

UNIVERSIDADE DO EXTREMO SUL CATARINENSE – UNESC

CURSO DE CIÊNCIA DA COMPUTAÇÃO

MÁRCIO JOSÉ CONTE

RECONHECIMENTO DE CÓDIGO DE BARRAS EM IMAGENS DIGITAIS

POR MEIO DO MODELO *PERCEPTRON* MULTICAMADAS

CRICIÚMA, DEZEMBRO DE 2006.

MÁRCIO JOSÉ CONTE

**RECONHECIMENTO DE CÓDIGO DE BARRAS EM IMAGENS DIGITAIS
POR MEIO DO MODELO *PERCEPTRON* MULTICAMADAS**

Trabalho de Conclusão do Curso para obtenção do
Grau de Bacharel em Ciência da Computação da
Universidade do Extremo Sul Catarinense

Orientadora: Prof^a. M. Sc. Merisandra Côrtes de Mattos
Co-orientador: Prof. M. Eng. Evânio Ramos Nicoleit

CRICIÚMA, DEZEMBRO DE 2006.

MÁRCIO JOSÉ CONTE

**RECONHECIMENTO DE CÓDIGO DE BARRAS EM IMAGENS DIGITAIS
POR MEIO DO MODELO *PERCEPTRON* MULTICAMADAS**

Submetido ao corpo docente do Departamento de Ciência da Computação da
Universidade do Extremo Sul Catarinense como um dos requisitos para obtenção do
grau de Bacharel em Ciência da Computação.

Prof^ª. M.Sc. Ana Cláudia Garcia Barbosa
Coordenadora do Curso de Ciência da Computação

Banca Examinadora:

Prof^ª. M.Sc. Merisandra Côrtes de Mattos (UNESC)
Orientadora

Prof. M.Eng. Evânio Ramos Nicoleit (UNESC)
Co-Orientador

Prof^ª. M.Sc. Priscyla Waleska T. A. Simões (UNESC)

Prof. John Cleber Jaracescki (UNESC)

Dedico aos meus queridos pais, José e Zenita, ao meu irmão, a minha noiva, aos meus avós, enfim, a todos os meus familiares que não mediram esforços em me apoiar. Sem eles, nada seria possível.

AGRADECIMENTOS

Espero que estas palavras consigam expressar a gratidão às pessoas que tornaram possível a realização desse sonho.

Agradeço:

Primeiramente a Deus, o doador da vida e de todos os meus dons, que permitiu esta conquista.

Aos meus pais, José e Zenita, talvez não com a intensidade que gostaria, pois palavras não são capazes de representar o tamanho da minha gratidão.

A minha noiva Daniela, que esteve comigo em todos os momentos desta caminhada.

Sem seu apoio, compreensão e amor, com certeza eu não teria conseguido.

Obrigado por estar ao meu lado nos momentos difíceis, por ouvir minhas teorias, mesmo sem entender nada, obrigado por compreender minhas ausências, e inquietações. Amo você, e quero lhe dizer muito obrigado mesmo sendo insuficiente para demonstrar o quanto estou grato. Obrigado, Dani!

Ao meu irmão Moacir, meus familiares e amigos que souberam entender minhas ausências nos finais de semana, enquanto eu estava estudando; podem ter certeza de que eu preferia estar com vocês. Mas valeu a pena!

Obrigado por aquele abraço, por aquela oração.

Aos Meus colegas de trabalho, que muitas vezes foram ouvintes de minhas dificuldades.

Obrigado pela compreensão e ajuda.

A minha orientadora, Merisandra Côrtes de Mattos e ao meu co-orientador, Evânio Ramos Nicoleit, obrigado por me socorrer tantas vezes, dividir seus conhecimentos e suas experiências comigo, se interessar pelo meu sucesso. Obrigado, professores!

*“O primeiro dever da
inteligência é desconfiar
dela mesma”*

(Albert Einstein)

RESUMO

A decodificação e o reconhecimento de imagens envolve uma área de pesquisa muito importante e promissora, no que diz respeito a classificação e identificação de objetos e informações nela contidas. Nem sempre, esta é uma tarefa simples de ser executada, pois a fim de se obter bons resultados, exige-se um grande conhecimento nas técnicas envolvidas e do contexto do problema a ser resolvido. A pesquisa na área de Inteligência Artificial tem evoluído bastante, principalmente nas últimas décadas, o que forneceu uma vasta gama de conhecimentos necessários para o desenvolvimento deste trabalho. Assim, esta pesquisa compreendeu o desenvolvimento de um protótipo para o reconhecimento de imagens de código de barras codificação EAN13, denominado VisionScan. Esta especificação é padrão na maioria dos produtos nacionais encontrados nos estabelecimentos comerciais. O VisionScan foi desenvolvido sob tecnologia Java e implementou, além de uma aplicação móvel para leitura das imagens, uma rede neural artificial do tipo *Perceptron* Multicamadas, utilizando o algoritmo *Backpropagation* no treinamento da mesma e técnicas do processamento digital de imagens tais como: aquisição, pré-processamento e segmentação. Visando a comunicação entre as aplicações, implementou-se um serviço Web que recebe as solicitações geradas no ambiente móvel e as encaminha ao reconhecedor de imagens. Concluindo a pesquisa, o VisionScan foi treinado e testado por meio de uma base de imagens geradas com o auxílio da API Barcode4j atingindo para estes exemplos uma taxa de acerto de 86%.

Palavras-chave: Redes Neurais Artificiais, Algoritmo *Backpropagation*, *Perceptron* Multicamadas, Processamento Digital de Imagens, Código de Barras Padrão EAN13.

ABSTRACT

The decoding and image identification involve a very important and profitable research area concerning to object classification-identification and information within it. Not always, it is a simple task to be executed, for in order to obtain good results, it is demanded a great knowledge of techniques and the context of the problem to be solved. The research on Artificial Intelligence field has been largely developing, mainly on the last decades, which provided the user with a wide variety of necessary tools for the development on this work area. Therefore, this research aimed the development of a prototype for the identification of images of bar code EAN13, named VisionScan. Such specification is standardized in most the national products found at stores. VisionScan has been developed based on Java technology and implemented both a mobile application for image reading, a MultiLayer *Perceptron* artificial neural network, using algorithm *Backpropagation* in its training and digital image processing techniques such as: acquisition, pre-processing e segmentation. It has also been implemented a Web service aiming the communication between the applications, which receive the requirements generated in the mobile environment and takes them to the image identifier. Concluding the research, the VisionScan has been taught and tested through an image base generated with the assistance of API Barcode4j with a assertion mark of 86%.

Key-words: Artificial Neural Network, *Backpropagation* Algorithm, MultiLayers *Perceptron*, Digital Image Processing, Standard Bar Codes EAN13.

LISTA DE ILUSTRAÇÕES

Figura 1. Esquema do neurônio de McCulloch e Pitts.....	23
Figura 2. Rede neural artificial organizada em camadas.	26
Figura 3. Rede <i>feedforward</i> e <i>feedback</i>	27
Figura 4. Representação artificial de um neurônio natural.	27
Figura 5. Rede de Kohonen com topologia hexagonal.	28
Figura 6. Rede de Hopfield.....	30
Figura 7. Modelo <i>perceptron</i> multicamadas.....	33
Figura 8. Representação numérica de uma imagem em tons de cinza de 10x10 <i>pixels</i> ..	37
Figura 9. Elementos do processo de análise da imagem.....	38
Figura 10. Composição do código de barras EAN-13.....	48
Figura 11. Passos executados no cálculo do DV.....	49
Figura 12. Diagrama de pacotes do VisionScan.....	52
Figura 13. Diagrama de caso de uso das funções do especialista.....	53
Figura 14. Diagrama de caso de uso das funcionalidades disponíveis ao usuário.....	54
Figura 15. Diagrama de atividade do VisionScan.....	55
Figura 16. Modelo de arquitetura usada no VisionScan.....	57
Figura 17. Tela inicial da aplicação cliente móvel.....	58
Figura 18. Tela de captura da imagem do código de barras.....	59
Figura 19. Tela de resultados obtidos no VisionScan.....	59
Figura 20. Taxa de acerto obtida para diferentes entradas na rede.....	66

LISTA DE SIGLAS

API	Aplication Program Interface
ART	Adaptive Resonance Theroy
ASL	Apache Software Licence
EAN	European Article Number
EDVAC	Eletronic Discrete Variable Automatic Computer
ENIAC	Eletronic Numerical Integrator and Computer
GPL	General Public License
GPRS	General Packet Radio Service
GTIN	Global Trade Identification Number
IA	Inteligência Artificial
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
INNS	International Neural Networks Society
IP	Internet Protocol
JAI	Java Advanced Imaging
MIDP	Mobile Information Device Profile
MLP	Multi Layer Perceptron
RN	Redes Neurais
RNA	Redes Neurais Artificiais
SNNS	Stuttgart Neural Network Simulator
TCP	Transmission Control Protocol
UML	Unified Modeling Language

SUMÁRIO

1 INTRODUÇÃO	14
1.1 OBJETIVO GERAL	15
1.2 OBJETIVOS ESPECÍFICOS	15
1.3 JUSTIFICATIVA.....	16
1.4 ESTRUTURA DO TRABALHO.....	17
2 INTELIGÊNCIA ARTIFICIAL CONEXIONISTA	19
2.1 REDES NEURAS ARTIFICIAIS	21
2.1.1 História das Redes Neurais Artificiais	24
2.1.2 Arquitetura das Redes Neurais Artificiais	26
2.1.3 Modelo de Redes Neurais Artificiais	27
2.1.3.1 Rede de Kohonen	28
2.1.3.2 Rede de Hopfield.....	29
2.1.3.3 <i>Perceptron</i> Multicamadas	31
2.1.4 O Processo de Aprendizagem nas Redes Neurais	31
3 O MODELO <i>PERCEPTRON</i> MULTICAMADAS	33
3.1 O ALGORITMO DE APRENDIZAGEM <i>BACKPROPAGATION</i>	34
4 IMAGENS DIGITAIS	37
4.1 PROCESSAMENTO DIGITAL DE IMAGENS	37
4.1.1 Aquisição	39
4.1.2 Pré-processamento	39
4.1.3 Segmentação	39
4.1.4 Representação e Descrição	40

4.1.5 Reconhecimento.....	41
4.1.6 Interpretação	42
5 EXEMPLOS DE FERRAMENTAS E PESQUISAS DESENVOLVIDAS COM AS TÉCNICAS DE REDES NEURAS ARTIFICIAIS.....	43
5.1 NEURON COLOR	43
5.2 SISTEMA DE IDENTIFICAÇÃO DE IMPRESSÕES DIGITAIS UTILIZANDO REDES NEURAS ARTIFICIAIS	44
5.3 RECONHECIMENTO AUTOMÁTICO DE PLACAS DE VEÍCULOS AUTOMOTORES POR MEIO DE REDES NEURAS ARTIFICIAIS.....	44
6 VISIONSCAN - RECONHECEDOR DE IMAGENS DE CÓDIGOS DE BARRAS POR MEIO DE REDES NEURAS.....	46
6.1 CÓDIGO DE BARRAS PADRÃO EAN13	47
6.2 METODOLOGIA	49
6.2.1 Levantamento Bibliográfico.....	50
6.2.2 Modelagem do VisionScan	51
6.2.2.1 Diagrama de Pacotes	51
6.2.2.2 Diagrama de Casos de Uso	53
6.2.2.3 Diagrama de Atividades	54
6.2.3 Definição da Arquitetura do VisionScan	56
6.2.4 Implementação da Interface Cliente	57
6.2.5 Implementação do Reconhecedor de Imagens	60
6.2.5.1 Biblioteca Java para Manipulação de Imagens	61
6.2.5.2 Modelagem Matemática do Algoritmo de Aprendizagem <i>Backpropagation</i>	61
6.2.5.3 Treinamento da Rede Neural	64
6.2.6 Implementação da Aplicação Servidor	65

6.3 TESTES E RESULTADOS OBTIDOS.....	65
CONCLUSÃO	67
REFERÊNCIAS	69
BIBLIOGRAFIA RECOMENDADA	72

1 INTRODUÇÃO

Nos últimos tempos, torna-se cada vez maior o número de pessoas que são influenciadas pelas facilidades do mundo moderno. A praticidade nos serviços prestados talvez seja hoje, um dos principais diferenciais que uma empresa pode oferecer aos seus clientes. O uso da telefonia móvel como opção em boa parte das transações bancárias é ótimo exemplo de tais falidades.

A possibilidade de utilizar um telefone celular no conforto de sua residência, fotografar um código de barras de uma mercadoria e a partir dessa imagem enviar solicitações para serviços diversos a um estabelecimento comercial de sua preferência a fim de se obter informações ou até mesmo a compra deste produto, pode ser uma das muitas possibilidades a serem exploradas por diversos clientes de uma determinada empresa.

Portanto, as empresas para que possam oferecer tais serviços, necessitam implementar meios para o reconhecimento destas imagens, sendo que o uso da inteligência artificial conexionista por meio das redes neurais artificiais, tem-se apresentado como uma boa alternativa na solução de problemas deste tipo (OSÓRIO, 1999).

A presente pesquisa compreendeu o desenvolvimento do protótipo de um sistema para o reconhecimento de imagens dos códigos de barras de determinados produtos com codificação padrão EAN13¹ obtidos por meio da câmera digital de um telefone celular. O reconhecimento destas imagens foi realizado aplicando-se um modelo de redes neurais do tipo *perceptron* multicamadas e, considerando que devido as limitações da computação móvel, as imagens podem ser recebidas com uma

¹ Código de barras linear numérico que representa os 13 dígitos do GTIN-13.

qualidade pequena e em situações adversas foram também utilizadas técnicas da computação gráfica como: pré-processamento, segmentação e representação, no que se refere a decodificação de imagens.

1.1 OBJETIVO GERAL

Desenvolver o protótipo de um sistema de redes neurais para o reconhecimento de imagens referentes a código de barras de produtos obtidos a partir da câmera digital de aparelhos celulares.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos desta pesquisa consistem em:

- a) compreender o processo de reconhecimento de imagens por meio de redes neurais;
- b) implementar um modelo de redes neurais artificiais;
- c) aplicar um algoritmo para a aprendizagem da rede neural;
- d) utilizar funções *Application Program Interface* (API) da tecnologia Java que auxilie no processo de transferência e decodificação de imagens;
- e) implementar uma aplicação cliente capaz de capturar imagens da câmera digital de um celular e enviá-las para o processamento no servidor;
- f) desenvolver uma camada responsável pela interface entre o cliente móvel, o reconhecedor de imagens e a aplicação servidora.

1.3 JUSTIFICATIVA

O protótipo de sistema desenvolvido neste trabalho teve como principal função, servir como ferramenta de apoio a sistemas de informação, provendo um modelo de acesso e identificação dos dados armazenados em suas bases.

Atualmente os sistemas de informação para realizar consultas em suas bases de dados, utilizam normalmente códigos, palavras-chave ou qualquer outro identificador que atenda ao contexto da busca e, com o reconhecimento de padrões em imagens, as características contidas na base de conhecimento² da rede, poderão também ser usadas como identificador de acesso aos dados.

Entretanto, para resolver com eficiência os problemas relacionados ao reconhecimento de imagens, o uso de redes neurais demonstra-se bastante interessante, pois propõem um modelo matemático com base na estrutura do cérebro humano (OSÓRIO, 1999).

As redes neurais consistem em um processador paralelamente distribuído e com unidades simples de processamento, capaz de armazenar conhecimento e torná-lo disponível para o uso (RUSSEL; NORVIG, 2004).

Conforme Osório (1999) esta utilização das redes neurais apresenta diversas vantagens, dentre as quais: aquisição de novos conhecimentos a partir do conjunto de treinamento, manipulação de diferentes tipos de dados como, por exemplo, quantitativos e aproximados; além de apresentarem potencial para a representação de conhecimento por meio de relações ponderadas entre as entradas da aplicação.

Mas para se obter um melhor resultado no processo de reconhecimento das imagens por meio de redes neurais é necessário primeiramente aplicar técnicas de

² Conjunto de imagens utilizadas para o treinamento da rede neural.

processamento de imagens, pois proporcionam o tratamento destas com o intuito de facilitar e melhorar o processo de reconhecimento (RUSSEL; NORVIG, 2004).

Contudo, o uso dos dispositivos móveis, como a câmera do celular, para obtenção das imagens, pode constituir-se em uma das possíveis utilizações para o protótipo VisionScan desenvolvido.

1.4 ESTRUTURA DO TRABALHO

O trabalho apresentado é composto por sete capítulos. No Capítulo 1 encontra-se uma contextualização do tema proposto, bem como os objetivos e justificativa para realização desta pesquisa.

No Capítulo 2, são abordados os conceitos fundamentais da inteligência artificial conexionista, indispensáveis para o entendimento do trabalho.

Os principais fundamentos das redes neurais artificiais são apresentados nos Capítulos 2 e 3, compreendendo um breve histórico que destaca os fatos mais importantes da área; alguns dos principais modelos de redes neurais artificiais (RNA) e seus respectivos algoritmos de treinamento, enfatizando-se o modelo de redes multicamadas e o algoritmo *backpropagation* que foram utilizados no desenvolvimento desta pesquisa.

O Capítulo 4 descreve respectivamente os conceitos de imagem digital e processamento digital de imagens, utilizados na obtenção e no tratamento das imagens que antecedem o processo de reconhecimento, enquanto no Capítulo 5 são abordadas algumas pesquisas, correlatas a esta, desenvolvidas em diferentes instituições e centros de pesquisas.

O Capítulo 6 apresenta o trabalho desenvolvido, sendo descritos os passos do processo de desenvolvimento desta pesquisa, bem como a exposição dos resultados obtidos.

E por fim, tem-se a conclusão, onde encontram-se também algumas sugestões para trabalhos futuros.

2 INTELIGÊNCIA ARTIFICIAL CONEXIONISTA

Segundo Russel e Norvig (2004), as definições de Inteligência Artificial (IA), encontradas na literatura científica, podem ser agrupadas em quatro categorias:

- a) sistemas que pensam como humanos;
- b) sistemas que agem como humanos;
- c) sistemas que pensam corretamente;
- d) sistemas que agem corretamente.

As duas primeiras categorias (a e b) são, essencialmente, empíricas³ e envolvem formulação de hipóteses e confirmação experimental, enquanto as outras duas (c e d) são teóricas e envolvem matemática e engenharia. Embora essas categorias de pesquisa muitas vezes pareçam antagônicas, os resultados obtidos em ambas têm contribuído bastante para o desenvolvimento da área de IA.

A área de IA apresenta, segundo Luger (2004) três abordagens, as quais são:

- a) **IA simbólica:** baseada na hipótese do sistema de símbolos físicos, segundo a qual um conjunto de estruturas simbólicas e de regras de manipulação dessas estruturas são os meios suficientes para se criar inteligência. Essa linha de pesquisa trata de problemas bem definidos e seu principal produto são os sistemas baseados em regras;
- b) **IA conexionista:** baseada na hipótese da causa-efeito, segundo a qual um modelo matemático do cérebro humano é suficiente para reproduzir a inteligência. Essa linha de pesquisa trata de problemas sem definição precisa, mas que são conhecidos por meio de exemplos, tem-se nesta abordagem as técnicas de Redes Neurais (RN);

³ Linha do conhecimento que se fundamenta apenas na experiência.

c) **IA evolucionária:** baseada na teoria evolutiva de Darwin⁴, pode-se modelar sistemas inteligentes simulando a evolução de uma população de indivíduos (soluções aleatórias), que carregam genes⁵ com informação suficiente para solução de um problema, usando operações genéticas de recombinação e mutação. Essa linha de pesquisa trata de problemas de otimização, utilizando-se para isso de algoritmos genéticos⁶.

Dentre essas abordagens da IA, esta pesquisa concentra-se na IA Conexionista, que de acordo com Russel e Norvig (2004) nasceu praticamente junto com a linha simbólica, mas devido a uma série de razões, as técnicas simbólicas de IA, baseadas na lógica, tiveram preferência na época. Entre estas razões, pode-se citar a falta de computadores suficientemente potentes para tratar a complexidade inerente ao método.

Mais tarde, os sucessos obtidos em aplicações práticas levaram a uma mudança de ênfase quanto ao objetivo das pesquisas de alcançar uma inteligência similar a humana. Originando-se, assim, uma linha de pesquisa que passa a se dedicar ao estudo das redes neurais vistas apenas como uma representação de funções matemáticas utilizando elementos computacionais aritméticos simples, sem maiores relações com a modelagem do sistema nervoso humano. Em contrapartida, outra parte da pesquisa passa a estudar as redes neurais com base no comportamento do cérebro humano, mas com uma grande simplificação do modelo original (HAYKIN, 2001).

⁴ Teoria da evolução moderna baseada na seleção natural.

⁵ Unidade responsável pela transmissão das características hereditárias de uma geração.

⁶ Parte da área dos sistemas inteligentes inspirados na simulação de processos naturais.

2.1 REDES NEURAIIS ARTIFICIAIS

Redes neurais artificiais são técnicas computacionais que propõem um modelo matemático baseado na estrutura neural de organismos inteligentes, mais especificamente o cérebro humano (TAFNER et al, 1996).

Segundo Haykin (1999), uma rede neural é um processador distribuído, constituído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento e torná-lo disponível para o uso.

A principal característica das redes neurais é a capacidade de aprender a partir de exemplos e, assim, classificar novos padrões. Desde o início da década de 80, as redes neurais têm sido aplicadas com sucesso a uma vasta gama de problemas. Dentre as aplicações que utilizam redes neurais pode-se citar: reconhecimento de objetos, mineração de dados, reconhecimento de fala, robótica, entre outras (BARRETO, 2001).

As redes neurais artificiais, também conhecidas como métodos conexionistas, possuem características peculiares de representação e de aquisição de conhecimentos, sendo considerado um método de nível sub-simbólico, onde o processamento da informação acontece em um nível em que os padrões representam conjuntos de dados que não podem ser associados a um significado imediato (FERNANDES, 2003).

De acordo com Barreto (2001), as características que tornam a metodologia das RNA interessantes para a solução de problemas são:

- a) capacidade de aprender por meio de exemplos e de generalizar este aprendizado de maneira a reconhecer instâncias similares que nunca haviam sido apresentadas como exemplo;

- b) bom desempenho em tarefas mal definidas, onde falta o conhecimento explícito sobre como encontrar uma solução;
- c) não requer conhecimento a respeito de eventuais modelos matemáticos dos domínios de aplicação;
- d) elevada imunidade ao ruído, isto é, o desempenho de uma RN não entra em colapso com a presença de informações falsas ou ausentes, como é o caso nos programas convencionais, mas piora de maneira gradativa;
- e) possibilidade de simulação de raciocínio *a priori* e impreciso, por meio da associação com a lógica *fuzzy*⁷.

Segundo Tafner, Xerez e Rodrigues Filho (1996) dentre as características das RNA, a de maior importância é o conhecimento que está codificado na estrutura da rede, onde se destacam as conexões entre as unidades que a compõe.

Nas redes neurais artificiais obtidas por simulação em computadores, associa-se a cada conexão um peso sináptico⁸, sendo que o aprendizado em uma RNA é realizado por um processo de adaptação destes pesos (BRAGA; CARVALHO; LUDERMIR, 2000).

Segundo Loesch e Sari (1996) o primeiro modelo de um neurônio artificial foi proposto pelo neurofisiologista Warren McCulloch e pelo matemático Walter Pitts em 1943. O trabalho de McCulloch e Pitts descreve um cálculo lógico das redes neurais que unifica os estudos de neurofisiologia e da lógica matemática. A Figura 1 mostra um esquema do neurônio artificial proposto por eles. A operação dessa unidade de processamento pode ser resumida da seguinte forma:

- a) sinais são apresentados as entradas (X1, X2, ..., Xp);

⁷ Lógica utilizada em ambientes onde a definição do problema apresenta imprecisões.

⁸ Valor numérico que caracteriza a força da conexão entre dois neurônios.

- a) cada sinal é multiplicado por um número, ou peso, que indica sua influência na saída da unidade (W_1, W_2, \dots, W_p);
- b) é realizada uma soma ponderada dos sinais, produzindo um nível de atividade (Σ);
- c) se este nível de atividade exceder um certo limite *threshold* a unidade produz uma determinada resposta de saída (Y), caso contrário; o neurônio permanece inativo.

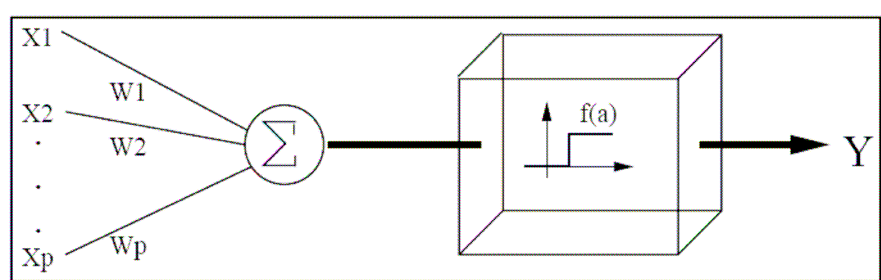


Figura 1. Esquema do neurônio de McCulloch e Pitts.

Fonte: Adaptado de BRAGA, A.; CARVALHO, A.; LUDERMIR, T. (2000)

As entradas do neurônio artificial podem ser comparadas com os estímulos do neurônio natural. Os pesos são valores que representam o grau de importância que cada entrada possui em relação aquele determinado neurônio. A função de ativação é responsável por decidir o que fazer com o valor resultante do somatório das entradas ponderadas.

A partir destes estudos de McCulloch e Pitts vários outros modelos se originaram. A seguir são apresentadas algumas notas históricas a respeito do estudo das RNA.

2.1.1 História das Redes Neurais Artificiais

Nos anos 40 surgiram os primeiros estudos sobre as RNA e propostas de modelos, os quais evoluíram bastante e se tornaram conhecidos (BRAGA; CARVALHO; LUDERMIR, 2000).

Segundo Haykin (2001) os estudos de Warren McCulloch e Walter Pitts tiveram muita repercussão no meio científico, tendo influenciado Von Neumann a usar chaves de atraso⁹ idealizadas, derivadas deste modelo de neurônio artificial, na construção do *Electronic Discrete Variable Automatic Computer* (EDVAC¹⁰) que foi desenvolvido a partir do *Electronic Numerical Integrator and Computer* (ENIAC¹¹).

Em 1949 Donald Hebb escreveu um livro intitulado “*The Organization of Behavior*”, suas idéias não eram completamente novas, mas foi o primeiro a propor uma lei de aprendizagem específica para as sinapses dos neurônios (LOESCH; SARI, 1996). Também proveniente deste período foi a construção do primeiro neuro computador, denominado *Snark*, por Marvin Minsky, em 1951. Embora nunca tenha executado qualquer função de processamento de informação interessante, o *Snark* serviu de inspiração para trabalhos futuros.

Em 1956, durante a 1ª Conferência Internacional de Inteligência Artificial, foi apresentado um modelo de RNA pelo pesquisador Nathaniel Rochester, da IBM. Seu trabalho consistia numa simulação de centenas de neurônios interconectados, por meio da construção de um sistema para verificar como a rede responderia aos estímulos ambientais. Entretanto, o primeiro neuro computador a obter sucesso foi o *Mark I Perceptron*, que surgiu no final de 1957, criado por Frank Rosenblatt, Charles Wightman e outros (BRAGA; CARVALHO; LUDERMIR, 2000).

⁹ Implementação de memória utilizada no primeiro projeto do EDVAC.

¹⁰ Computador que utilizava memória constituída por lâmpadas de mercúrio e foi o sucessor do ENIAC.

¹¹ Primeiro computador eletrônico de propósito geral.

Em 1969, Marvin Minsky e Seymour Papert publicaram o livro *Perceptron*, onde resumiram e criticaram seriamente a pesquisa sobre RN e, devido a importância e o respeito que a comunidade científica tinha por Minsky e Papert, uma retração nos investimentos e programas de pesquisa para essa tecnologia foi inevitável.

Um período de pesquisa silenciosa seguiu-se, quando poucos resultados foram publicados. Até que em 1982, o físico e biólogo do Instituto de Tecnologia da Califórnia John Hopfield deu um novo impulso as redes neurais, contestando, com sucesso, as teses matemáticas de Minsky e Papert (LOESCH; SARI, 1996).

Segundo Russel e Norvig (2004) com o trabalho de Hopfield e a criação do algoritmo *backpropagation* para o treinamento de redes com múltiplas camadas em 1986, as redes neurais ganharam novamente credibilidade. Houve, em termos de publicação, vários trabalhos, deixando claro que as pesquisas nunca pararam, pois nem todos esses trabalhos poderiam ter sido produzidos em tão pouco tempo.

Em 1987 ocorreu em São Francisco a Primeira Conferência de Redes Neurais em tempos modernos, a *International Conference on Neural Networks*, evento do IEEE e também foi formada a *International Neural Networks Society* (INNS).

A partir destes acontecimentos ocorreu a fundação do INNS Journal em 1989, seguido do Neural Computation e do IEEE Transactions on Neural Networks em 1990. Desde então, muitas universidades anunciaram a formação de institutos de pesquisa e programas de educação em neuro computação, estudando novos métodos e desenvolvendo novas arquiteturas conforme serão abordadas na próxima seção.

2.1.2 Arquitetura das Redes Neurais Artificiais

A maioria das arquiteturas neurais são tipicamente organizadas em camadas, apresentando suas unidades conectadas tanto com as camadas posteriores quanto anteriores (Figura 2). Estas camadas são normalmente classificadas em três grupos (HAYKIN, 2001):

- a) camada de entrada: recebe os padrões de entrada e repassá-os à camada seguinte;
- b) camada oculta: onde é processado o conhecimento da rede;
- c) camada de saída: o resultado final é concluído e apresentado.

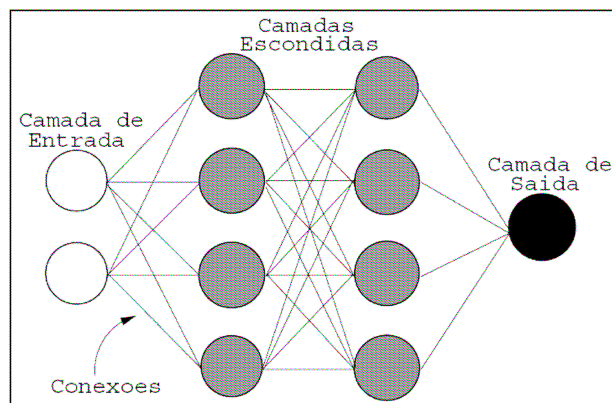


Figura 2. Rede neural artificial organizada em camadas.
Fonte: Adaptado de HAYKIN, S. (2000)

Segundo Haykin (2001), as conexões entre as camadas podem gerar várias estruturas diferentes (arquiteturas). Do ponto de vista dessas conexões, a arquitetura da rede pode ser caracterizada por dois aspectos, ilustrados na Figura 3:

- a) *feedforward*: quando o sinal de saída de um neurônio não é utilizado como entrada para as camadas anteriores ou para os neurônios da própria camada;
- b) *feedback*: quando o sinal de saída de um neurônio serve de entrada para neurônios da mesma camada ou de camadas anteriores.

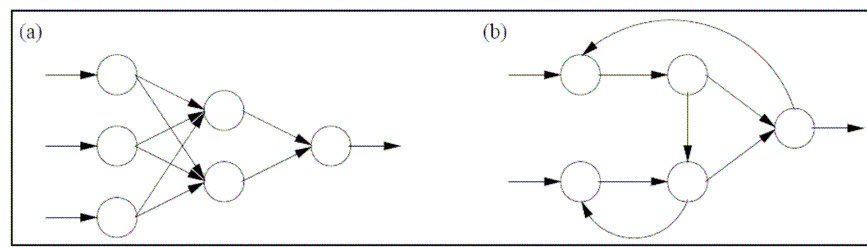


Figura 3. Rede *feedforward* (a) e *feedback* (b).
Fonte: Adaptado de HAYKIN, S. (2000)

Com o passar do tempo vários modelos de RNA foram criados com base nestas arquiteturas, sendo que alguns destes modelos serão apresentados a seguir.

2.1.3 Modelo de Redes Neurais Artificiais

Barreto (2001) explica que a construção de redes neurais artificiais é inspirada nos neurônios biológicos e nos sistemas nervosos, porém é importante compreender que atualmente as RNA possuem pequenas semelhanças com as redes neurais naturais (Figura 4).

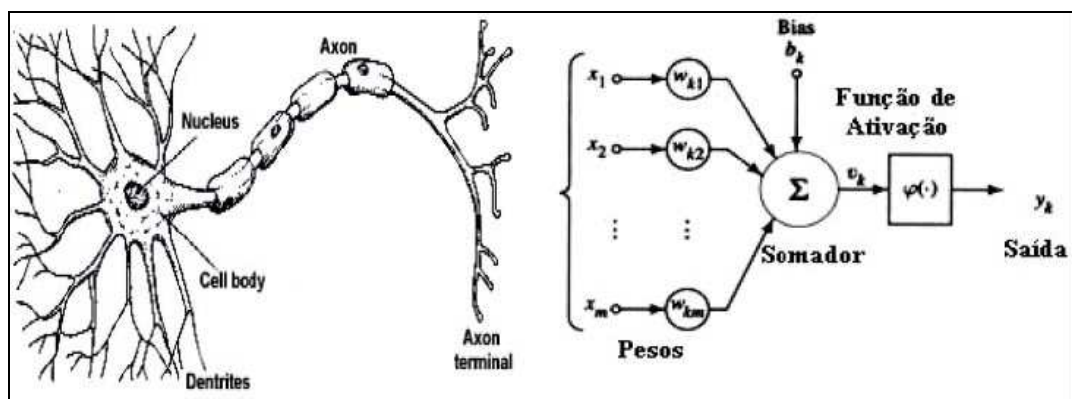


Figura 4. Representação artificial de um neurônio natural.
Fonte: BARRETO, J. (2001)

Na Figura 4 as entradas x_1, x_2, \dots, x_n , representam os sinais oriundos de outros neurônios. Os pesos $w_{k1}, w_{k2}, \dots, w_{kn}$ referem-se aos pesos sinápticos das

conexões entre os neurônios da camada anterior e da camada k . O Bias (b_k) é o limiar para a ativação do neurônio (CORDEIRO, 2002).

Dentre os modelos de RN, a seguir serão abordados alguns daqueles considerados clássicos conforme Haykin (2001), são eles: Kohonen, Hopfield e *Perceptron*.

2.1.3.1 Rede de Kohonen

O modelo de Kohonen é uma rede neural do tipo *feedforward* de treinamento não-supervisionado¹². Ao ser considerada uma rede de duas dimensões, o modelo de Kohonen não impõe nenhuma forma topológica, que pode ser triangular, retangular, hexagonal, entre outras. A Figura 5 mostra um exemplo de uma rede de Kohonen com topologia hexagonal (CORDEIRO, 2002).

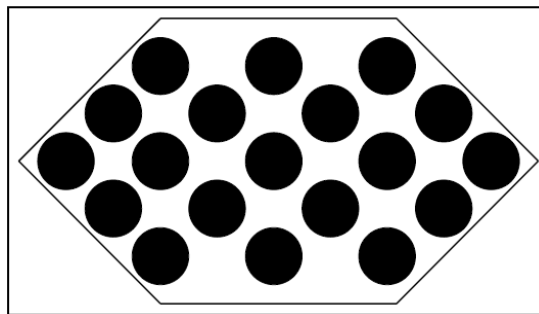


Figura 5. Rede de Kohonen com topologia hexagonal.
Fonte: Adaptado de CORDEIRO, F. (2002)

Segundo Cordeiro (2002), o esquema básico deste modelo tem a propriedade de modificar a si próprio. Os neurônios da camada competem entre si para serem os vencedores a cada modificação. O neurônio vencedor é aquele que gera a menor distância euclidiana¹³ entre os vetores de pesos e de entrada. Cada neurônio da rede representa uma saída. Todas as entradas são conectadas a todos os neurônios da

¹² Não há um crítico para supervisionar o processo de aprendizagem.

¹³ Distância geométrica em um espaço multidimensional (REIS; CARGNIN, 1997).

rede. Os pesos são inicializados com valores aleatórios baixos. Uma entrada X é apresentada à rede sem especificar a saída desejada. De acordo com a entrada, um neurônio Y deverá responder melhor e este será o vencedor. Os pesos do neurônio vencedor e de seus vizinhos serão ajustados.

Após todo o conjunto de treinamento ter sido apresentado à rede e os critérios de treinamento terem sido satisfeitos, considera-se a rede treinada. Na fase seguinte, chamada de teste, um conjunto de entradas é apresentado à RN sem haver alterações nos pesos.

Sendo assim, cada neurônio representa uma saída da rede. Isto significa que, se a rede possuir 10 neurônios, conseqüentemente, haverá 10 saídas possíveis para qualquer número de entradas que a mesma estiver sendo submetida. Além disso, cada neurônio está amplamente conectado com as entradas (TAFNER et al, 1996).

2.1.3.2 Rede de Hopfield

A rede de Hopfield é auto-associativa¹⁴ e consiste de um número de neurônios totalmente conectados, ou seja, cada neurônio está interligado a todos os outros, como mostra a Figura 6 (BRAGA; CARVALHO; LUDERMIR, 2000).

¹⁴ Rede capaz de calcular seu próprio valor de entrada.

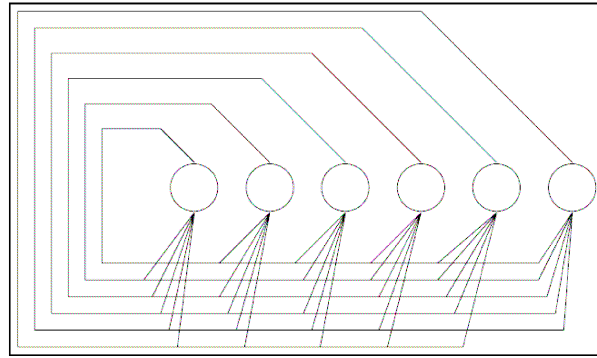


Figura 6. Rede de Hopfield
 Fonte: Adaptado de BRAGA, A.; CARVALHO, A.;
 LUDERMIR, T. (2000)

Ainda de acordo com Braga, Carvalho e Ludermir (2000) a rede de Hopfield tem como característica ser uma rede simetricamente pesada, isto é, os pesos das conexões de um neurônio para outro são os mesmos em ambas as direções. Cada neurônio tem um limiar e uma função-progresso¹⁵, assim o neurônio calcula a soma dos pesos de entrada e subtrai do valor limiar, passando este resultado para a função progresso que determina o estado de saída.

A rede aceita somente dois estados de entrada, que pode ser binário (0,1) ou bipolar (-1,+1). A principal característica da rede de Hopfield é que não existem conexões de entrada ou saída óbvias, ou seja, todos os neurônios são iguais (LOESCH; SARI, 1996).

Dessa forma, as entradas da rede são aplicadas para todos os neurônios de uma só vez. A rede procede em ciclo por meio de uma sucessão de estados, até convergir em uma solução estável, que ocorre quando os valores dos neurônios não alteram muito. Como cada neurônio está conectado a todos os outros, o valor de um afeta os demais (LOESCH; SARI, 1996).

¹⁵ Calcula o tempo decorrido no processo de aprendizagem.

2.1.3.3 *Perceptron* Multicamadas

De acordo com Loesch e Sari (1996), o modelo de RN *perceptron* foi o primeiro a ser desenvolvido. Possui um conceito de neurônio artificial que ainda hoje é usado. Cada neurônio calcula uma soma ponderada de suas entradas, e passa esta soma a uma função não-linear. Assim, determina-se quando uma entrada se encaixa ou não em certo padrão.

O modelo de RNA *perceptron* multicamadas foi utilizado nesta pesquisa, pois, segundo Loesch e Sari (1996) apresenta bom desempenho em situações que exijam tomada de decisões e por esse motivo, juntamente com seu algoritmo de aprendizagem, será comentado mais detalhadamente nos próximos capítulos.

2.1.4 O Processo de Aprendizagem nas Redes Neurais

Dentre as características das RN uma das mais importantes é a capacidade de aprender a partir de exemplos. Dessa forma, designa-se a sua aprendizagem quando a RN atinge uma solução para um problema desconhecido (RUSSEL; NORVIG, 2004).

De acordo com Haykin (2001), o tipo de aprendizagem é determinado pela maneira que a modificação dos parâmetros ocorre, tendo-se três paradigmas, os quais são:

- a) **aprendizagem supervisionada:** existe um mentor capaz de fornecer à rede neural uma resposta desejada para um vetor de treinamento inicial;
- b) **aprendizagem não-supervisionada:** não há um crítico para supervisionar o processo de aprendizagem, neste caso é fornecida

condição para realizar uma medida independente da tarefa cuja rede deve aprender;

- c) **aprendizagem por reforço:** é o mais geral dos três. Ao contrário de ser informado sobre o que fazer a partir de um instrutor, a rede aprende por meio do reforço. Em geral inclui o subproblema de aprender como o ambiente funciona.

De acordo com estes três paradigmas e o modelo de RN a ser implementado, tem-se algoritmos específicos, sendo que nesta pesquisa estudou-se apenas o algoritmo *backpropagation*, pois segundo Russel e Norvig (2004) é um dos mais utilizados para o modelo de RN *perceptron* multicamadas.

3 O MODELO *PERCEPTRON* MULTICAMADAS

A rede *perceptron* multicamadas é formada por neurônios do tipo *perceptron* derivados do modelo de McCulloch e Pitts.

Segundo Haykin (2000) a arquitetura deste modelo é organizada em camadas, conforme apresentado na Figura 7.

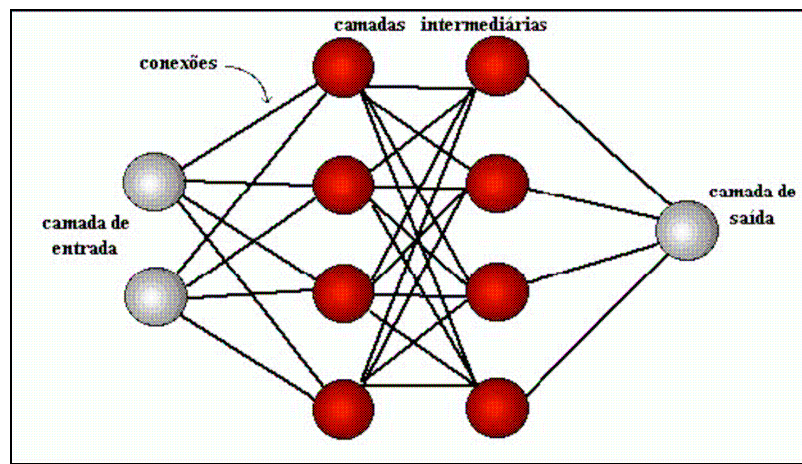


Figura 7. Modelo *perceptron* multicamadas
Fonte: Adaptado de HAYKIN, S. (2000)

O *perceptron* trabalha da seguinte forma: quando um neurônio artificial é ativado, ele recebe os valores obtidos nas entradas, realiza a soma ponderada, envia o resultado para a função de transferência e assim obtém o valor de saída. Este valor obtido é então comparado com a saída desejada, sendo que o erro é calculado pela diferença destes valores (HAYKIN, 2000).

O treinamento do modelo *perceptron* multicamadas é realizado pelo algoritmo *backpropagation*, desenvolvido por Rumelhart e colaboradores em 1986. Durante a sua utilização a rede opera em uma seqüência de dois passos (RUSSEL; NORVIG, 2004).

O primeiro passo é semelhante ao seu funcionamento na classificação, sendo que um padrão é apresentado à camada de entrada da rede e a atividade resultante flui por todas as camadas até que a resposta seja produzida pela camada de saída.

No segundo passo, a saída obtida é comparada a desejada para esse padrão particular e se não estiver correta, o erro é calculado e propagado a partir da camada de saída até a de entrada. Os pesos das conexões entre as unidades das camadas internas vão sendo modificados conforme o erro é retropropagado. Este ciclo é repetido até que o erro esteja dentro de um limite aceitável (HAYKIN, 2001).

Assim, os ciclos em que o conjunto de treinamento completo é apresentado à rede são chamados de épocas do treinamento, ou seja, cada vez que o conjunto completo é apresentado conta-se uma época (RUSSEL; NORVIG, 2004).

3.1 O ALGORITMO DE APRENDIZAGEM *BACKPROPAGATION*

O algoritmo *backpropagation* é baseado na regra de aprendizagem por correção de erro, ou seja, o erro das saídas é calculado e serve como entrada para a próxima camada, sendo constituído pela rede alimentada diretamente com múltiplas camadas. Ele é dividido basicamente em duas etapas: a propagação e a retropropagação (RUSSEL; NORVIG, 2004).

Conforme Fernandes (2003) a propagação consiste em disseminar os dados adquiridos pela camada de entrada por toda a rede, produzindo assim um conjunto de saída. O resultado desta saída é comparado com valores desejados. A partir de então, começa o processo de retropropagação ajustando os pesos segundo a diferença produzida, para fazer com que a resposta da rede se aproxime do desejado.

Os passos para execução do algoritmo *backpropagation* são apresentados a seguir (LOESCH; SARI, 1996):

- a) inicializa-se os pesos e os limiares com valores randômicos pequenos;
 b) apresenta-se a entrada $x_0, x_1, x_2, \dots, x_{n-1}$ e a saída desejada

$T_p = t_0, t_1, t_2, \dots, t_{m-1}$, onde n é o número dos nodos de entrada e m é o número dos nodos de saída;

- c) calcula-se a saída real. Cada camada calcula (1) e passa como entrada para a próxima camada. O valor final da camada de saída é y_{pj} ;

$$y_{pj} = f \left[\sum_{i=0}^{n-1} \omega_i x_i \right] \quad (1)$$

- d) adapta-se os pesos, iniciando pela camada de saída, em direção as camadas anteriores pela fórmula (2), onde $\omega_{ij}(t)$ representa os pesos a partir do nodo i até o nodo j no tempo t , η é a taxa de ganho ou taxa de aprendizagem e δ_{pj} é a taxa de erro para o padrão p no nodo j .

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \eta \delta_{pj} o_{pj} \quad (2)$$

- e) para as unidades de saída, calcula-se (3) enquanto para as unidades escondidas (4), onde \sum é o somatório sobre todos os k nodos na camada acima do nodo j .

$$\delta_{pj} = k o_{pj} (1 - o_{pj}) (t_{pj} - o_{pj}) \quad (3)$$

$$\delta_{pj} = k o_{pj} (1 - o_{pj}) \sum_k \delta_{pk} \omega_{jk} \quad (4)$$

Após a rede treinada e o erro aceitável no contexto da aplicação, a RN pode ser utilizada como uma ferramenta para classificação de novos dados. Para isto, ela deve ser utilizada apenas no modo progressivo (*feedforward*). Portanto, novas entradas serão

apresentadas a camada, processadas nas camadas intermediárias e os resultados apresentados na camada de saída, como no treinamento, mas sem a retropropagação do erro (HAYKIN, 2001). Dessa forma, a saída corresponde à interpretação da rede para a nova entrada apresentada.

O algoritmo *backpropagation* abordado nesta seção foi utilizado para o treinamento da rede *perceptron* multicamadas desenvolvida nesta pesquisa. Maiores explicações acerca da modelagem matemática são fornecidas no Capítulo 6.

No reconhecimento de imagens por meio de RNA além do entendimento acerca do modelo e do algoritmo de aprendizagem utilizado, segundo Russel e Norvig (2004) deve-se empregar técnicas do processamento digital de imagens, pois elas são muito importantes para se obter melhores resultados no reconhecimento por meio de RNA. Essas técnicas serão abordadas no próximo capítulo juntamente com as principais definições acerca deste assunto.

4 IMAGENS DIGITAIS

Imagem digital é uma representação matricial numérica discretizada de um objeto. Ou especificamente, uma função quantificada e amostrada, em duas dimensões, geradas por meios ópticos, disposta em uma grade padrão, retangular igualmente espaçada e quantificada em intervalos de amplitude (CASTLEMAN, 1996). Como mostra a Figura 8.

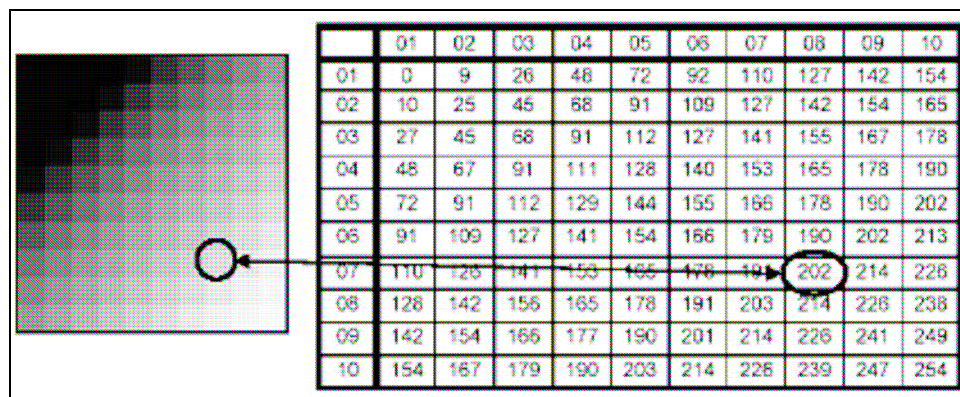


Figura 8. Representação numérica de uma imagem em tons de cinza de 10x10 *pixels*.
Fonte: CASTLEMAN, K (1996)

Segundo Gomes; Velho (1998), uma imagem pode ser definida como uma matriz cujos coeficientes, são uma função $f(x, y)$ onde o valor nas coordenadas espaciais x e y corresponde a cor associada a esta coordenada na imagem.

Para Gomes e Velho (1998) o processamento de imagens faz-se necessário a fim de obter recursos diversos, seja como produto final ou como parte essencial de um processo computacional envolvendo imagens digitais.

4.1 PROCESSAMENTO DIGITAL DE IMAGENS

Segundo Maria (2000), entende-se por processamento digital de imagens a manipulação de uma imagem por computador de modo que a entrada e a saída do

processo sejam imagens digitais. Como objetivo do uso do processamento digital de imagens, busca-se uma melhoria na representação de características das imagens para o usuário e o provimento de outros subsídios para a sua interpretação, inclusive gerando produtos que possam ser posteriormente submetidos a outros processamentos.

As técnicas de análise de imagem podem ser divididas em três áreas básicas:

- a) **processamento de baixo nível:** com funções que podem ser vistas como reações automáticas, ou seja, reações que não requerem comportamento inteligente;
- b) **processamento de nível intermediário:** com processos de extração e caracterização de componentes em uma imagem;
- c) **processamento de alto nível:** que envolve os processos de reconhecimento e interpretação.

Cordeiro (2002) considera a seguir os passos do processamento digital de imagens com foco no reconhecimento de imagens: aquisição; pré-processamento; segmentação; representação e descrição; reconhecimento; interpretação, conforme ilustrados na Figura 9.

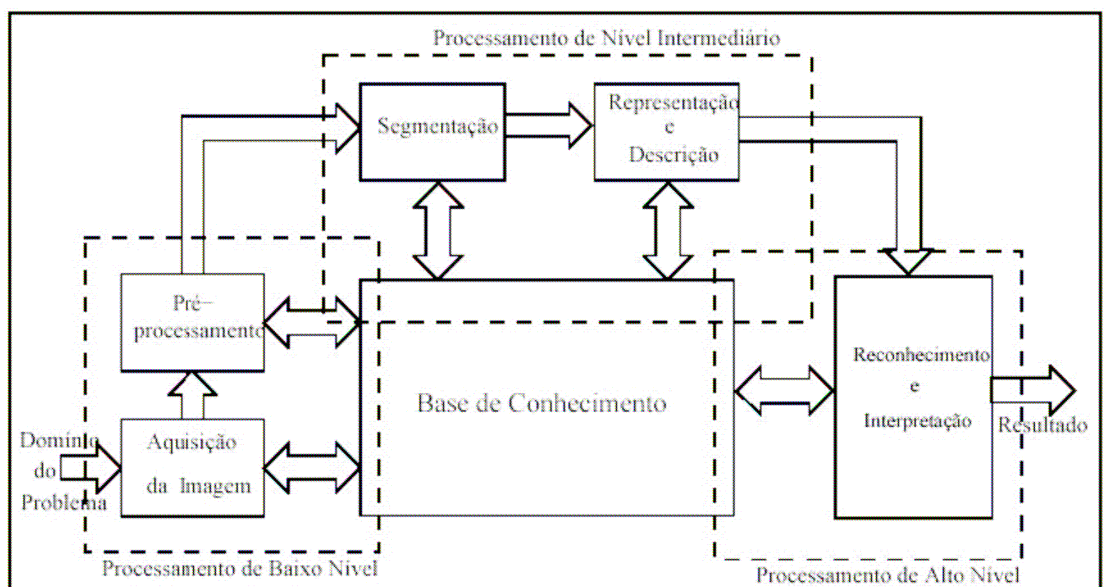


Figura 9. Elementos do processo de análise da imagem.
Fonte: Adaptado de GONZALEZ, R.; WOODS, R. (2000)

4.1.1 Aquisição

Para Cordeiro (2002), este é o primeiro passo e requer apenas um sistema de aquisição de imagens com a capacidade de receber o sinal elétrico produzido a partir deste sistema e convertê-lo em sua forma de imagem digital.

4.1.2 Pré-processamento

Após a aquisição e digitalização da imagem, o próximo passo é o pré-processamento. A função básica deste processo é a de melhorar a qualidade da imagem, com o objetivo de aumentar as chances de sucesso dos processos seguintes (GONZALEZ; WOODS, 1992).

Neste ponto, sobre as imagens, são utilizadas técnicas como o aumento de contraste, remoção de ruídos, realce, normalização, dentre outras, com o objetivo de conduzi-las a representações que possibilitem a simplificação do processamento principal.

4.1.3 Segmentação

Segundo Russell e Norvig (2004), a segmentação é o processo de subdividir uma imagem em grupos, baseando-se em características de semelhança entre um conjunto de *pixels*¹⁶. Ou seja, subdivide uma imagem digital de entrada em suas partes constituintes. Cada uma destas partes é uniforme e homogênea com respeito a alguma propriedade da imagem, como por exemplo, cor, textura e localização espacial.

¹⁶ Elemento básico de representação de uma imagem digital. Elemento de imagem.

Algoritmos de segmentação para imagens monocromáticas são geralmente baseados em duas propriedades: descontinuidade e similaridade.

Segundo Castleman (1996), o particionamento da imagem por descontinuidade é baseado no subconjunto dos pontos de um objeto que o separa do restante da imagem. As técnicas de segmentação nesta categoria buscam evidenciar os limites entre os objetos representados, por meio da detecção de pontos isolados e da detecção de linhas e bordas na imagem.

Na segunda categoria, similaridade, a segmentação é baseada nas técnicas de limiarização¹⁷, crescimento por regiões, união e divisão de regiões.

4.1.4 Representação e Descrição

O processo de segmentação, geralmente tem como saída dados brutos de *pixels*. Assim pode-se converter os dados para uma forma conveniente, possibilitando o processamento por computador. Para isso, dois tipos de representação podem ser utilizados: representação limite ou representação regional.

A representação via delimitação de áreas de objetos é apropriada quando o foco está em características de forma, como por exemplo, os cantos. Já a representação regional é apropriada quando o foco está em propriedades reflexivas, tais como textura ou cor (GONZALEZ; WOODS, 1992).

No entanto, em algumas aplicações, as duas representações podem existir, pois, a escolha e representação é apenas parte da solução para a transformação de imagens originais em uma forma conveniente para o processamento computacional

¹⁷ Converter imagens reduzindo a profundidade das cores, ou seja, reduzindo o número de *bits* utilizados na representação das tonalidades das imagens.

subseqüente. Assim, devem ser utilizados métodos para descrever as informações apropriadas, bem como, para que suas características de interesse sejam realçadas.

4.1.5 Reconhecimento

Após a separação das classes de objetos com características semelhantes passamos a uma fase na qual se tem interesse de identificar o que cada uma dessas classes representa.

O reconhecimento de padrões é utilizado para classificar formas de objetos tidos como padrões. A formação da linguagem, o modo de falar, o desenho das Figuras, o entendimento das imagens, envolvem padrões. Reconhecimento de padrões é uma tarefa complexa, onde busca-se avaliar as situações em termos de padrões nas circunstâncias que as constituem (BARRETO, 2001).

Segundo Cordeiro (2002), há muitas técnicas de reconhecimento de padrões desenvolvidas nas quais podem-se dividir em três abordagens:

- a) **estatística:** conjuntos de medidas de características são extraídos das imagens e métodos estatísticos são utilizados para separar as classes. Dentre os métodos utilizados pode-se citar classificadores bayesianos, métodos probabilísticos e regras de decisão;
- b) **estrutural:** padrões são representados em uma forma simbólica (*strings* e árvores). Os métodos de reconhecimento são baseados em casamento de informações e símbolos ou em modelos que tratam padrões de símbolos como sentenças, a partir de uma linguagem artificial;

c) **neural:** nesta abordagem, o reconhecimento é realizado utilizando-se RNA. Os autores Marques Filho e Viera Neto (1999), consideram o reconhecimento via redes neurais como um tipo particular de reconhecimento estatístico, já que as características também estão na forma de vetores e existe uma equivalência entre alguns modelos de redes neurais como, por exemplo, multicamadas e técnicas estatísticas fundamentais.

4.1.6 Interpretação

Conforme Barreto (2001) a interpretação envolve a fixação de significado a um grupo de objetos reconhecidos. Em outras palavras, o objetivo é dar significado às imagens.

A interpretação de imagens digitais é um processo extremamente complexo. As dificuldades aparecem tanto pelo volume de dados a serem processadas, quanto pela ausência de ferramentas adequadas de pré-processamento para os dados iniciais e posteriormente gerar os resultados esperados e satisfatórios. Não havendo estas ferramentas, somos forçados a considerar aproximações que proporcionem parcialmente possibilidade de êxito.

Essa insuficiência leva-nos a considerar duas restrições: limitar a generalidade do problema e a incorporação de conhecimento humano no processo (GONZALEZ; WOODS, 1992).

5 EXEMPLOS DE FERRAMENTAS E PESQUISAS DESENVOLVIDAS COM AS TÉCNICAS DE REDES NEURAIAS ARTIFICIAIS

Devido ao sucesso das redes neurais no reconhecimento de imagens e classificação de padrões, existe uma vasta gama de pesquisas envolvendo a técnica de redes neurais artificiais. A seguir são apresentadas algumas ferramentas relacionadas ao objetivo desta pesquisa.

5.1 NEURON COLOR

O Neuron Color é uma ferramenta capaz de realizar experimentos de processamento de imagens por meio de redes neurais artificiais. Esta pesquisa foi desenvolvida no ano de 1999 pelo acadêmico João Ricardo Bittencourt da Universidade do Vale do Rio dos Sinos (UNISINOS) em São Leopoldo-RS, em sua tese de mestrado. Implementado em linguagem de programação Java é atualmente distribuído livremente para a comunidade por meio da licença GNU GPL.

A versão atual desta ferramenta divide-se em três aplicações (BITTENCOURT, 2006):

- a) **Neuron Color Factory:** responsável pela criação de filtros neurais e pela geração de bases de exemplo;
- b) **Neuron Color View:** permite a visualização das imagens, aplicação dos filtros neurais e armazenamento das imagens transformadas;
- c) **Neuron Color Estatical:** compara duas imagens utilizando distâncias médias do matiz, saturação e luminosidade.

5.2 SISTEMA DE IDENTIFICAÇÃO DE IMPRESSÕES DIGITAIS UTILIZANDO REDES NEURAIIS ARTIFICIAIS

O protótipo desenvolvido nesta pesquisa implementou o reconhecimento de deltas e núcleo em impressões digitais. Foi desenvolvido em 2004 como trabalho final para graduação do acadêmico Caio Rafael Silva Matias do Centro de Ciências Exatas da Universidade Regional de Blumenau-SC.

Para o reconhecimento utilizou-se uma rede neural artificial *feedforward* multicamada com treinamento *backpropagation*.

Para as entradas da rede foram apresentados fragmentos de imagens, e como resposta, a classificação da entrada em três padrões, os quais indicam a qual região este fragmento de imagem pertence. Envolveu também a construção de um sistema de identificação das impressões digitais através da comparação de uma impressão digital com outras de usuários previamente cadastrados no sistema.

Os resultados obtidos neste trabalho não alcançaram o sucesso esperado, pois todos ficaram abaixo de 60%. O autor acredita que seria necessário a RNA ter um percentual de acerto acima dos 90% para ser validada como uma técnica relevante.

5.3 RECONHECIMENTO AUTOMÁTICO DE PLACAS DE VEÍCULOS AUTOMOTORES POR MEIO DE REDES NEURAIIS ARTIFICIAIS

Esta pesquisa foi desenvolvida com o intuito de desenvolver um sistema capaz de processar informações de forma rápida e precisa no que diz respeito ao reconhecimento de placas de veículos. Foi desenvolvida na graduação do acadêmico Bruno Clemente Guingo da Universidade Federal do Rio de Janeiro (UFRJ).

O sistema foi desenvolvido de forma modular, sendo que cada módulo ficou responsável por uma funcionalidade específica. A divisão utilizada foi a seguinte (GUINGO, 2006):

- a) módulo de obtenção da fotografia do veículo: nesta fase as imagens são capturadas por meio de câmeras espalhadas em vias públicas.
- b) módulo de localização da região da placa: recebe a foto inteira e tem o objetivo de fornecer a provável região da placa.
- c) extração da região da placa: separa entre as regiões candidatas a mais provável e guarda em um arquivo para a execução da próxima fase.
- d) segmentação de caracteres da placa: responsável por separar um a um os caracteres da placa e guarda-los em arquivos distintos.
- e) extração de características dos caracteres segmentados: captura as informações inerentes tal que o mesmo seja facilmente reconhecido pelo próximo módulo.
- f) reconhecimento: reconhece os caracteres que compõe a placa.

Os resultados obtidos nesta pesquisa se mostraram abaixo do desejado. O autor atribuiu tal desempenho a vários fatores, sendo que o mais importante deles foi a baixa qualidade das imagens disponíveis.

6 VISIONSCAN - RECONHECEDOR DE IMAGENS DE CÓDIGOS DE BARRAS POR MEIO DE REDES NEURAIAS

O protótipo reconhecedor de imagens de códigos de barras por meio de redes neurais, denominado de VisionScan, foi desenvolvido utilizando-se a tecnologia Java e tem como principal objetivo o reconhecimento de código de barras por meio de RN *perceptron* multicamadas e do algoritmo de aprendizagem *backpropagation*.

As imagens dos códigos de barras utilizadas pelo VisionScan podem ser provenientes de tecnologia móvel, como por exemplo, por meio de celulares, sendo que no desenvolvimento desta pesquisa utilizou-se um emulador de celulares fornecido pelo fabricante e imagens geradas com a API Java Barcode4j.

O VisionScan compreende três aplicações distintas divididas em camadas, sendo elas:

- a) **interface cliente:** desenvolvida utilizando-se *Java 2 Micro Edition* (J2ME) é a camada responsável pela aquisição e transferência das imagens para o servidor de aplicações, sendo também sua função receber o resultado final da consulta realizada ao VisionScan;
- b) **reconhecedor de imagens:** implementado por meio de *Java 2 Standard Edition* (J2SE) é onde acontece a maior parte do processamento, foco desta pesquisa. Nele a imagem digitalizada é recebida e processada, além de possuir o modelo de RN *perceptron* multicamadas responsável pelo reconhecimento das imagens digitais de códigos de barras;
- c) **aplicação servidor:** desenvolvida em *Java 2 Enterprise Edition* (J2EE) tem o intuito de fornecer recursos para a comunicação entre as camadas anteriores. Nesta aplicação foram implementados os serviços

responsáveis pelo recebimento da imagem digital, encaminhamento ao reconhecedor e busca da informação referente ao produto do código de barras reconhecido pela RN.

Partindo-se desta estrutura do VisionScan, a seguir encontram-se algumas explicações referentes ao domínio de aplicação do protótipo (código de barras no padrão EAN13¹⁸), bem como a metodologia onde estão demonstradas as etapas do desenvolvimento desta pesquisa.

6.1 CÓDIGO DE BARRAS PADRÃO EAN13

O código de barras é uma forma de representar a numeração, que viabiliza a captura automática dos dados por meio de leitura óptica nas operações automatizadas (GS1, 2006). O Sistema GS1 reconhece algumas simbologias de código de barras para representar as estruturas de numeração padronizada, sendo que nesta pesquisa utilizou-se a EAN/UPC.

A sigla EAN representa um modelo europeu para codificação de código de barras, este modelo propõe um método bastante sucinto de identificação de uma mercadoria, sendo formado por: país de origem, companhia e produto. O método de codificação escolhido denomina-se EAN13, pois a maioria dos produtos no Brasil apresenta código de barras pertencentes a este tipo de codificação.

A codificação dos tipos de barras deve obedecer algumas normas (GS1, 2006):

- a) só se codificam dígitos numéricos;
- b) cada dígito será codificado em uma sequência binária;

¹⁸ Código de barras linear numérico que representa os 13 dígitos do GTIN-13.

- c) um módulo equivale a 1 binário;
- d) um espaço equivale a 0 binário;
- e) todos os dígitos se compõe em 2 barras e 2 espaços em branco;
- f) a espessura da barra e o espaço será sempre proporcional de 1 a 4 vezes;
- g) o último caractere sempre será o dígito de controle;
- h) a espessura, tanto de um módulo, como de um espaço será de 0.33mm, o mesmo que 1 pixel. Permitindo uma espessura mínima de 0.23mm;

Mediante isso, pode-se dizer que de acordo com estas normas, cada dígito se codifica em um número de 7 *bits*, dentro das seqüências de zeros e uns.

Assim, 0111001 é válido uma vez que encontram-se duas subseqüências de cada. Já a seqüência 0100110 não é válida, posto que, possui três subseqüências de 0.

A Figura 10 explica graficamente a composição da numeração padrão EAN13.

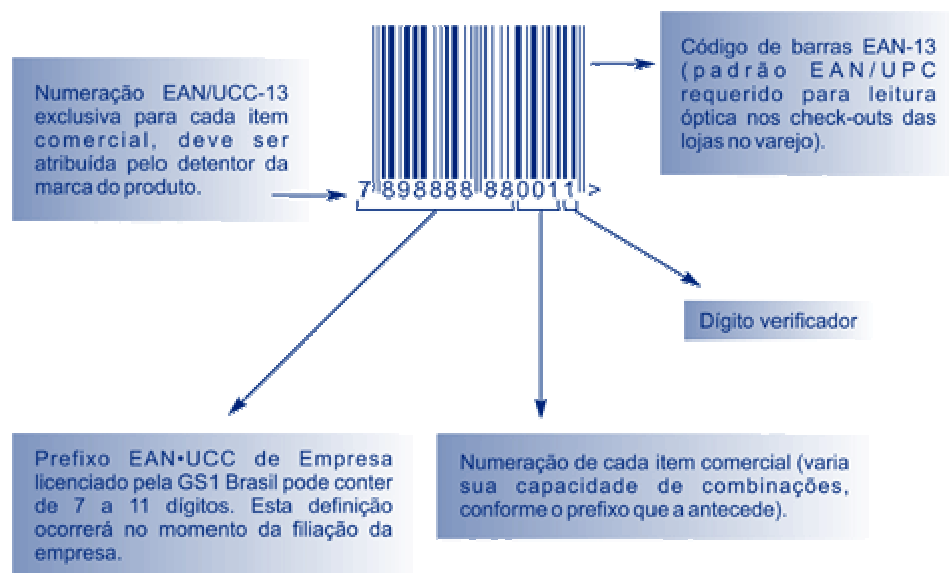


Figura 10. Composição do código de barras EAN-13
Fonte: GS1, Brasil (2006)

Como pode-se observar, outra parte importante do modelo EAN13 é o dígito de controle, que tem como função, verificar se o código é correto e pertence à especificação EAN. Este é o último número à direita, denominado de Dígito Verificador

(DV). Na Figura 11 encontram-se ilustrados os passos que devem ser executados para o cálculo do DV.

EAN-13	Posições dos dígitos												
	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12	N13
	Multiplicar o valor de cada posição por												
	1	3	1	3	1	3	1	3	1	3	1	3	
	Somar os produtos (resultados) das multiplicações												DV
	Subtrair o resultado da soma anterior do múltiplo de 10 mais próximo												

Figura 11. Passos executados no cálculo do DV
Fonte: Adaptado de GS1, Brasil (2006)

Portanto, deve-se adicionar todos os dígitos ímpares e ao resultado desta soma, acrescentar também a soma dos dígitos pares multiplicados por 3. O DV será o resultado da subtração entre o múltiplo de 10 mais próximo e o número encontrado nas somas anteriores conforme o exemplo a seguir:

Código: 0123456789005
 Dígitos em posição Ímpar -> 0 2 4 6 8 0 = 20
 Dígitos em posição Par -> (1 + 3 + 5 + 7 + 9 + 0) * 3 = 25 * 3 = 65
 Soma -> 20 + 65 = 85; Unidade = 5
 Dígito de Controle -> 10 - 5 = 5

6.2 METODOLOGIA

A metodologia utilizada no desenvolvimento desta pesquisa teve como etapa inicial a definição do escopo do problema a ser tratado, ou seja, prover o protótipo de uma ferramenta para o reconhecimento de imagens.

A seguir foi elaborada uma proposta de solução para o problema em questão, propondo-se o desenvolvimento de uma ferramenta capaz de reconhecer códigos de barras a partir de imagens digitais, como por exemplo, as obtidas em câmeras de aparelhos celulares.

Posteriormente a esta fase inicial da pesquisa executaram-se as seguintes etapas metodológicas: levantamento bibliográfico; definição dos códigos de barra a

serem utilizados; modelagem do VisionScan; definição da arquitetura do VisionScan; implementação da interface cliente, do reconhecedor de imagens e da aplicação servidor finalizando-se com os testes.

Após obter os conceitos necessários acerca do escopo da pesquisa, implementou-se a modelagem do protótipo por meio de Unified Modeling Language (UML), que foi uma etapa muito importante do projeto, onde um desenho do modelo utilizado na construção do sistema foi definido. Neste momento, também foi possível definir os tipos de código de barras e produtos a serem utilizados, no caso, a codificação EAN13.

A seguir, realizou-se a decodificação das imagens utilizando API *Java Advanced Imaging* (JAI) o que possibilitou uma maior facilidade no processo de reconhecimento.

A parte de maior ênfase neste trabalho foi o desenvolvimento do protótipo VisionScan e o treinamento da rede neural a partir de uma base de imagens de códigos de barras. Finalizada esta etapa, pode-se concluir o trabalho efetuando os testes de funcionamento e comunicação entre o cliente móvel e o reconhecedor de imagens.

6.2.1 Levantamento Bibliográfico

Esta etapa compreendeu o estudo e compreensão do tema realizado por meio de pesquisa no acervo bibliográfico disponibilizado na Unesc, bem como de artigos e teses na Internet. O estudo procurou referenciais teóricos envolvidos no desenvolvimento da pesquisa, ou seja, inteligência artificial, redes neurais artificiais, processamento de imagens, entre outros.

6.2.2 Modelagem do VisionScan

Fase onde elaborou-se o planejamento das atividades que o sistema executa, tendo-se utilizado a Linguagem Unificada de Modelagem ou *Unified Modeling Language* (UML) que é uma notação gráfica usada para visualização, especificação, construção e documentação de sistemas de softwares (PENDER, 2004).

A UML disponibiliza diversas formas de representações de diagramas que podem ser utilizados para modelar o sistema de acordo com as particularidades de cada implementação.

Na modelagem do protótipo VisionScan utilizou-se a ferramenta ARGOUML disponível gratuitamente para *download* no site <http://argouml.tigris.org/>. Dentre os diagramas da UML, optou-se nesta pesquisa pela utilização de três tipos: pacotes, casos de uso e atividades, conforme descrição apresentada nos itens a seguir.

6.2.2.1 Diagrama de Pacotes

De acordo com Booch et al (2000), procura-se usar diagramas de pacotes para organizar elementos do sistema em modelos, com o objetivo de facilitar a compreensão do sistema como um todo. Por meio deste, pode-se controlar os respectivos módulos que fazem parte de um grupo específico. Na modelagem do VisionScan usou-se este diagrama com o intuito de modularizar a visão do protótipo em áreas distintas, facilitando a compreensão e a implementação de futuras ferramentas.

Na Figura 12 estão ilustradas por meio do diagrama de pacotes, as divisões previamente estabelecidas para a implementação da ferramenta.

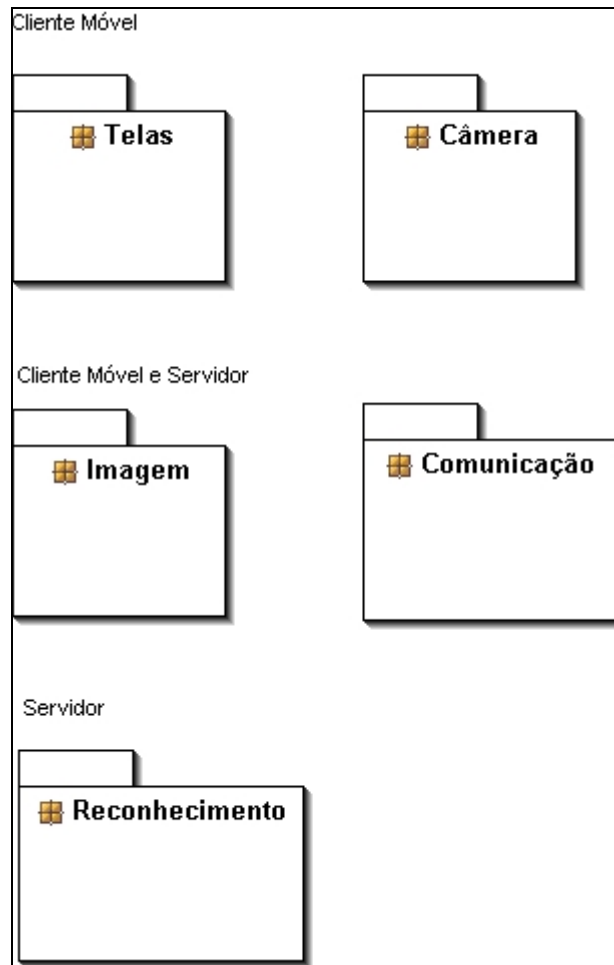


Figura 12. Diagrama de pacotes do VisionScan

As divisões ilustradas na Figura 12 foram definidas de acordo com a responsabilidade de cada componente nas aplicações. Estas responsabilidades serão descritas a seguir:

- a) **telas:** componentes responsáveis pela parte visual do sistema;
- b) **câmera:** componentes responsáveis pela captura da imagem no aparelho celular;
- c) **imagem:** componentes responsáveis pelo processamento da imagem deste o processo de aquisição até o reconhecimento;
- d) **comunicação:** componentes responsáveis pelo processo de envio e recebimento da imagem digitalizada;

- e) **reconhecimento:** componentes da rede neural responsáveis pelo processo de reconhecimento do código de barras.

6.2.2.2 Diagrama de Casos de Uso

Citando Medeiros (2004), por meio do diagrama de casos de uso pode-se demonstrar as funcionalidades do sistema desenvolvido, com as quais o usuário poderá interagir e o papel que estes desempenharão ao utilizar o sistema.

Na Figura 13 está ilustrado o diagrama de caso de uso descrevendo as funções do especialista no VisionScan.

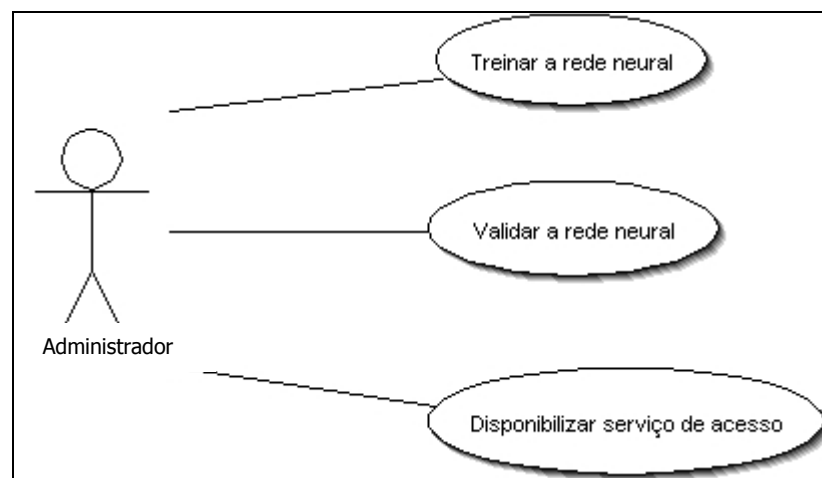


Figura 13. Diagrama de caso de uso das funções do especialista

O diagrama ilustrado na Figura 13 tem como objetivo demonstrar as ações do especialista no que diz respeito ao treinamento da rede e disponibilização do serviço aos usuários:

- treinar a rede neural:** processo pelo qual os exemplos são apresentados a rede;
- validar a rede neural:** processo relacionado aos testes referentes as taxas de acerto da rede para determinadas situações;

- c) **disponibilizar serviço de acesso:** processo onde o servidor web e o serviço responsável pela comunicação são habilitados.

A Figura 14 representa o diagrama de caso de uso referente as possíveis funcionalidades a serem executadas pelo usuário móvel do VisionScan. Dentre as principais funções estão:

- a) **capturar imagem:** o usuário, por meio da câmera digital fotografa um código de barras;
- b) **enviar solicitação:** envia requisição para o servidor web responsável pelo reconhecimento da imagem do código de barras;
- c) **receber resposta:** recebe a resposta processada no servidor.

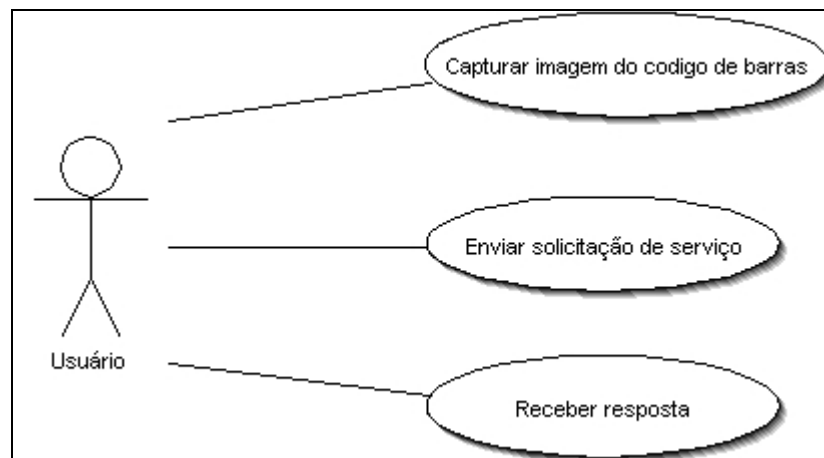


Figura 14. Diagrama de caso de uso das funcionalidades disponíveis ao usuário

6.2.2.3 Diagrama de Atividades

Segundo Medeiros (2004), este tipo de diagrama exhibe graficamente o fluxo do controle de uma atividade para a outra e procura descrever a seqüência das funções que o sistema executará, com suporte dinâmico do comportamento condicional e paralelo que essas possam apresentar.

O diagrama de atividades do protótipo VisionScan é apresentado na Figura

15.

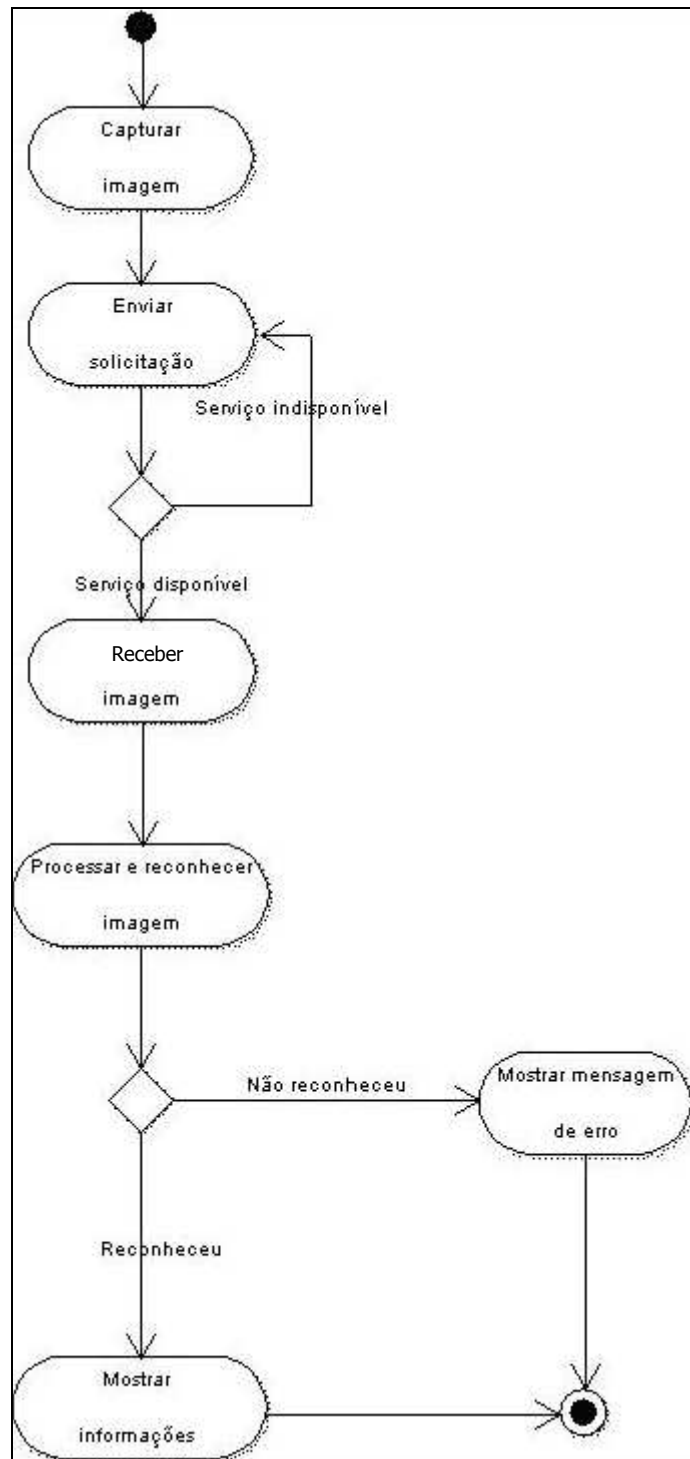


Figura 15. Diagrama de atividade do VisionScan

Conforme demonstrado na Figura 15, o início do processo acontece com a captura da imagem na câmera digital do aparelho celular. Logo após, esta imagem

digitalizada é enviada para um serviço web responsável pelo recebimento e repasse ao reconhecedor. O processo de reconhecimento envolve a manipulação da imagem e a identificação dos padrões por meio da RN devolvendo ao cliente móvel, no caso de sucesso, as informações sobre o produto.

6.2.3 Definição da Arquitetura do VisionScan

O VisionScan, protótipo que foi desenvolvido nesta pesquisa, trabalha interligado a seus clientes móveis por meio de uma aplicação desenvolvida sob a tecnologia J2EE, necessitando apenas que o servidor onde está instalado esteja de acordo com esta especificação.

O protótipo trabalha da seguinte forma: o administrador do sistema treina a rede utilizando uma base de imagens de códigos de barras previamente definidas e disponibiliza o serviço de acesso em seu servidor web. Por sua vez, os celulares enviam solicitações síncronas a este serviço e recebem como resposta os dados referentes ao código de barras decodificado. Entre o recebimento da solicitação pelo servidor e o envio da resposta ao cliente móvel, ocorre o processamento da imagem digital e o reconhecimento por meio da rede neural já treinada.

A Figura 16 mostra o modelo da arquitetura utilizada no desenvolvimento do protótipo VisionScan.

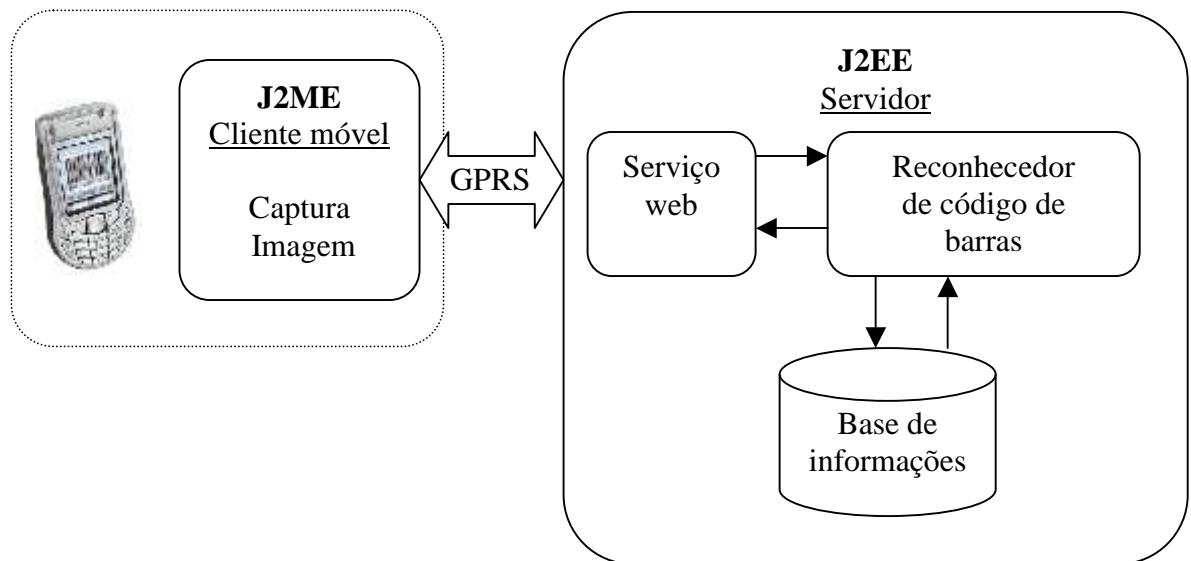


Figura 16. Modelo de arquitetura usada no VisionScan

A Figura 16 define o modelo da especificação J2EE para aplicações do tipo multicamada e cliente/servidor e especifica também, o posicionamento de cada módulo na estrutura, que serão explicados com maiores detalhes nas próximas seções.

6.2.4 Implementação da Interface Cliente

A interface cliente foi desenvolvida segundo os conceitos da tecnologia Java no que diz respeito a programação de aplicações para clientes móveis J2ME. Esta edição é uma das mais recentes lançadas para a tecnologia Java que fornece, além de uma máquina virtual, uma API completa para o desenvolvimento de aplicações para pequenos dispositivos, tais como: celulares, palms, entre outros, tendo como requisito mínimo para funcionamento deste protótipo, a especificação MIDP2.

A seguir serão ilustradas as telas desenvolvidas na aplicação cliente e a descrição de seu funcionamento.

A tela inicial (Figura 17) demonstra o emulador utilizado durante os testes, simulando a câmera fotográfica dos aparelhos celulares, além de implementar os comandos para início do reconhecimento e saída do sistema.



Figura 17. Tela inicial da aplicação cliente móvel

Na Figura 18 tem-se o momento em que o código de barras é recuperado no disco local e os *bytes* referentes a imagem digital são enviados para o VisionScan.



Figura 18. Tela de captura da imagem do código de barras

Finalmente, na Figura 19 são demonstrados os resultados do processamento do VisionScan, que refere-se ao nome do produto vinculado ao código reconhecido.



Figura 19. Tela de resultados obtidos no VisionScan

6.2.5 Implementação do Reconhecedor de Imagens

O processo envolvido no reconhecimento das imagens de códigos de barra tem início com o processo de tratamento da imagem recebida do dispositivo móvel. Esta etapa desenvolveu alguns dos passos referentes ao processamento digital de imagens citados no Capítulo 4, que foram:

- a) **aquisição:** ocorre na aplicação cliente, sendo responsável pela captura da imagem e sua recuperação por meio da câmera digital de um telefone celular. Nos testes executados utilizou-se o emulador de celulares, simulando-se portanto esta funcionalidade com a recuperação de uma imagem no diretório local;
- b) **pré-processamento:** nesta etapa utilizou-se recursos da computação gráfica com o intuito de melhorar a imagem recebida e obter melhores resultados no processo de reconhecimento. Os recursos utilizados foram: transformação da imagem em preto e branco e, quando necessário, aplicado filtro de contraste, além da vetorização;
- c) **segmentação:** foi uma das principais etapas desenvolvidas, facilitando consideravelmente o processo de reconhecimento. Nela as barras referentes ao código contido na imagem foram separadas e calculando-se as suas respectivas larguras. O resultado deste processamento foi utilizado como entrada para a RN implementada;

O reconhecedor, por sua vez, implementou uma RN do tipo *perceptron* multicamadas com o algoritmo de aprendizagem *backpropagation*. Como entrada para esta rede, foram utilizadas as larguras das barras obtidas pelo processo de segmentação,

sendo que, para validar o reconhecimento, também foi calculado o dígito verificador conforme apresentado na seção 6.1.

6.2.5.1 Biblioteca Java para Manipulação de Imagens

Conforme SUN (2006), *Java Advanced Imaging* (JAI) é uma biblioteca especializada no tratamento de imagens que estende a plataforma Java e permite o processamento de imagens com uma boa performance, possuindo um conjunto de classes que provê funcionalidades para este tipo de trabalho.

Essa API é extensível por permitir que novas operações sejam adicionadas de forma a se tornarem parte nativa da JAI. Ainda, conforme descrito por SUN (2006), JAI oferece vantagens como, independência de plataforma, orientação a objetos, flexibilidade, facilidade de extensão e eficácia.

Sendo assim, utilizou-se nesta pesquisa os recursos da API JAI no que se refere ao processamento digital de imagens para o pré-processamento e segmentação dos códigos recebidos do cliente móvel, processo executado antes do reconhecimento por meio da RN.

6.2.5.2 Modelagem Matemática do Algoritmo de Aprendizagem *Backpropagation*

Segundo Haykin (2001) o problema do OU-Exclusivo (XOR) é um exemplo clássico da utilização de uma rede MLP e do algoritmo de aprendizagem *Backpropagation*, por esse motivo, sua modelagem matemática foi realizada para que se compreendesse melhor este algoritmo.

Nesta etapa da metodologia foi necessário o estudo e entendimento do funcionamento do algoritmo *backpropagation* a fim de implementá-lo no VisionScan.

Tabela verdade do XOR:

0	XOR	0	=	0
0	XOR	1	=	1
1	XOR	0	=	1
1	XOR	1	=	0

O algoritmo *backpropagation* consiste na execução de várias etapas, porém, para demonstração se utilizará uma rede com cinco neurônios, sendo dois na camada de entrada, dois na camada oculta e 1 na de saída. Apenas a primeira iteração com os valores de entrada (0,0) será calculada e a taxa de aprendizagem será igual a 0.5. Essa estrutura apresentou uma boa performance na solução do problema proposto.

Primeiramente, a rede é inicializada com pesos aleatórios. Para o exemplo, escolheu-se qualquer valor entre -1 e 1.

Neurônio oculto 1:

$$\begin{aligned}w_2(0,1) &= 0.341232 \\w_2(1,1) &= 0.129952 \\w_2(2,1) &= -0.923123\end{aligned}$$

Neurônio oculto 2:

$$\begin{aligned}w_2(0,2) &= -0.115223 \\w_2(1,2) &= 0.570345 \\w_2(2,2) &= -0.328932\end{aligned}$$

Neurônio de saída:

$$\begin{aligned}w_3(0,1) &= -0.993423 \\w_3(1,1) &= 0.164732 \\w_3(2,1) &= 0.752621\end{aligned}$$

A próxima etapa consiste em calcular a soma da multiplicação dos valores de entrada pelos valores predefinidos nos neurônios da camada oculta.

$$\begin{aligned}x_1(0) &= 1 \text{ (bias)} \\x_1(1) &= 0 \\x_1(2) &= 0\end{aligned}$$

Neurônio 1:

$$(1 * 0.341232) + (0 * 0.129952) + (0 * -0.923123) = 0.341232$$

Neurônio 2:

$$(1 * -0.115223) + (0 * 0.570345) + (0 * -0.328932) = -0.115223$$

Com os valores obtidos no cálculo anterior, aplica-se a função de ativação *sigmóid* para limitá-los.

$$x_2(1) = 1/(1+e^{(-0.341232)}) = 0.584490$$

$$x_2(2) = 1/(1+e^{(0.115223)}) = 0.471226$$

Neste momento, tem-se as saídas calculadas para a camada oculta, devendo-se fazer o mesmo para a camada de saída utilizando-se $x_2(1)$ e $x_2(2)$ como entradas para esta camada.

$$x_2(0) = 1 \text{ (bias)}$$

$$x_2(1) = 0.584490$$

$$x_2(2) = 0.471226$$

Neurônio de Saída:

$$(1 * -0.993423) + (0.584490 * 0.164732) + (0.471226 * 0.752621) = -0.542484$$

$$\text{Conseqüentemente, } x_3(1) = 1/(1+e^{(0.542484)}) = 0.367610$$

As etapas descritas até agora demonstram apenas a metade do processo de treinamento. Dando seguimento, precisa-se ajustar todos os pesos a fim de se obter um resultado mais próximo do esperado, no caso 0, calculando os deltas para cada camada.

Inicialmente calcula-se para a camada de saída.

$$d_3(1) = x_3(1)(1 - x_3(1))(d - x_3(1))$$

$$= 0.367610 * (1 - 0.367610)(0 - 0.367610)$$

$$=-0.085459$$

Agora, com os valores calculados para a camada de saída, pode-se utilizá-los para a retropropagação do erro.

$$d_2(1) = x_2(1)(1 - x_2(1))w_3(1,1)d_3(1)$$

$$= 0.584490 * (1 - 0.584490)*(0.164732)*(-0.085459) = -0.0034190$$

$$d_2(2) = 0.471226 * (1 - 0.471226)*(0.752621)*(-0.085459) = -0.0160263$$

Ao executar a alteração dos pesos nas camadas, deve-se lembrar que o coeficiente de aprendizagem foi predefinido como 0.5 e que em alguns casos a mudança dos pesos será igual a zero devido a multiplicação pelas entradas, que neste exemplo é igual 0. Conseqüentemente serão demonstrados os cálculos apenas para os valores que mudam.

$$\delta w_2(0,1) = \eta * x_1(0) * d_2(1) = 0.5 * 1 * -0.0034190 = -0.017095$$

$$\delta w_2(1,1) = 0$$

$$\delta w_2(2,1) = 0$$

$$\delta w_2(0,2) = 0.5 * 1 * -0.0160263 = -0.0080132$$

$$\delta w_2(1,2) = 0$$

$$\delta w_2(2,2) = 0$$

$$\delta w_3(0,1) = 0.5 * 1 * -0.085459 = -0.042730$$

$$\delta w_3(1,1) = 0.5 * 0.584490 * -0.085459 = -0.024975$$

$$\delta w_3(2,1) = 0.5 * 0.471226 * -0.085459 = -0.020135$$

Assim, sucessivamente o processo é repetido até que o erro calculado esteja num valor aceitável ao contexto do problema ou o número de épocas de treinamento seja alcançado.

O exemplo de saída para a rede após o treinamento seria como mostrado abaixo. Utilizando arredondamento tem-se a tabela verdade para o problema XOR.

$$\begin{array}{l} 0 \text{ XOR } 0 = 0.017622 \\ 0 \text{ XOR } 1 = 0.981504 \\ 1 \text{ XOR } 0 = 0.981491 \\ 1 \text{ XOR } 1 = 0.022782 \end{array}$$

6.2.5.3 Treinamento da Rede Neural

A base de exemplos utilizada no treinamento da rede implementada fez uso de, aproximadamente, 300 imagens geradas com o auxílio da API Java Barcode4j, sendo que destas 250 foram utilizadas para o treinamento e o restante para a realização de testes.

A API Barcode4j implementa os recursos necessários para criação das imagens de código de barras em diversas codificações com base no seu referencial numérico (KRYVALIS, 2006). A mesma está disponível para uso não comercial sob licença *Apache Software Licence (ASL)*.

6.2.6 Implementação da Aplicação Servidor

A aplicação servidor foi desenvolvida seguindo os conceitos da tecnologia Java J2EE, que oferece um modelo de arquitetura distribuída e multicamada baseada no reuso de componentes e no uso de um modelo de segurança unificado e flexível no controle de transações.

Esta aplicação teve como objetivo implementar a comunicação entre o cliente móvel e o reconhecedor de imagens construindo uma camada de acesso por meio de um serviço web para estas funcionalidades.

A seguir serão apresentados os testes e resultados obtidos com modelo de RN *perceptron* multicamadas e o algoritmo de aprendizagem *backpropagation* no reconhecimento de imagens de códigos de barras.

6.3 TESTES E RESULTADOS OBTIDOS

Na execução dos testes do VisionScan, bem como no treinamento da rede, utilizou-se imagens geradas por meio da API Java Barcode4j, que foram selecionadas aleatoriamente, tendo-se o cuidado de utilizar no conjunto de testes códigos de barras que não participaram do processo de treinamento da RN.

A partir destes testes pode-se obter algumas análises no que diz respeito a escolha da arquitetura da rede e da forma de tratamento das entradas da mesma. Salientando-se como um dos pontos positivos da implementação do VisionScan a segmentação realizada nas entradas para cada número do código de barras a ser reconhecido.

Essa divisão possibilitou um melhor aproveitamento do conjunto de treinamento, que não precisou ser tão extenso, já que não ocorreu a aprendizagem de todo o código, mas sim de partes dele. Assim, reconheceu-se apenas a quantidade de barras responsável por um número específico. Em contrapartida, teve-se que especificar apenas uma codificação para o treinamento da rede neural (EAN13), pois conforme os variados padrões têm-se diferentes quantidades de barras envolvidas na montagem do código. Dessa forma, devido a esta variedade o processo de aprendizagem seria dificultado.

A estrutura foi definida utilizando-se uma rede organizada em 4 camadas, sendo elas: camada de entrada com 7 neurônios, duas camadas intermediárias também com 7 neurônios e a camada de saída com apenas um neurônio. Esta estrutura foi a que melhor adaptou-se as características do ambiente para uma taxa de erro igual a 0.015 levando aproximadamente 10 minutos para aprendizagem da rede.

Durante os testes do VisionScan obteve-se uma taxa de acerto no reconhecimento das imagens considerada satisfatória de acordo com a bibliografia estudada, alcançando uma média de 86%. Na Figura 20 pode-se observar a taxa de acerto obtida para as diferentes quantidades de entradas submetidas a rede neural.

Quantidade de entradas	Taxa de acerto
15	80%
30	83%
50	88%
150	94%
Média da taxa de acerto	86%

Figura 20. Taxa de acerto obtida para diferentes entradas na rede

Como pode-se observar, a rede obteve um melhor desempenho quando foi submetida a um número maior de entradas e entende-se que isso ocorreu porque no conjunto de testes também foram inseridas imagens que participaram do processo de treinamento.

CONCLUSÃO

Os métodos conexionistas constituem-se em uma das principais áreas de pesquisa para o reconhecimento de imagens por meio de redes neurais artificiais, sendo estes freqüentemente estudados em diferentes universidades do Brasil e do mundo.

No caso desta pesquisa o seu principal propósito consistiu no desenvolvimento de um reconhecedor de imagens com a capacidade de identificar um padrão específico de codificação de barras. O padrão EAN13 foi escolhido devido ao grande número de produtos encontrados no mercado brasileiro com esta codificação, sendo assim, o protótipo poderá ser utilizado nas mais diversas áreas, tais como: comércio, logística, entre outras.

Além deste também teve-se como objetivo desenvolver uma aplicação móvel para aparelhos celulares com tecnologia MIDP2 capaz de obter as imagens por meio da câmera digital e enviá-las para o processamento no servidor. Este foi um ponto de maior dificuldade devida necessidade de um servidor web registrado para atender as requisições vindas do celular.

Com a dificuldade encontrada na disponibilização deste servidor, os custos e a necessidade de solucionar este problema em tempo hábil, optou-se pela utilização de um emulador de celulares disponibilizado pelas próprias empresas de telefonia móvel, substituindo-se, dessa forma, a câmera digital por imagens previamente criadas por meio da API Java Barcode4j. Este emulador possibilitou a troca de informações via TCP/IP no computador local.

A ferramenta desenvolvida, denominada de VisionScan, foi submetida a vários testes a partir da base de imagens de códigos de barras, classificando-os corretamente na maioria dos casos a que foi submetido. De acordo com a bibliografia

estudada durante a pesquisa, os resultados obtidos podem ser considerados satisfatórios e atingiram os objetivos propostos.

Assim, no desenvolvimento desta pesquisa demonstrou-se que é possível por meio de um estudo detalhado obter-se bons resultados no reconhecimento de uma imagem de código de barras utilizando-se para isso de técnicas do processamento de imagem, do modelo *Perceptron* Multicamadas e do algoritmo de aprendizagem *Backpropagation*.

Dando continuidade a esta pesquisa sugere-se como trabalhos futuros:

- a) inclusão de novos módulos para o reconhecimento de outras codificações existentes no mercado nacional e internacional, com por exemplo: EAN8, Interleave25, GS1-128 entre outros;
- b) implementação de novos modelos de acesso ao reconhecedor de imagens;
- c) implementação de outros modelos de RN, utilizando-se por exemplo abordagens híbridas como redes neuro *fuzzy*. Essa abordagem pode melhorar os resultados obtidos com utilização da largura das barras nas entradas da rede.

REFERÊNCIAS

BARRETO, Jorge Muniz. **Inteligência Artificial no Limiar do Século XXXI**. 3.ed. Florianópolis: Duplic, 2001.

BITTENCOURT, Guilherme. **Inteligência Artificial: Ferramentas e Teorias**. Florianópolis: UFSC, 1998.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML: Guia do Usuário**. Tradução: Fábio Freitas da Silva. Rio de Janeiro: Campus, 2000.

BRAGA, Antônio de Pádua; CARVALHO, André Carlos Ponce de Leon Ferreira de; LUDERMIR, Teresa Bernarda. **Redes Neurais Artificiais: Teoria e Aplicações**. Rio de Janeiro: LTC, 2000.

CASTLEMAN, Kenneth R. **Digital Image Processing**. Upper Saddle River: Prentice Hall, Inc. 1996.

CORDEIRO, F. M. **Reconhecimento e Classificação de Padrões de Imagens com a Utilização de Redes Neurais Artificiais**. Universidade Federal de Santa Catarina, 2002.

COSTA, Ernesto; SIMÕES, Anabela. **Inteligência Artificial: Fundamentos e Aplicações**. Lisboa: FCA, 2004.

GS1, Brasil. **Sistema de Numeração EAN/UPC**. Disponível em:
<<http://www.eanbrasil.com.br/main.jsp>> Acesso em: 15 out. 2006.

FERNANDES, Anita Maria da Rocha. **Inteligência Artificial: Noções Gerais**. Florianópolis: Visual Books, 2003.

GUINGO, Bruno C. **Reconhecimento Automático de Placas de Veículos Automotores Através de Redes Neurais Artificiais**. Disponível em:
<http://www.inf.ucp.br/artigos/Artigo_2CBCComp.pdf> Acesso em: 02 nov. 2006.

GOMES, Jonas; VELHO, Luiz. **Computação Gráfica**. Rio de Janeiro: IMPA, 1998.

GONZALEZ, R. C. WOODS, R. E. **Digital Image Processing**. University of Tennessee Perceptics Corporation, 1992.

HAYKIN, Simon. **Redes Neurais: Princípios e Prática**. 2.ed. Porto Alegre: Bookman, 2001.

KRYSALIS. **Barcode4j**. Disponível em: < <http://barcode4j.krysalis.org/>> Acesso em: 17 out. 2006.

LOESCH, Claudio; SARI, Solange Teresinha. **Redes Neurais Artificiais: Fundamentos e Modelos**. Blumenau: FURB, 1996.

LUGER, George F. **Inteligência Artificial: Estruturas e Estratégias para a Resolução de Problemas Complexos**. 4. ed. Porto Alegre: Bookman, 2004.

MARIA, L. G. F. **Processamento Digital de Imagens**. INPE, Junho de 2000.

MARQUES FILHO, Ogê; VIEIRA NETO, Hugo. **Processamento Digital de Imagens**. Rio de Janeiro: Brasport, 1999.

MEDEIROS, Ernani. **Desenvolvendo Software com UML 2.0: Definitivo**. São Paulo: Makron Books, 2004.

OSÓRIO, F. S. **Sistemas Inteligentes Baseados em Redes Neurais Artificiais Aplicados ao Processamento de Imagens**. Santa Cruz do Sul. 1999.

REIS, Lisiane Albuquerque, CARGNIN, Moema Luz. **SDDEP – Uma aplicação na área médica utilizando raciocínio baseado em casos**. 1997. 95f. Monografia – Curso de Graduação em Ciências da Computação. Universidade Federal de Santa Catarina (UFSC). Florianópolis, 1997.

RUSSELL, Stuart J.; NORVIG, Peter. **Inteligência Artificial**. Rio de Janeiro: Elsevier, 2004.

SUN, Microsystems. **Java Advanced Imaging (JAI)**. Disponível em: <<http://java.sun.com/products/java-media/jai/>> Acesso em: 10 out. 2006.

TAFNER, Malcon A.; XEREZ, Marcos de; RODRIGUES FILHO, Ilson W. **Redes Neurais Artificiais**: Introdução e Princípios de Neurocomputação. Blumenau: EKO, 1996.

BIBLIOGRAFIA RECOMENDADA

DEITEL, H.M; DEITEL, P. J. **Java: Como Programar**. 4.ed. Porto Alegre: Bookman, 2003.

MICHAELIS. **Dicionário On-line da Língua Portuguesa**. Disponível em:
<<http://michaelis.uol.com.br/>> Acesso em: 8 nov. 2006.

MUCHOW, John W. **Core J2ME: Tecnologia & MIDP**. São Paulo: Pearson Educação, 2004.

PENDER, Tom. **UML: a Bíblia**. Rio de Janeiro: Elsevier, 2004.