICAS	80

0 INTRODUÇÃO

Neste capítulo são levantados alguns aspectos relativos ao tema, bem como a origem deste trabalho e as justificativas para o desenvolvimento do mesmo.

0.0 ORIGEM DO TRABALHO

A história da Identificação Humana é contada por historiadores, arqueólogos e exploradores que estudaram o Homem e a vida animal na crosta terrestre. Desde a mais remota Antigüidade, o Homem teve sua atenção voltada para a identificação. Os trogloditas usavam como meios de identificação, sinais diversos que serviam para marcar suas moradias. Aplicavam os desenhos palmares por meio de decalque sobre a argila, colorindo-o, e em seguida completavam o enfeite com cabeças dessecadas de animais ou até mesmo de inimigos abatidos em combates. Integrantes de tribos traziam desenhos coloridos pelo corpo, dentes de animais preso às orelhas, lábios e nariz. Tudo isto para que pudessem ser identificados uns pelos outros (APPOL, 2004).

Ao longo do tempo estes métodos de identificação evoluíram, principalmente pela necessidade crescente de se confirmar a identidade de um indivíduo, ou seja, de se ter certeza de que uma pessoa é quem realmente diz ser. Durante os últimos anos, as pessoas têm usado chaves, cartões, assinaturas e senhas para validar sua identidade. O grande problema destes artefatos é que podem ser esquecidos, roubados, perdidos, copiados, armazenados de maneira insegura e até utilizados por pessoas que não tenham autorização. Desta forma, não é surpreendente que um novo campo de atrações nas pesquisas recentes seja a biometria. Dentre as definições citadas no dicionário, biometria é “o ramo da ciência que estuda a mensuração dos seres vivos” (PROGLOBO, 1996).

Com base nos estudos da biometria e com a larga utilização de computadores, puderam-se criar sistemas computacionais confiáveis de identificação de indivíduos. Para tanto estes sistemas utilizam características biométricas que estudos científicos já comprovaram como sendo únicas. Entre estas características destaca-se, o reconhecimento através da face, da geometria das mãos, da assinatura, da voz, da íris, da retina e das impressões digitais associada ao acrônimo ID (AGUIAR, 2004).

Entre as características biométricas citadas acima, a mais utilizada é a identificação através das IDs, por ser mais barata e muito confiável. As IDs são reproduções fiéis dos desenhos digitais em alguma superfície plana. Desenhos digitais por sua vez, são as diversas linhas e entrelinhas formadas na pele dos dedos dos pés e das mãos dos seres humanos e de alguns outros animais dotados de mãos, tais como macacos, gambás, cotias, etc. No estudo das IDs destacam-se como sendo características importantes o(s) delta(s) e núcleo das mesmas. Pode-se definir o núcleo como sendo o centro de encontro das IDs, e o(s) delta(s) como sendo o(s) ângulo(s) ou triângulo(s) formados pelas linhas, localizados a direita e/ou esquerda da ID (KEHDY, 1968, p. 33).

Para extrair as características biométricas é necessário *hardware* especializado. No caso do desenho digital se faz necessário um sensor ou *scanner* biométrico. Os sistemas biométricos baseados em IDs utilizam como base a Datiloscopia agregada a técnicas e heurísticas próprias de computação para gerar um sistema que permita a autenticação da identidade de um indivíduo. O *hardware* normalmente apresenta para estes sistemas de identificação uma imagem digital, que por sua vez precisa ser tratada na maioria das vezes através de técnicas de pré-processamento de imagem, para que estes sistemas se tornem o mais confiáveis possível.

Para o desenvolvimento deste trabalho serão utilizadas técnicas de Redes Neurais Artificiais (RNA) para o reconhecimento das IDs e a comparação com outras IDs previamente armazenadas. Segundo Loesch (1996, p. 5), as RNA são uma tentativa de se copiar as redes neurais dos seres humanos. Nelas encontram-se também neurônios, mas artificiais, que interconectados, podem produzir resultados expressivos no reconhecimento de padrões. No caso deste trabalho, os padrões a serem identificados e comparados serão as IDs.

1.2 ÁREAS

As áreas relacionadas ao trabalho proposto são:

- a) Datiloscopia: o conteúdo desta área se faz necessário para definição das características das IDs, que serão amplamente utilizadas para o desenvolvimento do protótipo;
- b) Identificação Biométrica: seus conceitos vão ao encontro do protótipo, que é um sistema de identificação biométrica;

- b) Computação Gráfica: no que diz respeito à captura das imagens das IDs, e todo o pré-processamento destas imagem que servirão de entrada para a RNA;
- b) Inteligência Artificial (Redes Neurais Artificiais): a RNA implementada que será a principal tecnologia empregada no protótipo.

0.0 PROBLEMA

Atualmente, e cada vez mais, necessita-se ter o acesso restrito e controlado de pessoas a certos locais, que vão desde bancos, empresas internacionais, locais de jogos etc. Os sistemas de identificação tradicionais levantam enormes e conhecidos problemas na gestão de assiduidade e acesso. São comuns, por exemplo, os erros nos códigos a digitar e o esquecimento dos cartões. Além disso, são sistemas com uma elevada taxa de incerteza, já que os códigos podem ser divulgados a terceiros e os cartões podem ser emprestados ou até roubados (PROGLOBO, 1996).

Em razão desta dificuldade, criou-se os sistemas de identificação biométrica. Estes sistemas visam minimizar significativamente os problemas de controle de assiduidade e acesso, uma vez que utilizam as características físicas do indivíduo para identificação, e somente será possível a comprovação de sua identidade na presença física do mesmo.

0.0 JUSTIFICATIVAS

Inicialmente a identificação através das IDs eram feitas por indivíduos treinados, ou seja, uma ou mais pessoas tinham arquivos de IDs impressas em papel e, quando conhecido o nome do indivíduo, pegava-se sua ficha e comparavam-se as minúcias e outras características da ID já destacadas nesta ficha com a ID fornecida, utilizando para isto lupa e materiais como régua e lápis. Quando não se tinha o nome do indivíduo a ser identificado com sua ficha correspondente, como por exemplo, um crime sem testemunha, este trabalho se tornava demorado e de paciência, onde muitas vezes o indivíduo não podia ser identificado entre tantas fichas (GUMZ, 2002, p. 4).

Este processo de identificação tornou-se cada vez mais precário com o passar do tempo, e desde então os sistemas biométricos de identificação de indivíduos através das IDs vêm ganhando espaço no mercado de acesso continuamente. Entre os motivos geradores deste crescimento pode-se destacar que as IDs não são perdidas como cartões, nem divulgadas a

terceiros como senhas, e que os aparelhos de leitura de impressões digitais oferecem muito mais segurança que estes métodos, uma vez que é praticamente impossível falsificar uma ID humana (APPOL, 2004).

Neste trabalho pretende-se apresentar uma pesquisa referente às dificuldades que envolvem a criação de um sistema de identificação biométrica, propondo uma solução de identificação de indivíduos através do(s) delta(s) e núcleo das IDs. Através da utilização dos conceitos da Datiloscopia, dos conhecimentos dos sistemas biométricos, das técnicas de processamento de imagens digitais e das RNAs pretende-se criar um protótipo de identificação de indivíduos.

2.3 OBJETIVOS

O objetivo deste trabalho é desenvolver um sistema para identificação do(s) delta(s) e núcleo em IDs utilizando técnicas de RNAs.

Os objetivos específicos do trabalho são:

- a) realizar a identificação do(s) delta(s) e núcleo de IDs armazenadas em imagens *Bitmaps*;
- b) realizar a identificação do(s) delta(s) e núcleo de IDs de qualquer dedo das mãos.
- c) realizar a comparação destas IDs identificadas com outras previamente armazenadas.

2.4 ESTRUTURA DO TRABALHO

Este trabalho está organizado como descrito a seguir:

- a) no capítulo 1, é apresentada brevemente uma introdução ao assunto, bem como uma idéia a que este trabalho se propõe;
- b) no capítulo 2 serão apresentadas todas as áreas referentes a fundamentação teórica deste trabalho, sendo elas: Datiloscopia, Identificação Biométrica, Processamento de Imagens e Redes Neurais Artificiais. Neste capítulo são apresentados conceitos, características, técnicas, ferramentas e outros aspectos relevantes vinculados ao tema;

- c) no capítulo 3 será apresentado todo o processo de desenvolvimento do protótipo bem como as estruturas, algoritmos, técnicas, diagramas, fragmentos de códigos, etc, utilizados durante esta etapa;
- b) e, por último, no capítulo 4, serão apresentadas conclusões, análise de resultados, bem como sugestões para trabalhos futuros.

0 FUNDAMENTAÇÃO TEÓRICA

Nas próximas seções serão apresentados conceitos, técnicas e ferramentas mais relevantes ao desenvolvimento do trabalho, bem como o estado da arte sobre o tema e também trabalhos correlatos encontrados.

0.0 IMPRESSÕES DIGITAIS

Segundo APPES (2004) a Papiloscopia é o mais seguro método de identificação humana. A Papiloscopia divide-se em Datiloscopia, Quiroscopia e Poroscopia. A Datiloscopia é a ciência que estuda os desenhos papilares formados na polpa digital (ponta dos dedos). A Quiroscopia é ciência que estuda os desenhos formados pela palma das mãos. E a Poroscopia é a ciência que estuda os desenhos formados pela planta dos pés. Para o desenvolvimento deste trabalho será considerado apenas o estudo da Datiloscopia.

0.0.0 Datiloscopia

Segundo Tavares Júnior (1991, p. 19-20) alguns autores consideram o início do estudo da Datiloscopia no período pré-histórico, onde o Homem primitivo tinha o costume de marcar sua caverna com o desenho de suas mãos sobre a argila, passando após para países do Oriente, onde os homens eram obrigados a opor os dedos polegares impregnados de tintas em papéis para a emissão de documentos de divórcio. Mas foi entre os anos de 1664 e 1891 que a Datiloscopia teve seu estudo comprovado como ciência.

Em 1664 o anatomista italiano J. Marcello Malpighi fez estudos sobre as papilas da pele encontradas na palma das mãos e nas extremidades digitais. Seus estudos foram estendidos por vários pesquisadores, onde os que mais se destacaram por seu trabalho foram William Hershel que em 1.858 provou o postulado da imutabilidade; Arthur Kollmam que em 1.883 provou a formação dos desenhos digitais no sexto mês de vida fetal; Francis Galton que em 1.888 criou um sistema com trinta e oito tipos de IDs classificados em três grupos: arcos, presilhas e verticilos; Henry de Varigny que em 1.891 publicou um artigo sobre o sistema de Galton que foi traduzido para várias línguas; e Juan Vucetich, argentino que leu este artigo, e que criou uma extensão do sistema de Galton e implantou pela primeira vez a identificação de suspeitos de crime pela ID por volta de 1.891 com o nome de Icnofalangometria (TAVARES

JÚNIOR, 1991, p. 20-22). O termo Icnofalangometria foi modificado para Datiloscopia em 1894, por sugestão do Dr. Francisco Latzina (KEHDY, 1968, p. 37).

A Datiloscopia é a ciência que trata a identificação de pessoas através da comparação de suas IDs. O termo Datiloscopia deriva de dois elementos gregos *daktylos* = dedos e *skopêin* = examinar. A Datiloscopia divide-se em civil, criminal e clínica. A Datiloscopia civil trata da identificação das pessoas para expedição de documentos de identidade, tais como cédula de identidade, passaporte, etc. A Datiloscopia criminal reveste-se de três aspectos, a saber: indiciado em inquérito policial, expedição de documentos de idoneidade e aproveitamento das impressões papilares encontradas nos locais de crimes. Já a Datiloscopia clínica estuda as perturbações que se verificam nos desenhos digitais. Perturbações estas resultantes de algumas doenças, ou exercício de certas profissões (KEHDY, 1968, p. 25-27).

Segundo Kehdy (1968, p. 27) e APPES (2004), são os seguintes os postulados¹ da Datiloscopia:

- c) perenidade: é a propriedade que tem o desenho digital de existir desde o sexto mês de vida fetal até a putrefação cadavérica, quando se dá o descolamento da derme;
- c) imutabilidade: é a propriedade que tem o desenho digital de não mudar, desde o sexto mês de vida fetal até a putrefação cadavérica;
- a) variabilidade: é a propriedade que tem o desenho digital de variar de dedo para dedo e de pessoa para pessoa. Os gêmeos, quando do mesmo ovo, apresentam desenhos papilares semelhantes, mas nunca absolutamente iguais. A variedade é tão grande que em milhões e milhões de IDs já estudadas e fichadas em todo o mundo jamais se encontrou duas iguais;
- b) classificabilidade: é a propriedade que tem o desenho digital de poder ser classificado. Este por sua vez é classificado em tipo e subtipo.

0.0.0 Estudo da Pele

Pele é uma vasta membrana que recobre todo o corpo humano. Tavares Júnior (1991, p. 27) detalha que a pele é constituída por duas camadas: a derme e a epiderme. A derme é a parte principal e mais profunda da pele, e está localizada sob a epiderme. Na superfície da

¹ Postulado é uma proposição que é aceita sem demonstração.

pele ficam localizados pequenos relevos: as papilas. Estas existem nas superfícies palmares e plantares. Já a epiderme é uma fina membrana, transparente, que recobre a derme.

Segundo Kehdy (1968, p. 31) existem duas espécies de glândulas de secreção presentes na pele e que interessam no estudo da Dactiloscopia. São as glândulas sudoríparas e sebáceas. As glândulas sudoríparas existem em maior quantidade e estão presentes nas palmas das mãos, nas extremidades digitais e nas plantas dos pés. Já as glândulas sebáceas existem em maior quantidade na testa, no couro cabeludo, no nariz e nas orelhas, e produzem substâncias gordurosas. Estas glândulas e a secreção das mesmas é que permitem que sejam deixadas ID em locais de crimes, por exemplo.

Tavares Júnior (1991, p. 30) afirma que as papilas são situadas na derme e se espalham por todo o corpo, é por este motivo que pequenos cortes ou queimaduras leves não afetam os desenhos digitais. Define-se como desenho digital a combinação de cristas e sulcos papilares localizados na derme e espalhados pela epiderme, formando configurações diversas em IDs. E IDs por sua vez são a reproduções do desenho digital em um suporte liso e limpo.

APPOL (2004) afirma que, “o datilograma é produzido pelo entintamento da polpa digital ou por manchas de suor e gorduras sobre um suporte liso e polido, podendo ser também produzido por decalque em suporte mole que venha a endurecer. Em um datilograma, as linhas impressas representam cristas papilares (linhas pretas ou cinza escuro) e os espaços intercalados são os sulcos interpapilares (linhas brancas ou cinza claro)”.

Tavares Júnior (1991, p. 30) cita que as IDs compõem-se de: linhas pretas, linhas brancas, delta, ponto característicos, poros, linhas brancas albedactiloscópicas.

- b) linhas pretas: correspondem às cristas papilares;
- b) linhas brancas: correspondem aos sulcos interpapilares;
- b) delta: é o ângulo ou triângulo formado pelas cristas papilares;
- b) pontos característicos: são os acidentes que se encontram nas cristas papilares. Servem para identificar as impressões por serem permanentes;
- b) poros: são aberturas dos canais que expõem os produtos das glândulas sudoríparas e sebáceas;
- b) linhas brancas albedactiloscópicas: são formadas pela interrupção de duas ou mais cristas papilares, são formadas pelo enrugamento da pele.

Segundo Kehdy (1968, p. 34) as IDs com ausência de deltas, não possuem sistemas de linhas, as demais possuem três sistemas de linhas, conforme mostrados na Figura 1, sendo definidos como:

- b) região basilar: é formada pelo conjunto de linhas existentes entre a prega interfalangeana e a terceira linha abaixo do ramo descendente e ascendente do delta.
- b) região nuclear: é formada pelo conjunto de linhas que circunscvem o centro do datilograma, ou seguindo a diretriz superior até o ramo ascendente do delta.
- b) região marginal: é formada pelo conjunto de linhas do ápice e das laterais do datilograma até a linha imediata que acompanha a diretriz superior do delta.

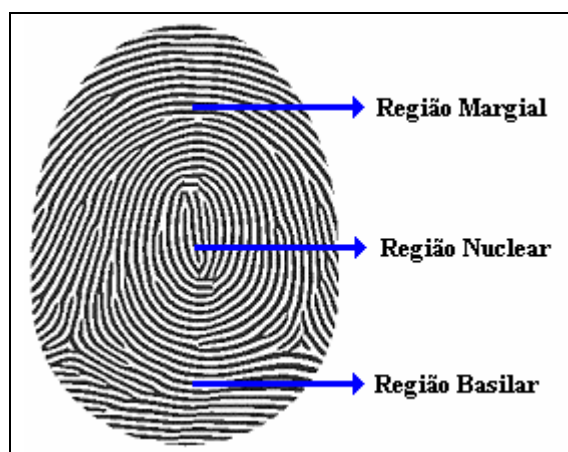


Figura 1 – Regiões dos sistemas de linhas.

As IDs são normalmente extraídas com papel, rolo e tinta, onde a superfície do dedo é molhada com o rolo entintado e imediatamente posta contra o papel. Assim são separados os indivíduos em fichas que podem ser decadatilares ou monodatilares (dez ou uma ID por ficha). Estas fichas são ordenadas por tipo de ID, quantidade e tipo de minúcias (KEHDY, 1968, p. 14).

0.0.0 Sistema Datiloscópico de Juan Vucetich

Foi por iniciativa de Félix Pacheco, que o Presidente da República Dr. Rodrigues Alves, em 05 de fevereiro de 1903, através do Decreto 4.764, introduziu a Datiloscopia como método mais simples e mais perfeito para identificação de indivíduos. Todo o processo de identificação seria subordinado à classificação dactiloscópica, de acordo com o Sistema

Vucetich, considerando-se, para todos os efeitos, a ID como prova mais concludente e positiva da identidade do indivíduo (APPES, 2004).

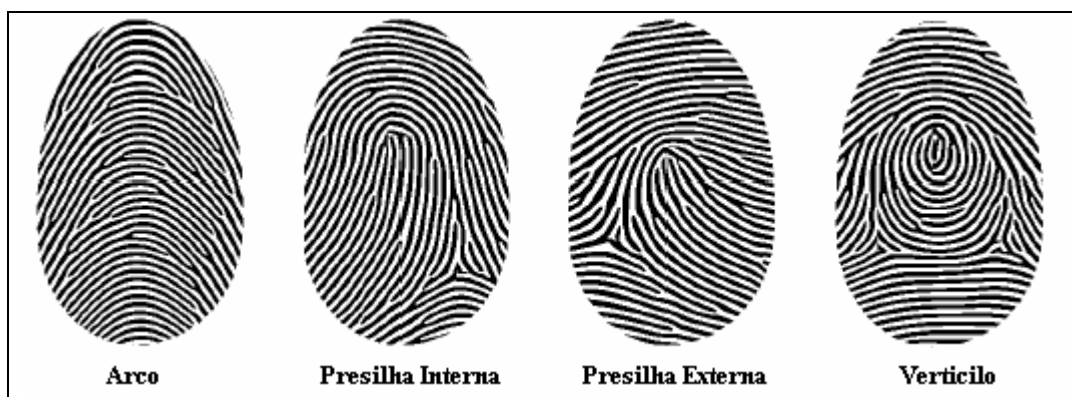
Segundo Tavares Júnior (1991, p. 37-38) e Kehdy (1968, p. 37), no sistema datiloscópico de Vucetich as IDs são classificadas nos tipos: arco, presilha interna, presilha externa e verticilo. O arco apresenta como características a ausência de delta, onde as linhas atravessam o campo da ID de um lado ao outro. A presilha interna tem um delta à direita do observador e as linhas nucleares correm para a esquerda do observador. Já a presilha externa tem um delta à esquerda do observador e as linhas nucleares correm para a direita do observador. E o verticilo tem dois deltas, sendo um à direita e outro à esquerda do observador, onde as linhas nucleares ficam encerradas entre os dois deltas, assumindo configurações variadas.

Os tipos fundamentais de Vucetich são representados abreviadamente por números e letras, a saber: Arco=A=1, presilha Interna=I=2, presilha Externa=E=3 e Verticilo=V=4. As letras são empregadas para os dedos polegares e os números para os demais dedos. Além dos símbolos acima, ainda são utilizados X quando há cicatriz e O para amputação, estes últimos dois símbolos são empregados para qualquer um dos dedos (KEHDY, 1968, p. 38).

Tavares Júnior (1991, p. 45) exhibe a fórmula datiloscópica como uma fração ordinária para se efetuar a classificação decadatilar, onde o numerador indica impressões da mão direita e o denominador da mão esquerda. Os símbolos da fórmula são dispostos na ordem natural dos dedos: polegar, indicador, médio, anelar e mínimo. Segundo Kehdy (1968, p. 40) existem 4^{10} combinações, o que resulta em 1.048.576 diferentes fórmulas datiloscópicas, que é o resultado dos 4 tipos fundamentais elevados aos dez dedos. Na fórmula 2-1 tem-se um exemplo de fórmula datiloscópica, mão direita: arco, verticilo, presilha externa, presilha interna e arco; e na mão esquerda: verticilo, arco, arco, presilha interna, arco.

$$\frac{A-4321}{V-1121} \quad (2-1)$$

Kehdy (1968, p. 40) afirma que os quatro tipos fundamentais de Vucetich não aparecem com a mesma frequência nos datilogramas. Através de estudos estabeleceu-se que 5% das IDs encontradas são do tipo arco, 60% do tipo presilhas e 35% do tipo verticilos. Na Figura 2 são apresentados os quatro tipos fundamentais das IDs no sistema Vucetich.



Fonte: Adaptado de Gumz (2002, p. 12).

Figura 2 – Tipos fundamentais de IDs no sistema de Juan Vucetich.

Em (KEHDY, 1968, p. 44) ainda são detalhados os subtipos das IDs, estes por sua vez são agregados aos tipos principais, a saber:

- b) arco: plano, angular, bifurcado à direita e bifurcado à esquerda;
- b) presilha interna: normal, invadida, dupla e ganchosa;
- b) presilha externa: normal, invadida, dupla e ganchosa; e
- b) verticilo: espiral, ovoidal, sinuoso e ganchoso.

Defrontam-se alguns casos onde o sistema de Vucetich apresenta tipos limites, quando uma impressão tem seu desenho encaixando-se em mais de um tipo. Os tipos limites são: arco e presilha interna, arco e presilha externa, arco e verticilo, presilha interna e verticilo e, presilha externa e verticilo (KEHDY, 1968, p. 36).

0.0.0.0 Minúcias

Segundo Costa (2000, p. 15), as minúcias são acidentes que se encontram nas cristas papilares e têm a finalidade de estabelecer a unicidade das IDs. Pode-se dizer que minúcias são terminações e bifurcações dos sulcos papilares.

Kehdy (1968, p. 61) e Tavares Júnior (1991, p. 32) afirmam que para se confirmar a identidade de uma ID, deverão ser coincidos no mínimo doze minúcias, as quais devem ser encontradas da mesma forma, localização e mesma quantidade nas duas impressões em

comparação. Também não pode haver minúcias que se encontram na impressão testemunha² e não estão presentes na impressão suspeita³.

0.0.0.0 Núcleo e Deltas

O núcleo de uma ID representa o “centro de gravidade” da impressão. Em IDs do grupo verticilo, o núcleo está localizado na parte superior da volta mais interna da linha datilar. Já em IDs dos grupos presilhas, o núcleo localiza-se no ombro da laçada mais interna da impressão (ANTHEUS, 2002).

O delta é o ângulo ou triângulo formado pelas cristas papilares, pela brusca divergência das linhas paralelas ou pela bifurcação de uma linha simples. Sua principal função é determinar o tipo da ID. Os deltas representam o ponto de divergência das linhas datilares mais internas que envolvem ou tendem a envolver a região nuclear de uma ID. Verticilos apresentam dois deltas, Presilhas Externas apresentam um delta à esquerda do observador e as Presilhas Internas um delta à direita do observador. O Arco normalmente não possui delta (APPOL, 2004). Na Figura 3, são mostrados núcleo e deltas.

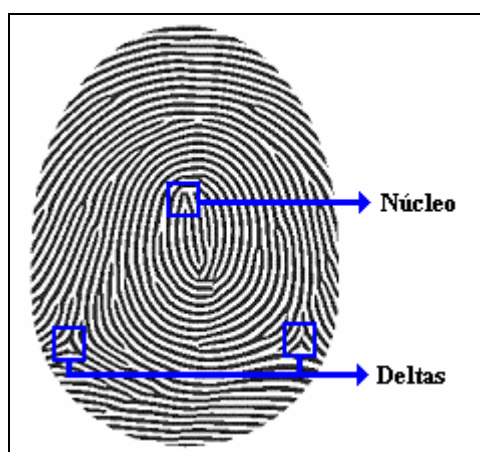


Figura 3 – Em destaque núcleo e deltas de uma ID.

² Impressão testemunha é a encontrada em local de crime.

³ Impressão suspeita é a tomada dos dedos do suspeito ou encontrada no arquivo datiloscópico.

0.0 BIOMETRIA

Segundo Bressan (2002, p. 3), todos os métodos de identificação de indivíduos são baseados em três conceitos fundamentais: segurança baseada em algo que o indivíduo possui, como uma chave de um cofre, segurança baseada em algo que o indivíduo sabe, como uma senha, e segurança baseada em quem o indivíduo é, como sua ID. A identificação através deste último conceito é chamada de identificação biométrica, pois usa características estudadas pela biometria.

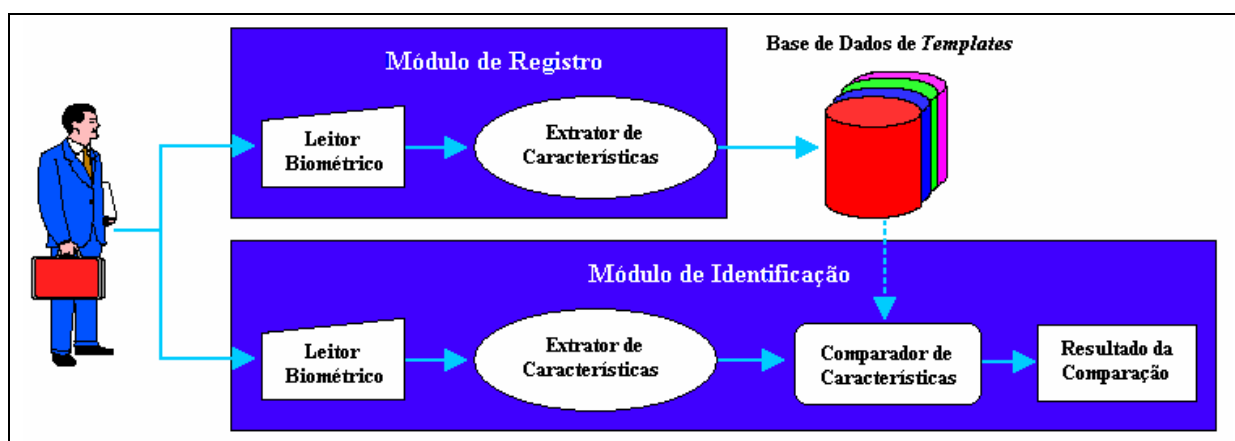
Como já citado anteriormente, segundo o dicionário, biometria é “o ramo da ciência que estuda a mensuração dos seres vivos”, mas no mundo da segurança, biometria se refere aos métodos automatizados para identificação de pessoas com base em suas características físicas, fisiológicas ou em seus aspectos comportamentais (AGUIAR, 2004).

0.0.0 Sistemas de Identificação Biométrica

Segundo Pankanti (2000, p. 2), um sistema biométrico é um sistema de reconhecimento de padrões capaz de estabelecer a identidade de um indivíduo através de características físicas ou comportamentais. Para que estas características possam ser utilizadas em um processo de identificação de usuário é necessária a utilização de tecnologias bastante avançadas. Bressan (2002, p. 3), enfatiza que qualquer característica humana, seja ela física, fisiológica ou comportamental, pode ser usada para identificação biométrica, desde que satisfaçam aos seguintes requisitos:

- b) universalidade: todas as pessoas devem possuir a característica a ser medida;
- b) singularidade: indica que esta característica não pode ser igual em pessoas diferentes;
- b) permanência: significa que a característica não deve variar com o tempo;
- b) desempenho: precisão da identificação, os recursos requeridos para conseguir uma precisão de identificação aceitável e ao trabalho ou fatores ambientais que afetam a precisão de identificação;
- b) aceitabilidade: indica o quanto as pessoas estão dispostas a aceitar os sistemas biométricos;
- b) proteção: facilidade ou dificuldade de enganar o sistema com técnicas fraudulentas.

Segundo Hong (1998, p. 3), os sistemas biométricos são divididos em duas partes: o módulo de registro e o módulo de identificação. No módulo de registro as características são obtidas por um leitor biométrico e processadas através de um módulo extrator de características gerando um *template* que será armazenado. Já o módulo de identificação deve fazer a captura das características biométricas no momento do acesso, gerando também uma representação através do módulo extrator, equivalente as armazenadas através dos *templates*. E finalmente, o resultado do módulo de identificação é alimentado ao módulo comparador de características que o confronta contra os *templates* armazenados para estabelecer a identidade. A tradução de *template* é modelo ou gabarito, mas como as palavras modelo e gabarito têm muitos significados em nosso idioma, a partir deste ponto será utilizado o termo *template* para indicar as características biométricas armazenadas pelo módulo de registro. Na Figura 4, apresenta-se uma representação macro de um sistema de identificação biométrica.



Fonte: Adaptado de Gumz (2002, p. 15).

Figura 4 – Representação macro de um sistema de identificação biométrica.

Como exemplos de aplicações biométricas têm-se as forenses que incluem identificação criminal, segurança de prisões e identificação de cadáveres; as civis que agregam identificação nacional, carteira de motorista, saúde, travessia de fronteiras; e as comerciais em que se destacam comércio eletrônico e serviços bancários como caixa automático, cartão de crédito e controles de acesso (GUMZ, 2002, p. 16).

0.0.0 Sistemas de Identificação Biométrica através das Impressões Digitais

Os sistemas de identificação biométrica por IDs são associados a sigla AFIS, do inglês *Automatic Fingerprint Identification System* ou sistema de identificação automática de IDs.

Segundo Bressan (2002, p. 8), os sistemas de identificação através de IDs, analisam basicamente a posição das minúcias, mas os sistemas mais modernos, além disto, estão verificando também outras características, tais como arcos, voltas, posição do(s) delta(s), núcleo, etc. Tudo isto para que estes sistemas tenham seus percentuais de acerto cada vez mais próximos de 100%.

Os *templates* citados anteriormente não necessariamente gravam a imagem da ID coletada pelo leitor biométrico, na maioria das vezes usam-se características específicas dos desenhos das impressões, como o número de linhas, redemoinhos, etc., de modo que o processo de comparação seja feito de uma forma mais rápida e precisa (BRESSAN, 2002, p. 3).

Segundo Bressan (2002, p. 8), existem vários tipos de sensores para identificação de ID. Alguns trabalham com calor, outros com pressão, etc., mas os mais comuns usam um pequeno CCD (que é o componente básico de um scanner) para obter imagens digitais da impressão. Estes tipos de sensores também são mais confiáveis que os de temperatura e pressão, pois não dependem das condições ambientais durante a utilização do sensor.

Mais detalhadamente podem-se citar os três tipos de leitores de IDs:

- b) ópticos: o dedo é colocado sobre uma plataforma de vidro e uma imagem do dedo é capturada. Estes dispositivos tornaram-se pequenos e baratos;
- b) ultra-som: o dedo é colocado sobre uma plataforma de vidro e uma varredura de ultra-som é efetuada;
- b) baseados em chip: o usuário coloca seu dedo direto em um chip de silício.

Hong (1998, p. 15) e Bressan (2002, p. 9) destacam que um problema ligado ao *AFIS* encontra-se na sua aceitabilidade pelo usuário civil, porque IDs têm sido associadas a investigações criminais e trabalho policial, fazendo com que estes usuários se sintam desconfortáveis, e também que a tecnologia requer uma grande quantidade de recursos computacionais além de que uma pequena parte da população não possibilite a identificação pelas IDs devido a razões genéticas, de idade, de ambiente ou ocupacionais como as estudadas pela datiloscopia clínica.

0.0 PROCESSAMENTO DE IMAGENS

O sentido da visão pode ser considerado o meio mais eficiente que os seres humanos dispõem para captar informações do ambiente que os cerca. Prova disso é a necessidade crescente de automatizar algumas atividades de nosso cotidiano e que pode ser considerada umas das responsáveis pelo grande crescimento das ferramentas de processamento de imagens. Estas ferramentas podem realizar tarefas que variam desde um simples aprimoramento de uma imagem até a compreensão de mecanismos complexos (FACON, 1993, p. 6).

A área de Processamento Digital de Imagens teve início em meados da década de 60, com a motivação criada pelos programas espaciais da NASA nos EUA. O aumento da capacidade de memória e velocidade dos computadores contribuiu para dar impulso à nova tecnologia, que passou a ser aplicada em outras áreas como: medicina, microscopia, ótica eletrônica, transmissão digital de sinais de TV ou fac-símile e em sensores visuais de robôs. Em 1975, a invenção do tomógrafo computadorizado por G. N. Hounsfield, abriu amplas perspectivas de utilização das técnicas digitais de imagens radiográficas (OSÓRIO, 1991, p. 24).

0.0.0 Imagem Digital

Imagens digitais podem ser definidas como sendo representações digitais de imagens capturadas do mundo real, através de sensores. Segundo Gonzalez e Woods (1992, p. 7), para esta captura, são necessários dois elementos. O primeiro é um dispositivo físico que seja sensível a uma banda de espectro de energia eletromagnética, tais como raios-X, ultravioleta, banda visível ou banda infravermelha, e que produza um sinal elétrico de saída proporcional a um nível de energia percebida. O segundo é chamado de digitalizador, que é um dispositivo que converta a saída elétrica para forma digital.

Segundo Gonzalez e Woods (1992, p. 5), uma imagem digital pode ser considerada como sendo uma matriz cujos índices de linhas e colunas identificam um ponto da imagem, e o correspondente valor do elemento da matriz identifica o nível de cor naquele ponto. Cada elemento dessa matriz é chamado de elemento da imagem ou elemento da figura, originários do inglês *pixel* ou *pels* que são abreviações de *picture elements*.

A qualidade das imagens capturadas e exibidas é definida pelo *dpi*, do inglês *dots per inch* ou pontos por polegada. O *dpi* trata da medição da quantidade de pontos em uma imagem digital por polegada, ou seja, quanto mais *pixels* a imagem apresentar por polegada, melhor será sua qualidade e definição (GUMZ, 2002, p. 20).

A dimensão de uma imagem digital, poderá ser unidimensional (1D = variável), bidimensional (2D = ponto), tridimensional (3D = ponto no espaço), quadridimensional (4D=ponto no espaço tempo) ou até N-dimensional. As dimensões superiores a 2D são mais utilizadas para visualização científica (GUMZ, 2002, p. 21).

0.0.0.0 Definição de Cores

Quanto a coloração, as imagens podem ser classificadas em dois grupos: monocromáticas e coloridas. As monocromáticas são imagens binárias, e apresentam somente duas cores, preto ou branco. As imagens coloridas podem ser representadas através de vários modelos, a saber: RGB (Red Green Blue), CMY (Cyan Magenta Yellow), HSI, RYB, YIQ, etc (BROWN; SHEPHERD, 1995, p. 106).

O sistema mais usado em monitores de vídeo coloridos é o RGB. No modelo RGB cada cor aparece nos seus componentes espectrais primários de vermelho verde e azul. Imagens no modelo RGB consistem em três planos de imagens independentes, um para cada cor primária (GONZALEZ; WOODS, 1992, p. 160).

O ser humano possui em seu sistema visual três tipos de sensores capazes de identificar três faixas diferentes de “espectros de energia”. Estas faixas correspondem as tonalidades de Vermelho (Red), Verde (Green) e Azul (Blue). Logo o ser humano vê na realidade a combinação resultante da mistura destas três cores básicas. Através de testes realizados com o ser humano chegou-se a conclusão que a utilização de 256 variações diferentes de intensidade em cada uma das cores básicas é capaz de gerar um número de cores superior a capacidade visual do ser humano, ou seja, fica praticamente impossível distinguir entre duas cores “vizinhas” (CASACURTA et al, 1998, p. 9-10).

Segundo Casacurta et. al. (1998, p. 10), no sistema RGB, o valor (0, 0, 0) equivale a cor preta com intensidade zero nas três componentes. O valor (255, 255, 255) equivale a cor branca onde as três componentes estão presentes com a sua intensidade máxima. As diferentes

combinações entre RGB serão capazes de gerar qualquer tipo de cor, sendo que se as três componentes tiverem sempre valores iguais será definida uma escala de tons de cinza do preto ao branco, é a chamada *gray scale*.

0.0.0.0 Formato de Imagens

Existem diferentes formatos de arquivos para o armazenamento de imagens, uma vez que se têm várias classes diferentes de representações de imagens. O armazenamento da imagem envolve basicamente três elementos principais: a forma como a imagem está representada, o tipo de compactação empregado e o cabeçalho contendo as informações a cerca desta imagem (resolução, número de cores, classe da imagem, *palette*, compactação, etc). Um mesmo tipo de arquivo pode inclusive permitir o armazenamento de diferentes classes de imagens e também permitir a utilização de vários métodos de compactação (CASACURTA et al, 1998, p. 14).

Segundo Facon (1993, p. 28), quanto maior o número de aplicações em que uma imagem digital pode ser tratada, diz-se que maior é a sua padronização. Normalmente programas gráficos podem ser classificados pela forma com a qual armazenam e apresentam as imagens. Para esta abordagem há duas categorias: formato de varredura e formato vetorial. Entre os formatos de imagens digitais mais utilizados estão: BMP, JPG, GIF, TIFF, PCX, CGM, ICO, RLE, TARGA, PostScript, entre outros (BROWN; SHEPHERD, 1998, p. 73).

0.0.0 Técnicas de Processamento de Imagens

Segundo Gonzalez e Woods (1992, p. 11), o processamento de imagens digitais geralmente é expresso de forma algorítmica e implementado em software, que uma vez aplicados a uma imagem geram um aprimoramento ou correção da mesma, no que diz respeito a eliminação de ruídos, realce de imagem, restauração de regiões da imagem, etc. Este processo normalmente é direcionado para uma solução, deste modo técnicas que funcionam em um problema, podem se mostrar totalmente inadequadas para outro problema.

Entre técnicas de processamento de imagens podem ser citadas algumas técnicas como a de restauração que busca compensar distorções específicas, normalmente geradas no momento de aquisição. Quando se pode identificar experimentalmente a função que representa a deformação ou construir um modelo matemático adequado, é possível buscar a

função inversa e aplicá-la sobre a imagem deformada. Em aplicações como: correção de foco, Imagens borradas por movimento. Já a Segmentação procura isolar regiões de *pixels*, e aplicar operações para “extração de atributos”. A operação de segmentação mais comum é a limiarização por um tom de corte. Tudo que está acima deste tom vira branco, tudo que está abaixo vira preto, obtendo-se uma imagem binária (SCURI, 1999, p. 55).

Gumz (2002, p. 22), detalha alguns filtros conhecidos, sendo eles, filtro de média (*mean filter*), filtro da mediana (*median filter*), filtro mínimo-máximo, suavização controlada por gradiente, suavização gaussiana (*gaussian smoothing*), suavização conservativa (*conservative smoothing*), remoção de manchas *Crimmins* (*Crimmins speckle removal*), filtros de frequência passa-baixa (*low-pass*) e passa-alta (*high-pass*), *laplacian of gaussian filter* e *unsharp filter*.

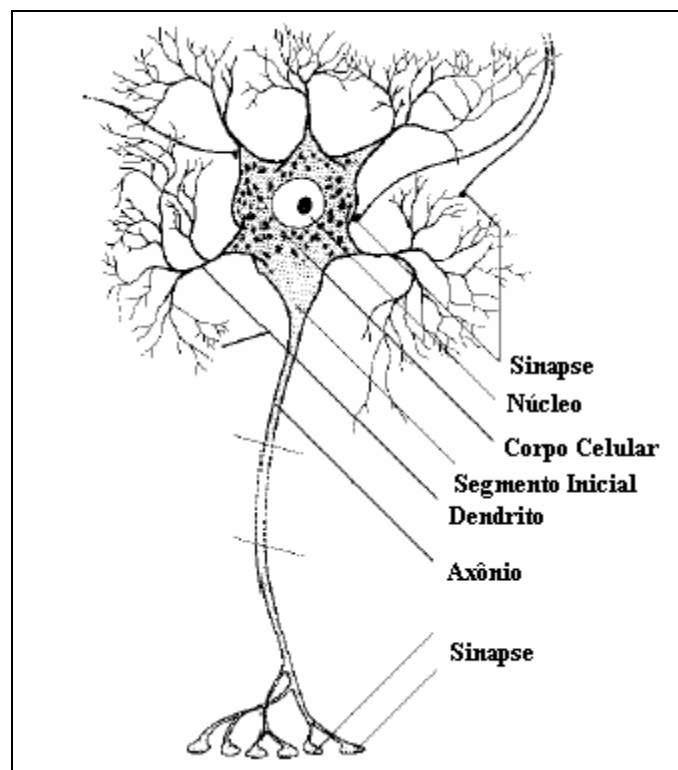
0.0 REDES NEURAIAS

Silva (1999, p. 16) comenta que o cérebro humano é o dispositivo mais complexo conhecido pelo Homem, e que a 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 (RNA).

0.0.0 Sistema Nervoso e Neurônio Biológico

O cérebro é uma das partes do sistema nervoso que provê uma grande quantidade de controle sobre as funções do corpo. Nele existem dois tipos de células: as glias, responsáveis pelo suporte e nutrição e os neurônios que são as células especializadas na atividade nervosa (LOESCH, 1996, p. 8). Tafner, Xerez e Rodrigues Filho (1996, p. 32) afirmam que o cérebro possui cerca de 10 bilhões de neurônios, e que cada neurônio é capaz de ter até 10.000 conexões com outros neurônios ou até mesmo com partes do próprio neurônio.

O neurônio biológico possui um corpo e diversas ramificações, que são chamadas de dendritos, que por sua vez conduzem sinais de entrada do neurônio para o corpo da célula. A partir do corpo da célula surge um filamento comprido denominado axônio, na extremidade deste filamento existem micro-ramificações, denominados terminais dos neurônios, que têm por objetivo transmitir um sinal do corpo para outra extremidade do neurônio. Os terminais dos neurônios são conectados aos dendrites de outros neurônios, cuja conexão dá-se o nome de sinapse (LOESCH, 1996, p. 8). Na Figura 5, é feita uma representação de um neurônio biológico.



Fonte: Osório e Bittencourt (2000, p. 8).

Figura 5 – Representação de um neurônio biológico.

Segundo Loesch (1996, p. 9) e Silva (1999, p. 17), os estímulos entram nos neurônios através das sinapses de outros neurônios, conectadas aos dendrites deste primeiro, e quando chegam ao corpo da célula, ocorre um processo de integração (soma), deste estímulos. Ao resultado desta soma pode ser gerado um impulso elétrico que começa no axônio e se propaga para todas as sinapses. O processo de controle da produção destes impulsos elétricos se dá normalmente através de uma função de limiar que produz uma saída, caso esta soma supere este limiar.

0.0.0 Redes Neurais Artificiais

Segundo Loesch (1996, p. 15), a história das redes neurais artificiais é recente, com pouco mais de meio século de estudos. Estes iniciaram por volta de 1940, quando as primeiras simulações foram feitas com papel e lápis por Donald Hebb e Karl Laschey. Seguidos por McCulloch e Pitts que propuseram uma modelagem matemática para os neurônios biológicos através dos fatos que: neurônios comportam-se como somadores algébricos; adicionam entradas excitatórias ou subtraem entradas inibitórias; neurônios possuem uma propriedade de limiar, entre outras.

Loesch (1996, p. 5) define as RNAs como sendo 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. Os benefícios obtidos pela utilização de uma RNA se dão graças a sua massiva estrutura distribuída e paralela, e sua habilidade de aprender e generalizar saídas de entradas que não estavam amostradas durante o seu treinamento. Estes benefícios permitem que ela resolva problemas complexos (HAYKIN, 2001, p. 4).

Existem vários tipos de RNA, 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). As RNA podem ser algumas vezes superiores a outros métodos porque os dados sobre os quais se efetuam conclusões são confusos, caóticos e inconsistentes (MASTERS, 1993, p. 6-7).

Uma RNA pode ser auto-associativa ou hetero-associativa. O mapeamento de entrada, ou auto-associação, permite a uma RNA ser treinada para correlacionar um padrão com ele mesmo. E o mapeamento de saída, ou hetero-associação, associa um padrão treinado com outro. Elas podem também aproximar funções já que a partir de uma entrada produzem uma saída o que os qualifica como uma função (MASTERS, 1993, p. 24, 68).

Gumz (2002, p. 24) comenta que, para que as RNAs possam resolver os problemas para o qual foram projetadas, as mesmas devem ser treinadas. Toda 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

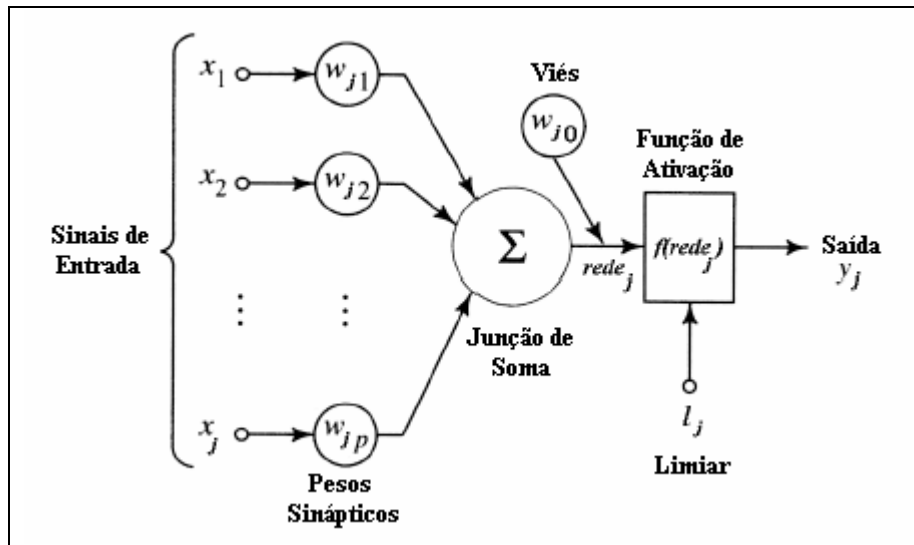
neurônios (ajuste sináptico) e configurar os valores de saída para cada caso verificando a quantidade de erro apontado por um cálculo sobre o valor de saída do neurônio.

0.0.0.0 Neurônio Artificial

Em Tafner, Xerez e Rodrigues Filho (1996, p. 56), encontram-se outros nomes para os neurônios artificiais, como “Elemento de Processamento” ou “Nodo”. Segundo o autor todo neurônio artificial tem um funcionamento muito parecido com o neurônio biológico, uma vez que todo neurônio artificial possui um ou mais sinais de entrada, e apenas um sinal de saída. Estas entradas dos neurônios artificiais podem ser comparadas aos estímulos recebidos pelo neurônio biológico.

Um importante atributo dos neurônios artificiais são os pesos associados ao mesmo. Os pesos (*weight*) são valores que representam o grau de importância que uma determinada entrada possui em relação ao neurônio. Sempre que as entradas são apresentadas ao neurônio, elas são multiplicadas pelos valores dos pesos, os resultados destas multiplicações são somados e apresentados a uma função de ativação (TAFNER; XEREZ; RODRIGUES FILHO, 1996, p. 56-58). O neurônio é ativado se, e somente se, o resultado da função de ativação alcançar um valor estipulado, chamado de limiar. Pode existir também um valor de entrada adicional, o *viés* ou *bias*, que é determinado no treinamento, com o objetivo principal de modificar o valor de limiar da função de ativação (LOESCH, 1996, p. 20).

Resumindo, um neurônio é formado por um conjunto de sinapses que contém os pesos, um somador para calcular os sinais de entrada multiplicados aos pesos (soma ponderada), e uma função de ativação para limitar a amplitude do sinal de saída para um valor numérico finito (HAYKIN, 2001, p. 36). A Figura 6 ilustra um neurônio artificial, onde os sinais de entrada $x_1..x_j$ podem ser saídas de outros neurônios; as variáveis $w_{j1}..w_{jp}$ armazenam os valores dos pesos sinápticos diretamente responsáveis pela saída esperada do neurônio após somatório ponderado; o valor w_{j0} é o viés que será somado a *rede_j*; *rede_j* é o resultado da soma ponderada da entrada; a função de ativação $f(rede_j)$ é a responsável pela ativação com a resposta da RNA sobre dada entrada se alcançado o valor de limiar l_j o qual impõe os valores limites máximos e mínimos que resultará numa saída y_j .



Fonte: Adaptado de Haykin (2001, p. 36).

Figura 6 – Representação do funcionamento de um neurônio artificial.

Segundo Masters (1993, p. 79-82), a saída do neurônio é calculada como:

$$rede_j = (w_{j0} + \sum_{i=1}^n x_i * w_{ji}) \quad (2-2)$$

aplicada em

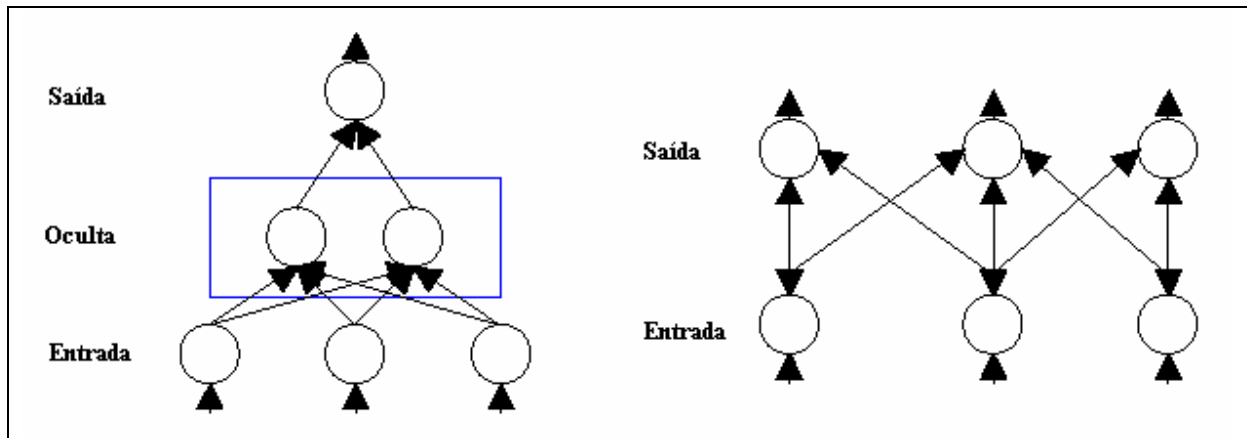
$$saída = f(rede_j) \quad (2-3)$$

Onde os pesos w_{ji} são primariamente responsáveis pelas características operacionais da rede, pela já conhecida soma ponderada com as entradas x_i . A função de ativação $f(rede_j)$ calcula o valor da saída do somatório ponderado de entradas da $rede_j$ e, w_{j0} é o valor do viés.

Segundo Tafner, Xerez e Rodrigues Filho (1996, p. 63), os neurônios artificiais sempre são dispostos em camadas, toda rede neural apresenta uma camada de entrada e uma camada de saída. Algumas RNA podem apresentar camadas intermediárias ou ocultas como são chamadas, estas sempre se situam entre a camada de entrada e a camada de saída e não têm contato com o mundo externo. As camadas de entradas apenas recebem os valores de entradas e repassam para as camadas seguintes, as camadas seguintes sendo ocultas ou de saída é que realizam o processamento da rede.

As ligações entre os neurônios das RNA podem ter diferentes fluxos, sendo *feedback* (realimentação), quando o sinal de saída de um neurônio servir de entrada para um outro, ou

para o mesmo neurônio, ou *feedforward* quando o fluxo seguir apenas em uma direção. Na Figura 7, apresenta-se os dois tipos de fluxo.



Fonte: Gumz (2002, p. 27).

Figura 7 – Representação dos Fluxos das RNA.

0.0.0.0 Rede Neural Artificial *Feedforward* Multicamada

Loesch (1996, p. 67) afirma que as RNA *feedforward* multicamadas (RNA-FM), ou *MLP* (*Multi Layer Perceptron - Perceptrons* de múltiplas camadas) como também são chamadas, são as mais utilizadas e difundidas dentre as RNAs. As *MLP* são uma extensão dos primeiros *Perceptrons*, também são hetero-associativas e podem possuir uma ou mais camadas ocultas, porém possuem capacidades muito mais potentes e genéricas que seu primeiro modelo. Segundo Welstead (1994, p. 59) estas são as RNAs mais utilizadas, onde estimativas apontam para 90% de todas as aplicações de RNA implementadas.

Numa RNA *feedforward* todos os neurônios de uma determinada camada são plenamente conectados aos neurônios da camada seguinte. Na Figura 7, no lado esquerdo tem-se uma RNA-FM, com uma camada de entrada, uma camada oculta e uma camada de saída. As RNA-FM podem apresentar mais de uma camada oculta, mas é conhecido que uma camada é suficiente para resolver problemas determinísticos (MASTERS, 1993, p. 79). A camada oculta além de não ter contato com o mundo externo tem a função de mediar a entrada externa e a saída da rede (HAYKIN, 2001, p. 19).

Haykin (2001, p. 183) enfatiza que este modelo de RNA têm sido aplicado com sucesso para resolver problemas difíceis através do seu treinamento supervisionado com um

algoritmo muito conhecido como *algoritmo de retropropagação de erro (backpropagation)*. Normalmente uma RNA-FM consegue aprender uma função ensinada. Se esta apresentar problemas, não é devido ao modelo em si, mas por um treinamento insuficiente, número pequeno de neurônios na camada oculta ou tentativa de aprender uma função que não é determinística (MASTERS, 1993, p. 87).

Segundo Loesch (1996, p. 77), apesar de muitas aplicações bem sucedidas, inconvenientes podem ocorrer durante o processo de treinamento, onde problemas complexos podem requerer dias e até semanas para que a RNA aprenda. Mesmo com todo este tempo de treinamento um RNA pode não aprender todos os pares de treinamento que lhes foram apresentados.

0.0.0.0 Funções de Ativação

A função de ativação de um neurônio é uma função não-linear que quando aplicada as entradas de um neurônio determina a saída do mesmo. As faixas de valores de algumas funções são limitadas de 0 a 1 enquanto outras de -1 a $+1$. As funções de ativação mais usadas são a função sigmóide binária, a função sigmóide bipolar, a função tangente hiperbólica e a função arco-tangente escalar (GUMZ, 2002, p. 39).

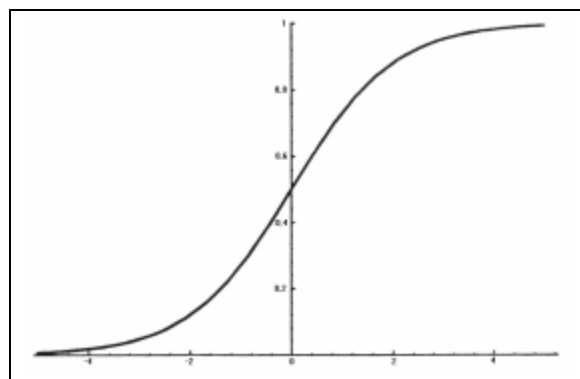
A maioria dos modelos de RNA-FM usam a função de ativação logística sigmóide binária (formato de S), que é representada na Figura 8. Ela pode ser definida como uma função real cujo domínio são os números reais, a derivada é sempre positiva e a faixa de valores é limitada (MASTERS, 1993, p. 81). A função de ativação logística sigmóide dá-se por:

$$f(x) = 1 / (1 + e^{-x}) \quad (2-4)$$

cuja derivada é

$$f(x) = f(x) * (1 - f(x)) \quad (2-5)$$

onde o valor máximo usado é + 1,0 e o valor mínimo é 0,0. Lembrando sempre que as funções sigmóides nunca alcançam seus teóricos mínimo e máximo, geralmente os neurônios são considerados totalmente ativados por volta de 0,9 e desativados por volta de 0,1. Onde se pode dar como entrada numa rede os valores extremos de 1,0 e 0,0; mas nunca tentar treinar a rede para alcançá-los. Em muitos casos têm-se verificado que a forma exata da função tem pouco efeito no poder final da rede, embora possa ter impacto significativo na velocidade de treino (MASTERS, 1993, p. 81).



Fonte: Masters (1993, p. 81).

Figura 8 – Função logística de ativação sigmóide binária.

0.0.0.0 Treinamento e Aprendizado

Segundo Tafner, Xerez e Rodrigues Filho (1996, p. 65), o treinamento das RNAs acontece basicamente de duas formas, sendo por aprendizado supervisionado e aprendizado não-supervisionado. No aprendizado supervisionado são apresentados a RNA pares de entrada e saída, ou seja, um conjunto de entradas e um conjunto de saídas desejadas para cada entrada. Já no aprendizado não-supervisionado não existem saídas desejadas, para o treinamento da rede são usados apenas os valores de entradas, e é através destes valores que a rede se organiza de modo a classificar as entradas.

A maioria das aplicações das RNAs são treinadas através do aprendizado supervisionado, onde o objetivo principal de treinar uma rede é ajustar seus pesos de tal forma que para um vetor de entrada produza-se um vetor desejado de saída, onde para cada entrada amostrada, emparelha-se uma saída (LOESCH, 1996, p. 71).

0.0.0.0.0 Aprendizado por Retropropagação (*backpropagation*)

O método de treino por retropropagação, ou *backpropagation*, foi o primeiro método de treino para a RNA-FM. Este método consiste basicamente de duas etapas executadas através das camadas da rede: um passo para frente a propagação e um passo para trás, a retropropagação. Na sua forma básica, a retropropagação é chamada por analistas numéricos de algoritmo de inclinação de gradiente, que determina a direção de um passo de cálculo.

Na propagação um vetor de entrada é aplicado aos pesos da rede e seu efeito é propagado para frente camada por camada até a camada de saída, produzindo um conjunto de valores de saídas como resposta da rede. Após é verificado se a resposta desejada da rede era igual a esperada de acordo com o vetor de saída, caso não seja é calculado um erro e este é retropropagado para correção de todos os pesos da rede (HAYKIN, 2001, p. 184). Este ajuste dos pesos é executado de acordo com uma taxa de aprendizado, em que se for muito pequena irá convergir muito devagar e se for muito grande, irá saltar e nunca irá convergir (MASTERS, 1993, p. 100).

Primeiramente o treino começa com todos os pesos da RNA iniciados com pequenos valores não-zero, freqüentemente gerados randomicamente ou por algoritmos específicos como o fortalecimento simulado (*simulated annealing*). Então pares de treinamento da coleção de amostras são apresentados a rede, um de cada vez (MASTERS, 1993, p.111). Uma medida do erro é feita pela rede, através da comparação de seus valores de saídas, após é calculado um valor que irá auxiliar na correção dos pesos da rede, designado como δ^4 , e por último os pesos da rede são atualizados de um modo a reduzir o erro. Todo este processo de apresentar as entradas à RNA, calcular as ativações das camadas, calcular os valores dos deltas e o erro, é chamado de *epoch*. As medidas de erro mais conhecidas são: erro quadrado médio (*mean square error*), erro absoluto médio (*mean absolut error*), erro de *Kalman-Kwasny*, entropia cruzada (*cross entropy*), erro máximo quadrado (*maximum squared error*), *mean log error*, além de outros (MASTERS, 1993, p. 40-49).

A medida mais comum do erro é dada pelo erro quadrado médio ou EQM, nas ativações de saída. A correta ativação alvo do neurônio de saída j é designada como d_j , e a

⁴ É importante salientar que quando houver referência ao termo “delta”, o mesmo pode significar um ponto característico da ID, ou o valor utilizado no algoritmo de aprendizado da rede. Para tanto deve-se analisar o contexto em que o termo encontra-se.

ativação observada é y_j . Se existirem n neurônios de saída, o erro para a simples apresentação é

$$\varepsilon = \sum_{j=1}^n (d_j - y_j)^2 \quad (2-6)$$

A fórmula para calcular o delta dos neurônios difere entre a camada de saída e as camadas ocultas. Para simplificar o cálculo pode-se dividir as fórmulas sendo o cálculo do erro da fórmula 2-7 para a camada de saída e 2-8 para as camadas ocultas.

$$\varepsilon_j^{(k)} = d_j - y_j \quad (2-7)$$

$$\varepsilon_j^{(k)} = \sum_{i=1}^{M_{k+1}} (\delta_i^{(k+1)} * w_{ij}^{(k+1)}) \quad (2-8)$$

Aplicando as fórmulas 2-7 e 2-8 na fórmula 2-9 tem-se o cálculo do valor do delta na camada k .

$$\delta_j^{(k)} = \varepsilon_j^{(k)} * f'(rede_j) \quad (2-9)$$

Onde k é o número da camada atual; $\delta_j^{(k)}$ é o valor do delta calculado na camada k ; w_{ij} é o peso que conecta o neurônio j desta camada oculta com o neurônio i da camada anterior. É importante lembrar que na fórmula 2-8 o $\delta_i^{(k+1)}$ refere-se ao valor do delta calculado na camada seguinte e que é utilizado para o cálculo do delta da camada atual.

Após realizar todo este cálculo é realizado o ajuste de todos os pesos da RNA, conforme a fórmula 4-5.

$$w_j^{(k)} = w_j^{(k)} + 2\mu \delta_j^{(k)} * x_j^{(k)} \quad (2-10)$$

Onde $w_j^{(k)}$ é o peso a ser ajustado do neurônio j na camada k ; μ é o valor da taxa de aprendizado; $\delta_j^{(k)}$ é o valor do delta calculado na camada k ; e o $x_j^{(k)}$ é o valor de entrada do neurônio j na camada k (LOESCH, 1996, p. XX).

2.4.2.5 Escolha da Estrutura das Redes

Normalmente a quantidade de neurônios existentes tanto na camada de entrada como na camada de saída é determinado pelo tipo de problema a ser resolvido, já a quantidade de neurônios na camada oculta podem variar consideravelmente. O neurônio na camada oculta será associado ao acrônimo NO de agora em diante.

Uma solução inicial para a quantidade de NOs de uma RNA é dada na fórmula (2-11).

$$O = \sqrt{m*n} \quad (2-11)$$

Onde m é a quantidade de neurônios de saída e n a de entrada e O a quantidade de NOs.

Após ser treinada e testada, a RNA pode não apresentar resultados satisfatórios, isto pode ocorrer devido a existência de muitos NOs ocasionando o superajustamento. Recomenda-se iniciar o treino com poucos neurônios artificiais nesta camada (GUMZ, 2002, p. 44).

2.5 CONTEXTO ATUAL DO TRABALHO

Analisando-se os trabalhos de Costa (2000), Hong (1998), Gumz (2002) entre outros verificou-se que, muitos sistemas biométricos que utilizam o reconhecimento através das IDs usam como base um fator importantíssimo que são as comparações das minúcias. Mas poucos trabalhos, pelo que se tem conhecimento, utilizam outras técnicas, como por exemplo a comparação entre as distância do(s) delta(s) e do núcleo.

2.5.1 Trabalhos Correlatos

Os trabalhos correlatos encontrados foram os trabalhos de:

- a) Hong (1998). Este é o trabalho mais completo, onde apresentou-se um estudo detalhado sobre os passos da construção de um *AFIS*: aquisição de ID, pré-processamento, mapa de direções, classificação e extração de minúcias.
- b) Silva (1999). Foi desenvolvido um protótipo para classificação de ID nos tipos fundamentais: arco, presilha interna, presilha externa e verticilo. Para tanto o autor

também utilizou a construção de mapa de direções. No protótipo foi utilizada uma RNA *perceptron feedforward* treinada por retropropagação.

- b) Costa (2000). Foram apresentadas fórmulas para pré-processamento de IIDs, conceitos e algumas técnicas para classificação e extração de minúcias. Efetuou-se apenas conclusões sobre o mapa de direções das cristas papilares.
- b) Pankanti (2000). Comentou vários métodos de identificação biométrica existentes, dando destaque especial ao *AFIS*, realizando a comparação entre os mesmos.
- b) Gumz (2002). Foi desenvolvido um protótipo para identificação e classificação das minúcias de uma ID. O trabalho é bem detalhado e mostra passo a passo o estudo realizado.
- b) Reis (2003). Neste trabalho é apresentada uma solução para detecção das minúcias e um cálculo para o encontro do núcleo de uma ID.

Todos os trabalhos relatam com bastante coerência o estudo/desenvolvimento de seus trabalhos. Em especial nos trabalho de conclusão de curso de Alex Sandro da Silva e Rafael Araújo Gumz foram sugeridos como extensão a identificação automática do(s) delta(s) e núcleo das ID.

0 DESENVOLVIMENTO DO PROTÓTIPO

Nas próximas seções serão apresentados os requisitos do protótipo, uma visão geral da solução proposta bem como o desenvolvimento do protótipo.

0.0 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

O protótipo do sistema deve atender aos seguintes requisitos:

- c) criar, treinar, validar e executar a RNA⁵;
- c) possibilitar o treinamento manual ou automático da RNA através de fragmentos de imagens;
- c) possibilitar salvar os pesos da RNA após o seu treinamento;
- c) possibilitar carregar os pesos da RNA para a continuação de um treinamento;
- c) possibilitar a validação manual ou automática da RNA através de fragmentos de imagens;
- c) possibilitar o cadastramento de usuários juntamente com sua IID;
- c) possibilitar carregar uma IID para comparação com as demais IIDs já cadastradas;

0.0 VISÃO GERAL

Antes da seção de especificação, se faz necessária uma abordagem prévia de como será o funcionamento geral do protótipo para que o leitor possa conhecer alguns conceitos necessários para as próximas seções.

Inicialmente o protótipo deve permitir que o especialista treine a RNA passando como entrada para a rede um vetor que será gerado com base nos fragmentos de imagens de tamanho 20x20 *pixels*. Estas imagens serão de “regiões dos deltas”, “regiões do núcleo” e “regiões quaisquer”, conforme é apresentado na Figura 9. Este treinamento como já citado na seção 3.1, poderá ocorrer de forma automática ou manual. Uma vez que a RNA esteja treinada serão gravados os pesos da mesma e a partir de então o usuário do protótipo poderá iniciar o cadastramento de usuários bem como executar a opção comparação das IIDs.

⁵ O termo RNA será utilizado a partir de agora para representar a RNA-FM, pois esta será a única rede implementada no protótipo.

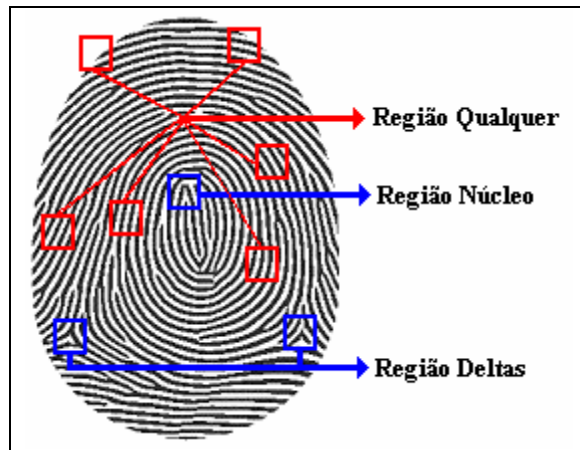


Figura 9 – Exemplo de regiões da IID passadas como entrada para a RNA.

Ao realizar o cadastramento de um novo usuário com sua respectiva IID, o sistema irá armazenar algumas informações importantes derivadas da IID, para a extração destas características foi utilizada uma heurística que de agora em diante iremos chamar de “heurística do cálculo da distância”. Através desta heurística é que o protótipo irá identificar se uma IID é igual a outra ou não. Para que se possa explicar o funcionamento desta heurística fazem-se necessárias algumas definições. Uma definição inicial é apresentada na Figura 10, onde dividiu-se uma IID em 5 regiões, chamados de agora em diante de “quadrantes”.



Figura 10 – Definição da divisão de quadrantes numa IID.

A idéia inicial da “heurística do cálculo da distância” é de varrer os quadrantes 3, 4 para obter as posições dos deltas bem como o quadrante 5 para obter a posição do núcleo. Esta varredura consistirá em passar fragmentos de imagens 20x20 para validação da RNA de todo o espaço dos quadrantes incrementando-se as linhas e colunas dos mesmos. Quando for encontrado o delta no quadrante 3 e 4, ou um núcleo no quadrante 5 deve-se guardar a posição

e interromper a varredura no quadrante atual. É importante lembrar que uma vez que a RNA esteja bem treinada a mesma irá encontrar obrigatoriamente um ou nenhum delta nos quadrantes 3 e 4, e um núcleo no quadrante 5. Caso isto não ocorra pode-se constatar que a IID está com pouca nitidez, ou que a RNA ainda se encontra com um desempenho insatisfatório.

Uma vez com as posições encontradas do(s) delta(s) e núcleo através das coordenadas iniciais do fragmento 20x20, irá se utilizar das coordenadas (x,y) das regiões para se calcular a distância entre delta(s) e núcleo, ou até entre os próprios deltas. No caso da ID ser uma presilha externa ou presilha interna apenas será traçada uma distância como pode ser visualizado na Figura 11 pelo valor A, em contrapartida quando a ID for um verticilo poderá se ter uma maior exatidão, pois poderão ser calculadas 3 distâncias para a comparação, entre cada um dos dois deltas e o núcleo e entre os próprios deltas, conforme a Figura 11, para os valores A, B e C. Quando a ID for um arco, não poderá se calcular nenhuma distância, ou seja, esta heurística apenas funcionará para 3 dos 4 tipos fundamentais.

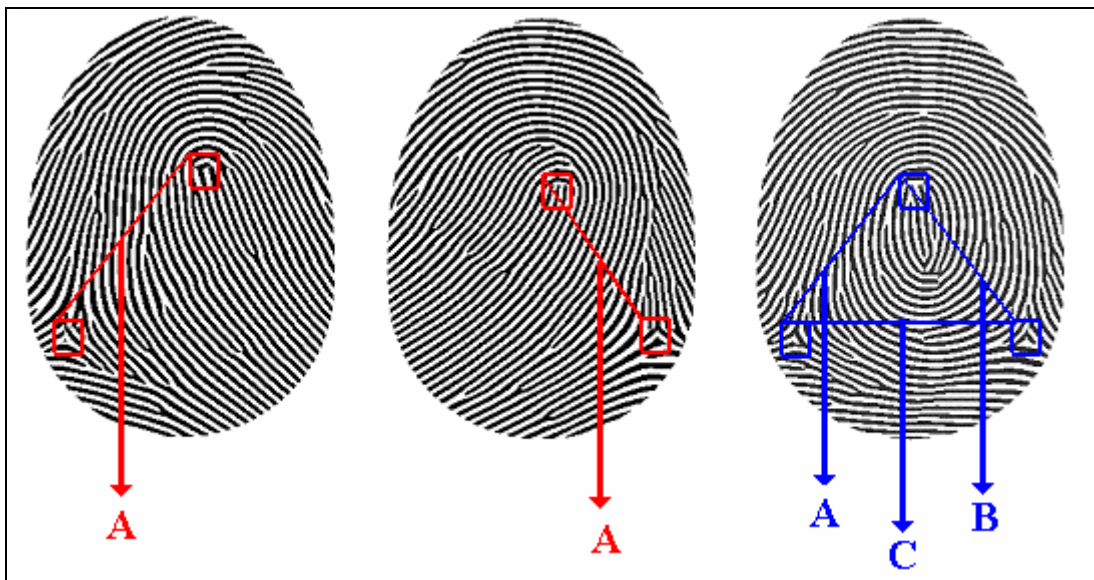


Figura 11 – Exemplo de distâncias calculadas para os 3 tipos fundamentais de ID.

0.0 ESPECIFICAÇÃO

A especificação do protótipo será apresentada através dos diagramas de casos de uso primário e secundário, e do diagrama de classe. Os diagramas apresentados seguirão a notação da UML. Assim como as classes implementadas no protótipo seguirão os conceitos de

orientação a objetos. Todos os desenhos dos diagramas foram realizados na ferramenta Rational Rose 2002 versão demo da Rational Rose Corporation. Maiores informações sobre a ferramenta pode ser obtidas no site Rational (2002).

0.0.0 Casos de Uso

Os casos de uso primários e secundários, mostrados dão uma idéia dos processos envolvidos no protótipo, conforme mostrados nas Figuras 12, 13 e 14.

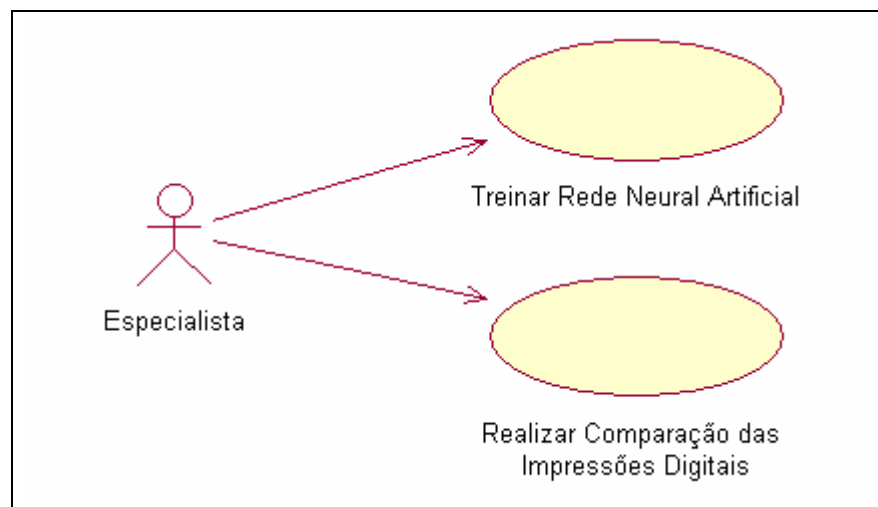


Figura 12 – Diagrama de caso de uso primário do ator “Especialista”.

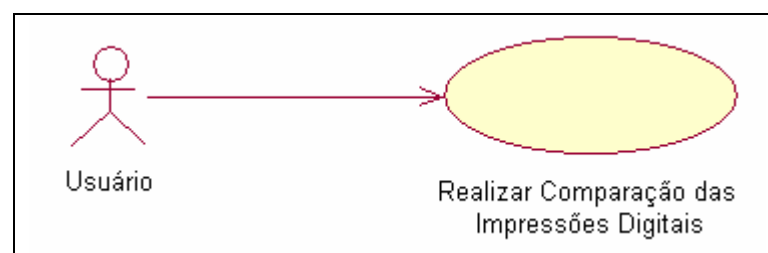


Figura 13 – Diagrama de caso de uso primário do ator “Usuário”.

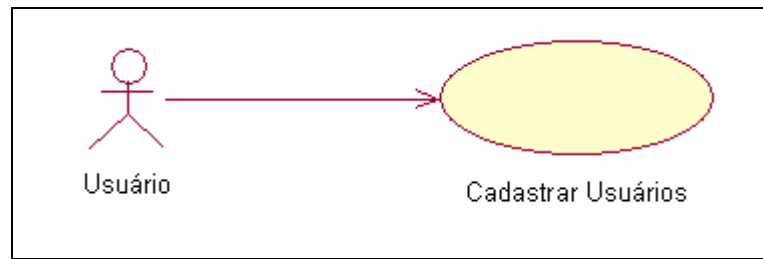


Figura 14 – Diagrama de caso de uso secundário do ator “Usuário”.

0.0.0 Diagramas de Classes

Na Figura 15 é apresentado o diagrama de classes do protótipo, onde podem ser visualizadas as 3 classes existentes no mesmo, sendo elas: Integração, RedeNeural e Usuário. A classe Integração é a classe que contempla todos os métodos que irão comunicar-se com as outras duas classes. Um exemplo destes métodos pode ser o método de conversão de um fragmento de imagem 20×20 pixels para os dados de entrada da Rede Neural. A classe Usuário comunica-se com a classe Integração pois junto com o cadastramento do usuário e sua digital, já são gravados os valores das distâncias calculados através do método da “heurística do cálculo da distância”. E por fim a classe RedeNeural contempla todos os métodos referentes ao treinamento e validação da rede. Como já citado qualquer método que deva auxiliar a rede, mas não faz parte do seu treinamento e validação foi encapsulado na classe Integração.

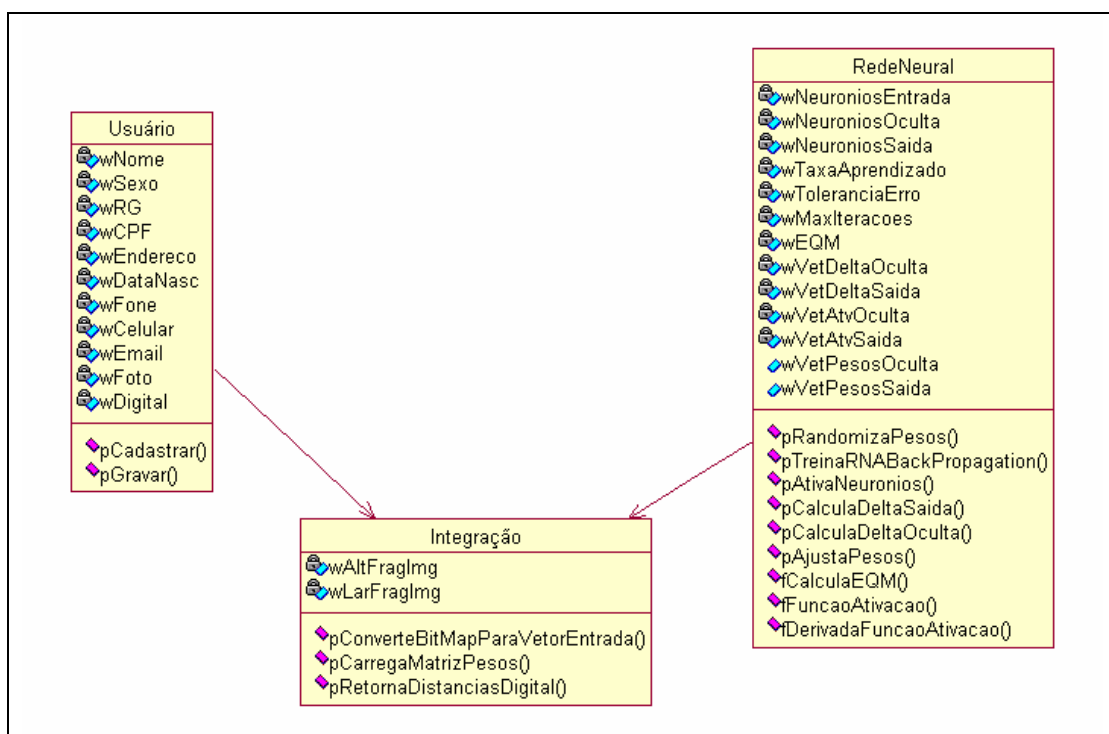


Figura 15 – Diagrama de Classes do protótipo.

Por não ser o foco principal do protótipo o cadastramento do usuário, apenas será detalhada a seguir a classe da Integração e RedeNeural.

0.0.0 Integração

A classe Integração é uma classe mediadora entre as classes Usuário e RedeNeural, nesta classe estão presentes todos os métodos necessários para o funcionamento do protótipo, mas que não poderiam fazer parte de nenhuma das outras duas classes. Como o próprio nome já diz, é uma classe que integra as demais classes servindo de “ponte” para a troca de mensagem entre elas.

Segue uma explicação dos principais atributos desta classe:

- c) *wAltFragImg*: Altura do fragmento das imagens a serem convertidos nos dados de entrada da RNA, utilizado *20 pixels*.
- c) *wLarFragImg*: Largura do fragmento das imagens a serem convertidos nos dados de entrada da RNA, utilizado *20 pixels*.

Seguem os principais métodos desta classe:

- c) *pConverteBitMapParaVetorEntrada*: este é o método que converte um fragmento de imagem *20x20 pixels* para os dados que servirão de entrada para a RNA;
- c) *pCarregaMatrizPesos*: método que carrega a matriz de pesos do arquivo para a memória;
- c) *pRetornaDistanciasDigital*: método que calcula as distâncias entre os pontos característicos das IIDs.

0.0.0 Classe da Rede Neural

Na classe RedeNeural a maioria dos atributos são privados, com exceção dos vetores de pesos da RNA que são acessados por vários pontos do sistema. Na classe também encontram-se alguns métodos privados que são os de manipulação interna da classe e outros públicos que são chamados de alguns pontos fora da classe.

É importante destacar dois tipos de dados muito importantes para a classe da Rede Neural, são eles os tipo: “TArray1D” e “TArray2D”. Estes tipos de dados como mostrados a

seguir, são tipos de *arrays* (vetores) dinâmicos, que podem ter seu tamanho alterado em tempo de execução, são eles:

- a) *TArray1D = Array of Double*: tipo usado para definir *arrays* dinâmicos de uma dimensão (1D) do tipo *Double*. O *Double* representa números reais com ponto flutuante ocupa 8 bits de espaço em memória
- b) *TArray2D = Array of Array of Double*: tipo usado para definir *arrays* dinâmicos de duas dimensões (2D) do tipo *Double*.

Segue uma explicação dos principais atributos da classe *RedeNeural*:

- a) *wNeuroniosEntrada*: quantidade de neurônios da camada de entrada;
- b) *wNeuroniosOculta*: quantidade de neurônios da camada oculta;
- c) *wNeuroniosSaida*: quantidade de neurônios da camada de saída;
- d) *wTaxaAprendizado*: valor da taxa de aprendizado utilizada durante a correção dos pesos da RNA;
- e) *wToleranciaErro*: valor da tolerância de erro a qual irá indicar se a RNA aprendeu ou não;
- f) *wMaxIteracoes*: valor máximo de iterações a qual a RNA será submetida. Espera-se que a rede sempre convirja antes deste valor;
- g) *wVetPesosOculta*: vetor que irá guardar os valores dos pesos entre os neurônios da camada de entrada e camada oculta;
- h) *wVetPesosSaida*: vetor que irá guardar os valores dos pesos entre os neurônios da camada oculta e camada de saída;
- i) *wVetAtivacoesOculta*: vetor que guarda os valores de ativações de todos os neurônios da camada oculta, ativados durante o processo de propagação;
- j) *wVetAtivacoesSaida*: vetor que guarda os valores de ativações de todos os neurônios da camada de saída, ativados durante o processo de propagação;
- k) *wVetDeltaSaida*: vetor que guarda os valores dos deltas de cada neurônio, calculados na camada de saída;
- l) *wVetDeltaOculta*: vetor que guarda os valores dos deltas de cada neurônio, calculados na camada oculta;

Segue uma explicação dos principais métodos da classe *RedeNeural*:

- a) *pRandomizaPesos*: método que randomiza os pesos da RNA com valores próximos a zero para a inicialização do treinamento;

- b) pTreinaRNABackPropagation: método principal de treinamento que chama os demais métodos para que a RNA seja treinada;
- c) pAtivaNeuronios: método que ativa os neurônios da camada oculta e camada de saída. Para tanto são utilizados os valores de entrada e os valores dos vetores de pesos;
- c) pCalculaDeltaSaida: método que calcula os valores dos deltas de todos os neurônios da camada de saída;
- c) pCalculaDeltaOculta: método que calcula os valores dos delta de todos os neurônios da camada oculta;
- c) pAjustaPesos: método que realiza o ajuste dos pesos nos vetores quando uma determinada amostra gerou um erro acima da tolerância de erro;
- c) fFuncaoAtivacao: método da função sigmóide binária, é utilizada na ativação dos neurônios;
- c) fDerivadaFuncaoAtivacao: método da derivada da função sigmóide binária, é utilizada para o cálculo do delta dos neurônios da camada de saída e oculta;
- c) fCalculaEQM: método que calcula o erro quadrado médio (EQM) de uma amostra de treinamento;
- c) fTestaRNA: método que testa a RNA após efetuado seu treinamento;

0.0.0 Estrutura da Rede

Inicialmente uma das preocupações numa aplicação utilizando RNA deve ser com relação a quantidade de camadas ocultas da rede e a quantidade de neurônios presentes tanto na camada de entrada, quanto na(s) camada(s) oculta(s) e camada de saída. Sabe-se que se um mesmo problema for dado a vários projetistas de RNAs todos podem apresentar soluções com RNAs estruturadas de maneira completamente diferentes. Para a implementação do protótipo definiu-se o uso de apenas uma camada oculta.

O número de neurônios utilizados na camada de entrada foi de 400, este valor é a representação de cada *pixel* dos fragmentos das IIDs de tamanho de 20x20 *pixels*. A quantidade de neurônios na camada de saída será igual a 1, que indicará através de faixas de valores a região dada na entrada da rede. E por último, definiu-se a quantidade de neurônios na camada seguindo a fórmula (2-11) descrita a seção 2.4, onde $O = \sqrt{400*1}$, totalizando 20 neurônios nesta camada.

Inicialmente se treinará a rede com 20 neurônios na camada oculta, caso esta rede mostre pouco poder de generalização, irá se seguir indicações de alguns autores, onde deverá se aumentar ou diminuir esta quantidade de neurônios um a um até que se encontre um melhor desempenho.

0.0.0 Parâmetros de Treino

Os parâmetros de treino são tão importantes quanto a definição da quantidade de neurônios presentes em cada camada. O algoritmo *backpropagation* implementado conta com 3 parâmetros extremamente importantes sendo eles: quantidade máxima de iterações, taxa de aprendizagem e valor da tolerância ao erro.

A quantidade máxima de iterações é o mais simples, ele é um critério de parada, onde se o erro desejado não for alcançado é usado para interromper o processamento após n iterações, este valor normalmente é estipulado em constantes muito altas. Para o protótipo o valor escolhido foi de 100.000 iterações. É importante lembrar que todo o conjunto de treinamento poderá ser apresentado para cada uma destas 100.000 iterações caso o erro gerado esteja acima do valor esperado.

O valor da tolerância do erro é outro critério de parada que encerra o ajuste dos pesos, uma vez que este for alcançado diz-se que a RNA aprendeu. Segundo Haykin (2001, p. 200), a taxa de variação do EQM é considerada pequena se esta se encontrar entre 0,1 e 1 por cento de cada *epoch*. Este valor de erro pode interromper prematuramente o processo de aprendizagem ou fazer com que a rede perca sua capacidade de generalização obrigando-a a treinar até que este erro seja alcançado. O valor escolhido para o protótipo foi de 0,0001.

Como último parâmetro e também muito importante temos a taxa de aprendizagem ou multiplicador gradiente (*learn rate*). Fausett (1994, p. 305) afirma que é recomendado um valor pequeno para este parâmetro para evitar maiores rompimentos na direção de aprendizado quando um padrão de entrada fora do comum é apresentado. Haykin (2001, p. 220) também sugere que os valores para esta variável deveriam encontrar-se entre 0,01 e 0,5. O valor escolhido para o protótipo foi de 0,1.

O algoritmo de treinamento requer um vetor de saída com valor alvo (desejado) para cada entrada, que é usado para calcular o EQM. Assim, para uma dada entrada espera-se que a

RNA responda com uma saída igual ou próxima da que lhe foi ensinada. Haykin (2001, p. 207) relata que é importante que os valores alvos sejam escolhidos dentro do intervalo da função de ativação sigmóide mas que se afastem de seus limites mínimos e máximos. Neste caso, a função sigmóide binária tem seus valores entre 0 e 1. Masters (1993, p. 82) também expõe que não se deve treinar uma RNA para alcançar seus extremos, as entradas podem conter estes valores, mas não as saídas. Com base nisto e após alguns testes, decidiu-se utilizar os seguintes valores:

- c) 0,15 para deltas;
- c) 0,45 para núcleo;
- c) 0,8 para regiões quaisquer.

A inicialização dos vetores de pesos foi feita pela randomização comum de valores entre $-0,5$ a $+0,5$. A escolha destes pesos iniciais tem forte influência sobre o aprendizado, porque deles serão calculadas ativações para encontrar os deltas e o EQM. Existem outras técnicas para inicialização dos pesos. Mas já é comprovado que para a maioria dos casos a randomização mostrou-se eficiente.

0.0 IMPLEMENTAÇÃO

Serão apresentadas a seguir todos os itens que comporam a parte de implementação do protótipo, sendo eles: técnicas, ferramentas, fragmentos de códigos, softwares utilizados, etc.

0.0.0 Técnicas e Ferramentas Utilizadas

Para realizar a implementação do protótipo utilizou-se a linguagem de programação *Object Pascal* no ambiente de desenvolvimento Borland Delphi, na versão 7.0. Maiores informações sobre a ferramenta pode ser obtidas em Cantu (2002). E o SGBD Interbase, em sua versão 5.0. A seguir serão apresentados fragmentos de códigos fontes das principais rotinas da Classe RedeNeural e dos principais processos do protótipo que não fazem parte da classe RedeNeural.

No Quadro 1 apresenta-se a rotina principal de treinamento da RNA, que faz a chamada para as demais rotinas de treinamento. Neste quadro pode ser percebido que para cada iteração é apresentado todo o conjunto de treinamento.

```

// Procedure que realiza o Treinamento da Rede Neural Artificial
procedure TRedeNeural.pTreinaRNABackPropagation(prVetEntradas, prVetSaidas: TArray2D;
        prQtdAmostras: Integer; var prRetPesosOculta: TArray2D;
        var prRetPesosSaida: TArray2D);

var
    wEQM          ,
    wAcumulaEQM   : Double;
    wIteracao     ,
    wAmostras     : Integer;
begin
    // Incicializa a Variável de Erro
    wAcumulaEQM := 0;

    // Realiza o laço para a quantidade de Iterações
    for wIteracao := 0 to wMaxIteracoes do
        begin
            // Incicializa a Variável de Erro
            wAcumulaEQM := 0;

            // Laço sobre todos as Amostras de Entradas
            for wAmostras := 0 to prQtdAmostras-1 do
                begin
                    wVetAtuEntrada := TArray1D(prVetEntradas[wAmostras]);
                    wVetAtuSaida := TArray1D(prVetSaidas[wAmostras]);
                    pAtivaNeuronios(wNroNodosEnt, wNroNodosOcu, wVetPesosOculta, wVetAtuEnt, wVetAtivOcu);
                    pAtivaNeuronios(wNroNodosOcu, wNeuroniosSai, wVetPesosSaida, wVetAtivOcu, wVetAtivSai);

                    // Chama a função que Calcula o Erro Quadrado Médio
                    wEQM := fCalculaEQM;

                    // Verifica se precisa ajustar para este caso de treinamento
                    if wEQM > wToleranciaErro then
                        begin
                            // Chama as procedures que calculam os Deltas.
                            pCalculaDeltaSaida;
                            pCalculaDeltaOculta;

                            // Chama as procedures que realizam o ajuste dos pesos nos Vetores de pesos.
                            pAjustaPesos(wVetDeltaSai, wVetAtvOcu, wVetPesosSai, wNroNodosSaida, wNroNodosOcu);
                            pAjustaPesos(wVetDeltaOcu, wVetAtuEnt, wVetPesosOcu, wNeuroniosOcu, wNeuroniosEnt);
                        end;

                    wAcumulaEQM := wAcumulaEQM + wEQM;
                end;

            // Realiza a média dos erros
            wAcumulaEQM := wAcumulaEQM / prQtdAmostras;

            // Verificar Tolerância ao Erro
            if (wAcumulaEQM < wToleranciaErro) then
                begin
                    prLabelTela.Caption := ' *** Rede Treinada com Sucesso ... *** ';
                    Break;
                end;
            end;

        // Se chegou neste ponto com valor de erro maior que a tolerância é pq a rede não aprendeu
        if wAcumulaEQM > wToleranciaErro then
            prLabelTela.Caption := ' *** Rede NÃO APRENDEU ... *** ';

        // Passo os pesos dos Vetores de Pesos para a tela
        prRetPesosOculta := wVetPesosOculta;
        prRetPesosSaida := wVetPesosSaida;
    end;

```

Quadro 1 – Rotina principal de treinamento da classe RedeNeural.

No Quadro 2 apresenta-se a rotina que realiza a ativação de todos os neurônios de todas as camadas da RNA. Este rotina é chamada para cada camada oculta e para a camada de

saída. A camada de entrada não chama esta rotina pois a ativação dos neurônios desta camada são as próprias entradas da rede.

```

procedure TRedeNeural.pAtivaNeuronios(prNrNodoCamadaAnt, prNrNodoCamadaAtu: Integer;
prVetPesosAntAtu: TArray2D; prVetAtvCamadaAnt: TArray1D;
var prVetAtvCamadaAtu: TArray1D);

var
    wContaAtu ,
    wContaAnt : Integer;
    wSoma      : Double;
begin
    // Esta procedure passa por cada Neurônio da Camada Atual e Multiplica todas as Entradas da
    // camada anterior pelos Pesos ligados a esta camada, o Resultado da Soma destas
    // multiplicações é passada para a função Logística de Ativação, que retorna o valor de
    // ativação para cada Neurônio da Camada Atual.

    // Laço sobre todos os Neurônios da Camada Atual
for wContaAtu := 0 to prNrNodoCamadaAtu-1 do
    begin
        // Inclui o Bias
        wSoma := (1 * prVetPesosAntAtu[wContaAtu,0]);

        // Laço sobre todos os Neurônios da Camada Anterior
        for wContaAnt := 0 to prNrNodoCamadaAnt-1 do
            wSoma := wSoma + (prVetAtvCamadaAnt[wContaAnt] *
                prVetPesosAntAtu[wContaAtu,wContaAnt+1]);

        // Carrega o Vetor de Ativação desta camada de acordo com o Valor da Função de Ativação
        prVetAtvCamadaAtu[wContaAtu] := fFuncaoAtivacao(wSoma);
    end;
end;

```

Quadro 2 – Rotina de ativação dos neurônios das camadas ocultas e de saída.

Nos Quadros 3 e 4 encontram-se fragmentos das rotinas de cálculo dos deltas da camada de saída e camada oculta respectivamente. Nestes quadros é possível verificar que os valores dos deltas calculados na camada de saída são utilizados para calcular os deltas dos neurônios da camada anterior, ou seja da camada oculta.

```

// Procedure que Calcula os valores do Delta de todos os neurônios da camada de saída
procedure TRedeNeural.pCalculaDeltaSaida;
var
    wContaAtu : Integer;
    wDelta     : Double;
begin
    // Esta função realiza o cálculo do Delta de cada neurônio da camada de saída que será
    // utilizado para a geração do vetor de Delta das Camadas Ocultas.

    // Laço sobre todos os Neurônios da Camada de Saída
for wContaAtu := 0 to wNeuroniosSaida-1 do
    begin
        // Calcula o Delta do Gradiente
        wDelta := (wVetAtuSaida[wContaAtu] - wVetAtivacoesSaida[wContaAtu]) *
            fDerivadaFuncaoAtivacao(wVetAtivacoesSaida[wContaAtu]);

        // Guarda o Delta de cada Neurônio
        wVetDeltaSaida[wContaAtu] := wDelta;
    end;
end;

```

Quadro 3 – Rotina de cálculo do delta dos neurônios da camada de saída da RNA.

```

// Procedure que Calcula os valores do Delta de todos os neurônios da camada Oculta
procedure TRedeNeural.pCalculaDeltaOculta;
var
    wContaAtu    ,
    wContaProx   : Integer;
    wDelta       ,
    wSoma        : Double;
begin
    // Esta função realiza o cálculo do Delta de cada neurônio da Camadas Ocultas.
    // Após o cálculo do Delta de todos os neurônios da camada Oculta poderá ser ajustados os pesos
    // dos vetores caso necessário

    // Laço sobre todos os Neurônios da Camada Oculta
    for wContaAtu := 0 to wNeuroniosOculta-1 do
        begin
            wSoma := 0;

            // Laço sobre todos os Neurônios da Camada de Saída
            for wContaProx := 0 to wNeuroniosSaida-1 do
                wSoma := wSoma + (wVetDeltaSaida[wContaProx] *
                    pVetPesosSaida[wContaProx,wContaAtu+1]);

            //wDelta := wSoma * fDerivadaFuncaoAtivacao(wVetSomaAtvOculta[wContaAtu]);
            wDelta := wSoma * fDerivadaFuncaoAtivacao(wVetAtivacoesOculta[wContaAtu]);

            // Guarda o Delta de cada Neurônio
            wVetDeltaOculta[wContaAtu] := wDelta;
        end;
    end;

```

Quadro 4 – Rotina de cálculo do delta dos nerônios da camada oculta RNA.

O ajuste dos pesos entre os neurônios da rede pode ser verificado no Quadro 5. Esta rotina é chamada duas vezes, uma para o ajuste dos pesos entre os neurônios da camada de entrada e camada oculta e uma segunda vez para o ajuste dos pesos entre a camada oculta e a camada de saída.

```

// Procedure que realiza as correções nos Pesos entre as Camadas da RNA
procedure TRedeNeural.pAjustaPesos(prVetDelta, prVetAtivacoes: TArray1D; var prVetPesos:
TArray2D;
                                prNrNodoCamadaAtu, prNrNodoCamadaAnt: Integer);
var
    wContaProx   ,
    wContaAtu    : Integer;
begin
    // Laço sobre toda a Matriz de Peso para as correções
    for wContaProx := 0 to prNrNodoCamadaAtu-1 do
        begin
            // Peso do Viés
            prVetPesos[wContaProx,0] := prVetPesos[wContaProx,0] + (2 * wTaxaAprendizado *
                prVetDelta[wContaProx] * 1);

            for wContaAtu := 0 to prNrNodoCamadaAnt-1 do
                prVetPesos[wContaProx,wContaAtu+1] := prVetPesos[wContaProx,wContaAtu+1] +
                    (2 * wTaxaAprendizado *
                        prVetDelta[wContaProx] *
                        prVetAtivacoes[wContaAtu]);
        end;
    end;

```

Quadro 5 – Rotina de ajuste dos pesos de todas as camada da RNA.

Nos Quadros 6 e 7 respectivamente são apresentadas as rotinas de cálculo do EQM, e as funções de Ativação Sigmóide Binário e a Derivada da função de ativação Sigmóide Binária.

```
// Procedure que calcula o EQM - Erro Quadrado Médio
function TRedeNeural.fCalculaEQM : Double;
var
    wContaSai : Integer;
    wSoma : Double;
begin
    wSoma := 0;

    // Laço sobre os Neurônios da Camada de Saída
    for wContaSai := 0 to wNeuroniosSaida-1 do
        // Diferença de Valores Esperado e Real elevado ao quadrado
        wSoma := wSoma + Sqr(wVetAtuSaida[wContaSai] - wVetAtivacoesSaida[wContaSai]);

    Result := wSoma / wNeuroniosSaida;
end;
```

Quadro 6 – Rotina de cálculo do Erro Quadrado Médio EQM.

```
// Função que Realiza o Cálculo da Função Logística de Ativação - "Função Sigmóide Binária".
function TRedeNeural.fFuncaoAtivacao(prValor: Double) : Double;
begin
    Result := 1.0 / (1.0 + Exp(-1 * prValor));
end;

// Função que realiza o Cálculo da Derivada da Função Logística de Ativação
function TRedeNeural.fDerivadaFuncaoAtivacao(prValor: Double) : Double;
begin
    Result := prValor * (1.0 - prValor);
end;
```

Quadro 7 – Rotinas da Função Sigmóide Binária e sua derivada.

A seguir no Quadro 8, é apresentado a rotina que converte um fragmento de uma IID de tamanho 20x20 *pixels* para a entrada da RNA. É possível verificar que os valores atribuídos são binários, o valor será igual a 1 caso o pixel seja preto do contrário o pixel certamente será branco e neste caso será atribuído o valor 0.2.

```
// Procedure que transforma uma imagem BitMap no vetor de Entrada para a RNA
procedure pConverteBitMapParaVetorEntrada(prImgEntrada: TImage; var prVetEntrada: TArrayOfD);
var
    wColuna ,
    wLinha ,
    wIndice : Integer;
begin
    // Inicializa e Limpa o Vetor
    SetLength(prVetEntrada, cNeuroniosEntrada);
    FillChar(prVetEntrada, 0, 0);

    wIndice := 0;

    // Laço sobre todas as linhas e colunas da imagem 20 x 20 Pixels
    for wLinha := 0 to 19 do
        for wColuna := 0 to 19 do
            begin
                // Verifica se o Pixel é Preto, se for carregado 1 para o vetor, senão carrega 0
                if prImgEntrada.Canvas.Pixels[wColuna, wLinha] = clBlack then
                    prVetEntrada[wIndice] := 1
                else prVetEntrada[wIndice] := 0.2;

                Inc(wIndice);
            end;
end;
```

Quadro 8 – Rotina de conversão de imagens 20x20 para os valores de entrada da RNA.

No quadro 9 é apresentado o algoritmo da “heurística do cálculo da distância”, utilizado para se traçar as distâncias entre os pontos característicos, conforme citado na seção 3.2.

```

// Procedure que retorna as distâncias entre o Núcleo e os Destas
procedure TIntegracao.pRetornaDistanciasDigital(var prDisDNE: Double; var prDisDND: Double;
                                             var prDisDeD: Double);

type
  TPosicaoReg = Record
    wPosX ,
    wPosY : Integer;
  end;

var
  wLinIni , wLinFim , wColIni , wColFim : Integer;
  wPosDE , wPosDD , wPosN : TPosicaoReg;

begin
  // ** Processos para o Quadrante 3 **
  wLinIni := Round(cAlturaImagem / 2);
  wLinFim := cAlturaImagem;
  wColIni := 1;
  wColFim := Round(cLarguraImagem / 2);

  // Chama a procedure passando os dados do quadrante 3 e o tipo de dado a ser buscado
  pRetornaPosicaoReg(wLinIni, wLinFim, wColIni, wColFim, 'D', wPosDE);

  // ** Processos para o Quadrante 4 **
  wLinIni := Round(cAlturaImagem / 2);
  wLinFim := cAlturaImagem;
  wColIni := Round(cLarguraImagem / 2);
  wColFim := cLarguraImagem;

  // Chama a procedure passando os dados do quadrante 4 e o tipo de dado a ser buscado
  pRetornaPosicaoReg(wLinIni, wLinFim, wColIni, wColFim, 'D', wPosDD);

  // ** Processos para o Quadrante 5 **
  wLinIni := Round(cAlturaImagem / 4 * 1);
  wLinFim := Round(cAlturaImagem / 4 * 3);
  wColIni := Round(cLarguraImagem / 4 * 1);
  wColFim := Round(cLarguraImagem / 4 * 3);

  // Chama a procedure passando os dados do quadrante 5 e o tipo de dado a ser buscado
  pRetornaPosicaoReg(wLinIni, wLinFim, wColIni, wColFim, 'N', wPosN);

  // Verifica quais posições retornou e calcula a distância
  // Para calcular as distâncias utiliza o teorema de Pitágoras (Triângulos retangulos)

  // Distância do Quadrante 3
  if (wPosDE.wPosX <> 0) and (wPosN.wPosX <> 0) then
    prDisDNE := Sqrt(Sqr(wPosN.wPosX - wPosDE.wPosX) + Sqr(wPosDE.wPosY - wPosDN.wPosY));

  // Distância do Quadrante 4
  if (wPosDD.wPosX <> 0) and (wPosN.wPosX <> 0) then
    prDisDNE := Sqrt(Sqr(wPosDD.wPosX - wPosN.wPosX) + Sqr(wPosDE.wPosY - wPosDN.wPosY));

  // Distância do Quadrante 5
  if (wPosDE.wPosX <> 0) and (wPosDD.wPosX <> 0) then
    begin
      // Verifica se está na mesma linha, pois neste caso a distância é o próprio valor de X
      if (wPosDE.wPosY = wPosDD.wPosY) then
        prDisDNE := wPosDD.wPosX - wPosDE.wPosX;
      else prDisDNE := Sqrt(Sqr(wPosN.wPosX - wPosDE.wPosX) + Sqr(wPosDE.wPosY -
wPosDN.wPosY));
    end;
  end;
end;

```

Quadro 9 – Rotina da heurística do cálculo da distância.

0.0.0 Imagens de Impressões Digitais

O tipo de arquivo utilizado por todas as rotinas do protótipo será o mapa de bits ou *bitmap* (BMP) porque não necessita de qualquer processamento para compactação ou descompactação. Durante o desenvolvimento deste trabalho verificou-se quatro formas distintas para obtenção das IIDs, a saber: tintadas em papel e capturadas através de *scanners* óticos; capturadas diretamente através de *scanners* biométricos; capturadas na Internet, na qual não se sabe qual o método de captura; ou ainda geradas artificialmente através de um software que será detalhado adiante.

Segundo Gumz (2002, p. 64), para a obtenção de IIDs tintadas em papel e capturadas através de *scanners* óticos com uma boa qualidade, não deve-se utilizar qualquer tipo de tinta, como as tintas de carimbos normais por exemplo, pois muitas delas são à base de água e entram facilmente nos sulcos interpapilares, fazendo com que se percam os detalhes das IDs. Para se ter um melhor resultado deve-se usar tinta tipográfica, que é uma tinta mais consistente, a mesma utilizada na impressão de jornais e tipográficos em geral. Esta pode ser aplicada sobre o dedo com um rolo comum de pintura sem a espuma de revestimento. O método de extração com tinta tipográfica permite IIDs mais limpas e nítidas, pois esta tinta gruda mais facilmente nas linhas do desenho digital e não borra com tanta facilidade.

Na Figura 16, apresenta-se duas IIDs capturadas através deste método, uma com tinta comum (esquerda) e a outra com tinta tipográfica (direita). Mesmo assim deve-se tomar cuidado pois este tipo de método ainda pode apresentar problemas como borramento e/ou manchas, causados pelo excesso ou falta de tinta, ou pressão inadequada ao papel (Costa, 2000, p. 8).



Fonte: Gumz (2002, p. 64)

Figura 16 – Exemplo de IIDs capturadas com tinta comum e tinta tipográfica

O método de captura das IIDs através de *scanners* biométricos pode ocorrer com leitores de diferentes tecnologias, como citados na seção 2.2.2. Sabe-se que em geral este método de captura deixa as IIDs com bastante nitidez e riqueza de detalhes. Em virtude do pouco tempo para o desenvolvimento deste trabalho não foi possível a realização de testes com estes tipos de leitores. Na Figura 13 é mostrada uma IID capturada através de um leitor biométrico.



Fonte: Costa (2000, p. 18)

Figura 17 – IID capturada através de um leitor biométrico

O terceiro método que encontrou-se para a obtenção de IIDs foi através da Internet. Inicialmente achou-se que seria muito fácil achar um grande número de IIDs publicadas nas centenas de páginas encontradas sobre o assunto, uma vez que a Internet é uma imensa fonte

de pesquisa, mas o resultado não foi bem este. Após várias horas de pesquisa encontrou-se poucos sites, e que forneciam pequenas bases de dados com poucos desenhos de IIDs. As bases de dados de diferentes IIDs encontradas podem ser baixadas de Fingerprints (2004?), FVC (2004).

Em FingerPrints (2004?) encontrou-se um grupo de 168 amostras, estas IIDs são de 8 dedos de 21 pessoas diferentes com tamanho fixo de 256×256 pixels. Neste grupo não estão presentes as IIDs dos dedos mínimos esquerdo e direito. Na Figura 18 são apresentadas alguns exemplos destas IIDs, mantidas por um laboratório de biometria italiano.



Figura 18 – Exemplos de IIDs capturadas na Internet.

Já em FVC (2004), que é um site de um campeonato mundial de validação de algoritmos de identificação biométrica através das IDs, encontrou-se 3 grandes grupos de IIDs, um para cada versão da competição, respectivamente para os anos de 2000, 2002 e 2004. Em cada um destes 3 grupos encontrou-se 320 IIDs no formato *bitmap* que são divididas em 4 séries de 80 imagens com 10 dedos em cada série, estes dedos em 8 diferentes posições, pressões, cores e tamanhos. Na Figura 19 encontram-se exemplos destas IIDs.



Figura 19 – Exemplos de IIDs capturadas na Internet.

A quarta e último forma encontrada para se obter as IIDs foi através de um software italiano de nome Sfinge. Este software gera IIDs artificialmente através de configurações simples, entre estas configurações destacam-se: o nível de nitidez, a quantidade de borrões, tamanho, grau de rotação, ângulo de inclinação entre outras. Para o processo de treinamento e validação da RNA do protótipo, utilizou-se apenas as IIDs geradas através deste software. Na Figura 20 e 21 são apresentadas algumas telas de configuração deste software para a geração da IID, onde na Figura 20 tem-se configurações quanto a sua altura e largura, do tipo quanto a sua classificação e um exemplo de como o sistema gera a IID inicial. Após a geração inicial podem ser aplicadas outras alterações na imagem, como inclusão de falhas, rotação de inclinação imagem entre outras, como é mostrado na Figura 21.

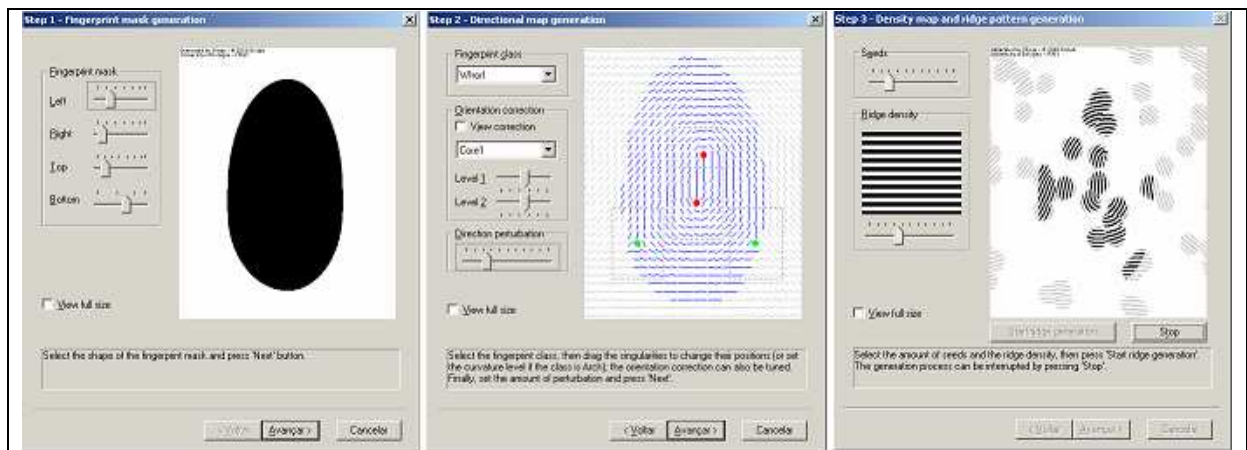


Figura 20 – Exemplos de configuração do Aplicativo Sfinge.

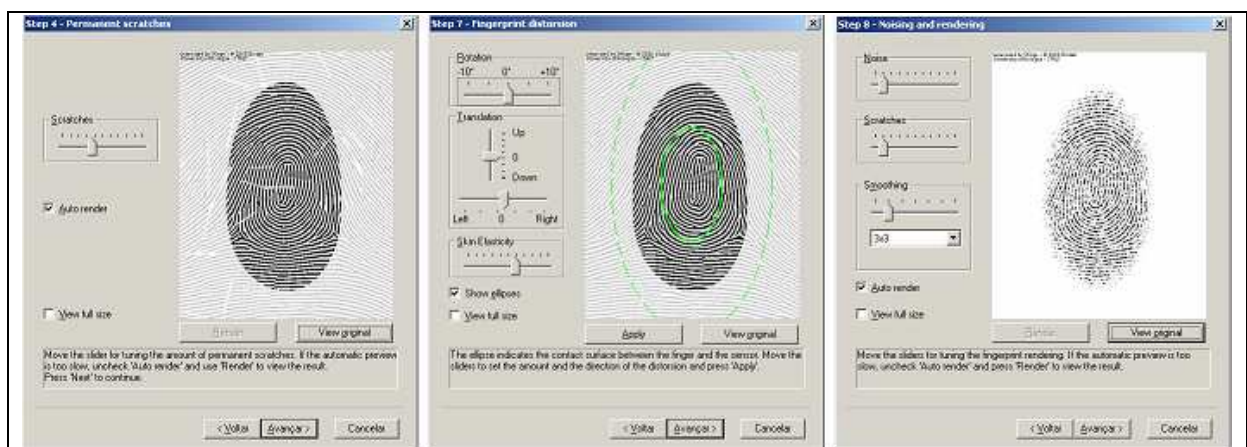


Figura 21 – Exemplos de configuração do Aplicativo Sfinge.

As imagens neste aplicativo podem ser geradas manualmente, ou seja, uma a uma conforme mostrado através das Figuras 20 e 21, ou geradas em grupos automaticamente pelo

sistema. Para este processo automático serão necessárias outras configurações como por exemplo a quantidade de IID a serem geradas, tipo destas IIDs, etc. Mas este processo somente está disponível na versão completa do aplicativo e que para versões acadêmicas custa em torno de 1.800 dólares. Na Figura 22, são apresentadas algumas IIDs geradas através deste aplicativo.



Figura 22 – Exemplo de IIDs geradas pelo aplicativo Sfinge.

0.0.0 Pré-Processamento em Imagens de Impressões Digitais

Após a geração das IIDs pelo software Sfinge como já mencionado na seção 3.4.2, realizou-se pré-processamentos extremamente simples nestas imagens. Primeiramente se eliminou os tons de cinza das mesmas transformando-as em monocromáticas, e após isto diminuiu-se a imagem para que seu tamanho ficasse na ordem de 80% de seu tamanho original. Isto se fez necessário, pois do contrário haveriam mais neurônios na camada de entrada da RNA e, escolheu-se este percentual pois esta alteração se mostrou muito boa não comprometendo os detalhes das imagens.

Estes pré-processamentos são tão simples, que os mesmos foram realizados na ferramenta MS-Paint da Microsoft, que é uma ferramenta extremamente simples e de conhecimento geral. Após estes processos as IIDs ficaram com 265 *pixels* de largura por 305 *pixels* de altura.

0.0.0 Treinamento da Rede Neural Artificial

Para o treinamento da RNA serão apresentados fragmentos de IIDs de tamanho 20x20 *pixels*. Estes fragmentos deverão ser de qualquer região da IID, pois quanto maior a heterogeneidade dos dados de treinamento, maior a probabilidade da RNA ter um bom desempenho. Todos os fragmentos de IIDs que serão as entrada da RNA deverão ter apenas três valores para a saída, a saber: deltas, núcleos e regiões quaisquer, como já citado na seção 3.3.6.

Fausett (1994, p. 300) expõe que treinar uma RNA utilizando entradas binárias torna seu treino relativamente lento, porque qualquer neurônio que receba uma entrada de 0 (zero) para um padrão em particular não pode aprender este padrão. Isso acontece na soma ponderada da execução da RNA. O caso exemplificado por Fausett foi o de uma RNA treinada para resolver o problema do XOR. Este caso trazia gráficos de erro, em relação a seus *epochs* que foram convincentes em demonstrar a quantidade de treinamento necessária. Fausett (1994, p. 299) então sugeriu que o treinamento pode ser melhorado se as entradas são representadas na forma bipolar e a função sigmóide bipolar seja utilizada como função de ativação ou que se modifique o valor das ativações dos neurônios de entrada. Seguindo a sugestão de Fausett, continuou-se a utilizar a função sigmóide binária como função de ativação, mas atribuiu-se 1 a cada entrada de cor preta e 0,2 a cada entrada de cor branca.

Segundo Masters (1993), Haykin (2001) entre outros, muitas vezes não será possível treinar a rede com toda a população, porque esta população pode ser inatingível. Este problema de população ocorre neste trabalho. Uma possível solução pode ser um treinamento com uma quantidade grande de casos onde estes casos estejam bastante embaralhados para o treinamento, isto para que a RNA não memorize alguns casos, uma vez que sempre deveria generalizá-los.

Como já citado anteriormente o treinamento da RNA poderá ser executado de duas formas: manual ou automático. No treinamento manual carrega-se uma IID e escolhem-se as regiões a serem passadas para treinamento da RNA, uma de cada vez, acompanhada de cada fragmento da IID define-se também a qual região ela pertence. Este tipo de treinamento é extremamente lento, mas mesmo assim foi disponibilizado este recurso no protótipo.

Já no treinamento automático define-se um conjunto de arquivos de extensões BMPs, para serem treinados. Uma vez que um destes arquivos está sendo apresentado a rede para treinamento deve-se saber qual o valor esperado na saída, e como o processo é automático e não pode ser interrompido, os arquivos devem obedecer a um padrão de nomes para que o sistema saiba qual valor carregar no vetor alvo. Para fragmentos de IIDs do tipo Delta deve ser criados arquivos com o nome “D01.Bmp”, para Núcleos “N001.Bmp” e para regiões Quaisquer “Q0001.Bmp”, o número de zeros após o caractere que difere os tipos das imagens pode variar indefinidamente.

Haykin (2001), Loesch (1996) e outros autores, afirmam que o treinamento de uma RNA não é um processo trivial, pois além de todo tempo investido para definir sua estrutura, muitas vezes são gastos dias, semanas e até meses para se acertar os parâmetros de treinamento para a mesma. Por isto muitas vezes é um processo de paciência e dedicação. Para se treinar uma RNA com sucesso é necessário muito tempo, pois deve-se treinar a rede com conjuntos de amostras de treinamento de tamanhos diferentes, além do que ainda pode-se mudar a taxa de aprendizagem e sua tolerância ao erro. Todos estes fatores misturados podem causar uma enorme bateria de treinamento.

Para o treinamento da RNA do protótipo pretendia-se formar 8 grupos de amostras, entretanto apenas conseguiu-se formar os 3 primeiros grupos devido ao tempo, pois a geração das amostras de deltas e núcleos foram feitas manualmente. Os grupos deveriam dobrar de tamanho gradativamente, sendo formados conforme a Tabela 1.

Tabela 1 – Quantidade de amostras por Grupo para cada tipo de região

Tipo de Região	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	Grupo 6	Grupo 7	Grupo 8
Deltas	25	50	100	200	400	800	1.600	3.200
Núcleos	25	50	100	200	400	800	1.600	3.200
Regiões Qualquer	50	100	200	400	800	1.600	3.200	6.400

Nas Figuras 23, 24 e 25 são apresentados alguns exemplos de fragmentos das IIDs que serviam de amostras de treinamento da RNA. As mesmas são respectivamente de regiões do núcleo, regiões dos deltas e regiões quaisquer, as imagens estão ampliadas 5 vezes.

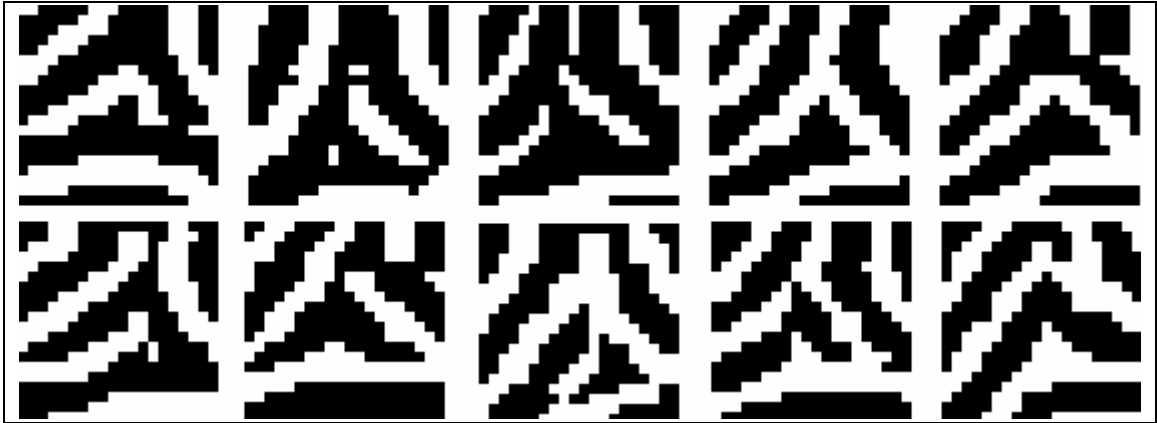


Figura 23 – Amostras de treinamento de regiões dos deltas ampliadas em 5 vezes.



Figura 24 – Amostras de treinamento de regiões dos núcleos ampliadas em 5 vezes.

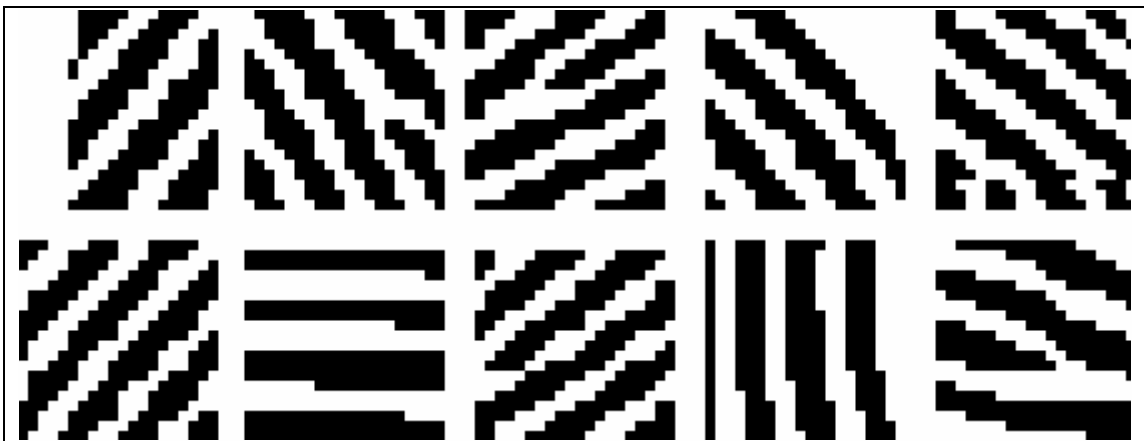


Figura 25 – Amostras de treinamento de regiões quaisquer ampliadas em 5 vezes.

Após o processo de treinamento da RNA seus pesos devem ser armazenados, para serem futuramente utilizados pelos processos de validação da RNA e utilização da RNA.

A validação da RNA é um processo muito importante, é nele que se verifica o quanto a rede aprendeu. Para realizar-se a validação da rede devem ser apresentadas à mesma amostras totalmente novas, e verificar se a mesma consegue enquadrá-las corretamente dentro dos grupos de saídas. Se o resultados da validação for maior ou igual ao resultado esperado diz-se que a RNA aprendeu, após isto ela já pode ser utilizada. Para o protótipo, a validação da RNA foi realizada com uma quantidade de 50 amostras, onde dentre estas haviam fragmentos de imagens de regiões dos 3 tipos possíveis.

0.0.0 Operacionalidade da Implementação

Ao executar o aplicativo do protótipo é apresentada a tela com os menus do sistema conforme a Figura 26. No menu “Rede Neural” são disponibilizadas as telas de “Treinamento” e “Validação” da RNA voltadas ao especialista; enquanto que os menus “Cadastro” com a opção “Usuários” e “Identificação” com a opção “Comparação das Impressões Digitais” são voltadas ao usuário comum.

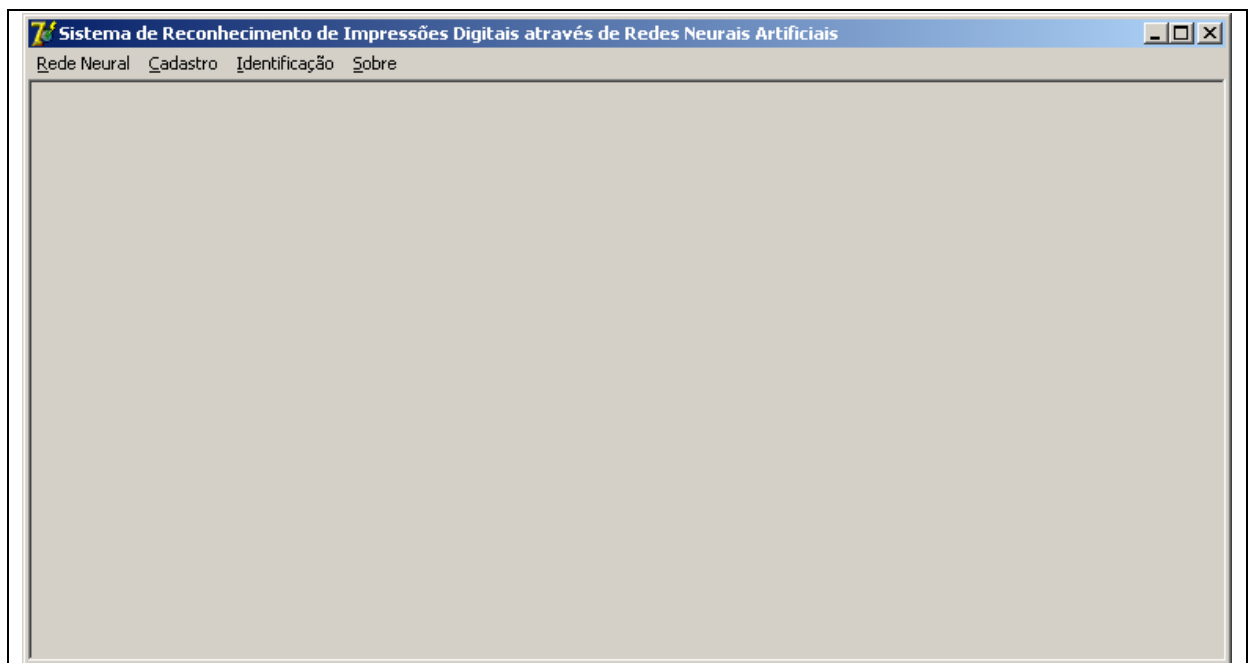


Figura 26 – Tela principal do protótipo.

A tela de treinamento da RNA é composta por 3 pastas, sendo: “Treinamento Automático”, “Treinamento Manual” e “Pesos”. Na Figura 27 é mostrada a pasta de “Treinamento Automático”, nesta tela é possível selecionar o tipo de inicialização dos pesos

entre “Não Inicializar”, “Zerar Pesos” ou “Randomizar Pesos”. Após a escolha desta opção deve ser selecionado o diretório em que está presente o conjunto de amostras com os fragmentos das IIDs 20x20 *pixels* com as quais a RNA será treinada, feito isto basta apertar o botão “Treinar Rede”. Por último será demonstrado no título acima da barra de progressão se a RNA aprendeu ou não, e a quantidade de iterações necessárias.

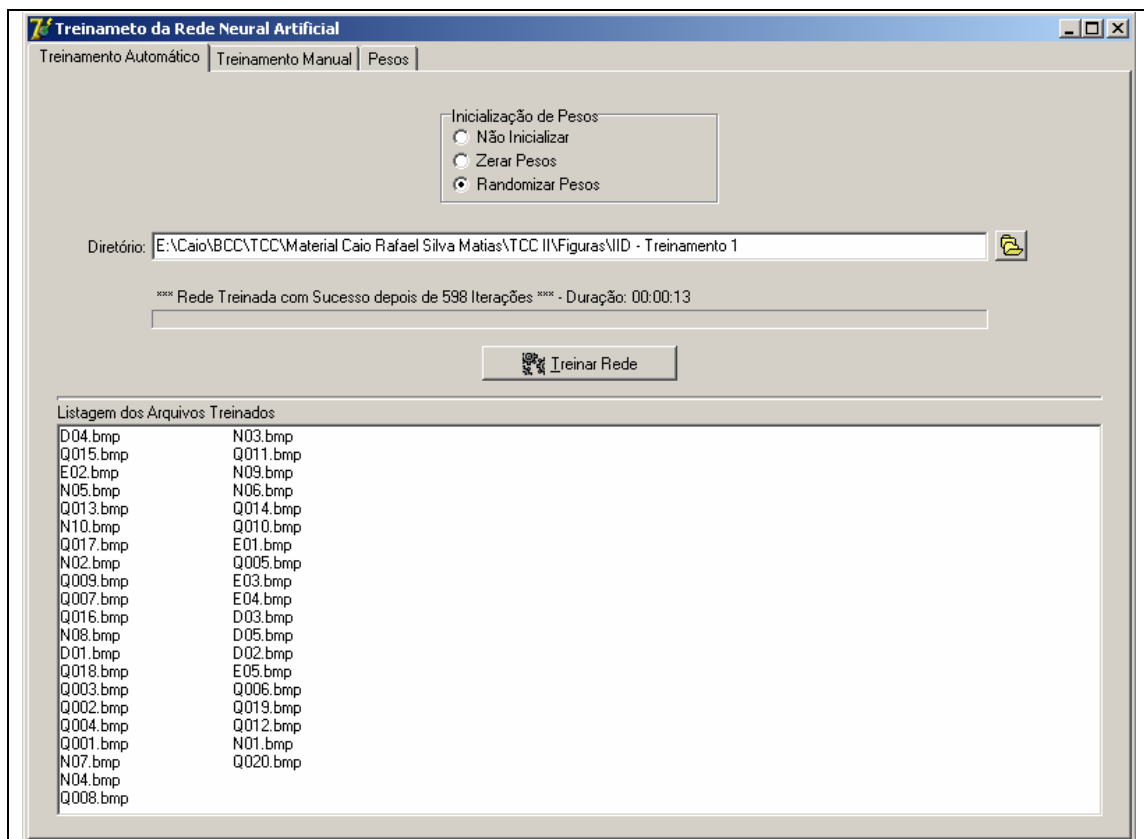


Figura 27 – Pasta Treinamento Automático da tela de Treinamento da RNA.

Na Figura 28 é mostrada a pasta de “Treinamento Manual” da RNA, onde como na pasta de “Treinamento Automático” deve-se escolher o tipo de inicialização de pesos e após, deve ser aberto uma IID e marcada a região que se deseja realizar o treinamento na RNA. Uma vez marcada a região deve-se definir o tipo a que esta região pertence: região dos deltas, região dos núcleos e região qualquer. Feito isto é só clicar no botão “Treinar Rede”, ainda existe nesta tela uma imagem ampliada em 5 vezes da região selecionada para o treinamento.

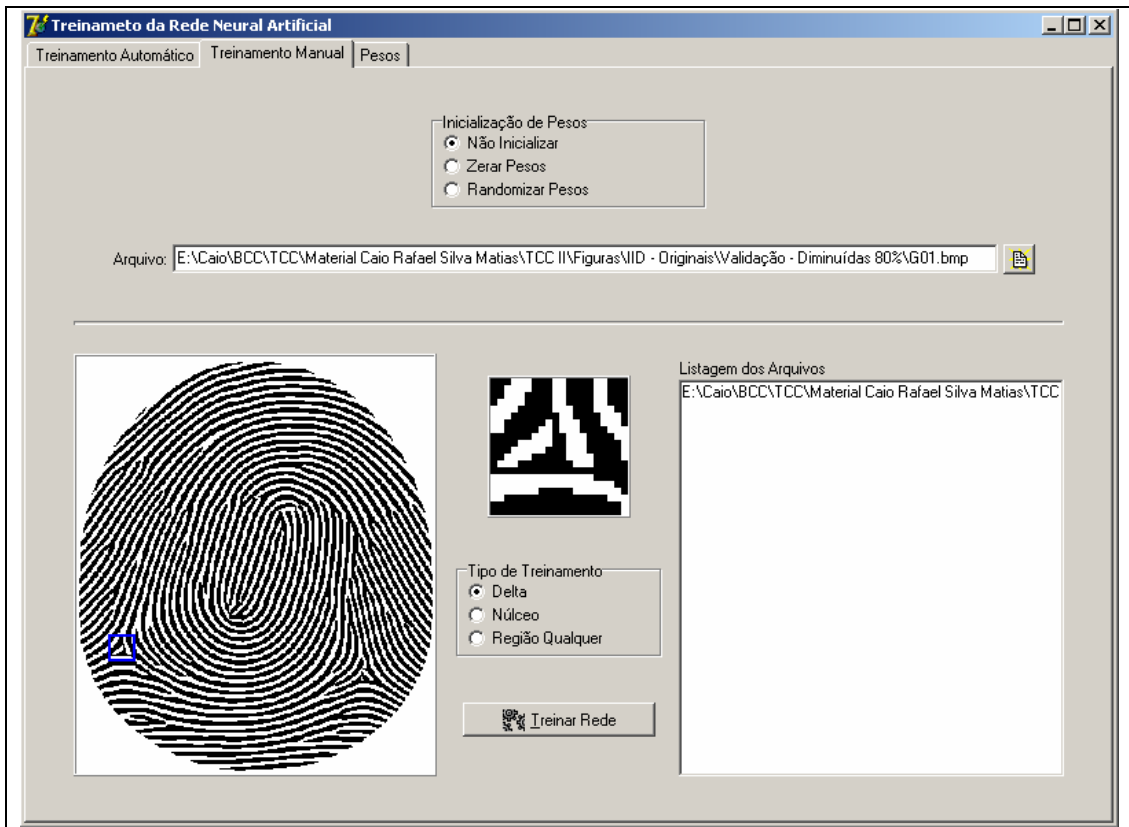


Figura 28 – Pasta Treinamento Manual da tela de Treinamento da RNA.

Tanto na tela de treinamento quanto na tela de validação da RNA existe uma pasta chamada “Pesos”, esta pasta apresenta os pesos gerados, quando na tela de treinamento, ou pode carregar os pesos armazenados quando na tela de validação. Na tela de treinamento estão disponibilizados tanto o botão “Salvar Matriz de Pesos”, quanto o botão “Carregar Matriz de Pesos”, já na tela de validação apenas o segundo botão está disponível.

Treinamento da Rede Neural Artificial

Treinamento Automático | Treinamento Manual | Pesos

Pesos entre Camadas: Entrada -> Oculta

	Col. 1	Col. 2	Col. 3	Col. 4	Col. 5	Col. 6
Lin. 1	-0,234710358613697	0,0927852646932592	0,392788185746616	0,493369095282212	0,0933750071053269	-0,4070241873014
Lin. 2	-0,4951495414291	-0,2893346570669	0,132678870302489	0,188761801186107	0,341845183956491	-0,1347808104373
Lin. 3	-0,139511696950544	0,391137291597127	-0,392036934029105	-0,308604143267091	-0,335865941369856	0,2474376046806
Lin. 4	0,213313459160785	-0,197122397895778	0,394863183031126	-0,0898341047146704	-0,105223547665168	0,2829855709671
Lin. 5	0,127602834868728	0,188701846078363	0,0872607219922453	-0,516074654568292	0,473291567813735	-0,5345513961768
Lin. 6	0,138626038626527	0,492356285066288	0,093956503959994	0,502740353041842	0,104283216663994	0,2046044736751
Lin. 7	-0,0176038675114651	-0,251372278828391	-0,355525088526621	-0,30203379959801	-0,295886541428865	0,5024739310027
Lin. 8	0,22272368511969	-0,488040056667701	-0,19351675998864	0,101919176936315	0,323077983579682	0,31854541325657
Lin. 9	-0,0887287650941993	0,369613104355583	0,257297275724911	-0,358456493426282	-0,296325557214509	-0,3055061340562
Lin. 10	-0,758232293783425	0,367168814928544	0,0739942374035022	-0,157175808870794	0,489862648904684	0,5020636162871
Lin. 11	-0,168476605789832	-0,0730573726340355	0,408425499993467	0,291445376868756	0,395939237370017	-0,1955722887909
Lin. 12	-0,284856078208797	-0,54126326405199	-0,334036597085586	0,0839205123012988	0,48271183373601	0,2912143447090
Lin. 13	0,0933140832677841	0,284035033728688	-0,516711443697851	0,291356688635908	-0,0989531974677917	0,40104950920362
Lin. 14	0,424314031915026	0,105750734200244	0,505750101195352	0,505670598206888	0,405715105321124	-0,3942890586720
Lin. 15	-0,344926250959417	-0,212696244119073	0,187336292215561	0,287077832277504	0,0888793991683564	0,1873889698415
Lin. 16	0,480781541542438	-0,355954500588546	-0,163306180228589	-0,35317719289345	0,414575131763016	-0,4935137784268
Lin. 17	0,730943258097182	-0,354705387167964	-0,356096736215149	-0,175402821544209	0,133316752145614	-0,1621071288417
Lin. 18	0,380293234047967	0,253624040031419	0,267219720894326	0,111503447173366	0,11405634495919	-0,2912213155537
Lin. 19	-0,11682218768896	0,12911492441957	0,435844404435876	0,129430874229801	-0,197708661760009	-0,1088804594457
Lin. 20	-0,0241082953606298	0,207805355974525	0,341017439294847	0,389862011967092	-0,530857326547927	0,1042775680133

Pesos entre Camadas: Oculta -> Saída

	Col. 1	Col. 2	Col. 3	Col. 4	Col. 5	Col. 6
Lin. 1	-0,273133356205917	-0,295789272769654	-1,00183647022443	1,21961593907315	1,4214637646153	-0,0610716550094

Gravar Matriz de Pesos Carregar Matriz de Pesos

Figura 29 – Pasta de Pesos da tela de Treinamento da RNA.

Nas Figuras 30 e 31 são mostradas as pastas da tela de “Validação” da RNA, que por sinal se parecem muito com a tela de “Treinamento” da RNA. Na pasta de “Validação Automática”, quase não há diferença, apenas existe uma botão para validação ao invés de um botão de treinar a RNA, na pasta de “Validação Manual” a única diferença é que ao marcar a região não se escolhe qual é o tipo da região a que o fragmento pertence, uma vez que é a RNA que envia uma mensagem com um dos três tipos possíveis. E por último na pasta de pesos a única diferença é que não existe um botão “Salvar Pesos”, mas apenas o botão “Carregar Pesos”.

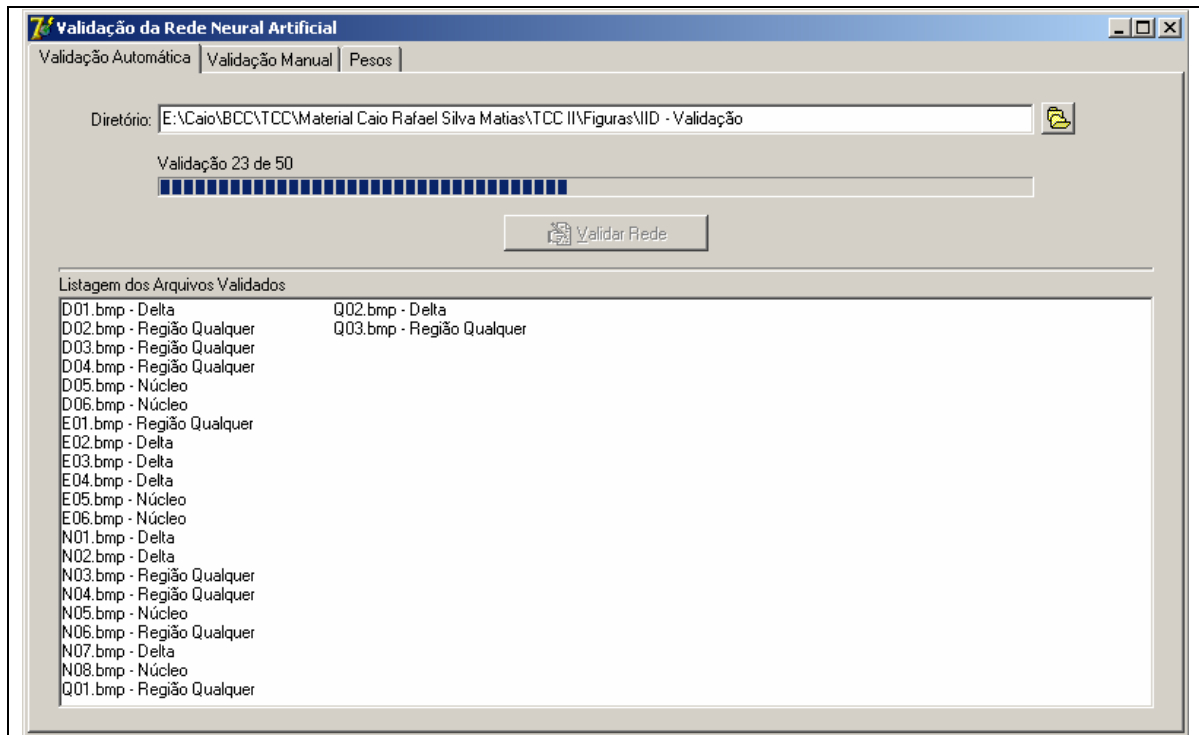


Figura 30 – Pasta de Validação Automática da tela de Validação da RNA.

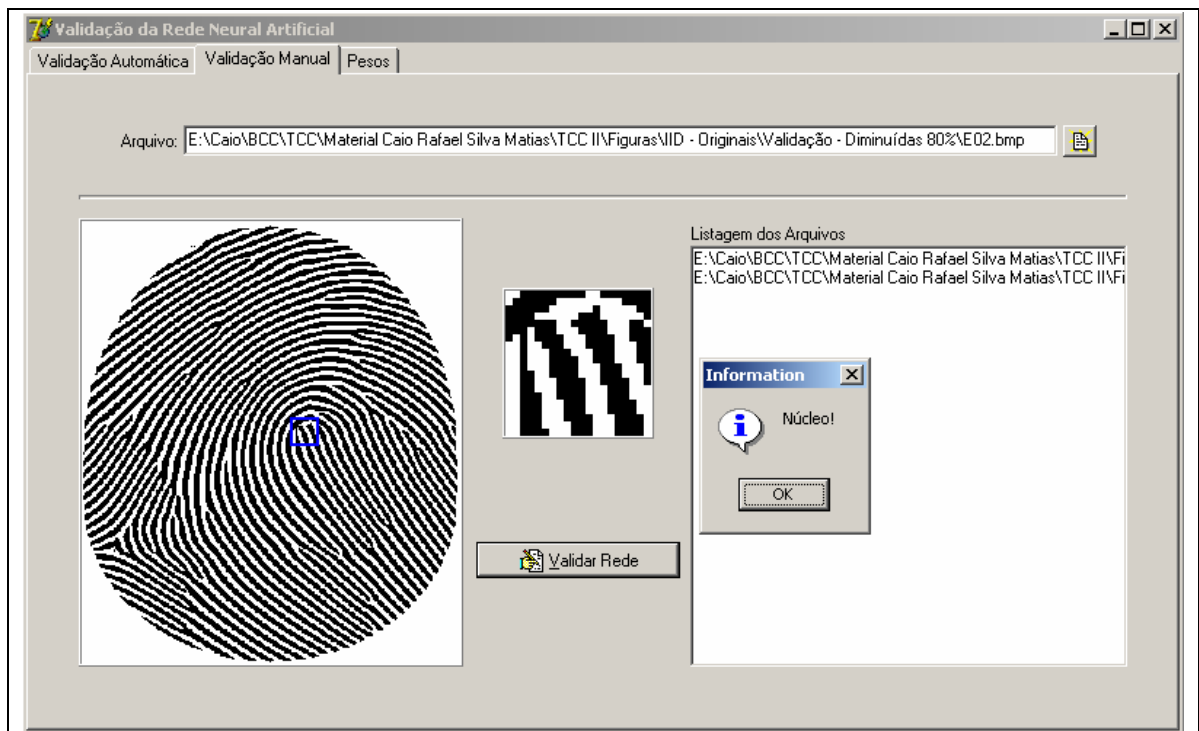
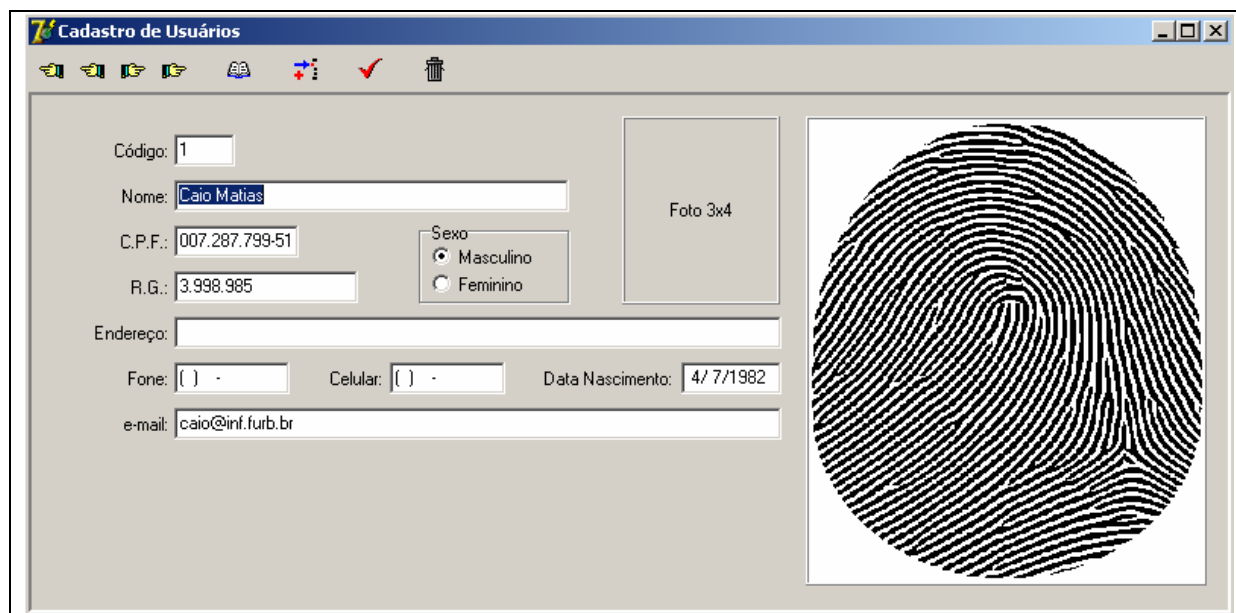


Figura 31 – Pasta de Validação Manual da tela de Validação da RNA.

Ainda tem-se no sistema as telas de cadastramento de usuários e de Identificação de usuários através de suas IDs, conforme mostrados nas próximas figuras. Na Figura 32 apresenta-se a tela de cadastro de usuários, onde podem ser visualizados seus campos básicos para o cadastramento, bem como a sua foto e imagem de sua Impressão Digital. Uma vez que

o usuário é cadastrado com sua IID o sistema calcula as distâncias entre os pontos característicos através da “heurística do cálculo da distância” conforme mencionado na seção 3.2 e armazena-os em 3 campos diferentes, um campo para cada distância.



The image shows a web-based user registration form titled "Cadastro de Usuários". The form contains the following fields and elements:

- Código: 1
- Nome: Caio Matias
- C.P.F.: 007.287.799-51
- R.G.: 3.998.985
- Endereço: (empty)
- Fone: () - () - ()
- Celular: () - () - ()
- Data Nascimento: 4/7/1982
- e-mail: caio@inf.furb.br
- Sexo: Masculino, Feminino
- Foto 3x4: (empty)
- A large circular fingerprint image is displayed on the right side of the form.

Figura 32 –Tela de cadastramento de usuários.

E por último na Figura 33 basta-se selecionar uma IID que automaticamente o sistema calcula as distâncias através da “heurística do cálculo da distância”, e a compara com os valores armazenados para os usuários já cadastrados no BD.

Normalmente em aplicações de identificação através das IDs, é considerado uma taxa máxima para o ângulo de rotação das IIDs capturada em relação a outras IIDs já armazenadas, uma vez que ao tomar uma nova ID para comparação, esta pode estar um pouco deslocada, para o protótipo esta taxa foi definida como um ângulo de 20°. Com base nesta definição, caso se tenha uma IID capturada com um ângulo de rotação de 15° por exemplo em relação a sua imagem inicialmente cadastrada, o resultado da “heurística do cálculo da distância” pode ser pouca coisa diferente para uma mesma IID, por isto irá se usar uma tolerância de 10% do resultado deste cálculo para identificar se uma IID é a mesma ou não.

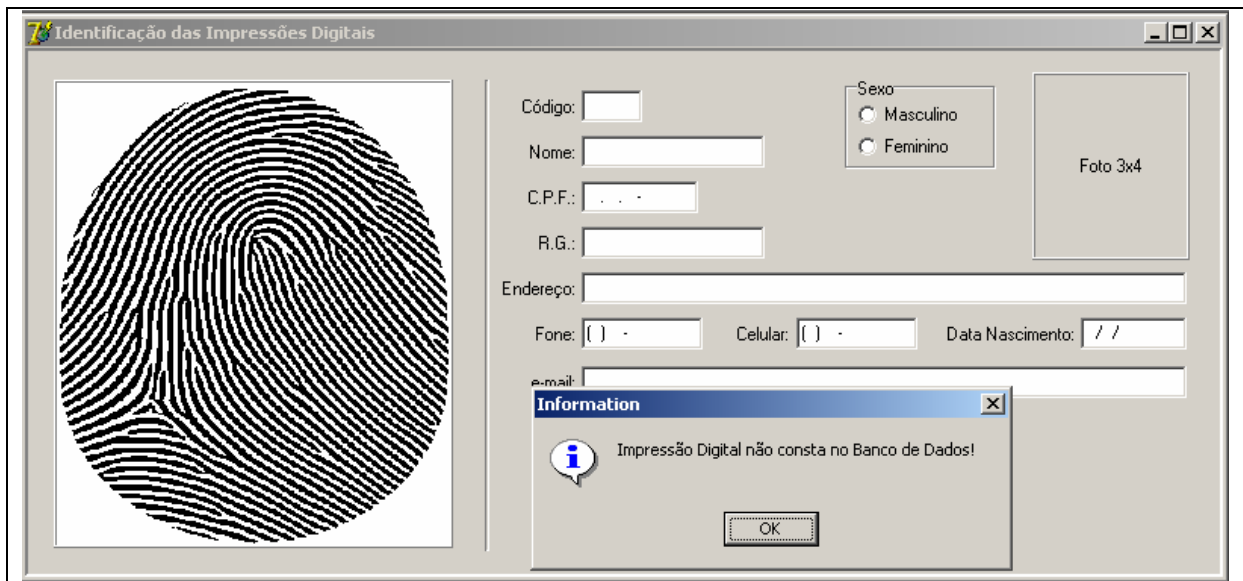


Figura 33 –Tela de identificação das IIDs através da comparação das distâncias.

0.0 RESULTADOS E DISCUSSÃO

Nesta seção serão abordados os resultados obtidos com o treinamento e validação da RNA, através dos diversos grupos de treinamento conforme apresentados na Tabela 1.

Como já citado anteriormente, os tamanhos das imagens utilizadas foram de 265x305 *pixels*, pois necessitava-se de uma alta qualidade das imagens capturadas, uma vez que a entrada para RNA eram fragmentos da própria imagem. Verificou-se que tanto a diminuição da imagem em 80% do seu tamanho original, quanto a transformação da imagem para monocromática não alteraram a qualidade da IID.

Como já citado anteriormente os fragmentos das imagens, tanto utilizadas no treinamento quanto na validação da RNA foram, parte recortados manualmente e parte gerados através de um software desenvolvido especialmente para este protótipo. Desta maneira conseguiu-se realizar a montagem das amostras de acordo com o desejado.

Inicialmente utilizou-se 20 neurônios na camada oculta, este valor foi o resultado da fórmula (2-11) contida na seção 2.4.2.5. Mas ao utilizar esta quantidade de neurônios na camada oculta verificou-se que a RNA não apresentou um bom desempenho. Com base nisto tentou-se alterar a estrutura da rede seguindo indicações de alguns autores, que é a de aumentar e diminuir gradativamente a quantidade de neurônios da camada oculta da rede e

verificar onde a rede encontra melhor desempenho. Na Tabela 2 e na Figura 34 são apresentados os resultados do teste com vários neurônios na camada oculta. No gráfico da figura 35, os números no eixo x , são a quantidade de NOs testadas, e no eixo y , o percentual de acerto da RNA.

Tabela 2 – Resultado da taxa de acerto da RNA com vários NO na camada Oculta

NO	Erro = 0.01	Erro = 0.001	Erro = 0.0001	Erro = 0.00001
2	30%	56%	50%	54%
3	54%	54%	46%	56%
4	52%	54%	50%	40%
6	56%	40%	56%	56%
8	58%	54%	54%	54%
10	48%	52%	56%	54%
15	58%	50%	38%	48%
20	46%	50%	46%	56%
30	42%	44%	56%	56%
40	52%	58%	58%	56%
60	46%	50%	56%	52%

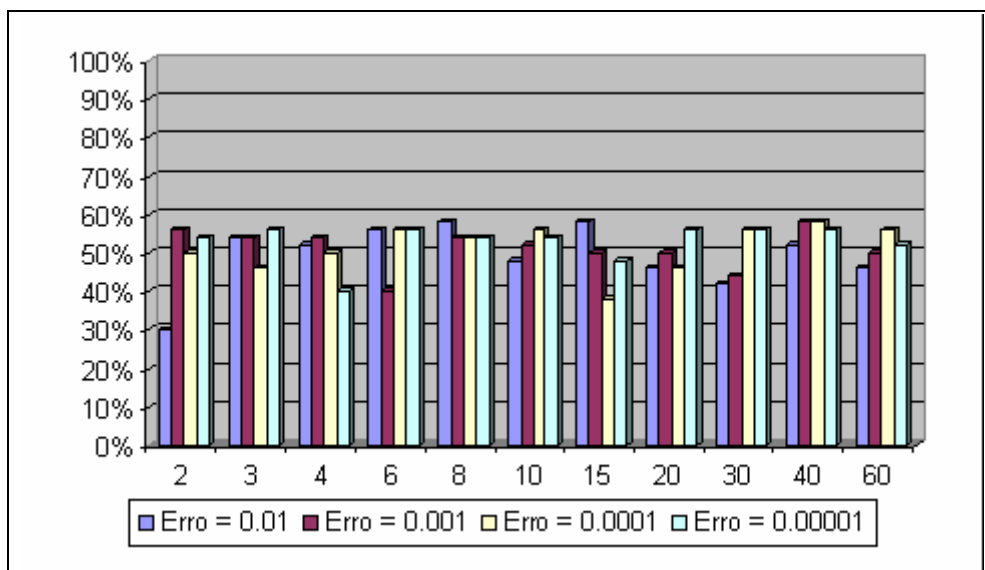


Figura 34 – Gráfico de testes com diferentes valores em parâmetros da RNA.

Após a escolha da melhor quantidade de neurônios x a melhor taxa de tolerância do erro, testou-se os 3 primeiros Grupos de Treinamento da RNA, os resultados estão na Tabela 3. A quantidade de NO escolhida foi igual a 40 e a taxa de tolerância ao erro igual a 0,0001

Tabela 3 – Resultados por grupos de Amostras

Grupos	Erro = 0.0001
Grupo 1	52%
Grupo 2	52%
Grupo 3	58%
Grupo 4	-
Grupo 5	-
Grupo 6	-
Grupo 7	-
Grupo 8	-

Infelizmente os resultados obtidos pela RNA nos grupos de amostras treinados, sendo eles os grupos de 1 a 3 conforme detalhados na Tabela 3, não obtiveram muito sucesso, pois todos os resultados ficaram com uma taxa de acerto entre 52% e 58%.

Para que a “heurística do cálculo da distância” fosse uma técnica realmente relevante como uma das etapas na identificação de um indivíduo através de suas IDs, acredita-se que seria necessário que a RNA tivesse um percentual de acerto acima dos 90%, o que está um pouco longe dos 58% obtidos pela mesma.

Em percentuais talvez esta diferença não pareça tão grande, mas ainda se está longe de uma solução ideal. Ao se considerar este percentual de 58% será verificado por exemplo que em uma busca do delta no Quadrante 3, conforme definição na seção 3.3 poderá ser encontrado um número de deltas entre 5 e 10 por exemplo ao invés de 1, uma coisa totalmente impraticável com a solução proposta. Pois ao encontrar um “falso delta” estaria traçando uma distância incorreta para comparação das IIDs.

Com base em que o melhor percentual da RNA foi o de 58%, acredita-se que este desempenho insuficiente possa ter ocorrido em função de alguns aspectos, a saber: falta de pré-processamento das entradas e quantidade pequena de treinamento da rede.

A falta de um melhor pré-processamento nos fragmentos das IIDs que foram as entradas para a rede, pode ter sido um dos motivos da baixa taxa de acerto da mesma, uma vez que a RNA possa ter tido dificuldades em generalizar as amostras de entradas já que eram valores binários entre 0,2 e 1. Em virtude de um não pré-processamento nos fragmentos das IIDs, teve-se que usar uma quantidade de neurônios bastante grande na camada de entrada, e por este motivo talvez a rede não tenha tido capacidade de generalizar as amostras e acabou memorizando-as, já que no máximo foram treinados apenas 400 fragmentos. Este número de

amostrar pode ser muito pequeno se compararmos aos 400 neurônios de entrada e 20 neurônios na camada oculta. Uma das primeiras soluções a serem tentadas seriam o aumento nas amostras de treinamento para a RNA.

4 CONCLUSÕES

Nesta seção serão apresentadas as conclusões referentes a todo o processo envolvido tanto no estudo, como no desenvolvimento deste protótipo.

4.1 OBJETIVOS DO TRABALHO

A proposta inicial do trabalho era de criar um protótipo que através de RNAs pudesse realizar a identificação de deltas e núcleos em IDs. Este requisito foi atendido parcialmente uma vez que a rede retornava muitas vezes um resposta inválida para determinados fragmentos de IIDs.

Um dos outros requisitos era a criação de um sistema onde, uma vez com a as informações de posição do(s) deltas(s) e núcleo pudesse realizar a comparação entre outras IDs previamente armazenadas, este requisito foi atendido totalmente.

4.2 FERRAMENTAS

O Rational Rose embora utilizado apenas para a criação dos diagramas de caso de uso e o diagrama de classes, foi uma ferramenta importante e de fácil utilização. Outra ferramenta utilizada foi o MS-Paint, embora considerada uma ferramenta extremamente simples mostrou-se suficiente para realização dos tratamentos nas IIDs. Também utilizou-se o software Sfinge que é um excelente gerador de IIDs artificiais e que conta com muitos recursos, o que facilitou muito no desenvolvimento do protótipo.

A ferramenta mais utilizada durante todo o desenvolvimento deste protótipo foi a ferramenta Borland Delphi 7. Além de ser a ferramenta mais conhecida pelo autor, é extremamente intuitiva, de fácil utilização e com muitos componentes visuais que facilitaram muito a criação do aplicativo. O Delphi conta com muitos componentes nativos e pode-se dizer que para muitos tipos de aplicações se encontra quase tudo que necessita-se disponível na própria ferramenta.

4.3 PROTÓTIPO

O protótipo desenvolvido pode apresentar as principais vantagens:

- a) possibilidade de se realizar o treinamento da RNA automaticamente e manualmente;
- d) possibilidade de se realizar a validação da RNA automaticamente e manualmente;
- d) permite gravar as matrizes de pesos da RNA para uma posterior continuação de seu treino;
- d) um ambiente gráfico de fácil utilização;

As principais desvantagens:

- d) a não implementação de uma rotina mais específica de pré-processamento da imagem;
- d) a baixa taxa de aprendizado apresentado pela RNA, em torno de 58%;

0.0 REDE NEURAL ARTIFICIAL *FEEDFORWARD* MULTICAMADA

As Redes Neurais, em especial as RNAs *Feedforward* multicamada, embora sejam recentes, mostraram-se como sendo técnicas bastante interessantes e avançadas. O entendimento das Redes Neurais não é um assunto trivial, embora já existam muitos livros publicados, necessita-se de um estudo mais dedicado, principalmente por conterem nestes livros muitas notações matemáticas.

Uma outra dificuldade era a qualidade das imagens adquiridas, onde apenas conseguiu-se IIDs com bastante detalhes em figuras muito maiores se comparadas as capturadas através de leitores biométricos, acredita-se que isto tenha prejudicado a rede devido ao grande número de neurônios na camada de entrada.

Pode-se destacar entre as vantagens das RNAs como sendo:

- d) a possibilidade de generalizar as informações separando as mesmas em grupos;
- d) as rotinas referentes ao aprendizado não são difíceis de ser implementadas.

As principais desvantagens são:

- d) dificuldade no entendimento do funcionamento;
- d) não é possível saber como a rede aprendeu ou onde exatamente se encontra o problema do não aprendizado;
- d) não existem regras para a criação de sua estrutura ou seja, cada projetista deve estruturar a RNA conforme suas experiências.

4.5 CONCLUSÕES FINAIS

Chegando ao final deste trabalho pode-se dizer que o mesmo foi muito enriquecedor, primeiramente pela opção de se implementar uma RNA, ao invés de se utilizar componentes prontos. Com certeza esta escolha consumiu um tempo precioso na parte do desenvolvimento, mas foi um momento único de extrema dedicação e aprendizado constante para o autor.

O assunto de identificação biométrica não é um assunto trivial, por isto dedicou-se muito tempo ao estudo do mesmo. Com certeza é uma área extremamente interessante que requer muito estudo e persistência. Pelas dificuldades encontradas no desenvolvimento acredita-se que muitos sistemas biométricos possam utilizar outras técnicas que não as RNA.

Enfim, embora não tenham sido alcançados todos os pontos desejados com o protótipo, encerra-se este trabalho com o sentimento de dever cumprido, uma vez que o papel dos cientistas são pesquisar, testar e provar se coisas funcionam ou não. Este foi o sentimento que o autor teve ao final deste trabalho, e que pretende dar continuidade pois o mesmo se interessou muito pelo assunto. Um dos outros motivos ao qual se chega a esta etapa extremamente satisfeito é a possibilidade de poder-se deixar material de pesquisa a novos alunos. Pois acredita-se que um dos sentidos da vida seja este, passar aos outros aquilo que se tem e que se fez de melhor.

4.6 EXTENSÕES

Serão apresentadas sugestões para a continuação deste trabalho e para outros na construção de *AFIS*.

- a) utilizar para a captura das IIDs leitores biométricos, para tentar melhorar ainda mais a definição das imagens;
- b) inclusão de técnicas de pré-processamento nas IIDs para minimizar a quantidade de neurônios na camada de entrada, pois quanto maior for a quantidade, mais amostras devem ser apresentadas a rede e mais tempo de treinamento será necessário;
- c) realizar o treinamento da RNA com mais amostras;
- d) desenvolvimento de melhores modelos de dados para os *templates* para armazenar apenas características das IDs ao invés do desenho da ID;

- e) unificar este trabalho com outras técnicas como a de identificação de minúcias automáticas e posicionamento das mesmas, para se ter uma solução mais completa;
- d) incluir algoritmos para realizarem a identificação do(s) delta(s) e núcleo na IID independente do ângulo de rotação da imagem.

REFERÊNCIAS BIBLIOGRÁFICAS

- ANTHEUS. **Análise e reconhecimento em impressões digitais**. Curitiba, 2002. Disponível em: <<http://www.arid.com.br/Tecnologia/Arid1.htm>>. Acesso em: 02 ago 2004.
- AGUIAR. **Consultoria em informática**. Duque de Caxias, 2004. Disponível em: <<http://www.aguiarsoftware.com.br>>. Acesso em: 02 ago 2004.
- APPES – **Associação dos Papiloscopistas do Espírito Santo**. Espírito Santo, 2004. Disponível em: <<http://www.appes.com.br>> . Acesso em: 03 ago 2004.
- APPOL – **Associação dos Papiloscopistas do Rio de Janeiro**. Rio de Janeiro, 2004. Disponível em: <<http://www.appol.com.br>>. Acesso em: 02 ago 2004.
- BRESSAN, Nadja Mench. **Biometria**. 2002. 16 f. Disponível em: <<http://www.uces.br/ccet/demc/vjbrusam/inst/biometria.pdf>>. Acesso em: 12 ago 2004.
- BROWN, Wayne C; SHEPHERD, Barry J. **Graphics file formats: reference and guide**. Greenwich: Manning Publications, 1995. 472 p.
- CANTU, Marco. **Dominando o Delphi 6: a bíblia**. São Paulo: Makron Books, 2002. 934 p.
- CASACURTA, Alexandre; OSÓRIO, Fernando; FRANZ, Figueroa; MUSSE, Soraia Raupp. **Computação gráfica - introdução**. 1998. 78 f. Disponível em <http://www.unipan.br/odair/CGSM/Apostilas/CS_Unicinos.Pdf>. Acesso em: 21 agosto 2004.
- COSTA, Silvia Maria Farani. **Classificação e reconhecimento de impressões digitais**. 2000. 123 f. Dissertação (Mestrado em Engenharia Elétrica) – Escola Politécnica da Universidade de São Paulo, São Paulo. Disponível em: <<http://sim.lme.usp.br/publicacoes/exames/pdf/QualiSi.pdf>>. Acesso em: 05 out. 2004.
- FINGERPRINTS. **Small subset of our fingerprint database**. Cesena, [2004?]. Disponível em: <<http://bias.csr.unibo.it/research/biolab/Fingdb.zip>>. Acesso em: 20 out. 2004. Biometric System Lab, Department of Computer Science, University of Bologna, Cesena, Italy.
- FVC. **Fingerprint verification competition: first international competition for fingerprint verification algorithms**. Cesena, 2004. Disponível em: <<http://bias.csr.unibo.it/fvc2004>>. Acesso em: 19 out. 2004. Biometric System Lab, Department of Computer Science, University of Bologna, Cesena, Italy.
- FACON, Jacques. **Processamento e análise de imagens**. Córdoba: CEFET, 1993. 198 p.
- FAUSETT, Laurene V. **Fundamentals of neural networks: architectures, algorithms, and applications**. Englewood Cliffs: Prentice Hall International, 1994. 461 p.

GONZALEZ, Rafael C.; WOODS, Richard E. **Processamento de imagens digitais**. Tradução Roberto Marcondes César Filho e Luciano da Fontoura Costa. São Paulo: Edgar Blucher Ltda, 1992. 509 p.

GUMZ, Rafael Araújo. **Protótipo de um sistema de identificação de minúcias em impressões digitais utilizado 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. **Redes neurais: princípios e práticas**. Tradução de Paulo Martins Engel. Porto Alegre: Bookman, 2001. 900 p.

HONG, Lin. **Automatic personal identification using fingerprints**. 1998. 242 f. Dissertação (Doutorado em Filosofia) – Departamento de Ciências da Computação, Michigan State University, Ann Arbor. Disponível em: <<http://www.cse.msu.edu/publications/tech/TR/MSUCPS-98-24.ps.gz>>. Acesso em: 14 ago. 2004.

KEHDY, Carlos. **Elementos de criminalística**. 3. ed. São Paulo: Sugestões Literárias, 1968. 267 p.

LOESCH, Cláudio. **Redes neurais artificiais, fundamentos e modelos**. Blumenau: Editora da FURB, 1996. 166 p.

MASTERS, Timothy. **Practical neural network recipes in C++**. San Diego: Academic Press, 1993. 493 p.

OSÓRIO, Fernando Santos. **Um estudo sobre reconhecimento visual de caracteres através de redes neurais**. 1991. 310 f. Dissertação (Mestrado em Computação).

OSÓRIO, Fernando Santos; BITTENCOURT, João Ricardo. **Sistemas inteligentes baseados em redes neurais artificiais aplicados ao processamento de imagens**. 2000. 30 f.

PANKANTI, Sharath. **Biometrics: promising frontiers for emerging identification market**, **Howthorne**, 2000. 16 f. Disponível em: <<http://www.cse.msu.edu/publications/tech/TR/MSU-CSE-00-2.ps.gz>>. Acesso em: 20 ago 2004.

PROGLOBO. **Sensores em biometria**. São Paulo, 1996. Disponível em: <<http://web01.proglobo.pt:8080/backo/html/SolucoesGA.htm>>. Acesso em: 11 maio 2004.

RATIONAL. **Rational Rose: a Rational suite product**. Cupertino – Califórnia, 2002. Disponível em: <<http://www.rational.com/products/rose/index.jsp>>. Acesso em: 29 jan. 2004. Rational Software Corporation.

SCURI, Antonio Escaño. **Fundamentos da imagem digital**. 1999. 67 f. Disponível em: <<http://www.tecgraf.puc-rio.br/~rtoledo/cg1/apostila%20imagem%20digital.pdf>>. Acesso em: 21 ago 2004.

SILVA, Alex Sandro. **Protótipo de software para classificação de impressão digital**. 1999. 81 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

TAVARES JÚNIOR, Gilberto da Silva. **A papiloscopia nos locais do crime**. São Paulo: Ícone, 1991. 159 p.

TAFNER, Malcon A.; XEREZ, Marcos de; RODRIGUES FILHO, Ilson W. **Redes neurais artificiais introdução e princípios da neurocomputação**. Blumenau: Editora da FURB, 1996. 196p.

WELSTEAD, Stephen T. **Neural network and fuzzy logic applications in C/C++**. New York: John Wiley & Sons, 1994. 494 p.