



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

**DETECÇÃO DE BORDAS E NAVEGAÇÃO AUTÔNOMA UTILIZANDO REDES
NEURAS ARTIFICIAIS**

Ana Paula Abrantes De Castro

Dissertação de Mestrado em Computação Aplicada, orientada pelo Dr.
José Demisio Simões Da Silva

INPE
São José dos Campos
2003

(xx.xxx.xx(xxx.x))

Castro, A. P. A.

Detecção de Bordas e Navegação Autônoma Utilizando
Redes Neurais Artificiais/ A. P. A. Castro - São José dos
Campos: INPE, 2003.

154p. - (INPE-xxxx-TDI/xxx).

1. Redes Neurais. 2. Visão Computacional. 3. Detector de
Bordas. 4. Navegação Autônoma.

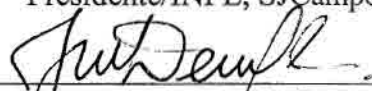
Aprovada pela Banca Examinadora em
cumprimento a requisito exigido para a
obtenção do Título de **Mestre em**
Computação Aplicada.

Dr^a Leila Maria Garcia Fonseca



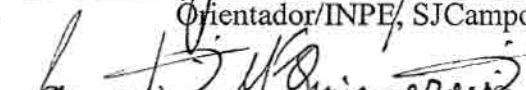
Presidente/INPE, SJCampos-SP

Dr. José Demisio Simões da Silva



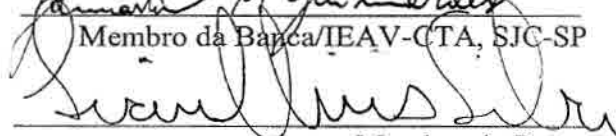
Orientador/INPE, SJCampos-SP

Dr. Lamartine Nogueira Frutuoso Guimarães



Membro da Banca/IEAV-CTA, SJC-SP

Dr. Ivan Nunes da Silva



Membro da Banca

Convidado UNESP, Bauru-SP

Candidata: Ana Paula Abrantes de Castro

São José dos Campos, 2 de junho de 2003.

Aos meus pais, oferto o amor e a promessa de tê-los sempre em meu coração; a gratidão, desde já, declara ser eterna, porque estou ciente de que o meu “tudo” não é suficiente para agradecer. A eles dedico este trabalho como singela retribuição pelo amor, carinho e dedicação que, com a ajuda de Deus, me incentivaram e apoiaram nesta difícil empreitada.

AGRADECIMENTOS

Primeiramente a Deus, que é a razão de tudo, que através de sua infinita bondade, nos concede incontáveis oportunidades para o crescimento.

Ao meu orientador e amigo Dr. José Demisio Simões da Silva, a quem eu tenho profunda admiração, agradeço pela confiança, pela valorosa orientação de forma segura e amiga, e principalmente pela pessoa especial que demonstrou ser. Também não poderia deixar de agradecer às horas de conversas que tivemos.

Aos membros da banca examinadora pela disposição em analisar este trabalho.

A todos os meus familiares, que souberam compreender a dedicação e, por vezes, o isolamento necessário à conclusão deste trabalho perdoando minha omissão como filha, irmã, neta e amiga, deixo aqui registrado, o meu agradecimento.

À minha querida mamãe Aparecida e ao meu querido papai Agostinho, que com muito amor e carinho sempre me incentivaram muito, me apoiando e acreditando em minha capacidade.

Ao meu querido namorado Elcio Hideiti Shiguemori, pelo amor e pela compreensão, ficando sempre ao meu lado em todos os momentos difíceis. Auxiliando-me no decorrer do mestrado, tirando as minhas dúvidas e por ter ficado longas horas ao meu lado quando eu mais precisava, fazendo-me rir quando a minha vontade era apenas de chorar.

Agradeço também à minha querida avó Cida pelas orações e pelo carinho.

Aos meus queridos irmãos Renato e Junior pela paciência e compreensão

A minha sogra e amiga Yurico Shiguemori por ter me acompanhado nessa jornada tão difícil, me apoiando e incentivado.

Ao Sr. Noboru Gondo que com muito carinho sempre me incentivou e torceu por mim.

A tia Elza e a tia Dica que me apoiaram e torceram por mim.

As minhas Cunhadas Erica e a Adriana e ao meu amigo Silvio, por estarem sempre torcendo pelo meu sucesso.

A minha melhor amiga Isabela Neves Drummond, pelas conversas, pelos desabafos e pela amizade.

Aos meus amigos do INPE, Leonardo, Fabrício, Karine, Isabela 2 e Camel, que me acompanharam e apoiaram este trabalho.

Ao professor Demisio e ao Elcio pelas valorosas correções no texto.

Ao Felipe, pelo empréstimo do veículo para realização do projeto.

Ao Dr. Francisco de Assis Tavares Ferreira da Silva pela ajuda e pelo incentivo, a minha entrada na área de pesquisa.

Aos professores do INPE pelo conhecimento compartilhado, em especial ao Dr. Haroldo Fraga de Campos Velho, ao Dr. Lamartine Nogueira Frutuoso Guimarães e Dra. Leila Maria Garcia Fonseca.

Às secretárias Cristina e Tâmara pelo auxílio ao longo do curso.

Enfim, agradeço a todas aqueles que, de alguma maneira, colaboraram e me ajudaram a vencer mais esta etapa da vida.

O meu...

MUITO OBRIGADO!!!!

RESUMO

Esta dissertação estuda o uso de Redes Neurais Artificiais na detecção de bordas em imagens e no controle de um veículo em navegação autônoma. São apresentados vários modelos de redes neurais que são estudados e testados, com o objetivo de buscar o modelo mais adequado para a tarefa de detecção de bordas, segundo critérios de desempenho que comparam as redes neurais artificiais com algoritmos tradicionais na área de visão computacional, como por exemplo, o operador de Canny. Para o controle da navegação autônoma utilizam-se modelos de redes neurais com aprendizagem supervisionada, treinadas para simular os processos envolvidos na navegação realizada por um humano. O desempenho das redes neurais na navegação é comparado com um sistema baseado em lógica nebulosa usado como base para o treinamento. São usadas diferentes imagens teste na experimentação das redes neurais no processo de detecção de bordas. No trabalho é proposta uma metodologia para medir a qualidade das imagens de borda geradas pelos operadores de redes neurais. Os resultados encontrados mostram-se promissores, com as redes neurais apresentando desempenho similar ao método de Canny.

EDGE DETECTION AND AUTONOMOUS NAVIGATION USING ARTIFICIAL NEURAL NETWORK

ABSTRACT

This work is about the study of Artificial Neural Networks (ANN) Systems for edge detection and robot autonomous navigation. Different ANNs are studied and tested in a search for the most adequate model for edge detection according to a performance criterion that compares the ANN based detectors to the standard algorithms available in the literature, such as Canny operator. Supervised neural network models are used to simulate the human navigation control processes in autonomous navigation. Their performances are compared to a fuzzy logic control system developed earlier, whose parameters are used in the neural network training processes. Different images are used to test the neural network edge detectors. A methodology for measuring the quality of the edge images produced by the neural network operator is proposed. The results show that the neural network operators have a performance comparable to existing Canny standard operator, and are thus promising operators for edge detection.

SUMÁRIO

Pág.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS E ABREVIATURAS

CAPÍTULO 1 - INTRODUÇÃO	19
CAPÍTULO 2 - REDES NEURAIS ARTIFICIAIS.....	23
2.1 FUNDAMENTOS.....	23
2.2 DEFINIÇÕES E CONCEITOS	26
2.2.1 NEURÔNIO ARTIFICIAL	27
2.2.2 FUNÇÃO DE ATIVAÇÃO.....	28
2.3 ARQUITETURAS DAS REDES NEURAIS.....	29
2.4 PROCESSOS DE APRENDIZAGEM.....	30
2.4.1 Aprendizado Supervisionado	30
2.4.2 Aprendizado Não-Supervisionado	31
2.4.3 Aprendizado Competitivo.....	31
2.4.4 Aprendizado Hebbiano	32
2.5 REDES PERCEPTRON.....	33
2.6 REDE PERCEPTRON DE MÚLTIPLAS CAMADAS	34
2.7 RBF	36
2.7.1 Centros Fixos Seleccionados Aleatoriamente.	38
2.7.2 Seleção Auto-organizada de Centros.....	39
2.8 MEMÓRIA BIDIRECIONAL ASSOCIATIVA	40
2.9 TEORIA DA RESSONÂNCIA ADAPTATIVA	42
2.9.1 Rede ART1	42
2.9.2 Rede ART2	46
2.10 APRENDIZAGEM POR QUANTIZAÇÃO VETORIAL.....	51
2.10.1 LVQ1	52
2.10.2 LVQ2.....	53
2.11 MEMÓRIA ASSOCIATIVA NEBULOSA.....	54
2.12 MAXNET	56
CAPÍTULO 3 - DETECÇÃO DE BORDAS.....	59
3.1 FUNDAMENTOS.....	59
3.2 IMAGEM DIGITAL	60
3.3 SISTEMAS DE PROCESSAMENTO DE IMAGENS	61
3.4 SEGMENTAÇÃO DE IMAGENS	63
3.4.1 Segmentação por Região	63
3.4.2 Segmentação por Contorno.....	65
3.4.3 Detecção de Bordas.....	65
3.4.3.1 Operador Laplaciano	66
3.4.3.2 Operador de Roberts.....	68

3.4.3.3 Operador Prewit	69
3.4.3.4 Operador de Sobel	70
3.4.3.5 Operador Canny	71
3.5 MÉTODOS DE VISÃO COMPUTACIONAL	73
CAPÍTULO 4 - DETECÇÃO DE BORDAS UTILIZANDO REDES NEURAIAS ARTIFICIAIS	75
4.1 REDES NEURAIAS EM DETECÇÃO DE BORDAS	75
4.2 ASPECTOS DE IMPLEMENTAÇÃO DOS EXPERIMENTOS COM REDES NEURAIAS	77
4.2.1 Padrões de Treinamentos	78
4.2.2 Treinamento das Redes Neurais	79
4.2.3 Imagens Utilizadas nos Experimentos	81
4.2.4 Extração da Janela	83
4.2.5 Ativação das Redes Neurais (Detecção Bordas)	84
CAPÍTULO 5 - ANÁLISE DOS RESULTADOS PARA OPERADOR NEURAL 85	85
5.1 ANÁLISE QUALITATIVA	85
5.1.1 Resultado para Imagem da Lena em Níveis de Cinza	86
5.1.2 Resultado para Imagem Bloco em Níveis de Cinza	91
5.1.3 Resultado para Imagem Pista em Níveis de Cinza	94
5.1.4 Resultado para Imagem do SPOT em Níveis de Cinza	98
5.1.5 Resultado para Imagem Lena RGB	101
5.1.6 Resultado para Imagem Peppers RGB	104
5.1.7 Resultado para Imagem Bloco RGB	106
5.1.8 Resultado para Imagem Casa RGB	108
5.1.9 Resultado para Imagem Pista Real RGB	110
5.1.10 Resultado para Imagem Satélite RGB	112
5.1.11 Operador de Canny	113
5.1.12 Operador Nebuloso	117
5.2 ANÁLISE QUANTITATIVA	119
CAPÍTULO 6 - NAVEGAÇÃO AUTÔNOMA UTILIZANDO REDES NEURAIAS ARTIFICIAIS	129
6.1 REDES NEURAIAS EM CONTROLE DA NAVEGAÇÃO AUTÔNOMA	129
6.2 IMPLEMENTAÇÃO DO CONTROLADOR NEURAL	131
6.2.1 Extração das Características da Pista	133
6.2.2 Controle da Navegação por Redes Neurais Artificiais	136
6.3 ANÁLISE DOS RESULTADOS DO CONTROLADOR NEURAL	136
CAPÍTULO 7 - CONCLUSÃO	141
REFERÊNCIAS BIBLIOGRÁFICAS	145

LISTA DE FIGURAS

2.1 – Esquema da Célula Neural.....	26
2.2 – Representação de um Neurônio Artificial.	27
2.3 – Funções de Ativação: (a) Sinal, (b) Rampa, (c) Logística e (d) Tangente Hiperbólica.....	28
2.4 – Rede com uma Única Camada.	29
2.5 – Rede de Três Camadas	30
2.6 – Redes de Camadas Múltiplas	35
2.7 – Arquitetura da RBF com uma Camada Intermediária (escondida). Os neurônios da camada intermediária são funções de base radial.	37
2.8 – Arquitetura de uma Memória Associativa Bidirecional	41
2.9 – Arquitetura Básica da Rede ART, com uma camada de entrada e uma de saída, duas unidades de controle e um mecanismo de reset.	43
2.10 – Arquitetura Básica da Rede ART2.....	47
2.11 – Aprendizagem por Quantização Vetorial	52
2.12 – Arquitetura da Memória Associativa Nebulosa (FAM)	55
2.13 – Arquitetura da Rede Maxnet.....	56
3.1 – (a) Sistema de coordenadas de referência em uma imagem; (b) Representação matricial de um trecho de uma imagem digital.....	60
3.2 – Passos Fundamentais em Processamento de Imagens Digitais.....	62
3.3 – (b) Primeira e (c) Segunda derivada para detecção de borda	67
3.4 – Exemplo de máscara que implementa o operador Laplaciano.....	67
3.5 – Operador Laplaciano: (a) Lena NC, (b) Lena RGB.	68
3.6 – Exemplos de Máscara que Implementa o Operador Roberts.....	68
3.7 – Operador Roberts: (a) Lena NC e (b) Lena RGB.....	69
3.8 – Exemplos de Máscara que Implementa o Operador Prewitt.....	69
3.9 – Operador Prewit: (a) Lena NC e (b) Lena RGB.	70
3.10 – Exemplos de Máscara que Implementa o Operador Sobel.....	70
3.11 – Operador Sobel: (a) Lena NC e (b) Lena RGB.....	71
4.1 – Diagrama Hierárquico do Sistema	77
4.2 – Exemplos dos 14 Padrões de Bordas Usados no Treinamento das Redes Neurais.	78
4.3 – Padrões de Bordas	79
4.4 – Imagens RGB Utilizadas na Dissertação: (a) Lena RGB, (b) Peppers e.....	81
(c) Blocos RGB, (d) Casa RGB.	81
4.5 – Imagem de Pista (a) Real, (b) e (c) Experimento.....	81
4.6 – Imagem em Níveis de Cinza (a) Blocos, (b) Lena.	82
4.7 – (a) Imagem SPOTPAN e (b) Imagem Plantações.....	82
4.8 – Exemplo de Extração da Janela.....	83
4.9 – Detecção de Bordas por RN.....	84
5.1 – Resultado das Redes Neurais treinadas com 14 padrões binários e com limiar da variância igual a 100: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.	86

5.1 – Conclusão.....	87
5.2 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 100: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.	87
5.2 – Conclusão.....	88
5.3 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 50 para a imagem Lena NC: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.	88
5.3 – Conclusão.....	89
5.4 – Resultado das Redes Neurais treinadas com 26 padrões bipolares e com limiar da variância igual a 50 para a imagem Lena NC: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.	90
5.5 – Resultado da variação do limiar de Variância: (a) LV = 1, (b) LV = 2, (c) LV = 3, (d) LV = 5, (e) LV = 15, (f) LV = 25.	91
5.6 – Resultado das Redes Neurais treinadas com 14 padrões binários e com limiar da variância igual a 100 para a imagem bloco em níveis de cinza: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.	92
5.7 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 100: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.	92
5.7 – Conclusão.....	93
5.8 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 50 para a imagem bloco NC: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.	93
5.8 – Conclusão.....	94
5.9 – Resultado das Redes Neurais treinadas com 14 padrões binários e com limiar da variância igual a 100: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.	95
5.10 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 100 para a imagem pista em níveis de cinza: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.	96
5.11 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 50 para a imagem pista NC: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.	97
5.12 – Resultado das Redes Neurais treinadas com 14 padrões binários e com limiar da variância igual a 100: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.	98
5.12 – conclusão.....	99
5.13 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 100 para a imagem Spot em níveis de cinza: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.	99
5.13 – Conclusão.....	100
5.14 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 50 para a imagem spot NC: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.	100

5.14 – Conclusão.....	101
5.15 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 100 para a imagem Lena RGB: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.....	102
5.16 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 50 para imagem Lena RGB: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.	103
5.17 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 100 para a imagem peppers RGB: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.....	104
5.17 – Conclusão.....	105
5.18 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 50 para imagem peppers RGB: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.....	105
5.18 – Conclusão.	106
5.19 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 100 para a imagem blocos RGB: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.....	106
5.19 – Conclusão.....	107
5.20 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 50 para imagem bloco RGB: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.	107
5.20 – Conclusão.....	108
5.21 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 100 para a imagem casa RGB: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.	108
5.21 – Conclusão.....	109
5.22 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 100 para imagem casa RGB: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.	109
5.23 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 100 para a imagem pista real RGB: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.	110
5.24 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 50 para imagem da pista RGB: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.....	111
5.25 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 100 para a imagem satélite RGB: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.....	112
5.26 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 50 para imagem satélite RGB: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.	113
5.27 – Resultado do Operador Canny com $\sigma = 0.5$: (a) Lena, (b) Blocos, (c) Pista, (d) Spot_pan.	114
5.28 – Resultado do Operador Canny com $\sigma = 0.5$: (a) Lena, (b) Peppers, (c) Blocos, (d) Casa, (e) Pista, (f) Plantação.....	115

5.29 – Resultado do Operador Canny com $\sigma = 1.0$: (a) Lena, (b) Peppers, (c) Blocos, (d) Casa, (e) Pista, (f) Plantação.....	116
5.30 – Resultado do Operador Fuzzy: (a) Lena, (b) Blocos, (c) Pista, (d) Spot_pan. ..	117
5.31 – Resultado do Operador Nebuloso: (a) Lena, (b) Peppers, (c) Blocos, (d) Casa, (e) Pista, (f) Plantação.....	118
5.32 – Procedimento para cálculo de limiar a partir de histograma.....	120
5.33 – Resultados da aplicação da limiarização sobre as imagens geradas pelo operador de Canny com $\sigma = 0.5$, sobre imagens em níveis de cinza: (a)Lena, Limiar=12; (b) Blocos, Limiar=10; (c) Pista, Limiar=8; (d)Spot_pan, Limiar=33.	121
5.34 – Imagem da Lena em níveis de cinza Limiarizada: (a) Método de Rosin e (b) Método de Histere.	122
5.35 - Resultados das redes com os tipos de bordas encontrados: (a) ART 1 e seu histograma, (b) ART 2 e seu histograma, (c) LVQ 1 e seu histograma, (d) LVQ 2 e seu histograma, (e) FAM e seu histograma, (f) MAXNET e seu histograma.	124
5.35 – Continua.....	125
5.35 - Conclusão.	126
6.1 – Sistema de Navegação Autônoma.....	133
6.2 – Ângulos Formados pelas Faixas Laterais com a Linha Perpendicular à Linha de Visão do Motorista.	134
6.3 – Exemplo de Imagem da Estrada na Visão de um Motorista.....	135
6.4 – Sistema Neural	136
6.5 – Veículo usado nos experimentos.....	137
6.6 – (a) Imagem da pista pela câmera; (b) Imagem com informação de bordas das faixas.....	137
6.7 – Imagem de borda com alguns pontos confusos.....	138
6.8 – Imagem da Pista com alguns pontos confusos.....	139

LISTA DE TABELAS

3.1 – Métodos de detecção de bordas.....	74
4.1 – Detecção de Bordas por RNA.....	76
5.1 - Diferenças Entre as Imagens de Bordas de Canny e os Operadores Neurais com Limiar 100.....	123
5.2 - Diferenças Entre as Imagens de Bordas de Canny e os Operadores Neurais com Limiar 50.....	126
5.3 – Tempo de Processamento de cada Algoritmo sobre as Diferentes Imagens, Utilizando um Limiar de Variância de 100 para as Redes Neurais (Em segundos).	127
6.1 – Navegação Autônoma por RNA	130
6.2 – Exemplos de ângulos extraídos da imagem.....	138
6.3 – Erro quadrático médio em 29 segundos de navegação.....	139

LISTA DE SIGLAS E ABREVIATURAS

Adaline	- Adaptive Linear Element
ANN	- Artificial Neural Networks
ART	- Adaptative Resonance Theory
BAM	- Bidirectional Associative Memory
D	- Direção
DD	- Direção Direita
DE	- Direção Esquerda
FAM	- Fuzzy Associative Memory
FPCNN	- Feedback Pulse-Coupled Neural Network
FPGAs	- Field Programmable Gate Arrays
FSOFM	- FeedBack Self-Organization Feature Maps
LVQ	- Learning vector Quantization
MAX	- Máximo
Maxnet	- Rede de Máximos
MBNN	- Model-Based Neural Network
MIN	- Mínimo
MLP	- Multi-Layer Perceptron
NC	- Níveis de Cinza
PCNN	- Pulse-Coupled Neural Network
PDP	- Parallel Distributed Processing
RBF	- Radial Based Function
RF	- Rádio Frequência

- RNA - Redes Neurais Artificiais
- RPMC - Rede Perceptron de Múltiplas Camadas
- SOFM - Self-Organization Feature Maps
- V - Velocidade
- VHF - Very High Frequency

CAPÍTULO 1

INTRODUÇÃO

Nesta dissertação são reportados resultados de estudos, pesquisas e experimentações do uso de Redes Neurais Artificiais (RNAs) em detecção de bordas e na navegação robótica autônoma, como continuidade dos trabalhos realizados em Castro et al. (2001), que desenvolveu um modelo computacional adaptável para um sistema de navegação autônoma baseado em imagens, utilizando decisões de controle baseado em lógica nebulosa. No sistema a direção e a velocidade a ser empregada pelo veículo são obtidas, do sistema de decisão nebuloso, com base nas informações extraídas das imagens do ambiente de navegação, utilizando um operador gradiente para localizar as regiões de auto-contraste da imagem, entre a pista e as faixas laterais, permitindo o cálculo das direções das faixas laterais para a tomada de decisão de navegação. O operador gradiente mostrou-se sensível a ruídos (nas imagens) oriundos do processo de transmissão por rádio frequência, desde a câmera a bordo do veículo.

Com objetivo de buscar operadores mais tolerantes a falhas, nesta dissertação propõem-se o estudo, a implementação e a análise de modelos de RNAs como operadores para extração das informações do ambiente e no controle do veículo em navegação.

Alguns modelos de redes neurais artificiais (RNA) já são utilizados na segmentação de imagens [(Lin et al. 1992), (Jorgensen et al. 1996), (Young et al. 1997), (Chai et al. 2001)] e na navegação autônoma de robôs (ou veículos em geral) [(Pormeleau, 1991), (Pormeleau, 1994), (Jochem, et al. 1995), (Pormeleau, 1995), (Baluja, 1996), (Davis, 1996), (Cao et al. 1999), (Huntsberger e Rose, 2000)]. Embora existam diversos modelos de redes neurais, que utilizam diferentes formas de aprendizagem, em geral, os trabalhos concentram-se naquelas mais relatadas na literatura. Nesta dissertação são reportados resultados de estudos e testes do uso de RNA diferentes na detecção de bordas em imagens e no controle de um veículo em navegação autônoma.

Os estudos e experimentos separados em dois problemas distintos, o uso de RNA em detecção de bordas e na navegação autônoma. Na detecção de bordas procura-se investigar a eficiência das redes neurais comparando-se os resultados obtidos com aqueles algoritmos de detecção considerados ótimos na literatura de visão computacional. Na navegação autônoma dois modelos de redes são utilizados para o controle da navegação sendo os seus resultados comparados com o sistema de navegação nebulosa desenvolvido em Castro et al. (2001).

A detecção de bordas é uma das tarefas de segmentação mais importantes como pré-processamento para o reconhecimento e caracterização de objetos presentes em uma imagem. A literatura de visão computacional reporta diversas abordagens algorítmicas para a detecção de bordas (González e Woods, 2000). O amadurecimento das técnicas de inteligência artificial tem permitido novas abordagens para esse problema. A complexidade do problema de reconhecimento e de classificação de imagens, tem tornado o uso dessas técnicas cada vez mais freqüente. Em particular, os modelos de RNAs têm sido mais explorados (Kulkarni, 2001).

Na tarefa de detecção de bordas são avaliados nove modelos de RNAs, com aprendizagem supervisionada e não supervisionada, com o objetivo de buscar o modelo mais adequado, comparação ao operador de Canny (Canny, 1986), tomado nesta dissertação como referência de melhor detector de bordas.

Os padrões utilizados nos treinamentos das redes neurais são padrões considerados como possíveis elementos de bordas, partindo-se do princípio que uma linha de borda é formada por pequenos elementos de bordas. As redes treinadas podem ser utilizadas para recuperar as informações de bordas em imagens. No total foram utilizados 26 padrões de treinamento, definidos em uma janela de tamanho 3x3, codificados em função do tipo de rede neural, como padrões binários (0 e 1) ou bipolares (-1 e 1). A escolha desses padrões e a operação básica dos neurônios, que compõem alguns modelos de redes utilizadas, mostram que as redes neurais comportam-se como filtros de diferenciação, semelhantes aos operadores de bordas encontrados na literatura, como por exemplo, Sobel, Roberts e Canny.

Nos experimentos, foram usadas diferentes imagens teste. Tomando-se o operador de Canny como referência, por ser o mais citado na literatura como sendo um operador ótimo para detecção de bordas, adota-se algumas medidas de similaridade para medir a qualidade de imagem de borda produzida pelas redes neurais, em relação aquelas produzidas pelo operador de Canny.

No controle da navegação autônoma utilizaram-se dois modelos de redes neurais com aprendizagem supervisionada, treinadas para simular os processos envolvidos na navegação, quando realizada por humanos. O desempenho das redes neurais é comparado com o sistema de navegação, baseado em lógica nebulosa de Castro et al. (2001), cuja parametrização é usada como base para o treinamento das redes neurais.

Os algoritmos de redes neurais mencionados são testados dotando um robô (veículo controlado remotamente) que move-se autonomamente por um ambiente do qual as imagens fornecem as informações necessárias para a tomada de decisão para a navegação, como em Castro et al. (2001). O sistema automaticamente se adapta ao ambiente e corrige a sua trajetória com base nas informações fornecidas pelas imagens.

Os resultados encontrados mostram-se promissores, com as redes neurais apresentando desempenho aproximado aos métodos clássicos em visão computacional, na tarefa de detecção de bordas. Ressalte-se que cada modelo de rede neural exige adequação dos dados de treinamento devido às suas características de aprendizagem. A principal vantagem em se utilizar redes neurais para as tarefas investigadas nesta dissertação é a possibilidade de implementação fácil em hardware programável, permitindo assim a exploração do paralelismo intrínseco das redes neurais artificiais em aplicações em tempo real.

No capítulo 2 é feita uma revisão breve das redes neurais artificiais apresentando as arquiteturas e os algoritmos de treinamento das redes neurais estudadas. No capítulo 3, são descritos alguns operadores de visão computacional para extração de bordas e são apresentados alguns trabalhos que utiliza operador Canny. O Capítulo 4 apresenta a metodologia e alguns trabalhos encontrados na literatura que utilizam redes neurais artificiais para detecção de bordas. No capítulo 5 são apresentados alguns resultados

obtidos para detecção de bordas mostrando uma comparação em relação aos operadores. O Capítulo 6 apresenta a metodologia, alguns trabalhos encontrados na literatura e os resultados obtidos pelas redes neurais artificiais para controle de um veículo na navegação autônoma. Finalmente, no capítulo 8 são apresentados algumas conclusões e possibilidades de futuros trabalhos na mesma linha de pesquisa.

CAPÍTULO 2

REDES NEURAIS ARTIFICIAIS

Neste capítulo apresenta-se uma breve introdução as redes neurais artificiais, no qual, introduz-se os modelos e algoritmos das redes neurais utilizadas nesta dissertação.

2.1 FUNDAMENTOS

As Redes Neurais Artificiais são inspiradas no funcionamento dos neurônios biológicos. A pesquisa e o desenvolvimento das várias aplicações destes sistemas deram origem à área de computação neural, também conhecida como neurocomputação.

O primeiro neurônio artificial foi desenvolvido por McCulloch e Pitts em 1943. McCulloch, um psiquiatra e neuroanatomista passou 20 anos tentando representar um evento no sistema nervoso e Pitts um matemático que associou-se com McCulloch em 1942. Juntos, publicaram um trabalho em 1943, descrevendo um cálculo lógico das redes neurais, unificando os estudos de neurofisiologia e da lógica matemática. O trabalho se concentrou muito mais na descrição de um modelo artificial de neurônio e apresentar suas capacidades computacionais, do que em apresentar técnicas de aprendizado.

Alguns anos após a publicação do trabalho de McCulloch e Pitts, o aprendizado de redes neurais biológicas e artificiais voltou a ser objeto de estudo. Em 1949, o psicólogo Donald Hebb postulou a base de aprendizado nas redes neurais, apresentando uma formulação explícita de uma regra de aprendizagem fisiológica para a modificação sináptica demonstrando, que a aprendizagem de redes neurais é obtida através da variação dos pesos de entrada dos neurônios, apresentando uma teoria que explica o aprendizado em neurônios biológicos baseado no reforço das ligações sinápticas entre neurônios excitados.

Em 1958, Frank Rosenblatt criou o *Perceptron* mostrando que ao se acrescentar sinapses ajustáveis ao modelo de rede neural proposto por McCulloch e Pitts, este poderia ser treinado para classificar padrões linearmente separáveis.

A topologia descrita por Rosenblatt, consiste em estruturas de ligação entre os neurônios e um algoritmo de treinamento para executar determinados tipos de funções. Outro modelo similar ao Perceptron também foi desenvolvido na mesma época, o *Adaline* (*Adaptive Linear Element*) criado por Bernard Widrow em 1960. Os modelos do tipo Perceptron, incluindo o Adaline, têm como base o aprendizado supervisionado por correção do erro.

Em 1956 Taylor iniciou o trabalho sobre a memória associativa por Taylor em 1956. Em 1969, Willshaw, Buneman e Longuet-Higgins publicaram um artigo sobre a memória associativa não-holográfica, apresentando dois modelos de redes: um sistema ótico simples realizando memória de correlação e uma rede neural intimamente relacionada com ele, inspirada na memória óptica. Outras contribuições significativas ao desenvolvimento da memória associativa incluem artigos de Anderson em 1972, Kohonen em 1972 e Nakano em 1972. E no mesmo ano introduziram a idéia de uma memória por matriz de correlação, baseada na regra de aprendizagem do produto externo (Haykin, 2001).

Em meados de 70 surgiram os mapas auto-organizáveis utilizando a aprendizagem competitiva. Em 1976, Willshaw e Von Der Malsburg publicaram o primeiro artigo sobre a formação de mapas auto-organizáveis e em 1980 Grosseberg, baseado na aprendizagem competitiva, estabeleceu um novo princípio de auto-organização, conhecido como Teoria da Ressonância Adaptativa (*ART, Adaptive Resonance Theory*) (Haykin, 2001).

Após um período de estagnação, as pesquisas em redes neurais ressurgiram em 1982, com a publicação do artigo de Hopfield, que propôs um modelo fundamentado em um tipo de rede, diferente dos modelos baseados na competição entre os neurônios, onde o aprendizado era não supervisionado. Outros modelos similares ao modelo de Hopfield surgiram em seguida como: a máquina de Boltzmann, proposta por Hinton em 1986; e a

Memória Associativa Bidirecional (BAM, *Bidirectional Associative Memory*) proposta por Kosko (1988).

Também na década de 80 reapareceram as redes baseadas em Perceptron com o modelo de rede perceptron de múltiplas camadas, que utilizava aprendizado supervisionado implementado por um algoritmo chamado *Backpropagation* que resolveu em grande parte os problemas levantados por Minsky e Papert em 1962. O algoritmo de aprendizado por *backpropagation* foi desenvolvido quase ao mesmo tempo por diferentes pesquisadores, como o Werbos em 1974, Parker em 1985, LeCun em 1985, Rumelhart em 1985, Hecht-Nielsen em 1987 e Baldi e Hornik em 1989. Mas foi Rumelhart e Hinton em 1986 que tornaram este algoritmo famoso com a sua obra "*Parallel Distributed Processing – PDP*". Além dos modelos de Hopfield e do modelo de redes perceptron de múltiplas camadas com *backpropagation* (chamada de *Multi-Layer Perceptron – MLP*), outro modelo importante que surgiu na década de 80 foi o modelo de Kohonen (1982), implementando o aprendizado competitivo com uma auto-organização da rede neural, criando os chamados "mapas de atributos auto-organizáveis" (SOFM - *self-organization feature maps*).

As redes neurais artificiais são sistemas paralelos distribuídos, compostos por neurônios que calculam determinadas funções matemáticas, normalmente não-lineares. Os neurônios podem ser distribuídos em uma ou mais camadas, e interligadas por um grande número de conexões (pesos), as quais armazenam o conhecimento representado no modelo e servem para ponderar a entrada recebida por cada neurônio da rede.

Uma das características importantes das redes neurais é a capacidade de aprender através de exemplos e de generalizar a informação. Nos modelos hetero-associativos, à medida que se apresenta um exemplo à rede, esta é ajustada para relacioná-lo com uma saída. Essa forma constitui um modelo que representa o relacionamento entre as variáveis de entrada e saída, capazes de extrair informações novas de forma explícita através de exemplos (Haykin, 2001).

As redes neurais são tolerantes a falhas, ou seja, tendem a tomar decisões corretas quando apresentados dados incompletos.

2.2 DEFINIÇÕES E CONCEITOS

O Neurologista Ramón Cajal identificou em 1911, a célula nervosa, ou neurônio, que é composto pelo corpo da célula, os dendritos e o axônio. O corpo, ou soma, é o centro dos processos metabólicos da célula nervosa; os dendritos são unidades estruturais e funcionais elementares que fazem as interações entre os neurônios, ou seja, faz o transporte das informações vindas de outros neurônios para dentro da célula e as informações são somadas no corpo celular processando assim uma outra informação, que sai da célula através do axônio.

O neurônio biológico pode ser visto como um dispositivo computacional elementar básico do sistema nervoso, com muitas entradas e uma saída. As entradas ocorrem através das conexões sinápticas, que conectam a árvore dendrital aos axônios de outras células nervosas. Por onde chegam pulsos elétricos conhecidos como impulsos nervosos que constituem a informação que o neurônio processará para produzir como saída um impulso nervoso no seu axônio (Figura 2.1).

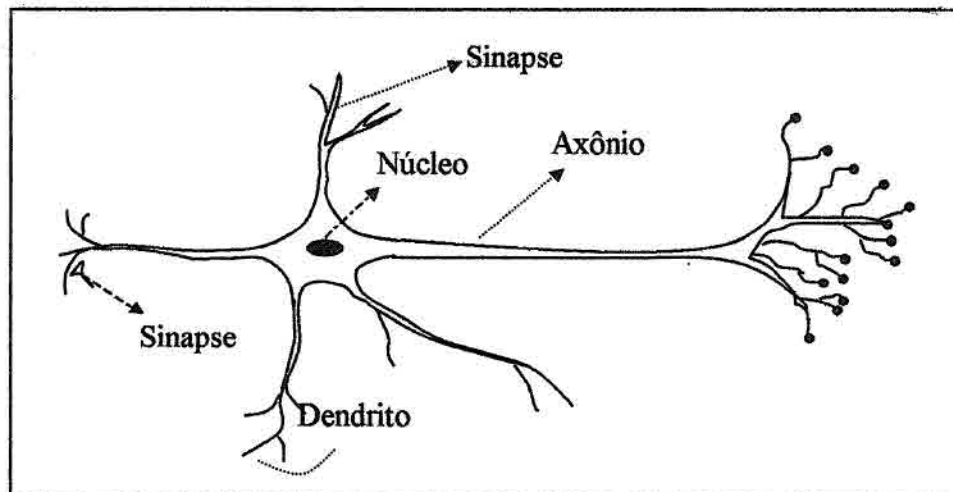


FIGURA 2.1 – Esquema da Célula Neural

FONTE: Modificado de Braga et al. (2000 p. 6).

A sinapse é a unidade funcional básica que une os neurônios, e controlam a transmissão de impulsos, ou seja, o fluxo da informação entre os neurônios na rede neural (Braga et al. 2000).

2.2.1 NEURÔNIO ARTIFICIAL

Os neurônios artificiais são baseados nos neurônios biológicos. Os neurônios biológicos são células nervosas que recebem e transformam estímulos em outros estímulos, enquanto neurônios artificiais transformam várias entradas em uma única saída (Haykin, 2000). A Figura 2.2 mostra a representação esquemática de um neurônio artificial.

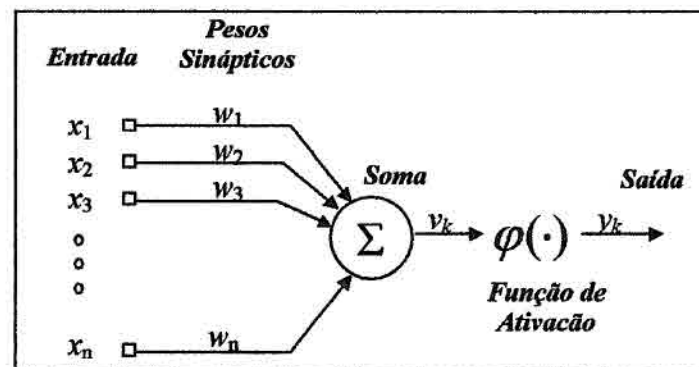


FIGURA 2.2 – Representação de um Neurônio Artificial.
FONTE: Haykin (2000 p. 36).

A função básica de um neurônio artificial é adicionar entradas e produzir uma saída. Se o somatório das entradas for maior que um certo valor limite (*threshold*), haverá uma saída; caso contrário, não será produzida uma saída. A estrutura de um neurônio artificial é semelhante à estrutura dos neurônios biológicos. As informações chegam através dos dendritos, são somadas no corpo celular, e o neurônio processa uma nova informação.

Pode-se dizer que, um neurônio só terá dois estados na saída: ligado ou desligado, dependendo das entradas. Cada entrada j é ponderada por um peso sináptico, representado por w_j para $j=1, 2, \dots, n$ e n o número total de entradas.

Os pesos são coeficientes adaptáveis dentro da rede e determinam a intensidade do sinal de entrada. A soma ponderada das entradas é efetuada pela Equação 2.1.

$$v_k = \sum_{j=1}^n w_{kj} x_j \quad (2.1)$$

Na equação (2.1) os componentes do vetor de entrada são x_j para $j=1, 2, \dots, n$ e o peso sináptico correspondente é representado pelo w_{kj} , que conecta a entrada j ao neurônio k . O sinal de atividade interna total (v_k) do neurônio v é o produto interno entre o vetor \bar{x} e o vetor \bar{w} .

2.2.2 FUNÇÃO DE ATIVAÇÃO

Função de ativação restringe a amplitude da saída de um neurônio e impõe um intervalo permissível de amplitude do sinal de saída a um valor finito, sendo um intervalo normalizado escrito como em intervalo unitário fechado $[0,1]$ ou alternativamente $[-1, 1]$.

A partir do modelo de McCulloch e Pitts, foram derivados vários outros modelos que permitem a produção de uma saída qualquer (não necessariamente zero ou um) com diferentes funções de ativação. Na figura 2.3 têm-se alguns exemplos de funções de ativação alternativas àquela de McCulloch e Pitts. A Figura 2.3-(a) é a função sinal; a Figura 2.3-(b) é a função rampa; a Figura 2.3-(c) é a função logística; e Figura 2.3-(d) é a função tangente hiperbólica.

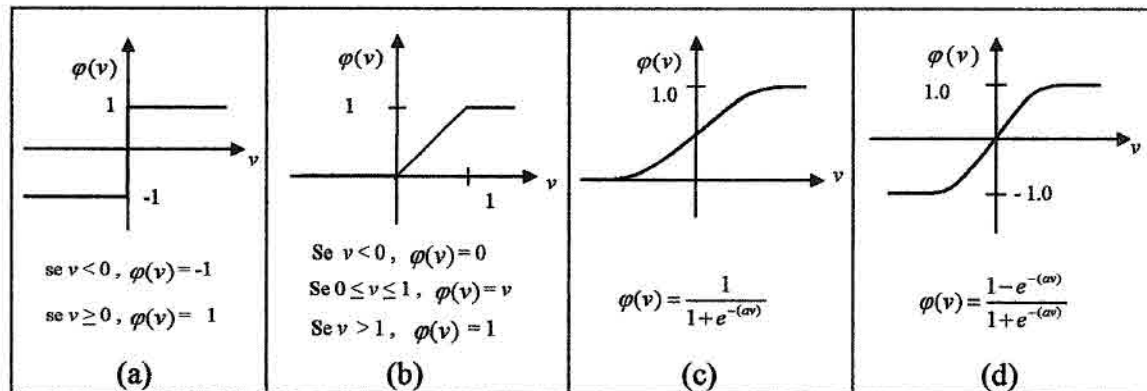


FIGURA 2.3 – Funções de Ativação: (a) Sinal, (b) Rampa, (c) Logística e (d) Tangente Hiperbólica.

FONTE: Modificado de Braga et al. (2000 p. 10).

2.3 ARQUITETURAS DAS REDES NEURAIS

Através da combinação dos neurônios artificiais pode-se gerar diferentes arquiteturas de redes. As arquiteturas das redes são compostas por: número de camadas da rede, números de neurônios em cada camada, tipo de conexão entre os neurônios e topologia da rede.

Os neurônios são geralmente organizados em camadas, podendo ter uma ou mais camadas. Quando existe realimentação entre essas camadas, a rede é chamada de recorrente.

A Figura 2.4 mostra uma arquitetura simples onde os neurônios formam uma rede com uma única camada. Nela as entradas projetam-se diretamente sobre a camada de saída. A rede é chamada de “*feed-forward*” (alimentação para frente) por não ter realimentação na rede.

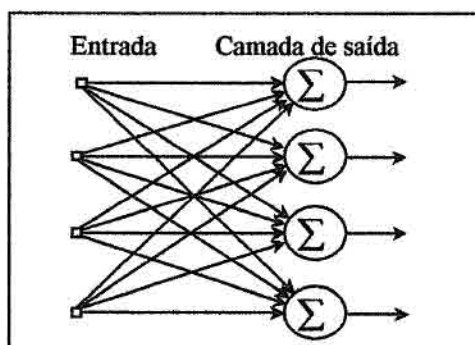


FIGURA 2.4 – Rede com uma Única Camada.
FONTE: Modificada de Haykin (2000 p. 47).

Os neurônios também podem estar dispostos de modo a formar uma arquitetura composta com duas ou mais camadas. Na Figura 2.5 tem-se um exemplo, onde a primeira camada é a entrada, a última camada de saída e as outras, que não têm contato com o meio externo, são denominadas camadas escondidas, compostas pelos chamados neurônios escondidos.

Quando todas as saídas de uma camada estão conectadas a todos os neurônios da próxima camada a conexão é total, na falta de alguma dessas conexões, a rede é referida como parcialmente conectada.

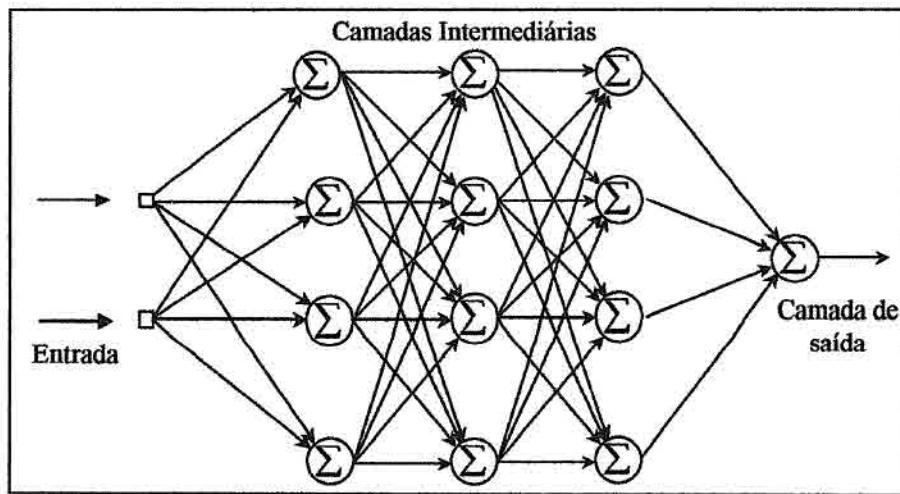


FIGURA 2.5 – Rede de Três Camadas
 FONTE: Modificada de Haykin (2000 p.48).

2.4 PROCESSOS DE APRENDIZAGEM

O processo de aprendizagem, também conhecido como treinamento, consiste no processo de adaptação dos pesos sinápticos das conexões e dos níveis de limiar (*bias*¹) dos neurônios em resposta às entradas. A ativação consiste no processo de receber uma entrada e produzir uma saída de acordo com a arquitetura final da rede.

2.4.1 Aprendizado Supervisionado

Esse método de aprendizado é o mais comum no treinamento das redes neurais. É assim chamado porque a entrada e saída desejada para a rede são fornecidas anteriormente, e a rede ajusta os parâmetros (pesos) de forma a encontrar uma ligação entre os pares de entrada e saída fornecidos. A cada padrão de entrada submetido à rede, compara-se a resposta desejada com a resposta calculada e ajustam-se os pesos das conexões para minimizar o erro. A soma dos erros quadráticos de todas as saídas é, normalmente, utilizada como medida de desempenho da rede e também como função de custo a ser minimizada pelo algoritmo de treinamento.

Esta adaptação por correção de erros minimiza a diferença entre a soma ponderada das entradas pelos pesos e a saída desejada, ou seja, o erro da resposta atual da rede. A

¹ Bias é um parâmetro externo, que tem o efeito de aumentar ou diminuir a entrada da função de ativação.

minimização da função custo resulta na regra de aprendizagem normalmente referida como *regra delta* ou *regra de Widrow-Hoff*, assim denominada em homenagem aos seus criadores Widrow e Hoff em 1960 (Haykin, 2001).

2.4.2 Aprendizado Não-Supervisionado

No aprendizado não-supervisionado não há exemplos rotulados da função, que a rede deva aprender. Para este tipo de aprendizagem somente os padrões de entrada estão disponíveis para a rede, ao contrário do aprendizado supervisionado. A rede se auto-organiza para que cada neurônio responda a diferentes conjuntos de entrada (Haykin, 2001; Braga et al. 2000).

2.4.3 Aprendizado Competitivo

Na aprendizagem competitiva os neurônios de saída de uma rede neural competem entre si para serem ativados. Nesse tipo de aprendizagem um único neurônio de saída é ativado em um determinado instante e seus pesos atualizados no treinamento. Tal característica torna a aprendizagem competitiva adequada para descobrir características marcantes que podem ser utilizadas para classificar um conjunto de padrões de entrada. Existem três elementos básicos em uma regra de aprendizagem competitiva:

- Um conjunto de neurônios iguais, exceto por alguns pesos sinápticos distribuídos aleatoriamente, e que *respondam diferentemente* a um dado conjunto de padrões de entrada.
- Um *limite* de robustez para cada neurônio.
- Um mecanismo que permita que os neurônios possam competir pelo direito de responder a um dado subconjunto de entradas, de forma que somente *um* neurônio de saída, ou somente um neurônio do grupo, seja ativado em um determinado instante. O neurônio que vence a competição é denominado um *neurônio vencedor* (Haykin, 2001).

Na forma mais simples de aprendizagem competitiva, a rede neural tem uma única camada de neurônios de saída, estando cada neurônio totalmente conectado aos nós de entrada. A rede pode incluir conexões de realimentação entre os neurônios. O aprendizado por competição é à base do modelo ART (Grosberg, 1987) e dos mapas de Kohonen (Kohonen, 1982).

2.4.4 Aprendizado Hebbiano

O aprendizado Hebbiano é a mais antiga e mais famosa de todas as regras de aprendizagem. A regra de aprendizado por Hebb propõe que o peso de uma conexão sináptica seja ajustado se houver sincronismo entre os níveis de atividade das entradas e saídas. Quando dois neurônios, em lados distintos da sinapse, são ativados de forma sincronizada deve haver fortalecimento da sinapse. Entretanto, se os neurônios forem ativados assincronamente, a sinapse será enfraquecida ou mesmo eliminada. Existem quatro propriedades fundamentais que caracterizam uma sinapse:

- *Mecanismo Dependente do Tempo:* Este mecanismo se refere ao fato de que as modificações em uma sinapse hebbiana dependem do tempo exato de ocorrência dos sinais pré-sinápticos e pós-sinápticos.
- *Mecanismo Local:* Uma sinapse é um local de transmissão onde sinais portadores de informação estão em contigüidade espaço-temporal. Esta informação localmente disponível é utilizada por uma sinapse hebbiana para produzir uma modificação sináptica local que é específica para a entrada. É a partir do mecanismo local que sinapses hebbianas efetuam o aprendizado não supervisionado.
- *Mecanismo Interativo:* dentro do aprendizado hebbiano, não se pode analisar atividades pré-sinápticas de forma isolada; qualquer modificação na sinapse hebbiana depende da interação entre os dois tipos de atividade.
- *Mecanismo Conjuncional ou Correlacional:* a sinapse hebbiana pode ser chamada de sinapse conjuncional pelo fato de a ocorrência conjunta de

atividades pré e pós-sinápticas ser suficiente para que haja uma modificação. Além disso, pode também ser chamada de sinapse correlacional porque uma correlação entre estas mesmas atividades também é suficiente para gerar mudanças.

Pode-se formular a aprendizagem hebbiana em termos matemáticos a partir da equação 2.2, que consiste em uma regra para a mudança do peso sináptico w_{kj} no passo de tempo t :

$$\Delta w_{kj}(t) = \eta y_k(t) x_j(t) \quad (2.2)$$

onde η é uma constante positiva que determina a taxa de aprendizagem, $y_k(t)$ é a saída do neurônio k e x_j corresponde à entrada associada ao peso w_{kj} .

A regra de Hebb é classificada como aprendizado não-supervisionado, pois não existe supervisor externo que verifica a qualidade da resposta da rede para ajustar os pesos. Neste caso, o treinamento da rede é feito independentemente da resposta atual da rede, através de um mecanismo local à sinapse.

2.5 REDES PERCEPTRON

O Perceptron foi proposto por Rosenblatt em 1958, como um modelo de rede neural com aprendizagem supervisionada. O perceptron trata-se de uma rede simples de uma única camada com pesos e bias ajustáveis. Rosenblatt provou que se os padrões usados para treinar o perceptron são retirados de duas classes linearmente separáveis, então o algoritmo do perceptron converge e posiciona a superfície de decisão na forma de um hiperplano entre as duas classes. A prova de convergência do algoritmo é conhecida como teorema de convergência do perceptron. Um único neurônio perceptron limita-se a realizar classificação de padrões de apenas duas classes. Expandindo a camada de saída do perceptron para incluir mais de um neurônio, pode-se classificar mais de duas classes, no entanto as classes devem ser linearmente separáveis (Haykin, 2001).

No processo de adaptação, ou aprendizado, deseja-se obter o valor do incremento $\Delta \bar{w}$ a ser aplicado ao vetor de pesos \bar{w} para um valor atualizado $\bar{w}(n+1) = \bar{w}(n) + \Delta \bar{w}(n)$.

O erro entre a resposta esperada e a resposta produzida pela rede da equação 2.3,

$$e_k(n) = d_k(n) - y_k(n) \quad (2.3)$$

onde $d_k(n)$ e $y_k(n)$ são os sinais de saída desejado e obtido respectivamente no neurônio k no instante n .

Os cálculos dos novos pesos e dos novos bias são obtidos por:

$$w(n+1) = w(n) + \eta x(n)e(n) \quad (2.4)$$

e

$$b(n+1) = b(n) + \eta e(n)(-1) \quad (2.5)$$

onde $w(n)$, $x(n)$, $e(n)$ e $b(n)$ são respectivamente, os pesos, a entrada, o erro e o bias no instante n e η é a taxa de aprendizagem.

2.6 REDE PERCEPTRON DE MÚLTIPLAS CAMADAS

Uma classe importante de redes neurais são as redes de múltiplas camadas. As redes perceptron de múltiplas camadas têm sido aplicadas para resolver problemas difíceis, sendo treinadas de forma supervisionada com um algoritmo, conhecido como retropropagação do erro (error back-propagation) (Haykin, 2001).

Um Perceptron de Múltiplas Camadas é composto por uma camada de entrada, uma ou mais camadas escondidas e uma camada de saída, como mostra a Figura 2.6. O algoritmo de treinamento funciona em dois passos: um passo para frente (ativação da rede) e um passo para trás, a retropropagação do erro.

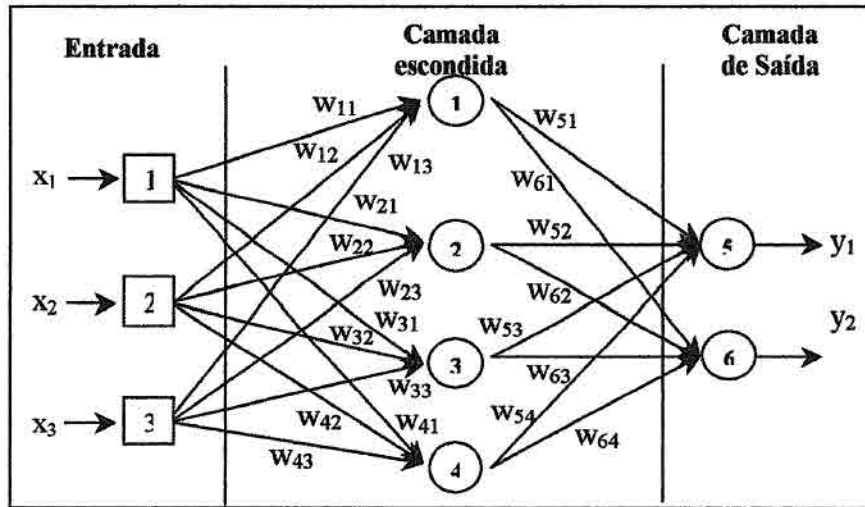


FIGURA 2.6 – Redes de Camadas Múltiplas

Na fase de ativação da rede o sinal de entrada se propaga para frente, camada por camada, até que o vetor saída seja obtido na última camada. O número de neurônios na camada escondida varia de acordo com a aplicação apesar de não existir um método para determiná-lo a priori.

O funcionamento do neurônio é semelhante aquele do neurônio de McCulloch e Pitts. Mas a ativação de cada neurônio é dada por uma função φ (ver equação 2.6) que deve ser uma função diferenciável. Nesta rede é possível variar o número de neurônios na camada escondida e também o número de camadas.

$$y_k(n) = \varphi_k(v_k(n)) \quad (2.6)$$

No processo de propagação, todos os pesos sinápticos da rede são fixos. Durante o passo para trás, os pesos sinápticos são ajustados de acordo com uma regra de correção do erro. A resposta real da rede é subtraída de uma resposta desejada (alvo) para produzir um sinal de erro (Equação 2.3), que é então propagado para trás, através da rede.

Os pesos sinápticos são ajustados para fazer com que a resposta real da rede se aproxima da resposta desejada, num sentido estatístico (Haykin, 2001). O processo de

aprendizado termina quando se atinge um erro desejado ou quando se atinge um número máximo de épocas de aprendizado.

Os novos pesos são obtidos pela equação:

$$w_{kj}^L(n+1) = w_{kj}^L(n) + \eta \delta_k^L(n) y_j^{L-1}(n) \quad (2.7)$$

onde $y_j^{L-1}(n)$ é saída do neurônio j na camada $L-1$, na iteração n , $w_{kj}^L(n)$ é o peso sináptico do neurônio k da camada L , que é alimentado pelo neurônio j da camada $L-1$, η é a taxa de aprendizagem e $\delta_k^L(n)$ é o gradiente do neurônio k da camada L .

O gradiente da última camada (ou seja, se L é uma camada de saída) é obtido pela equação:

$$\delta_k(n) = -\frac{\partial \varepsilon(n)}{\partial v_k(n)} = -e_k(n) \varphi'_k(v_k(n)) \quad (2.8)$$

O gradiente nas camadas escondidas é dado pela equação:

$$\delta_k(n) = \varphi'_k(v_k(n)) \sum_j \delta_j(n) w_{jk}(n) \quad (2.9)$$

Nas equações (2.8) e (2.9), $\varphi'_k(v_k(n))$ apresenta o valor da derivada da função de ativação do neurônio φ , na atividade interna do neurônio no instante n , $v_k(n)$.

2.7 RBF

Powell (1985) desenvolveu o modelo de Rede de Função de Base Radial (RBF) com o intuito de resolver o problema de ajuste de curva num espaço de alta dimensionalidade, onde sua generalização é equivalente ao uso da superfície de múltiplas dimensões para interpolar os dados de teste.

Uma rede de Funções de Base Radial (RBF) do tipo feed-forward (alimentação para frente) possui a camada de entrada, uma camada escondida e a camada de saída (Figura 2.7). O algoritmo de aprendizado da RBF é supervisionado, no entanto as conexões

entre as camadas de entrada e a escondida, diferentemente das redes perceptron de múltiplas camadas, representam centros de classes, podendo ser fixos. Apenas os pesos entre a camada escondida e a saída são ajustados de maneira semelhante à equação (2.7) onde L é a camada de saída e $L-1$ é a camada escondida.

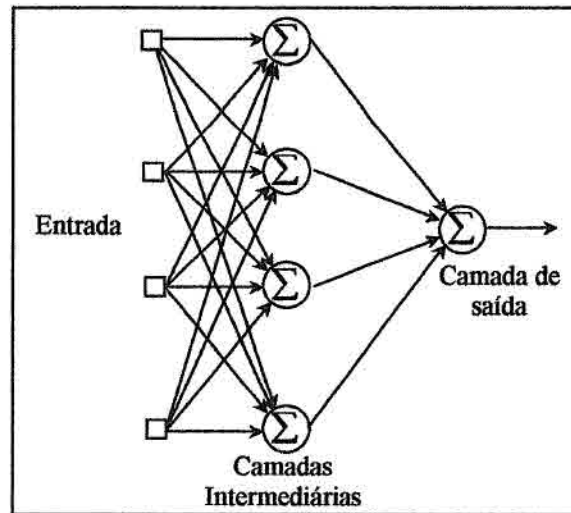


FIGURA 2.7 – Arquitetura da RBF com uma Camada Intermediária (escondida). Os neurônios da camada intermediária são funções de base radial.
 FONTE: Modificado Braga et al. (2000 p.194)

A ativação da rede é dada (saída da rede) pela Equação 2.10,

$$y_k(n) = \sum_{j=1}^m w_{kj} \varphi_j(v_j(n)) \quad (2.10)$$

onde w_{kj} é o peso do neurônio de saída k para o neurônio escondido j , m é o número de neurônios e $\{\varphi_j(\bar{x}) | j = 1, 2, \dots, m\}$ é o conjunto de funções de base, dadas por:

$$\varphi_j(\bar{x}) = G\left(\|\bar{x} - \bar{t}_j\|\right), \quad j=1, 2, \dots, m \quad (2.11)$$

onde \bar{t}_j para $j=1, 2, \dots, m$ é o centro correspondente ao vetor j . Assim, pode-se redefinir $y_k(n)$ como:

$$y_k(n) = \sum_{j=1}^m w_{kj} G(\bar{x}, \bar{t}_j) \quad (2.12)$$

onde G é uma função de base radial como uma Gaussiana (2.13), Multiquadráticas (2.14), entre outras (Haykin, 2001).

$$\varphi(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right) \text{ para um } \sigma \text{ (desvio padrão)} > 0 \text{ e } r \in \mathfrak{R} \quad (2.13)$$

$$\varphi(r) = (r^2 + c^2)^{1/2} \text{ para um } c > 0 \text{ e } r \in \mathfrak{R} \quad (2.14)$$

A aprendizagem consiste na determinação de um novo conjunto de pesos que minimize o custo $\varepsilon(n)$ definido por:

$$\varepsilon(n) = \sum_{k=1}^N \left(d_k - \sum_{j=1}^m w_{kj} G\left(\|\bar{x}_k(n) - \bar{t}_j\|\right) \right)^2 \quad (2.15)$$

Algumas estratégias de aprendizado podem ser implementadas numa Rede de Funções de Base Radial dependendo de como os centros das Funções são especificados (Haykin, 2001). Duas delas são apresentadas na seção 2.8.1 e 2.8.2.

Como mencionado anteriormente, os pesos entre a saída e a camada escondida são atualizados pela equação (2.7), entretanto a saída da RBF é uma combinação linear das funções de base radial na camada escondida, dada pela equação (2.10). Assim para esta rede φ' , que aparece na equação (2.7) é igual a 1.

2.7.1 Centros Fixos Selecionados Aleatoriamente.

Na estratégia mais simples, assume-se centros (\bar{t}_j) fixos para as funções. Os centros são escolhidos aleatoriamente do conjunto de treinamento. Empregando-se funções Gaussianas isotrópicas, com desvio padrão que é fixado de acordo com o espalhamento dos centros, a uma função de base radial centrada em \bar{t}_k , é definida como:

$$G\left(\|\bar{x} - \bar{t}_j\|\right) = \exp\left(-\frac{M}{d^2} \|\bar{x} - \bar{t}_j\|^2\right), \quad j=1,2,\dots,m. \quad (2.16)$$

onde M é o número de centros e d é a distância máxima desejada entre os centros escolhidos. O desvio padrão (σ) de todas as funções de base radial são fixadas em:

$$\sigma = \frac{d}{\sqrt{2M}} \quad (2.17)$$

Esta fórmula assegura que as funções de base radial individual não sejam pontiagudas ou planas demais.

Com uma alternativa para Equação 2.17, pode-se usar centros escalonados individualmente com larguras maiores em áreas de menor densidade de pontos, o que requer experimentação com os dados do treinamento.

Assim, os únicos parâmetros que devem ser aprendidos nesta abordagem são os pesos dos neurônios na camada de saída da rede.

2.7.2 Seleção Auto-organizada de Centros

Esta estratégia de aprendizado tem um processo híbrido constituído de dois estágios diferentes. O primeiro consiste na aprendizagem auto-organizada, para estimar os centros das funções de base radial da camada escondida. O segundo consiste na aprendizagem supervisionada na camada de saída.

No estágio de aprendizagem auto-organizada, um algoritmo de agrupamento junta dados similares, que colocam os centros das funções de base radial nas regiões de espaço de entrada. Um algoritmo pode ser o agrupamento de k -médias (Duda e Hart, 1973). Nele os valores dos centros são iniciados aleatoriamente, e um vetor de entrada é selecionado do conjunto de treinamento, depois é feito um casamento de similaridade entre esse vetor e o centro de cada classe, utilizando a distância euclidiana mínima:

$$j^* = \arg \min \|\bar{x}(n) - \bar{t}_j(n)\|, \quad j = 1, 2, \dots, m \quad (2.18)$$

onde “arg min” corresponde ao argumento da distância mínima entre a entrada $\bar{x}(n)$ e o centro da j -ésima função de base radial t_j na iteração n , j^* representa o índice do centro com a menor distância.

O ajuste dos centros das funções base radial é obtido por:

$$\bar{t}_j(n+1) = \begin{cases} \bar{t}_j(n) + \eta[\bar{x}(n) - \bar{t}_j(n)], & \text{se } j = j^* \\ \bar{t}_j(n), & \text{c.c.} \end{cases} \quad (2.19)$$

onde η é a taxa de aprendizagem.

No processo de ajuste dos pesos e limiares da camada de saída, usa-se o algoritmo a equação (2.7), como no caso anterior.

2.8 MEMÓRIA BIDIRECIONAL ASSOCIATIVA

A BAM (Kosko, 1987) é uma arquitetura de duas camadas que combina N pares de padrões hetero-associativos (\bar{X}_p, \bar{Y}_p) , $p = 1, 2, \dots, N$, codificados como padrões espaciais binários (0, 1) ou bipolares (-1, 1), usando a regra de aprendizagem de *Hebb* (Haykin 2001). Os vetores das equações (2.20) e (2.21) representam o p -ésimo par de padrões.

$$\bar{X}_p = (x_{p1}, x_{p2}, \dots, x_{pn}) \quad (2.20)$$

e

$$\bar{Y}_p = (y_{p1}, y_{p2}, \dots, y_{pn}) \quad (2.21)$$

A arquitetura de uma memória associativa bidirecional (BAM) é mostrada na Figura 2.8.

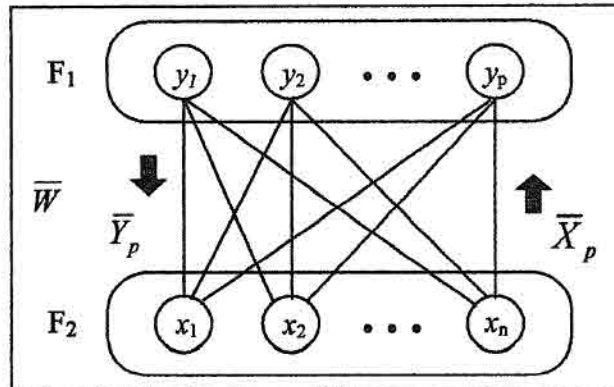


FIGURA 2.8 – Arquitetura de uma Memória Associativa Bidirecional
 FONTE: Modificado Kosko (1988 p.53)

Na Figura 2.8, a BAM é composta por duas camadas de neurônios (F_1 e F_2) com conexões sinápticas bidirecionais entre elas. A rede envia sinais para frente e para trás entre as duas camadas até que todos os neurônios alcancem o equilíbrio, ou seja, até que ativação de cada neurônio permaneça constante por vários passos.

O número total de nós da memória na BAM é a soma dos números de nós de cada camada. O número de conexões é o produto do número de unidades em cada camada ($n_1 = n^\circ$. de neurônios na camada $F_1 \times n^\circ$. de neurônios na camada F_2).

A aprendizagem é Hebbiana, na BAM é dada por:

$$\bar{W} = \sum_{p=1}^N (\bar{X}_p)(\bar{Y}_p) \quad (2.22)$$

onde \bar{W} é a matriz de pesos resultante.

Após a fase de aprendizagem, quando a BAM é exposta a um padrão, ela tenta recuperar o respectivo padrão associado. As equações que regem a ativação são:

$$\bar{Y}_p = \bar{X}_p \bar{W} \quad (2.23)$$

e

$$\bar{X}_p = \bar{Y}_p \bar{W}^T \quad (2.24)$$

onde \bar{W} e \bar{W}^T são as matrizes pesos e sua transposta respectivamente. Uma vez que o padrão \bar{Y}_p é fornecido, a BAM tenta “relembrar” o padrão \bar{X}_p e vice-versa. Estes procedimentos de “relembrar” padrões são executados pelas Equações (2.23) e (2.24).

2.9 TEORIA DA RESSONÂNCIA ADAPTATIVA

A ART (*Adaptive Resonance Theory*), foi desenvolvida por Carpenter e Grossberg em 1976 como uma alternativa para solucionar o problema conhecido como o dilema da estabilidade-plasticidade. São redes do tipo auto-organizável (*self-organizing*), que utilizam o paradigma de aprendizado não-supervisionado (Braga et al. 2000).

Estes são modelos que apresentam forte inspiração biológica e cognitiva. A primeira rede ART proposta (ART1) utiliza padrões codificados com valores binários. Posteriormente foi desenvolvida a rede ART2 que permite também o uso de valores contínuos de entrada. Outras versões do modelo ART podem ser encontradas em Fausett (1994).

Os modelos de redes ART têm sido utilizados em várias aplicações como: reconhecimento de padrões, robótica, diagnóstico médico, sensoriamento remoto e processamento de voz, visão computacional, etc (Braga et al. 2000) (Rangsanseri et al. 2001).

2.9.1 Rede ART1

A rede ART1 tem uma arquitetura composta por duas camadas (Figura 2.9): uma camada de entrada, que processa os dados de entrada, e uma camada de saída, que agrupa os padrões de treinamento em *clusters*. As camadas estão conectadas por dois conjuntos de conexões que unem cada neurônio de uma camada a todos os neurônios da outra. O primeiro conjunto representado por conexões *feed-forward*, \bar{w} , assume valores reais e segue da camada de entrada para a camada de saída. O segundo conjunto, que contém as conexões de *feedback*, \bar{t} , assume valores binários e conecta os neurônios da camada de saída aos neurônios de entrada. Assim, o neurônio i da camada de entrada está conectado ao neurônio j da camada de saída através do peso w_{ji} .

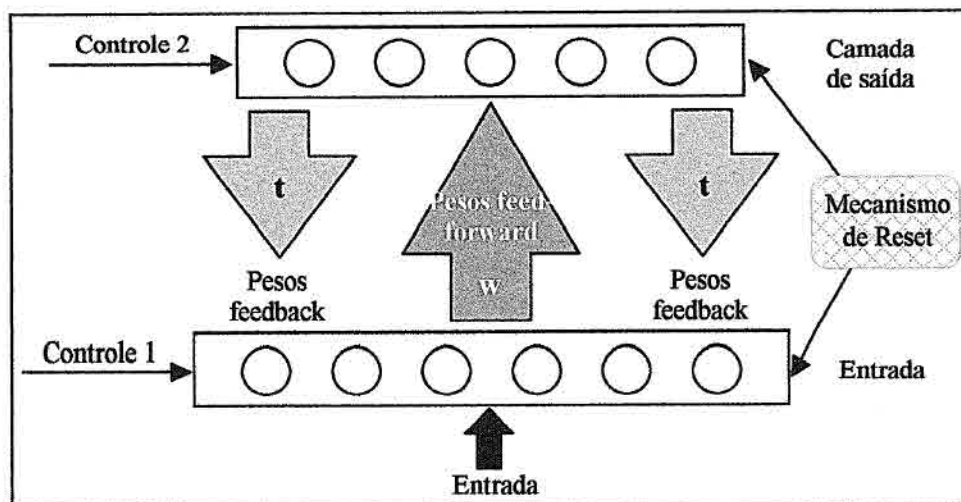


FIGURA 2.9 – Arquitetura Básica da Rede ART, com uma camada de entrada e uma de saída, duas unidades de controle e um mecanismo de reset.

FONTE: Modificada de Beale e Jackson (1990 p.168)

Da mesma forma, o neurônio j da camada de saída está conectado ao neurônio i da camada de entrada através do peso t_{ij} . Entre os neurônios da camada de saída existem pesos que atuam como inibidores laterais.

Os pesos *feed-forward* e *feedback* se adaptam dinamicamente, para possibilitar o aprendizado de novos padrões. Cada vetor de pesos *feedback* associado a cada neurônio da camada de saída pode ser visto como um protótipo ou exemplo dos padrões pertencentes ao *cluster* representado por este neurônio.

Um conjunto de pesos bipolares não-adaptáveis existe entre os neurônios da camada de saída. O peso entre dois neurônios da camada de saída é -1 , se este liga dois neurônios distintos; e $+1$ se liga a sua saída a si mesmo.

Cada camada da rede ART1 possui uma unidade externa de controle, sendo C_1 para a camada de entrada e C_2 para a camada de saída. Estas unidades de controle C_1 e C_2 determinam o fluxo de dados para as camadas de entrada e saída, respectivamente. Além disso, C_2 tem a função de habilitar ou desabilitar neurônios da camada de saída.

O mecanismo de *reset* é responsável por verificar a semelhança entre um vetor de entrada e um dado vetor protótipo, utilizando um limiar de vigilância (ρ). Este limiar de

vigilância determina se um padrão de entrada pode ser incluído em um dos *clusters* existentes.

O algoritmo de aprendizado da rede ART1 é não-supervisionado e pode ser ativado a qualquer momento, permitindo que a rede aprenda novos padrões continuamente. O treinamento da rede ART1 pode ser rápido ou lento. No primeiro caso, os pesos *feed-forward* são setados para seus valores ótimos em poucos ciclos, geralmente em apenas um ciclo de treinamento. No aprendizado lento, os pesos são ajustados lentamente em vários ciclos de treinamento, possibilitando um ajuste melhor dos pesos da rede aos padrões de treinamento.

Com relação aos parâmetros durante o treinamento, a rede ART se apresenta muito sensível, sendo o parâmetro mais crítico aquele que controla a resolução do processo de classificação, que é o limiar de vigilância (ρ). Se ρ assume um valor baixo (< 0.4), a rede permite que padrões não muito semelhantes sejam agrupados no mesmo *cluster*, criando poucas classes. Caso contrário, atribuindo a ρ um valor alto, pequenas variações nos padrões de entrada levarão à criação de novas classes.

Antes do treinamento, os pesos e os parâmetros da rede devem ser inicializados. Todos os pesos de *feedback* (\bar{t}) são inicializados com valor 1, indicando que todo neurônio da camada de saída está inicialmente conectado a todo neurônio de entrada através de uma conexão *feedback*. O limiar de vigilância (ρ) é ajustado para um valor dentro do intervalo $0 < \rho < 1$. Os valores iniciais do vetor de pesos *feed-forward* (\bar{w}) são determinados pela equação:

$$w_{ji}(0) = \frac{1}{1 + N_n}, \forall i, j \quad (2.25)$$

onde N_n é o número de neurônios de entrada da rede.

O processo de aprendizado da rede ART1 envolve algumas fases, tais como, reconhecimento, comparação e busca (Braga et al. 2000).

Na fase de reconhecimento, cada neurônio da camada de entrada recebe três sinais: um sinal do vetor de entrada, um sinal de *feedback* da camada de saída e um sinal da unidade de controle C_1 . Um neurônio da camada de entrada somente será ativado se receber pelos menos dois sinais excitatórios. Esta regra é denominada *regra dos dois terços (2/3)* e é aplicada a cada neurônio da camada de entrada, gerando um vetor resultante que é comparado com todos os vetores de pesos *feed-forward*, para encontrar o mais semelhante. Esta comparação é realizada calculando o produto interno entre o vetor resultante e o vetor de pesos *feed-forward* para cada neurônio da camada de saída, conforme Equação 2.26:

$$y_j = \sum_i w_{ji} x_i \quad (2.26)$$

onde y_j é a ativação do neurônio de saída j , definido pelo produto interno entre seu vetor de pesos \bar{w}_j e seu vetor de entrada \bar{x} .

O neurônio da camada de saída que tiver o maior valor de ativação é selecionado como provável *cluster* para armazenar o novo padrão. A inibição lateral faz com que somente o neurônio vencedor da camada de saída fique ativo. O neurônio selecionado envia à camada de entrada seu vetor de pesos \bar{t} .

Na fase de comparação tem-se o valor da unidade de controle C_1 igual a 0, uma vez que um neurônio foi ativado na fase anterior. Aplica-se novamente a regra dos 2/3, resultando na ativação ou não de cada neurônio da camada de entrada. É gerado um novo vetor, o *vetor comparação* z , resultante da operação AND entre o vetor de pesos do neurônio selecionado e o vetor de entrada.

O mecanismo de reset é responsável por testar a similaridade entre o vetor de entrada e o vetor comparação. O teste é feito calculando a razão P , por meio da Equação 2.27, entre o número de valores iguais a 1 de ambos os vetores e comparando essa razão com o limiar de vigilância (ρ).

$$P = \frac{\left\| \sum x_i z_{ij} \right\|}{\left\| \sum x_i \right\|} \quad (2.27)$$

Se P for maior que o limiar de vigilância (ρ), então o padrão de entrada é incluído no *cluster* do neurônio ativo na camada de saída, sendo ajustados os vetores de pesos do neurônio vencedor para incorporar as características do vetor de entrada corrente. Caso contrário outro neurônio da camada de saída deverá ser encontrado para armazenar o vetor de entrada, e a rede entra na fase de busca.

Durante a fase de busca, a rede procura selecionar um novo neurônio da camada de saída para representar o vetor de entrada corrente. O neurônio rejeitado da camada de saída tem seu valor de saída igual a 0, impedido de participar desta seleção. O vetor de entrada é rerepresentado, e a rede entra novamente na fase de comparação. O processo se repete até que seja encontrado um neurônio de saída que melhor se assemelhe ao vetor de entrada corrente, dentro dos limites do limiar de vigilância ρ . Se nenhum neurônio de saída for encontrado, o vetor de entrada é então considerado de uma classe desconhecida, sendo alocado um neurônio de saída “livre” (aquele que não está associado ainda a nenhum *cluster*) para representá-lo.

2.9.2 Rede ART2

A ART2 é uma rede do tipo auto-organizável para o reconhecimento estável em seqüências arbitrárias de padrões de entrada analógicos (Carpenter e Grossberg, 1987). Uma diferença entre as redes ART2 e ART1, é que a ART2 utiliza vetores de entrada com valores contínuos, enquanto a ART1 utiliza vetores de entrada com valores binários. Isso implica em modificações no algoritmo de treinamento da rede.

A arquitetura típica da ART2 (Figura 2.10) é composta por duas camadas, a camada F_1 e a camada F_2 . A camada F_1 é a mais complexa, sendo necessária porque os vetores de entrada com valores contínuos arbitrários podem se arbitrariamente semelhantes. A camada F_1 na ART2 inclui uma combinação da normalização e supressão do ruído, além da comparação dos sinais de baixo para cima e de cima para baixo, necessária para o mecanismo de reset.

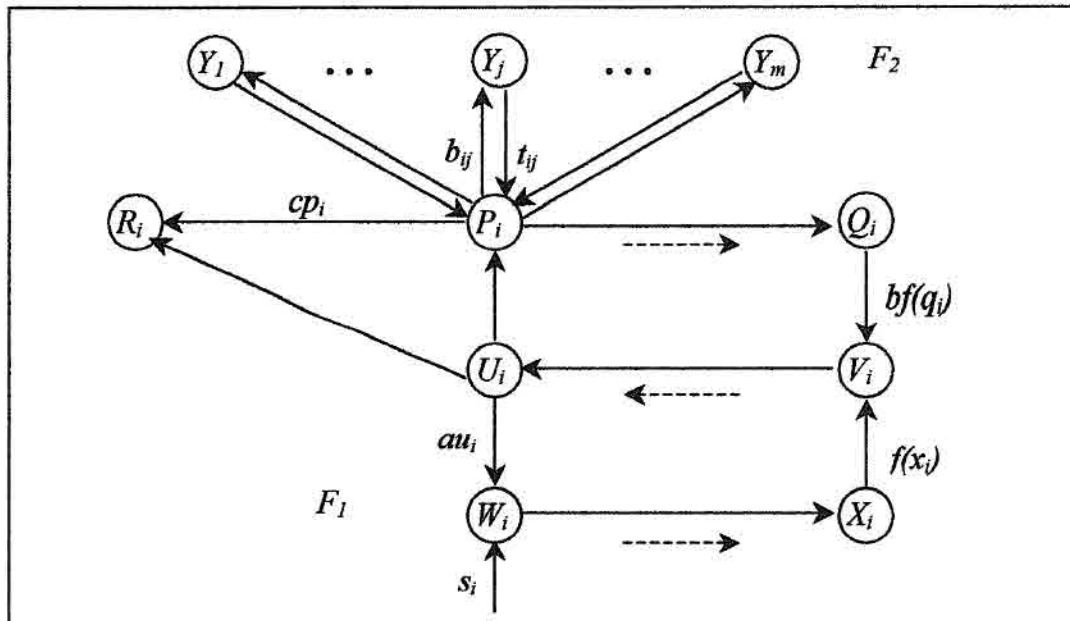


FIGURA 2.10 – Arquitetura Básica da Rede ART2
 FONTE: Modificada de Fausett (1994 p.248)

A camada F_1 consiste em seis tipos de unidades W , X , U , V , P e Q , existindo n unidades dessas seis unidades. A camada F_2 contém somente um tipo de unidade, denotada por Y , e serve como uma camada competitiva. A conexão é total entre as camadas F_1 e F_2 . A ação da camada F_2 é exatamente igual à ação na ART1, cada unidade compete com as demais para ser o neurônio vencedor que aprenderá o padrão de entrada. Como na ART1, a aprendizagem ocorre sempre quando o vetor peso para baixo (top-down) da unidade vencedora é suficientemente similar ao vetor de entrada. Os testes para o mecanismo de reset da ART1 e ART2 são diferentes, a função das unidades U é similar às unidades da camada F_1 da ART1.

Na ART2 alguns processamentos dos vetores de entrada são necessários por que as magnitudes dos componentes desses vetores são valores reais. A rede ART2 trata de componentes com valores pequenos como ruído e, portanto não distingue entre padrões que são apenas versões reduzidas deles mesmos (Fausett, 1994).

Uma tentativa de aprendizagem consiste na apresentação de um padrão de entrada. O sinal de entrada contínuo (Equação 2.28) é enviado enquanto todas as ações são executadas. No começo da aprendizagem, todas as ativações são ajustadas para zero.

$$\bar{s} = (s_1, \dots, s_i, \dots, s_n) \quad (2.28)$$

O ciclo computacional (para uma tentativa de aprendizagem) na camada F_1 é originado com a ativação da unidade U_i (a ativação de V_i normalizada com valor aproximadamente 1). Em seguida, um sinal é enviado de cada unidade U_i para suas unidades associadas W_i e P_i . As ativações das unidades W_i e P_i são então computadas. A unidade W_i soma o sinal que recebe de U_i e da entrada S_i . P_i soma o sinal que recebe de U_i e o sinal *top-down* que recebe se houver uma unidade de F_2 ativa. As ativações de X_i e Q_i são versões normalizadas dos sinais W_i e P_i , respectivamente. Uma função de ativação é aplicada em cada uma destas unidades antes que o sinal seja enviado a V_i , que então soma os sinais que recebe simultaneamente de X_i e Q_i ; completando um ciclo de atualização da camada de F_1 .

Com a função de ativação da Equação 2.29, as ativações das unidades U e P encontram o equilíbrio após duas adaptações da camada F_1 .

$$f(x) = \begin{cases} x & \text{if } x \geq \theta \\ 0 & \text{if } x < \theta \end{cases} \quad (2.29)$$

Esta função trata qualquer sinal que é menor que θ como ruído e oculta isto. O valor do parâmetro θ é especificado pelo usuário. A supressão do ruído ajuda a rede a alcançar uma formação estável do agrupamento. A rede está estável quando a primeira unidade de cluster escolhido para cada padrão de entrada é aceito e nenhum reajuste acontece. Também na aprendizagem lenta, os vetores de peso para os clusters convergirão para valores estáveis.

Após as ativações das unidades de F_1 terem atingido o equilíbrio, as unidades P enviam seus sinais para a camada F_2 , onde um processo competitivo escolhe um neurônio vencedor para aprender o padrão de entrada. As unidades U_i e P_i na camada F_1

também enviam sinais para a unidade de *reset* R_i correspondente. O mecanismo de *reset* pode checar o *reset* a cada tempo recebendo um sinal de P_i , como as computações necessárias são baseadas no valor desse sinal e o sinal da unidade R_i mais recente foi recebido de U_i . Entretanto, isto só precisa ser feito quando P_i receber primeiro sinal de cima para baixo, uma vez que os valores de parâmetros sejam escolhidos, tal que nenhum reajuste acontecerá se nenhuma unidade de F_2 for ativada, ou depois que aprendizagem começar.

Depois das condições do *reset* terem sido conferidas, a unidade do agrupamento candidato ou será rejeitada, como não similar o bastante ao padrão de entrada, ou será aceita. Se a unidade do agrupamento for rejeitada, será inibida, e a unidade de cluster com a próxima entrada, com contribuição para a rede é escolhida como candidata. Este processo continua até que uma unidade de cluster aceitável é escolhida. Quando uma unidade de cluster é escolhida isso passa pelas as condições do *reset* e aprendizagem irá acontecer (Fausett, 1994).

Existem dois tipos de aprendizagem, a aprendizagem lenta e a aprendizagem rápida. Na aprendizagem lenta, somente uma interação da adaptação das equações dos pesos ocorre em cada processo de aprendizagem. É necessário apresentar um número grande de cada padrão, mas a computação relativamente pequena é realizada em cada processo. Por conveniência, essas apresentações repetidas são consideradas como épocas nos algoritmos. Entretanto, não há qualquer exigência para que os padrões sejam apresentados na mesma ordem ou que exatamente os mesmos padrões sejam apresentados em cada ciclo por eles.

Na aprendizagem rápida, em algumas épocas é necessário um grande número de interações para as adaptações dos pesos da camada F_1 . Na aprendizagem rápida, a colocação dos padrões em clusters estabiliza-se, mas os pesos mudarão para cada padrão apresentado.

O algoritmo de treinamento pode ser usado tanto para a aprendizagem rápida como para aprendizagem lenta. Na aprendizagem rápida, as iterações da mudança dos pesos seguem através das adaptações da ativação da F_1 até que o equilíbrio seja encontrado.

Na aprendizagem lenta, somente uma iteração da adaptação dos pesos é executada, mas um número grande de tentativas de aprendizagem é necessário para que a rede estabilize-se (Carpenter e Grosseberg, 1987). As equações (2.31 - 2.35) apresentam as adaptações para a camada de ativação F_1 .

$$u_i = \frac{v_i}{e + \|v\|} \quad (2.30)$$

onde u_i e v_i são os vetores da camada F_1 e e é um parâmetro pequeno para prevenir divisão por zero quando a norma do vetor ($\|v\|$) for zero.

$$w_i = s_i + au_i \quad (2.31)$$

onde w_i e u_i são os vetores da camada F_1 e s_i é a entrada da rede e a é um peso fixo da camada F_1 .

$$p_i = u_i + dt_{ji} \quad (2.32)$$

onde p_i é um vetor da camada F_1 , t_{ji} é o peso inicial de cima para baixo e d é a ativação da unidade F_2 vencedora.

$$x_i = \frac{w_i}{e + \|w\|} \quad (2.33)$$

$$q_i = \frac{p_i}{e + \|p\|} \quad (2.34)$$

$$v_i = f(x_i) + bf(q_i) \quad (2.35)$$

onde x_i , w_i , p_i , q_i e v_i são vetores da camada F_1 , f a função de ativação e b é o peso inicial de baixo para cima.

2.10 APRENDIZAGEM POR QUANTIZAÇÃO VETORIAL

Kohonen propôs a rede neural artificial denominada Aprendizagem por Quantização Vetorial (Learning Vector Quantization - LVQ) como um método de classificação de padrão, no qual cada unidade de saída representa uma classe ou uma categoria. Os pesos para cada unidade de saída são considerados como um vetor referência da classe que a unidade representa. A LVQ usa um algoritmo de aprendizagem supervisionada que usa a informação sobre as classes para mover ligeiramente os vetores de referência para melhorar a qualidade das regiões de decisão do classificador (Simpson,1989; Fausett, 1994; Kohonen et al. 1996).

A arquitetura da rede LVQ é similar à rede de Kohonen, possuindo duas camadas. O número de unidades na camada de entrada é igual ao número de atributos de entrada usados para classificação, e o número de unidades na camada de saída representa as classes (ou categorias) de saída. A rede LVQ é uma rede totalmente conectada, ou seja, cada unidade de saída é conectada a todas as unidades de entrada. O vetor de peso para uma unidade de saída representa um exemplar (referência) da categoria que a unidade representa. (Fausett, 1994; Kulkarni, 2001).

Após o treinamento, a rede LVQ classifica um vetor de entrada atribuindo-o a unidade com vetor de pesos (vetor de referência) mais semelhante ao vetor de entrada.

Existem algumas variações do algoritmo LVQ, chamado LVQ2, LVQ2.1, e LVQ3 (Fausett, 1994). No algoritmo original do LVQ, somente o vetor de referência que é mais próximo ao vetor de entrada é atualizado, ou seja, os pesos são atualizados se o vetor de referência da unidade vencedora pertencer à mesma classe do vetor de entrada. Nesta dissertação foi utilizada apenas a rede LVQ original (denominada aqui de LVQ1) e a LVQ2.

2.10.1 LVQ1

A arquitetura da rede neural LVQ1 está mostrada na Figura 2.11 e é similar à arquitetura dos mapas auto-organizáveis de Kohonen. Cada unidade de saída representa uma classe conhecida.

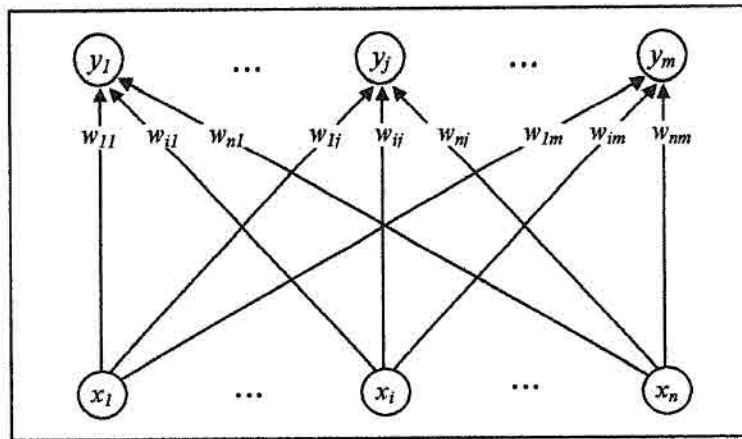


FIGURA 2.11 – Aprendizagem por Quantização Vetorial
FONTE: Modificada de Simpson (1990 p.87).

A motivação para o algoritmo da rede LVQ1 é achar a unidade de saída que mais se aproxime do vetor de entrada. Na direção da saída, se x e w_c pertencerem à mesma classe, então move-se os pesos ao novo vetor de entrada ou se x e w_c pertencerem a classes diferentes, então não se move os pesos para esse vetor de entrada.

Um método simples de inicializar o vetor peso é pegar os primeiros vetores de treinamento m e usá-los como vetores de treinamento. Outro método utilizado é fixar os pesos iniciais e fazer classificações randômicas (Kulkarni, 2001).

Outro método possível para inicializar os pesos é usar clusterização por K-means ou mapa auto-organizáveis no lugar dos pesos. Cada vetor de peso é então calibrado para determinar os padrões de entrada que são armazenados, buscando a classe que tem o maior número pertencentes a esses padrões de entrada e fixando essa classe a um vetor peso (Fausett, 1994).

Para achar a unidade vencedora, apresenta-se o vetor de entrada $\bar{x} = (x_1, \dots, x_i, \dots, x_n)$ e procura-se a unidade vencedora usando o critério da distância Euclidiana entre os

vetores de entrada e o vetor de pesos $\bar{w}_j = (w_{j1}, \dots, w_{ji}, \dots, w_{jn})$ para cada unidade de saída j :

$$\|\bar{w}_i - \bar{x}\| < \|\bar{w}_j - \bar{x}\| \text{ para } j = 1, 2, \dots, n, \quad i \neq j \quad (2.36)$$

A unidade de saída j vencedora do processo anterior tem o vetor de pesos adaptados pela equação:

$$\bar{w}_j(n+1) = \bar{w}_j(n) + \alpha(t)[\bar{x} - \bar{w}_j(n)] \quad (2.37)$$

para a categoria incorreta é utilizada por:

$$\bar{w}_i(n+1) = \bar{w}_i(n) - \alpha(t)[\bar{x} - \bar{w}_i(n)] \quad (2.38)$$

onde $\alpha(t)$ é a taxa de aprendizagem na iteração t que pode sofrer redução entre uma e outra.

2.10.2 LVQ2

O algoritmo LVQ2 é uma versão do algoritmo da rede LVQ1. Neste último apenas a unidade mais próxima do vetor de entrada tem os pesos adaptados, enquanto que no algoritmo LVQ2 quando a unidade vencedora não representa a categoria correta e a unidade seguinte (mais próxima) representar a categoria correta, é realizado o treinamento de ambas as unidades. Assim, o vetor de pesos do vencedor é adaptado de modo que este se afaste do vetor de entrada, e o vetor de pesos que conecta a unidade seguinte (mais próxima) é adaptado de modo que essa unidade se aproxime do vetor de entrada.

Uma condição imposta pelo algoritmo LVQ2 é que a distância (d_v) entre o vetor de entrada (\bar{x}) e o vetor de pesos para a unidade vencedora (\bar{w}_v) e a distância (d_s) entre o vetor de entrada e o vetor de pesos da unidade seguinte (\bar{w}_s) sejam aproximadamente iguais. Isso significa que os vetores de pesos das unidades são adaptados somente se o vetor de entrada satisfizer as condições mostradas nas Equações 2.39 e 2.40:

$$\frac{d_v}{d_s} > 1 - \varepsilon \quad (2.39)$$

e

$$\frac{d_s}{d_v} < 1 + \varepsilon \quad (2.40)$$

onde o valor de ε depende do número de amostras de treinamentos. Quando todas as condições anteriores são satisfeitas, os vetores pesos são adaptados segundo as equações:

$$\bar{w}_s(n+1) = \bar{w}_s(n) + \alpha(t)[\bar{x}(n) - \bar{w}_s(n)] \quad (2.41)$$

$$\bar{w}_v(n+1) = \bar{w}_v(n) - \alpha(t)[\bar{x}(n) - \bar{w}_v(n)] \quad (2.42)$$

2.11 MEMÓRIA ASSOCIATIVA NEBULOSA

A rede Memória Associativa Nebulosa (Fuzzy Memory Association - FAM) foi introduzida por Kosko em 1987 como uma arquitetura composta por duas camadas para ser um classificador nebuloso heteroassociativos que armazena pares de padrões espaciais (A_k, B_k) com valores nebulosos no intervalo unitário. O k -ésimo par de padrões é representado pelo vetores $A_k = \{a_1^k, \dots, a_n^k\}$ e $B_k = \{b_1^k, \dots, b_p^k\}$. A rede usa aprendizagem Hebbiana nebulosa; aprende *offline*; e opera em tempo discreto. A topologia da rede FAM está mostrada na Figura 2.12 onde a camada F_A recebe o vetor A_k com n componentes e a camada F_B recebe o vetor B_k com p componentes.

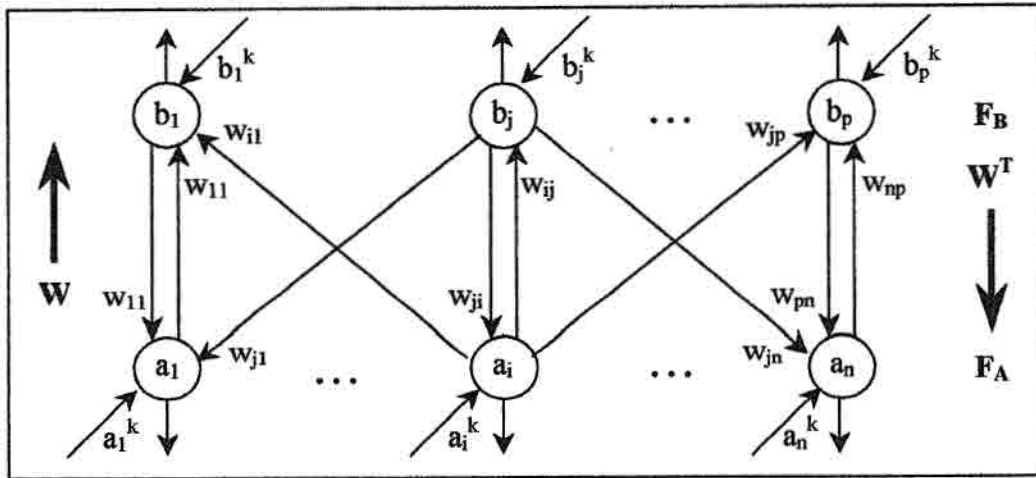


FIGURA 2.12 – Arquitetura da Memória Associativa Nebulosa (FAM)
 FONTE: Modificada de Simpson (1990 p. 84)

As operações básicas da lógica nebulosa associada com o funcionamento da rede FAM são *MIN* e *MAX* (mínimo e máximo) que representam a interseção e a união, respectivamente. Considerando os conjuntos nebulosos $A_k = \{0.5, 0.2, 0.8\}$ e $B_k = \{0.9, 0.3\}$, o valor do segundo elemento de A_k é dado por $M_A(a_2^k) = 0.2$ e $M_B(b_1^k) = 0.9$. Exemplos da aplicação dos operadores *MIN* e *MAX* são $MIN(M_A(a_1^k), M_A(a_2^k)) = 0.2$ e $MAX(M_B(b_1^k), M_B(b_2^k)) = 0.9$.

A aprendizagem Hebbiana nebulosa é implementada pela equação (2.43)

$$w_{ij}^k = MIN(M_A(a_i^k), M_B(b_j^k)) \quad (2.43)$$

Embora as equações que recuperam os padrões sejam bidirecionais, a FAM não usa a direção de realimentação (*feedback*), todo o fluxo de informação é simplesmente para frente (*feed-forward*). A Equação 2.44 mostra a recuperação de cada componente do padrão B apresentado na camada F_B .

$$b_j = MAX_{i=1}^n [MIN(a_i, w_{ij})] \quad (2.44)$$

onde a_i é o valor do i -ésimo componente do vetor apresentado em F_A , b_j é o valor do j -ésimo componente do vetor apresentado em F_B . A Equação 2.45 é a equação de recuperação dos componentes de um vetor na camada F_A .

$$a_i = \underset{j=1}{\overset{p}{\text{MAX}}} [\text{MIN}(b_j, w_{ji})] \quad (2.45)$$

É interessante notar a troca da ordem dos índices i e j nas equações 2.44 e 2.45. Este processo de recuperação mapea todas as entradas em subconjuntos das associações armazenadas na memória (Kosko, 1987).

2.12 MAXNET

A rede MAXNET pertence a uma classe de redes neurais que usam mecanismos de competição entre os neurônios para realçar o contraste nas ativações dos neurônios. Na situação mais extrema, chamada de vencedor leva tudo (*Winner-take-all*), somente o neurônio que tem a maior ativação é ativado (Fausett, 1994).

A MAXNET usa um mecanismo competitivo para seleccionar o valor máximo de um conjunto de dados. A topologia da MAXNET é de uma rede recorrente com conexões excitatórias (de auto-realimentação com peso 1) e inibitórias que representam realimentações cruzadas de valor fixo $-\varepsilon$ (coeficiente de inibição lateral).

Os m neurônios da MAXNET são completamente interconectados, com pesos simétricos e fixos, ou seja, o treinamento da rede consiste da atribuição de valores a ε . A arquitetura da rede MAXNET é apresentada na Figura 2.12.

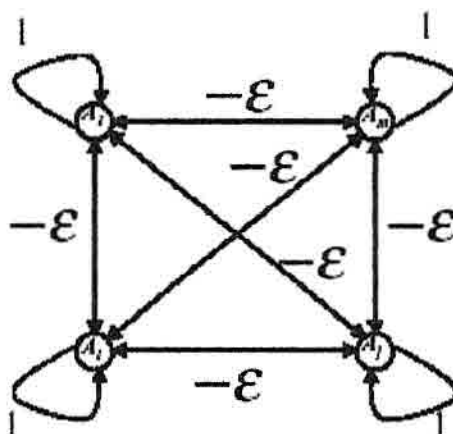


FIGURA 2.12 – Arquitetura da Rede Maxnet
FONTE: Silva, 2002.

A função de ativação de cada unidade na MAXNET é mostrada pela equação 2.46 onde x a atividade interna do neurônio.

$$f(x) = \begin{cases} x & \text{se } x > 0 \\ 0 & \text{caso contrário} \end{cases} \quad (2.46)$$

A inicialização dos pesos na MAXNET é dada pela equação 2.47, onde ε é a inibição entre as unidades.

$$w_{ij} = \begin{cases} 1 & \text{se } i = j \\ -\varepsilon & \text{se } i \neq j \end{cases} \quad 0 < \varepsilon < \frac{1}{m} \quad (2.47)$$

Inicialmente a rede é apresentada a um vetor $A = (a_1, a_2, a_3, \dots, a_m)$. A ativação de cada neurônio é realizada pela equação.

$$a_j^{Novo} = f \left[a_j^{Antigo} - \varepsilon \sum_{k \neq j} a_k^{Antigo} \right] \quad (2.48)$$

onde ε é a inibição inicial e a_j é a entrada do neurônio A_j .

A função f é dada pela equação 2.46 e a rede repete o ciclo até se ter um único valor diferente de zero no vetor A .

CAPÍTULO 3

DETECÇÃO DE BORDAS

Neste capítulo faz-se uma breve introdução ao processamento de imagens, apresentando alguns métodos de pré-processamento e de extração de informações em imagens. Dado o escopo do trabalho, enfatiza-se as técnicas de segmentação em especial aquelas envolvendo detecção de bordas.

3.1 FUNDAMENTOS

Os métodos de processamento digital de imagens têm 2 (dois) objetivos principais: melhoramento de imagens para interpretação humana e processamento de dados (pixels) para percepção autônoma pelas máquinas. O primeiro objetivo data do início dos anos 20, com a necessidade de se enviar fotos jornalísticas via cabo submarino de Nova York a Londres; e hoje, é aplicada para a resolução de uma variedade de problemas em diversos campos. O segundo tem como foco de interesse, a extração de informações de imagens em uma forma adequada ao processamento computacional, para problemas como reconhecimento automático de caracteres, máquinas com visão industrial para acompanhamento da montagem de produtos, reconhecimento de alvos militares, entre outros (Gonzalez e Woods, 2000).

A área de processamento digital de imagens ganhou força devido à evolução da tecnologia digital e ao desenvolvimento de novos algoritmos para lidar com sinais bidimensionais. Estes fatos têm permitido uma gama de aplicações cada vez maior, em diversas áreas, como medicina, sensoriamento remoto, microscopia ótica, eletrônica, transmissão digital de sinais, arqueologia, astronomia, geografia, física, biologia, medicina nuclear, apoio à lei e defesa e aplicações industriais, etc. Uma boa introdução à área de processamento de imagens, pode ser encontrada em Gonzalez e Woods (2000).

A área de Processamento Digital de Imagens teve uma grande evolução em meados da década de 1960, com o programa espacial norte-americano. As técnicas de Processamento de Imagens serviram de base para o realce e restauração de imagens em missões como Surveyor para a Lua, a série Mariner de missões para Marte e para os vôos tripulados da Apollo para a Lua (Gonzalez e Woods, 2000).

3.2 IMAGEM DIGITAL

Uma imagem monocromática analógica pode ser descrita por uma função matemática bidimensional $f(x, y)$ da intensidade luminosa, sendo seu valor, em qualquer ponto de coordenadas espaciais (x, y) , proporcional ao brilho (ou nível de cinza) naquele ponto. A função $f(x, y)$ representa o produto da interação entre a iluminância $i(x, y)$ que ilumina o objeto e as propriedades de reflectância ou de transmitância próprias do objeto (Gonzalez e Woods, 2000). Essas propriedades podem ser representadas pela função $r(x, y)$ cujo valor exprime a fração de luz incidente que o objeto vai refletir no ponto (x, y) .

O processamento de imagens por computador requer a representação digital da imagem para o tratamento computacional. Uma imagem digital resulta da discretização de uma imagem $f(x, y)$ nas coordenadas espaciais (x, y) e no brilho. A representação de uma imagem digital é uma matriz cujos índices das linhas e das colunas identificam um ponto na imagem e o valor do elemento da matriz identifica o nível de cinza naquele ponto, como pode ser observado na Figura 3.1.

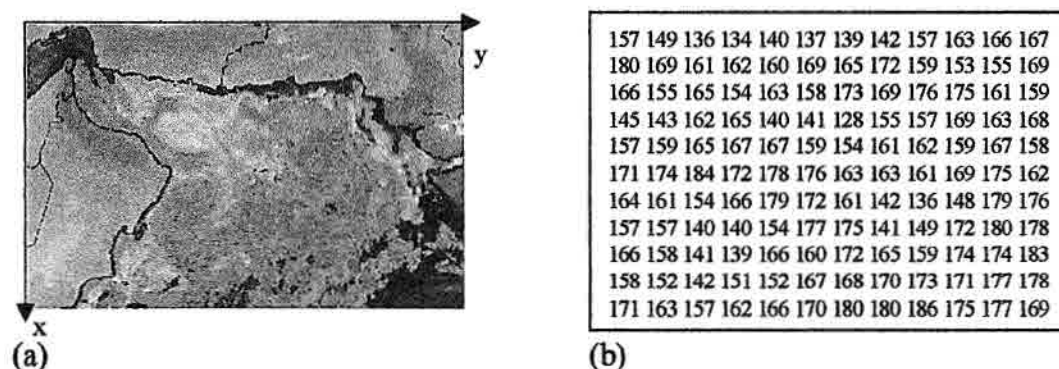


FIGURA 3.1 – (a) Sistema de coordenadas de referência em uma imagem; (b) Representação matricial de um trecho de uma imagem digital.

FONTE: <http://xtreme.gsfc.nasa.gov/oceancolor/>

Os elementos da matriz são chamados de *elementos da imagem*, *elementos da figura* ou *pixels*. Considerando que o processo de discretização envolve a definição de parâmetros de amostragem e quantização, o número de amostras e de níveis de cinza para gerar uma boa imagem digital, influencia a qualidade da imagem quanto a sua representação da cena original. (Para uma análise detalhada da influência de tais parâmetros na formação de imagens digitais ver Gonzalez e Woods (2000) capítulo 7).

Além da imagem monocromática, existe a imagem que possui informações adquiridas em intervalos ou bandas de frequências distintas. Para sua representação é necessária uma função $f(x, y)$ para cada banda. Este é o caso das imagens padrão RGB¹, onde a cor em cada ponto (x, y) é definida através de três grandezas: luminância, matiz e saturação. A luminância está associada com o brilho da luz, a matiz com o comprimento de onda dominante e a saturação com o grau de pureza (ou intensidade) da matiz. A maioria das cores visíveis pelo olho humano pode ser representada como uma combinação das três cores primárias, vermelho, verde e azul. Assim, uma representação comum para uma imagem colorida utiliza três bandas R, G, B com profundidade 1 byte por pixel Gonzalez e Woods (2000).

3.3 SISTEMAS DE PROCESSAMENTO DE IMAGENS

As técnicas de Processamento Digital de Imagens permitem analisar imagens monocromáticas e RGB, além de possibilitarem a integração de vários tipos de dados. Uma das áreas de aplicação que requer a integração de diferentes tipos de dados é a de sensoriamento remoto, quando usa imagens obtidas por satélites em diferentes regiões do espectro eletromagnético. A Figura 3.2 mostra a seqüência de tarefas necessárias para executar o processamento digital de imagens.

¹ RGB (Red, Green e Blue)

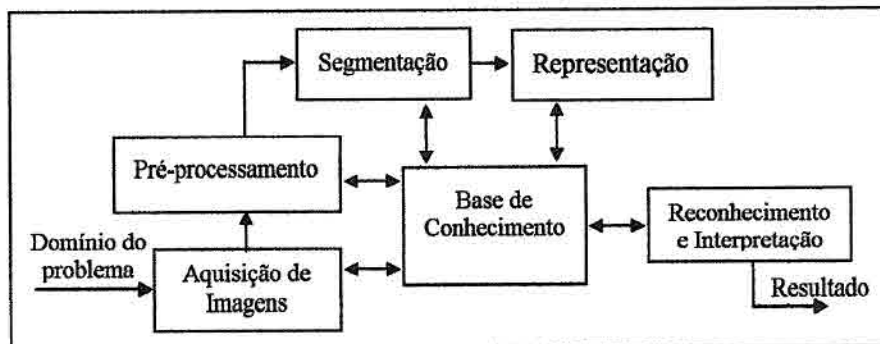


FIGURA 3.2 – Passos Fundamentais em Processamento de Imagens Digitais
 FONTE: Adaptado de Gonzalez e Woods (2000. p.5).

A primeira tarefa é a aquisição da imagem por um equipamento sensor com capacidade de digitalização. A imagem digitalizada resultante é uma matriz de valores discretos inteiros (*pixels*).

No pré-processamento da imagem corrige-se defeitos e imperfeições resultantes do processo de aquisição da imagem, através de técnicas de realce de contraste, remoção de ruído, isolamento de regiões, etc.

No processo de segmentação, uma imagem é dividida em partes ou objetos constituintes, através da detecção de descontinuidades (contornos) e/ou de similaridade (regiões) na imagem, que são armazenadas em forma de segmentos. A saída do estágio de segmentação é constituída tipicamente por dados em forma de *pixels*, correspondendo tanto à fronteira de uma região como a todos os pontos da mesma. Nos dois casos é preciso converter os dados para uma forma adequada ao processamento computacional. A representação por fronteira é adequada para as características da forma externa, tais como cantos ou pontos de inflexão. A representação por região é adequada para as características das propriedades internas, tais como textura ou a forma do esqueleto. Essas representações coexistem, em aplicações de reconhecimento de caracteres, que utilizam algoritmos baseados na forma da borda e em outras propriedades internas.

Na representação, os dados iniciais são transformados para uma forma adequada para o processamento computacional subsequente. Para enfatizar as características de interesse,

deve-se especificar um método que descreva os dados, através da extração de características que resultam em alguma informação quantitativa de interesse ou que sejam básicas para discriminação entre classes de objetos.

O reconhecimento é o processo que atribui um rótulo a um objeto, baseado na informação fornecida pelo seu descritor. A interpretação atribui um significado a um conjunto de objetos reconhecidos (rotulados).

A base de conhecimento tem as informações sobre o domínio do problema, podendo ser simples ou complexa em função da aplicação. Os módulos de processamento são guiados pela base de conhecimento, como pode ser visto na Figura 3.2, onde as flechas unidirecionais indicam as saídas de alguns módulos de processamento são conhecimento para outros (Gonzalez e Woods, 2000).

3.4 SEGMENTAÇÃO DE IMAGENS

Nas seções 3.1 e 3.3, foi feita uma introdução superficial sobre o Processamento Digital de Imagens. Como mencionado anteriormente, uma introdução mais detalhada pode ser encontrada em Gonzales e Woods (2000).

O foco desta dissertação é a segmentação de imagens, enfatizando em particular o uso de redes neurais artificiais como operadores de detecção de bordas. Assim, nesta seção apresenta-se uma pequena introdução sobre segmentação de imagens, enfatizando-se alguns operadores de detecção de bordas publicados na literatura de processamento de imagens e de visão computacional. Mais detalhes sobre técnicas de segmentação de imagens podem ser encontrados em (Gonzalez e Woods, 2000), (Nalwa, 1993), (Jain, 1989) e (Mascarenhas e Vellasco, 1984), por exemplo.

3.4.1 Segmentação por Região

Na segmentação dos objetos de uma imagem o particionamento é realizado por algoritmos que buscam as diferenças entre objetos com base nos conceitos de similaridade e descontinuidade que permitem a distinção entre agrupamentos de regiões de pixels contínuos.

Segundo Gonzalez e Woods (2000), o processo de segmentação não é formalizado. Os algoritmos existentes se baseiam em estudos empíricos e podem ser ajustados para diferentes tipos de imagem. Em geral, o processo de segmentação tenta executar por computador um processo cognitivo sofisticado realizado pela visão humana.

A segmentação é um passo importante na identificação de objetos simples como linhas ou regiões em imagens. A segmentação por regiões pode ser feita de dois modos: através do agrupamento de pontos vizinhos com características semelhantes, ou através da determinação da fronteira (borda) da região. Em ambos os casos, considera-se que uma imagem resulta da união de subconjuntos disjuntos de *pixels* (Mascarenhas e Velasco, 1984).

Uma região na imagem é um conjunto de pontos ligados onde, de qualquer ponto da região pode-se chegar a qualquer outro ponto através das ligações na região. Geralmente as regiões detectadas, são regiões homogêneas, ou seja, regiões que apresentam a propriedade de continuidade do nível de cinza, isto é, invariância do nível de cinza na região.

Gonzalez e Woods (2000) relacionam e discutem vários métodos usados para particionar a imagem em regiões. Alguns deles são: limiarização, divisão, fusão e crescimento de regiões. O princípio da limiarização consiste em separar o primeiro plano do fundo e tem como objetivo marcar os *pixels* que pertencem às regiões do primeiro plano com um valor simples e as regiões do fundo com intensidade diferente, gerando uma imagem binária. O método de divisão e fusão permite agrupar pixels para criar regiões homogêneas, manipulando a imagem através de divisões e fusões até se obter regiões que satisfaçam uma medida de homogeneidade estabelecida. No crescimento de regiões busca-se agrupar pixels com mesmas propriedades seguindo-se uma busca linha a linha, pixel a pixel da imagem ou em todas as direções, partindo-se de um *pixel* tomado como semente.

Uma revisão técnica sobre os métodos de segmentação por região citada, pode ser encontrada em Gonzalez e Woods (2000) e Mascarenhas e Velasco (1984).

3.4.2 Segmentação por Contorno

O contorno em uma imagem está associado à mudança brusca do nível de cinza entre duas regiões relativamente homogêneas e aparece como uma seqüência de pontos, uma linha, um segmento, uma curva ou uma variação forte do nível de cinza médio. Estes elementos podem ser detectados usando operadores como, detectores de pontos interessantes e detectores de bordas. Gonzalez e Woods (2000) descrevem uma técnica para detecção de pontos interessantes baseada na vizinhança do ponto em análise.

Como mencionado anteriormente, nesta dissertação há interesse apenas na análise do desempenho de operadores de detecção de bordas em imagens. Portanto, na seção 3.4.2.1 são descritas algumas das várias técnicas, disponíveis na literatura de processamento de imagens e de visão computacional, como parâmetro para comparar com a modelagem usando redes neurais, como proposto nesta dissertação.

3.4.3 Detecção de Bordas

As bordas em uma imagem são locais onde ocorrem mudanças súbitas do nível de cinza, entre regiões relativamente homogêneas. Estes locais são caracterizados pelos contornos dos objetos e são úteis para segmentar e identificar objetos da cena.

As mudanças súbitas de níveis de cinza entre regiões correspondem aos locais de alta derivada espacial, que contêm alta frequência espacial (Kulkarni, 2001). Os operadores de detecção de bordas devem gerar imagens, de modo que tais locais apareçam com alta intensidade e as regiões internas dos objetos apareçam com baixa intensidade.

É importante distinguir o conceito de borda (*edge*) e o de contorno (*boundary*): um contorno é uma linha fechada formada pelas bordas de um objeto (Kulkarni, 2001); uma borda tem uma característica local, em geral, restrita a um pequeno subconjunto de *pixels* da imagem. Pela própria natureza, um contorno apresenta a propriedade de continuidade, enquanto uma borda não possui esta propriedade. Apesar de um contorno ser formado pela bordas de uma imagem, nem sempre é possível obtê-lo a partir das bordas detectadas.

A teoria de detecção de bordas foi explorada por Marr e Hildreth (1980) que descreveram um método para determinar as bordas usando a detecção de cruzamentos por zeros ("zero-crossings"), que consiste em encontrar o ponto onde a segunda derivada cruza o zero entre valores positivos e negativos. Esta operação pode ser implementada aplicando-se o operador Laplaciano sobre uma imagem suavizada por um filtro Gaussiano (Laplaciano da Gaussiana) (Jain, 1989).

A detecção de linhas é um processo que exige a busca por *pixels* semelhantes que devem ser testados para determinar se são partes de uma linha comum. A definição de borda exige que os operadores de detecção busquem realçar ou captar as diferenças entre *pixels* vizinhos, no sentido de vizinhanças 4-conectadas e 8-conectadas (Gonzalez e Woods, 2000). Existem diferentes operadores descritos na literatura para detecção de bordas, sendo os mais simples sendo aqueles em que as bordas são detectadas por operadores baseados em gradientes (Sobel, Roberts, Laplaciano) seguidos por uma operação de limiarização (Gonzalez e Woods, 2000).

A seguir são apresentados os princípios envolvidos em alguns operadores baseados em gradientes citados na literatura. Um maior detalhamento é apresentado para o operador de Canny (1986) considerado como um dos melhores algoritmos de detecção de borda na literatura, e que será tomado como referência para efeitos de comparação dos operadores detectores baseados em redes neurais apresentados nesta dissertação.

3.4.3.1 Operador Laplaciano

Os métodos de detecção de bordas através da estimação de gradientes funcionam adequadamente em imagens que possuem transições abruptas de níveis de cinza, semelhantes a uma função degrau. A região de transição pode ser ampla, como mostrado na Figura 3.3 (b). A localização da borda pode ser estimada tomando-se a primeira derivada parcial. Com a segunda derivada parcial, a borda pode ser estimada como sendo o local de cruzamento por zero da segunda derivada (parte inferior da Figura 3.3). Assim, pode-se afirmar que as primeiras derivadas avaliam as bordas em uma imagem, enquanto as segundas derivadas permitem a sua localização (Jain, 1989).

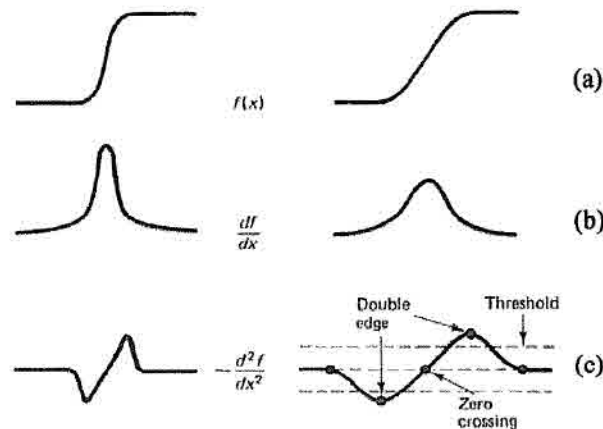


FIGURA 3.3 – (b) Primeira e (c) Segunda derivada para detecção de borda
 FONTE: Jain (1989 p. 352)

Como comentado anteriormente, o operador Laplaciano (Gonzalez e Woods, 2000) implementa um método que estima a segunda derivada, calculando localmente a intensidade da borda através da equação 3.1, na qual, f representa a imagem e x e y são a coordenadas da linha e do pixel na imagem, respectivamente.

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (3.1)$$

A operação da equação 3.1 deve ser efetuada para cada pixel da imagem. Isso pode ser feito por uma operação de convolução da imagem com a máscara da Figura 3.4 (Gonzalez e Woods, 2000).

0	-1	0
-1	4	-1
0	-1	0

FIGURA 3.4 – Exemplo de máscara que implementa o operador Laplaciano

Esta máscara realça os pontos de contorno que correspondem aos cruzamento por zero, encontrados pelo operador Laplaciano da equação 3.1. Uma operação de limiarização sobre a norma do gradiente é necessária para eliminar os contornos correspondentes ao ruído. A Figura 3.5 mostra o resultado da aplicação do operador laplaciano em duas imagens, uma em níveis de cinza e outra RGB.

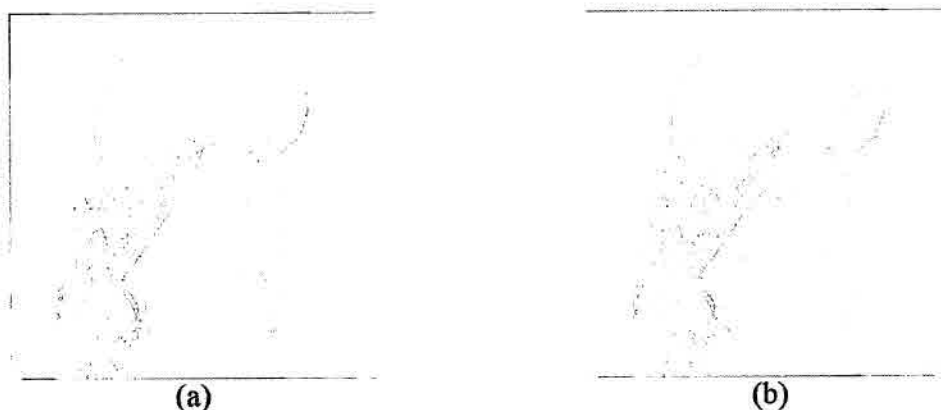


FIGURA 3.5 – Operador Laplaciano: (a) Lena NC, (b) Lena RGB.

3.4.3.2 Operador de Roberts

O operador de Roberts (1965) é o mais antigo e simples dos algoritmos de detecção de bordas, que se baseia em uma operação de vizinhança em direções diagonais. Assim, para um ponto $f(x,y)$ na imagem são calculados os gradientes.

$$G_x = f(x+1,y) - f(x,y+1) \quad (3.2)$$

$$G_y = f(x,y) - f(x+1,y+1) \quad (3.3)$$

Estas operações podem ser implementadas pela convolução da imagem com as máscaras mostradas na Figura 3.6.

G_x	G_y								
<table border="1" style="border-collapse: collapse;"> <tr><td style="padding: 5px;">0</td><td style="padding: 5px;">1</td></tr> <tr><td style="padding: 5px;">-1</td><td style="padding: 5px;">0</td></tr> </table>	0	1	-1	0	<table border="1" style="border-collapse: collapse;"> <tr><td style="padding: 5px;">1</td><td style="padding: 5px;">0</td></tr> <tr><td style="padding: 5px;">0</td><td style="padding: 5px;">-1</td></tr> </table>	1	0	0	-1
0	1								
-1	0								
1	0								
0	-1								

FIGURA 3.6 – Exemplos de Máscara que Implementa o Operador Roberts

Para determinar se o pixel avaliado é um pixel de borda, a norma do gradiente é calculada pela equação 3.4. Se a magnitude calculada é maior do que o menor valor de entrada (definido de acordo com a natureza e qualidade da imagem que está sendo processada), o pixel é considerado parte de uma borda.

$$|G| = \sqrt{G_y^2 + G_x^2} \quad (3.4)$$

A Figura 3.7 apresenta o resultado da aplicação do operador Roberts em duas imagens, uma em níveis de cinza e outra RGB.

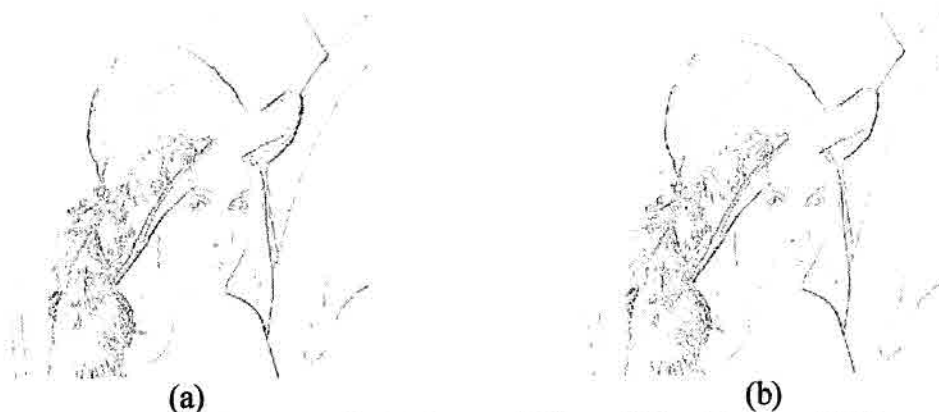


FIGURA 3.7 – Operador Roberts: (a) Lena NC e (b) Lena RGB.

3.4.3.3 Operador Prewitt

Os operadores de Prewitt detectam bordas com orientações pré-definida na horizontal e vertical. Os operadores estimam valor da borda pelas equações abaixo.

$$G_v = \begin{matrix} f(x+1,y-1) - f(x-1,y-1) + \\ f(x+1,y) - f(x-1,y) + \\ f(x+1,y+1) - f(x-1,y+1) + \end{matrix} \quad (3.5)$$

$$G_H = \begin{matrix} f(x-1,y-1) - f(x-1,y+1) + \\ f(x,y-1) - f(x,y+1) + \\ f(x+1,y-1) - f(x+1,y+1) + \end{matrix} \quad (3.6)$$

As operações das equações (3.5) e (3.6) podem ser implementados pela convolução da imagem com a Figura 3.8 (Gonzalez e Wood, 2000).

G _v		
-1	0	1
-1	0	1
-1	0	1

G _H		
1	1	1
0	0	0
-1	-1	-1

FIGURA 3.8 – Exemplos de Máscara que Implementa o Operador Prewitt

Os valores encontrados para G_v e G_H são analisados pelo cálculo da norma dada pela equação 3.7.

$$G = \sqrt{G_v^2 + G_H^2} \quad (3.7)$$

A Figura 3.9 apresenta o resultado da aplicação do Operador Prewitt em duas imagens, uma em níveis de cinza e outra RGB, após operações de limiarização.

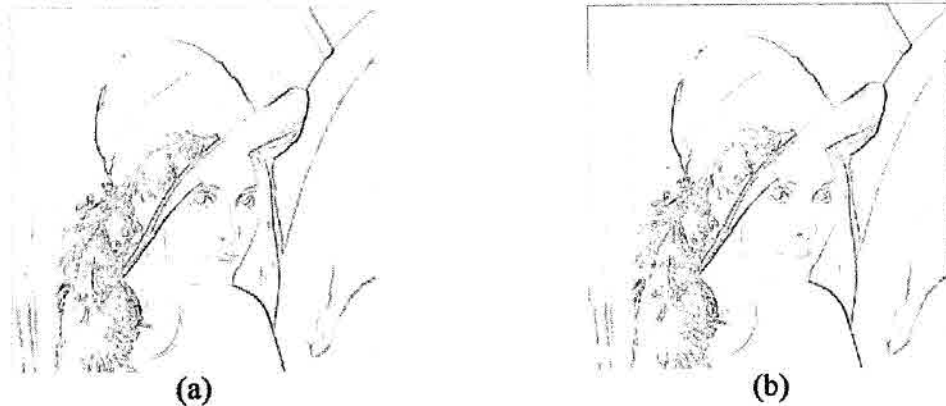


FIGURA 3.9 – Operador Prewitt: (a) Lena NC e (b) Lena RGB.

3.4.3.4 Operador de Sobel

O operador de Sobel considera que os *pixels* mais próximos do centro da máscara apresentam uma maior influência, no cálculo dos gradientes. As equações 3.8 e 3.9 implementam o operador de Sobel.

$$G_v = \begin{matrix} f(x-1,y-1) - f(x+1,y-1) + \\ f(x-1,y) - f(x+1,y) + \\ f(x-1,y+1) - f(x+1,y+1) \end{matrix} \quad (3.8)$$

$$G_H = \begin{matrix} f(x-1,y-1) - f(x-1,y+1) + \\ f(x,y-1) - f(x,y+1) + \\ f(x+1,y-1) - f(x+1,y+1) \end{matrix} \quad (3.9)$$

As operações das equações 3.8 e 3.9 podem ser executadas pelas máscaras da Figura 3.10 (Jain, 1989).

1	0	-1
2	0	-2
1	0	-1

1	2	1
0	0	0
-1	-2	-1

FIGURA 3.10 – Exemplos de Máscara que Implementa o Operador Sobel

As fórmulas empregadas para encontrar a norma do gradiente são a mesma do operador Roberts. A Figura 3.11 mostra o resultado obtido pelo operador Sobel, para duas imagens, uma em níveis de cinza e outra RGB.



FIGURA 3.11 – Operador Sobel: (a) Lena NC e (b) Lena RGB.

3.4.3.5 Operador Canny

Em 1986, John Canny definiu um conjunto de critérios para um detector de bordas ótimo, baseado nos trabalhos de Marr e Hildreth (1980) e desenvolveu um algoritmo que considera bordas do tipo degrau (“step edge”) sujeitas a ruído branco Guassiano. Um operador ótimo deve ser avaliado pelos seguintes critérios:

1. **Erro de localização:** Localiza e marcam todas as bordas realmente existentes e evita os pontos que não são bordas.
2. **Localização:** Minimiza a distância entre as bordas encontradas e as bordas reais;
3. **Resposta:** Identifica apenas um ponto de borda.

O operador de Canny realiza a detecção em alguns passos. O primeiro passo consiste em eliminar o efeito do possível ruído existente. Para isso, a imagem é suavizada com um filtro passa baixa do tipo Gaussiano, cuja equação característica é dada na equação (3.10).

$$G(x, y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)} \quad (3.10)$$

Após a filtragem para eliminação (ou minimização) do ruído, o operador concentra-se na etapa de encontrar bordas ideais. Esta fase é realizada pela convolução da função da imagem com operadores construídos pela primeira derivada de operadores gaussianos em 1D. A imagem é convolvida nos sentidos das colunas (pixels) e das linhas da imagem. A equação (3.11) mostra a função Gaussiana unidimensional usada para implementar o operador para realizar a convolução.

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (3.11)$$

Na equação (3.11), σ é o desvio padrão e x representa a coordenada do ponto.

A primeira derivada na equação (3.12) utilizada para realçar as bordas, mesmo na presença de ruído.

$$G'(x) = \frac{-x}{\sqrt{2\pi}\sigma^3} e^{-\frac{x^2}{2\sigma^2}} \quad (3.12)$$

As imagens resultantes das operações de convolução do operador 1D (Equação 3.12), nas direções das linhas e das colunas (pixels) da imagem, são então combinadas resultando no cálculo da norma do gradiente em cada ponto. A norma do gradiente em cada ponto (x,y) é dada pela equação (3.13).

$$N(x,y) = \sqrt{G_x^2 + G_y^2} \quad (3.13)$$

Após o cálculo da norma dos gradientes em todos os pontos da imagem, o algoritmo localiza os pontos de bordas como aqueles que apresentam valores de intensidade máxima, ou, de maneira inversa, elimina os pontos de intensidade mínima.

O passo final do algoritmo consiste no processo de limiarização, uma vez que a imagem de bordas resultante apresenta valores diferentes de intensidade de gradientes. No algoritmo de Canny utiliza-se um processo de limiarização por histerese que consiste na determinação de dois valores de limiares: um limiar inferior e um superior. Os pixels

situados acima do limiar superior são considerados como pontos de bordas e aqueles situados abaixo do limiar inferior são descartados. Os demais pixels são considerados bordas se estiverem nas proximidades de um pixel de borda.

A convolução relativamente simples de ser implementada tem um alto custo computacional. Os esforços computacionais do detector de bordas de Canny são altos e os resultados são, geralmente, pós-processados para maior clareza. Entretanto, o algoritmo é eficiente no processamento de imagens com ruídos ou com bordas difusas.

Outros detalhes desse operador podem ser encontrados em Canny (1986).

3.5 MÉTODOS DE VISÃO COMPUTACIONAL

Muitos algoritmos e métodos de visão computacional e processamento de imagens têm sido utilizados como ferramentas na tarefa de extração de características de imagens em várias áreas científicas e de aplicações, como por exemplo, medicina, sensoriamento remoto, robótica, cartografia, etc.

A análise de imagens em diferentes áreas de aplicação requer uma fase de pré-processamento para segmentação de imagens para a qual diferentes abordagens podem ser empregadas. No escopo desta dissertação, enfatiza-se as técnicas de segmentação relacionadas com a detecção de bordas. Na Tabela 3.1 tem-se um resumo de alguns trabalhos abordando o problema de detecção de bordas. É possível encontrar tabelas mais completas em alguns trabalhos na literatura, como por exemplo, em Heath et al. (1998). Na Tabela 3.1 encontram-se alguns trabalhos que utilizaram o operador Canny como detector de bordas, em imagens em níveis de cinza (NC). A idéia da utilização dessa tabela é apenas mostrar os métodos e os tipos de imagens utilizados pelos autores.

Nesta dissertação, entretanto, o operador de Canny é enfatizado por ser considerado um operador padrão (Shin et al. 1994) e por ser tomado como operador de referência.

TABELA 3.1 – MÉTODOS DE DETECÇÃO DE BORDAS.

Autores	Ano	Métodos	Imagem	
Ding, L. e Goshtasby, A.	2001	Canny, Minor e Canny Modificado	Quadros e Texto impresso	NC ¹
Neves, S. C. M. e Pelaes, E. G.	2001	Gradiente, Sobel, Roberts e Laplaciano	Lena	NC
Heath et. al.	1998	Canny, Nalwa-Binford, Sarkar-Boyer e Sobel	Câmera, cafeteira, flor, avião, cone, escada, pasta e lixeira.	NC
Cho e Cho	1994	Operador Fuzzy	Lena, peppers, casa.	NC
McIvor, A. M.	1993	Canny.	Bloco	NC
Canny, J.	1986	Operador Circular	Robô Dalek	NC

Um outro operador apresentado na Tabela 3.1 é o operador nebuloso (Cho e Cho, 1994) que utiliza inferência nebulosa para tomada de decisão sobre a existência de bordas em uma região analisada de uma imagem.

¹ As siglas na Tabela 3.1 podem ser encontradas no início da dissertação em siglas e abreviaturas.

CAPÍTULO 4

DETECÇÃO DE BORDAS UTILIZANDO REDES NEURAIIS ARTIFICIAIS

Este capítulo descreve alguns aspectos da implementação do uso de redes neurais na detecção de bordas, como proposto nesta dissertação, procurando enfatizar a metodologia para o treinamento das redes neurais. É apresentada também uma revisão bibliográfica de trabalhos que usam redes neurais na detecção de bordas.

4.1 REDES NEURAIIS EM DETECÇÃO DE BORDAS

As redes neurais artificiais (RNA), por serem inspiradas no funcionamento do cérebro, possuem características muito úteis no reconhecimento de padrões. Através de um processo de treinamento, as redes neurais passam a ser capazes de reconhecer padrões, mesmo que os dados utilizados nesse treinamento sejam não-lineares, incompletos ou até mesmo contraditórios; a habilidade de manipular estes dados imprecisos faz com que as redes neurais sejam extremamente eficazes em tarefas onde especialistas não estão à disposição, ou mesmo quando um conjunto de regras não pode ser facilmente formulado.

Já há algum tempo as redes neurais artificiais vêm sendo usadas para tarefas tradicionais de processamento de imagens (Kulkarni, 2001). A implementação de um sistema de visão artificial, baseado em redes neurais, faz uso das características intrínsecas das redes, sendo capaz de extrair padrões de conjuntos de dados complexos. Vários trabalhos as têm empregado para fazer segmentação de imagens, como pode ser observado na Tabela 4.1.

TABELA 4.1 – DETECÇÃO DE BORDAS POR RNA¹

Autores	Ano	RNA	Função	Aprendizagem	Imagem	Tipo
Chai et al.	2001	Competitiva	Gaussiana	Não-supervisionado	Médicas (Tumores)	NC
Wong et al.	2000	MBNN	-	Não-supervisionado	Águia, flor e cenas naturais.	NC
Bittencourt et al.	2000	RPMC	-	Supervisionado	Igreja	NC
Davis et al.	1999	RPMC Counter propagation SOFM BAM	Sigmóide	Supervisionado Não-supervisionado Não-supervisionado Supervisionado	Raio X.	NC
Ribeiro e Lima	1999	RPMC	Sigmóide	Supervisionado	Fotos áreas de tráfegos	NC
Lindblad et al.	1997	PCNN FPCNN	-	Supervisionado	Aviões	NC
Young et al.	1997	Hopfield	Degrau	Não-supervisionado	Objetos, chaves.	NC
Campadelli	1997	Hopfield	Sigmóide	Não-supervisionado	Flat bed	RGB
Ridder et al.	1996	RPMC	Sigmóide	Supervisionado	Textos	NC
Prewer	1995	RPMC	Sigmóide	Supervisionado	Lena, casa e sintéticas.	NC
Basak et al.	1994	Celular	Rampa Sigmóide	Supervisionado	Satélites Faces.	NC

A Tabela 4.1 tem o objetivo de mostrar que o uso das redes neurais na tarefa de detecção de bordas tem sido explorado com o decorrer dos anos. Entretanto, podemos observar que apenas alguns modelos de redes são utilizados para detecção de bordas, sendo a rede de perceptrons de múltiplas camadas (PMC) a mais utilizada, como mostra a tabela 4.1. Muitas redes neurais também são utilizadas para segmentação de imagens [(Ahmed e Farag, 1997), (Chen e Wang, 2002), (Verikas et al, 1997), (Goltsev, 1995)], classificação de imagens [(Blume et al, 1997), (Srinesavan e Kanal, 1997), (Reddick et al, 1997)], reconhecimento de padrões [(Silva, 1992), (Kohonen et al, 1988), (Kohonen, 1990)]. Ranganseri et al. (2001), utiliza a rede ART1 e ART2 para segmentação de imagens multi-espectrais.

¹As siglas na Tabela 4.1 podem ser encontradas no início da dissertação em Siglas e abreviaturas.

4.2 ASPECTOS DE IMPLEMENTAÇÃO DOS EXPERIMENTOS COM REDES NEURAIS

Nesta dissertação, descreve-se o uso de redes neurais como alternativa para realizar tarefas na área de processamento digital de imagens. O objetivo principal é desenvolver algoritmos de processamento de imagens que possuam características similares ao cérebro humano quando realizam tarefas de visão computacional e processamento de imagens.

O diagrama hierárquico da Figura 4.1 mostra a metodologia adotada para testar os diferentes modelos de redes neurais utilizados na tarefa de detecção de bordas em imagens. Os módulos principais são comentados nas subseções seguintes.

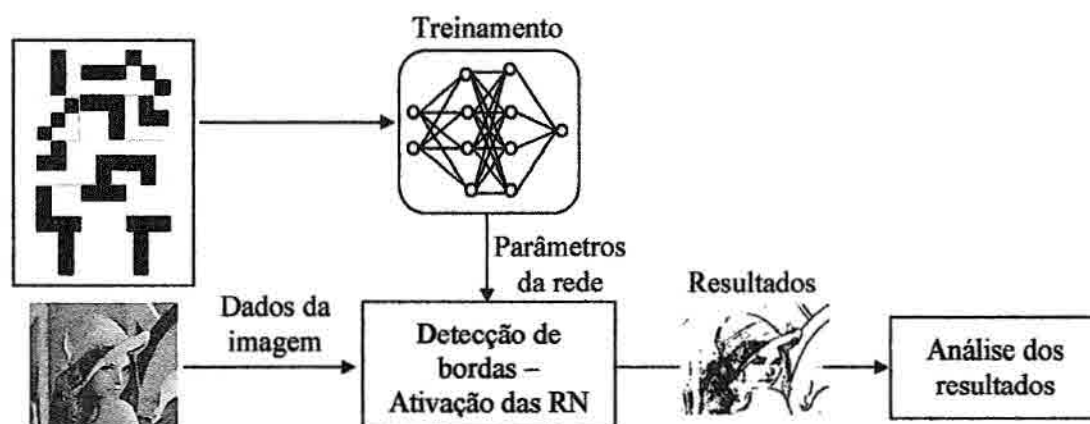


FIGURA 4.1 – Diagrama Hierárquico do Sistema

Os modelos de redes neurais utilizadas neste trabalho são: Perceptron de múltiplas Camadas (RPMC), Rede de Funções de Base Radial (RBF), Memória Associativa Bidirecional (BAM), Teoria da Ressonância Adaptativa (ART1 e ART2), Aprendizagem por Quantização Vetorial (LVQ1 e LVQ2), MaxNet e Memória Associativa Nebulosa (FAM), descritas no capítulo 2.

4.2.1 Padrões de Treinamentos

A abordagem para usar redes neurais na detecção de bordas utilizada nesta dissertação, consiste em treinar as redes neurais a reconhecer elementos de bordas (padrões) na imagem, assumindo que uma borda resulta da união de elementos básicos.

Assim, os padrões utilizados nos treinamentos das redes neurais são padrões considerados como possíveis elementos de bordas. A Figura 4.2 ilustra alguns exemplos de elementos de bordas, para uma janela de dimensões 3 x 3, considerados no treinamento das redes neurais nesta dissertação.

Foram experimentados dois conjuntos de elementos de bordas, um com 14 e outro com 26 padrões. Os padrões de treinamento, dependendo da rede, são codificados como elementos binários (0,1) ou bipolares (-1,1).

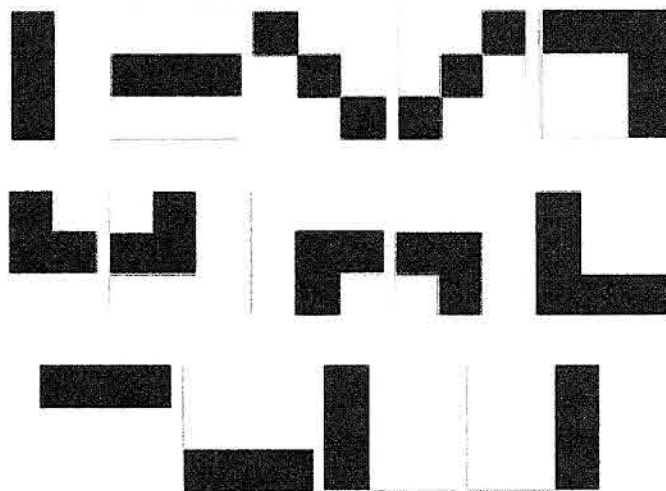


FIGURA 4.2 – Exemplos dos 14 Padrões de Bordas Usados no Treinamento das Redes Neurais.

A Figura 4.3 mostra 12 padrões adicionais que juntos com os padrões na Figura 4.2 formam o conjunto de 26 padrões de bordas.

A codificação dos padrões permite a identificação do elemento de borda pela análise da energia presente em uma janela da imagem (ao se usar padrões binários), e a implementação de filtros diferenciais (ao se usar padrões bipolares) como nos filtros de

Sobel, de Roberts e de Canny. O uso de janelas de dimensões maiores que 3x3 exige a redefinição dos elementos de bordas para o treinamento das redes.

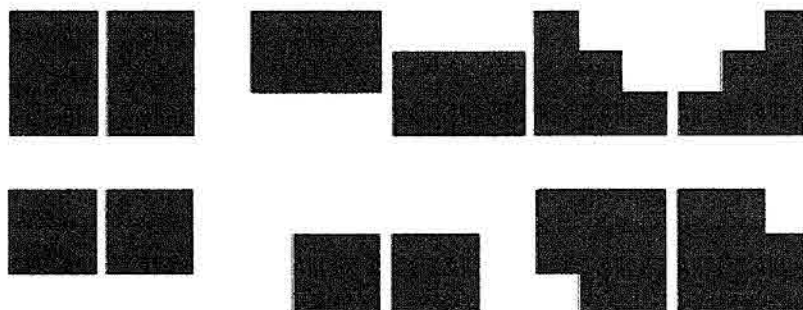


FIGURA 4.3 – Padrões de Bordas

4.2.2 Treinamento das Redes Neurais

No treinamento das redes neurais, a forma de inserção dos padrões de bordas apresentados considera as características de cada modelo de rede usado. Nove modelos de redes foram experimentados: RBF (Funções de Base Radial), RPMC (Rede perceptron de múltiplas camadas), ART1, ART2 (Teoria de Ressonância Adaptável 1 e 2), LVQ1, LVQ2 (Aprendizagem por Quantização Vetorial 1 e 2), FAM (Memória Associativa Nebulosa), BAM (Memória Associativa Bidirecional), MaxNet (Rede de Máximos).

A parametrização utilizada no treinamento depende das características próprias das redes, envolvendo diferentes variáveis de treinamento, assim são estabelecidos os seguintes parâmetros, para cada rede:

- Rede perceptron de múltiplas camadas (RPMC):
 - o número de camadas escondidas,
 - o número de neurônios em cada camada,
 - taxa de aprendizagem e
 - tipos de funções de transferência.
- Rede de Funções de Base Radial (RBF):
 - o número de neurônios na única camada escondida,
 - variação da função de ativação na última camada,
 - taxa de aprendizagem e
 - método para ajustar as funções de base radial existentes entre a camada de entrada e a camada escondida.

- Rede Memória Bidirecional Associativa (BAM):
 - n°. de neurônios na camada de entrada é igual ao n°. de neurônio na camada de saída, com conexões bidirecionais.
- Rede 1 da Teoria da Ressonância Adaptativa (ART1)
 - n°. de neurônios nas camadas da rede,
 - limiar de vigilância.
- Rede 2 da Teoria da Ressonância Adaptativa (ART2)
 - n°. de neurônios nas camadas da rede,
 - limiar de vigilância,
 - taxa de aprendizagem,
 - parâmetro de supressão de ruído,
 - nível de ativação do neurônio vencedor
- Rede Aprendizagem por Quantização Vetorial (LVQ1, LVQ2).
 - número de neurônios em cada camada,
 - taxa de aprendizagem.
- Rede MaxNet
 - n°. de neurônios na camada,
 - coeficiente de inibição lateral ($-\varepsilon$)
- Rede Memória Associativa Nebulosa (FAM)
 - n°. de neurônios na camada de entrada é igual ao n°. de neurônios na camada de saída.

Para alguns modelos de redes, os padrões são apresentados na sua entrada e um algoritmo de treinamento realiza a aprendizagem, como por exemplo, no caso da RPMC e da RBF. Para outros, os padrões são apresentados como sendo os próprios pesos, restringindo a fase de treinamento da rede à inserção dos padrões na matriz de pesos da rede, como por exemplo, no caso das redes ART1 e ART2.

A saída das redes após o treinamento depende do modelo de rede utilizado. Em alguns modelos as saídas estão associadas aos tipos de bordas. Em outros modelos a saída da rede é o próprio elemento de borda.

4.2.3 Imagens Utilizadas nos Experimentos

As imagens utilizadas nesta dissertação, são imagens em níveis de cinza e coloridas (RGB) com dimensões variando em 315×315 à 512×512 . As imagens escolhidas são as que estão presentes na literatura de visão computacional, por exemplo, a imagem da Lena [Neves e Peales (2001), Hsieh et al. (1997)], e a imagem *peppers* [Hsieh et al. (1997), Tsang e Tsang (1997)] (Figura 4.4).

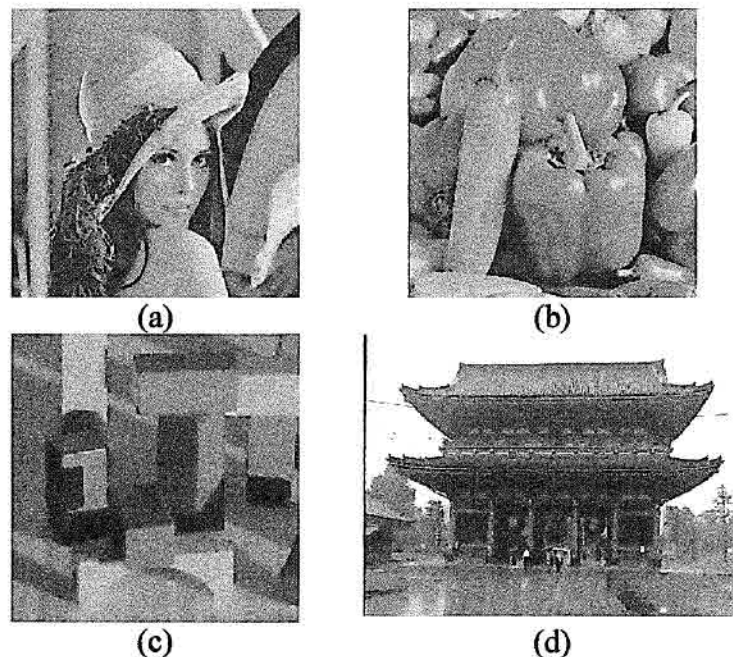


FIGURA 4.4 – Imagens RGB Utilizadas na Dissertação: (a) Lena RGB, (b) Peppers e (c) Blocos RGB, (d) Casa RGB.

Outros dados utilizados são imagens de pista reais encontradas na internet e a imagem da pista utilizada no artigo de Castro et al. (2001)(Figura 4.5).

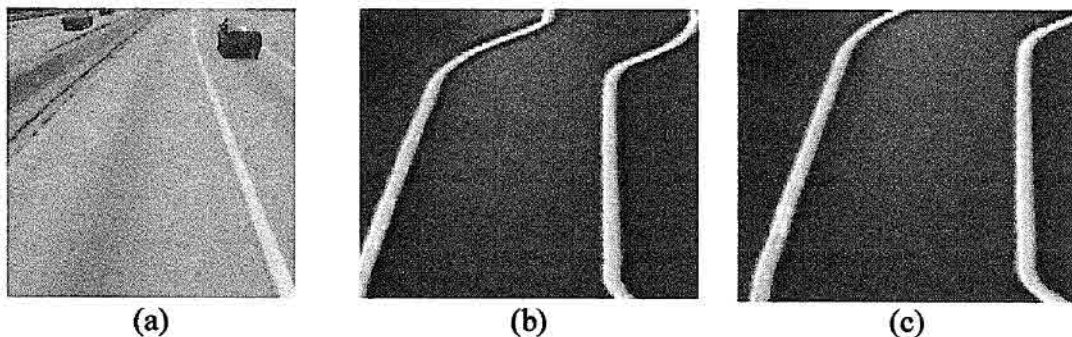


FIGURA 4.5 – Imagem de Pista (a) Real, (b) e (c) Experimento.

Fonte:(a)<http://vasc.ri.cmu.edu/idb/html/road/july4_92/run7a/index.html> e (b) e (c) Castro et al. (2001).

A Figura 4.6 mostra outras imagens em níveis de cinza utilizadas nos experimentos conduzidos nesta dissertação.

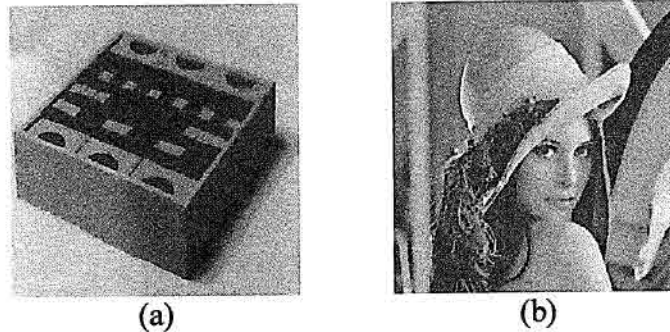


FIGURA 4.6 – Imagem em Níveis de Cinza (a) Blocos, (b) Lena.

A Figura 4.7.a mostra um trecho de uma imagem do satélite SPOT (Banda Pancromática) em níveis de cinza usada por Federov (2002) e a Figura 4.7.b mostra uma imagem multiespectral do satélite LANDSAT, contendo plantações usada por Gleriani (2002).

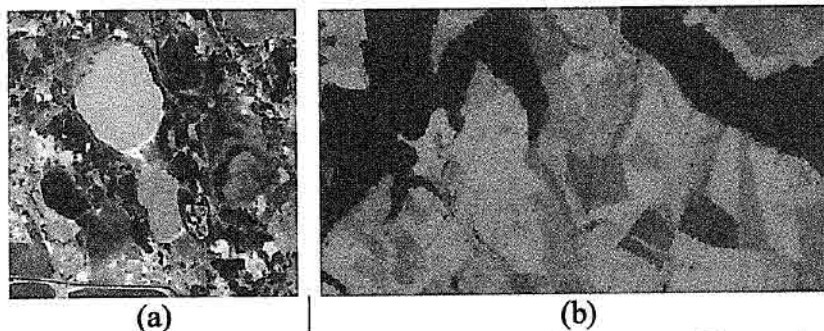


FIGURA 4.7 – (a) Imagem SPOTPAN e (b) Imagem Plantações
FONTE: Fedorov, 2002 e Gleriani (2002).

As imagens multiespectrais coletadas por satélites tais como, Landsat, SPOT ou similares, são extremamente valiosas para a extração dos dados destinados às várias aplicações de recursos naturais. A obtenção das informações espectrais registradas pelos sistemas nas diferentes partes do espectro eletromagnético, visando à identificação e discriminação dos alvos de interesse, depende principalmente da qualidade da representação dos dados contidos nas imagens (Fonseca, 2000). Por esses motivos, foram escolhidas algumas imagens de satélites (Figura 4.7) para testar as redes neurais.

4.2.4 Extração da Janela

No processo de detecção de bordas nas imagens nesta dissertação, utiliza-se uma janela deslizante 3×3 sobre a imagem principal, para extração de estruturas locais de bordas (Figura 4.8). Os dados da janela são normalizados antes de serem apresentados às redes. Dependendo do modelo de rede usado, quando um padrão de borda é detectado uma matriz binária equivalente ao padrão de borda detectado é associada a uma área de mesma dimensão numa imagem de saída, ou um número representando a identificação do padrão de borda encontrado é associado ao elemento central da janela.

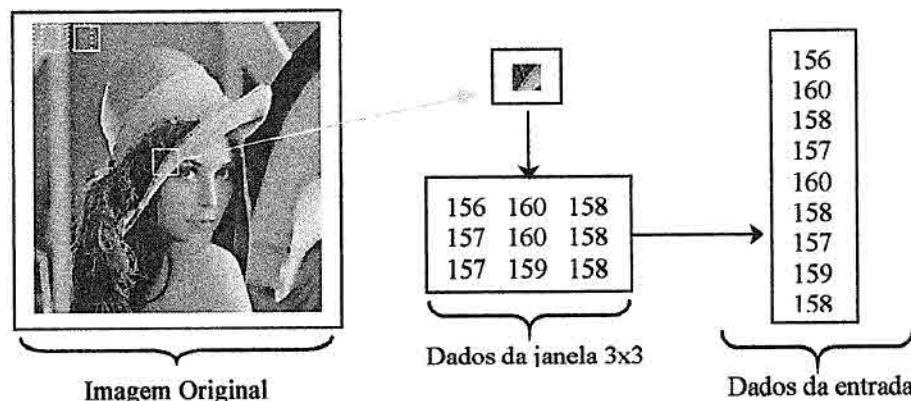


FIGURA 4.8 – Exemplo de Extração da Janela

A aplicação das redes neurais na janela da imagem é precedida pelo cálculo da variância, que indica a presença ou não de elementos de bordas. A variância é utilizada neste contexto como uma medida de contraste da janela, medindo a dispersão quadrática média dos pixels na janela. A Equação (4.1) calcula o valor do pixel médio na janela. A variância calculada pela Equação (4.2) é utilizada para uma pré-seleção das janelas da imagem que devem ser submetidas ao processamento pelas redes. A janela é selecionada se apresentar um valor de contraste superior a um determinado limiar. Se não for detectado um padrão em alguma janela, associa-se uma janela 3×3 de zeros à área equivalente da imagem de saída ou o número zero ao ponto central da janela, dependendo do modelo de rede.

Seja a média e a variância (var) dadas por:

$$media = \frac{\sum_{x=1}^n \sum_{y=1}^m f(x, y)}{n \times m} \quad (4.1)$$

$$Var = \frac{\sum_{x=1}^n \sum_{y=1}^m (f(x, y) - media)^2}{n \times m} \quad (4.2)$$

Nas Equações 4.1 e 4.2, f representa a imagem sendo processada, y é a coordenada da linha, x é a coordenada do pixel na linha, n é o número de linhas e m número de pixels por linha.

4.2.5 Ativação das Redes Neurais (Detecção Bordas)

Depois de treinadas, as redes são ativadas para detecção de bordas (Figura 4.9). Cada rede recebe como entrada a janela da imagem de onde se deseja extrair bordas. Os parâmetros para o funcionamento das redes são aqueles obtidos durante o treinamento. As imagens de bordas encontradas são binárias, enfatizando as áreas da imagem que apresentam contraste significativo, segundo um critério de decisão dependente dos objetivos da aplicação.

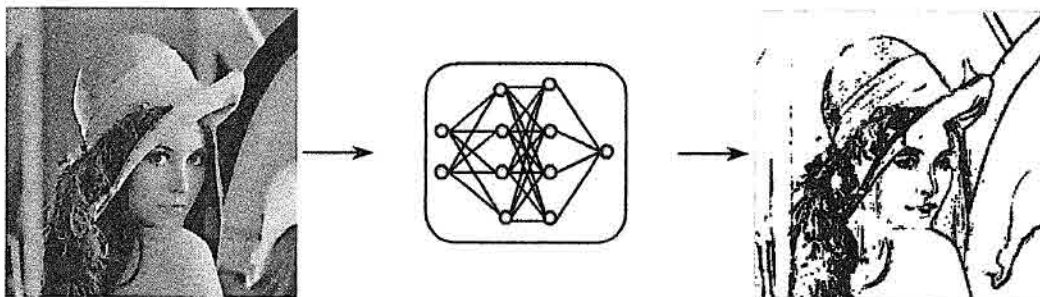


FIGURA 4.9 – Detecção de Bordas por RN

CAPÍTULO 5

ANÁLISE DOS RESULTADOS PARA OPERADOR NEURAL

Neste capítulo são apresentados os resultados obtidos na detecção de bordas utilizando técnicas de redes neurais artificiais, como proposto nesta dissertação. Os resultados obtidos são comparados aqueles produzidos pelo operador de Canny (1986) e pelo operador nebuloso de Cho et al. (1994).

5.1 ANÁLISE QUALITATIVA

Como mencionado no capítulo 4, as redes neurais foram treinadas com os padrões de bordas codificados em binários e bipolares. A detecção de bordas consiste na ativação da rede sobre janelas das imagens com dimensão 3 x 3. Os algoritmos utilizados foram adaptados para permitir a extração de bordas em imagens em níveis de cinza e em imagem RGB. Os mesmos conjuntos de imagens foram utilizados para ativação de todos os modelos de redes neurais.

O parâmetro limiar de variância, utilizado nas redes neurais, foi modificado para diferentes valores. Esse parâmetro evita a ativação das redes nas janelas que possuem informação de contraste não-significativa.

Os resultados apresentados consideram também a alteração do número de padrões utilizados no treinamento das redes neurais. Nas figuras deste capítulo são apresentados os resultados para treinamento das redes com 14 e com 26 padrões.

Em todos experimentos utilizando redes neurais, as imagens de bordas são imagens binárias ou com níveis associados ao tipo de elemento de borda encontrado na janela. Portanto, na visualização as imagens são escalonadas para permitir um bom realce dos resultados.

5.1.1 Resultado para Imagem da Lena em Níveis de Cinza

As Figuras 5.1(a)-(i) apresentam os resultados obtidos pelas redes neurais BAM, ART1, ART2, RPMC, RBF, LVQ1, LVQ2, FAM e Maxnet, respectivamente, sobre a imagem da Lena em níveis de cinza.. O valor do limiar igual a 100 foi encontrado de forma empírica. Este limiar significa que as janelas com variância abaixo de 100 são consideradas homogêneas, não apresentando, portanto informação de contraste significativo. Pode-se observar que a rede ART1 apresentou falhas em algumas bordas quando comparada às outras redes. Uma análise visual entre as imagens, mostra que as outras redes tiveram um desempenho qualitativo melhor, detectando bordas mais definidas (cheias).

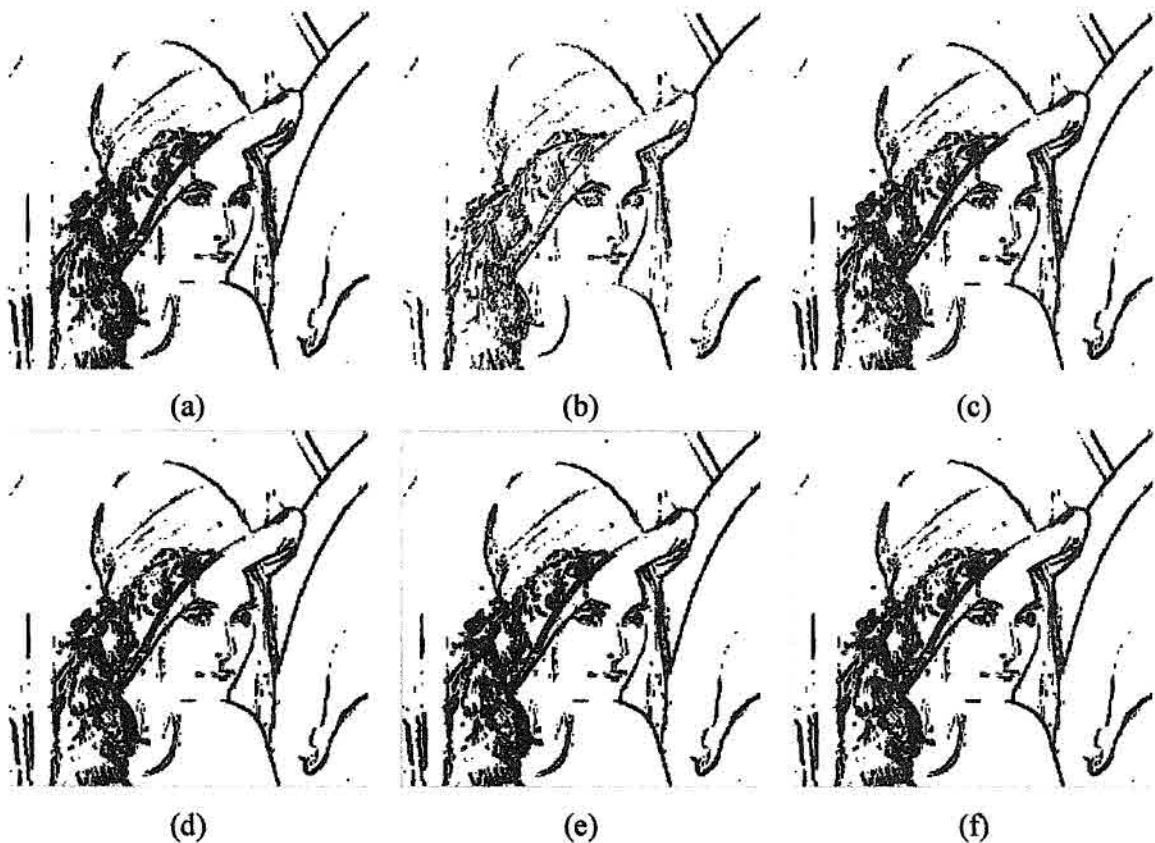


FIGURA 5.1 – Resultado das Redes Neurais treinadas com 14 padrões binários e com limiar da variância igual a 100: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.



FIGURA 5.1 – Conclusão.

Dos resultados da Figura 5.1 é possível ser que as redes localizaram as bordas mais salientes na imagem. As Figuras 5.2 (a)-(i) mostram os resultados das ativações das redes neurais, treinadas com 26 padrões binários, sobre a imagem da Lena em níveis de cinza.

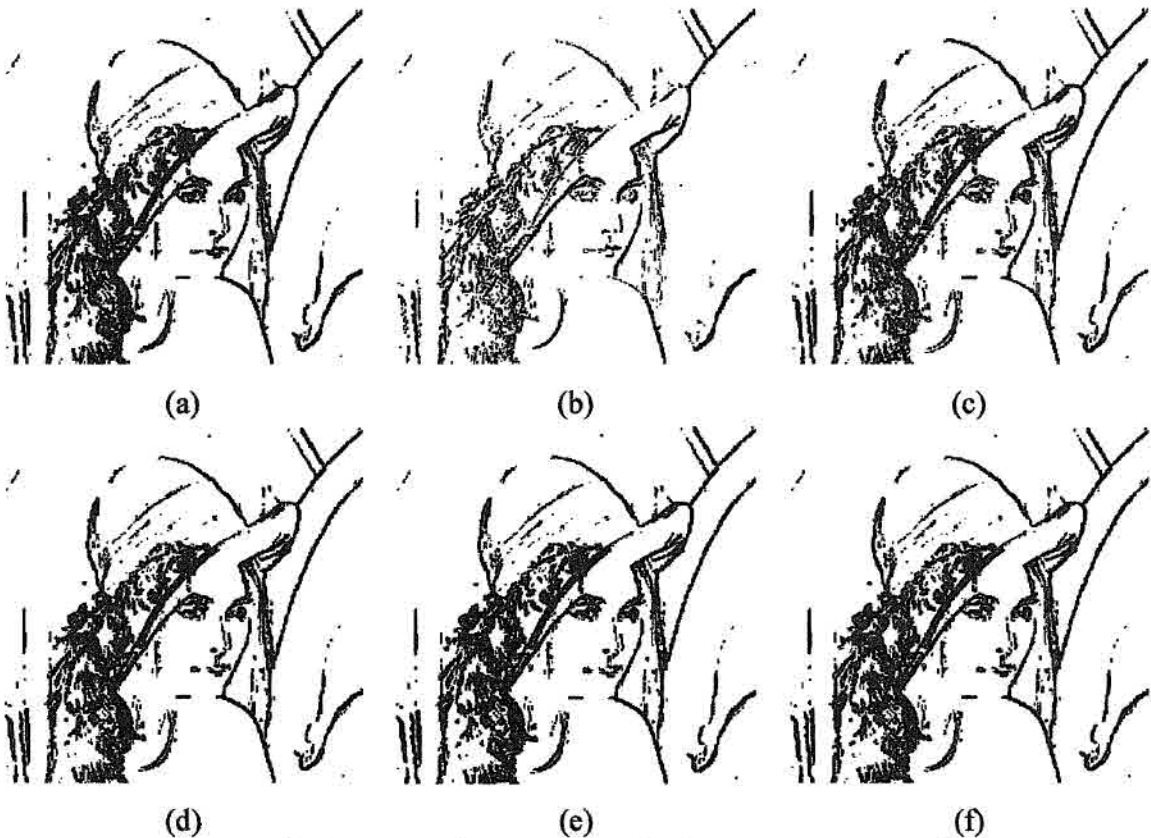


FIGURA 5.2 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 100: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.

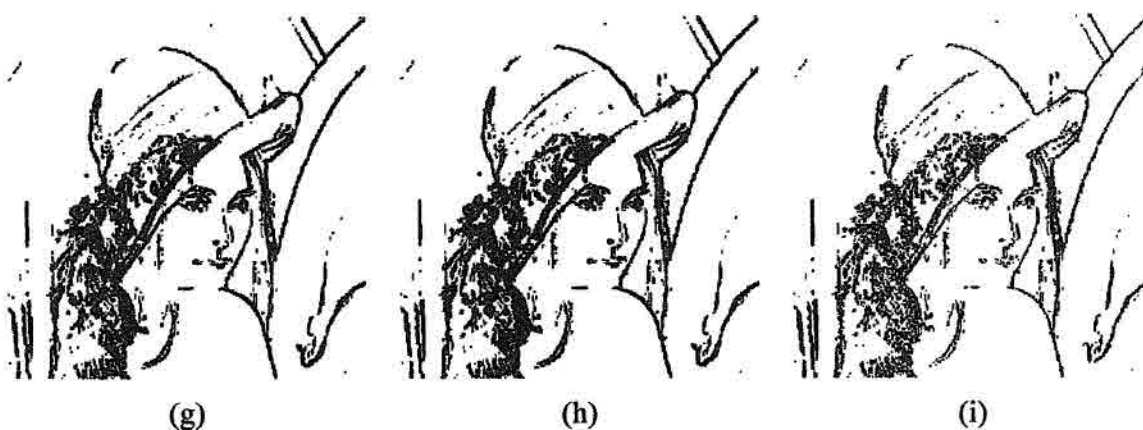


FIGURA 5.2 – Conclusão.

Comparando-se os resultados das Figuras 5.2(a)-(i) pode-se observar que o número de padrões apresentados à rede não produz modificações visíveis nas imagens de bordas obtidas na maioria das redes. A rede ART1, treinada com 26 padrões, apresentou uma densidade maior de bordas. Nas ativações das redes treinadas com 26 padrões de bordas foi utilizado o limiar de variância igual a 100, com o mesmo objetivo de evitar processamento em janelas com pouco contraste.

Com o objetivo de melhorar os resultados obtidos mostrados nas Figuras 5.1 e 5.2 as redes (treinadas com 26 padrões) foram ativadas alterando-se o valor do limiar de variância para 50.

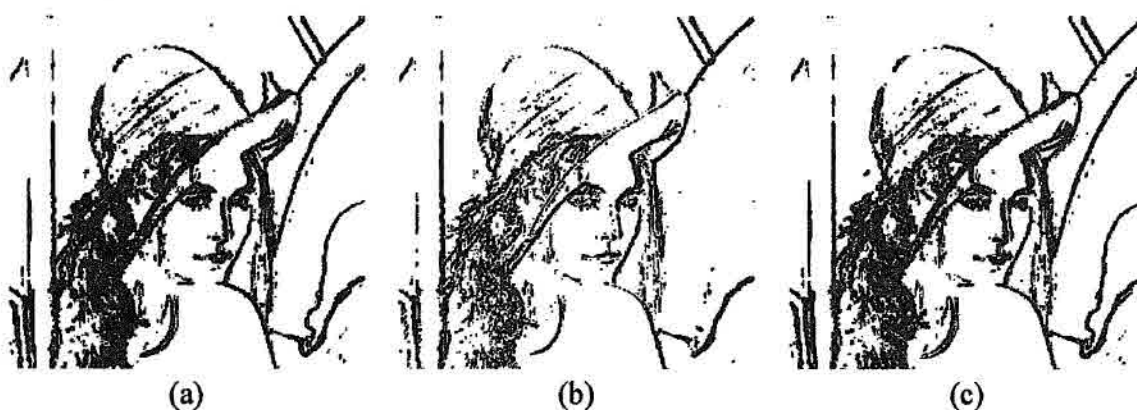


FIGURA 5.3 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 50 para a imagem Lena NC: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.

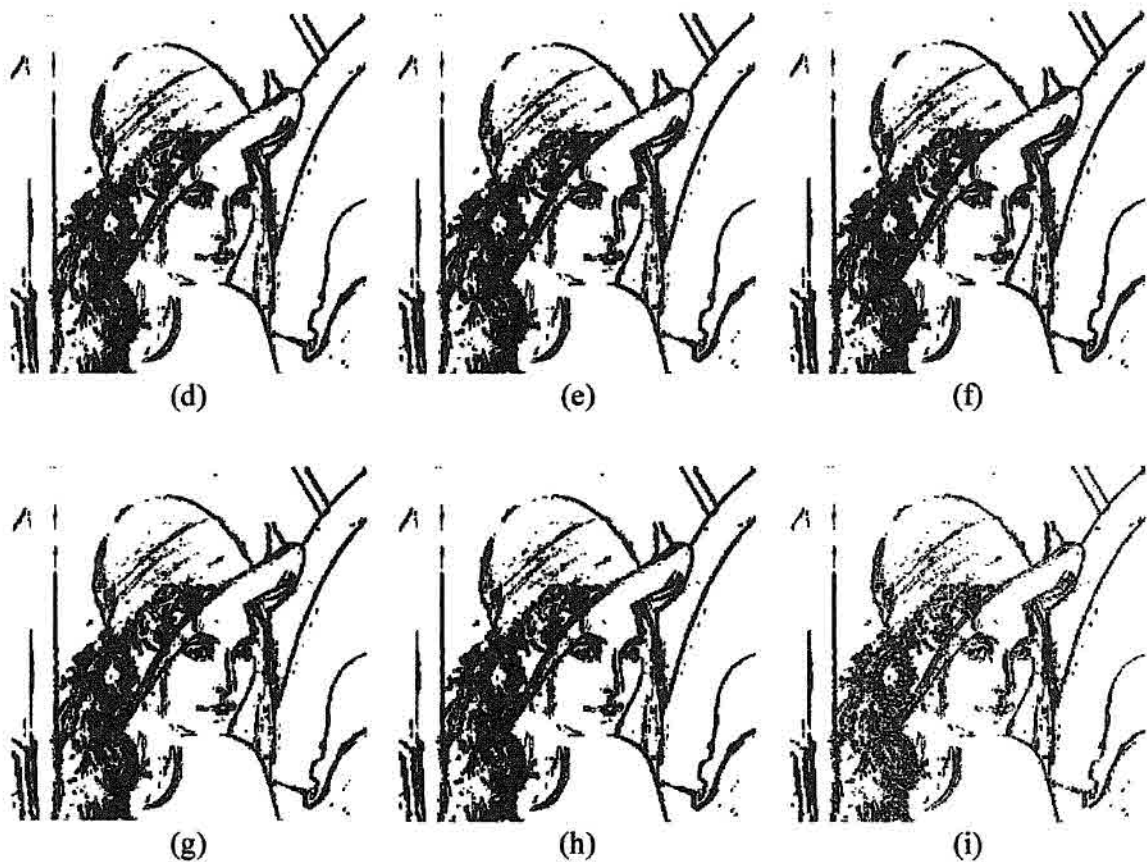


FIGURA 5.3 – Conclusão.

Nas Figuras 5.3(a)-(i) pode-se observar que há influência no resultado devido à mudança no parâmetro limiar de variância, resultando em um aumento no número de pontos de bordas encontrados.

Nas Figuras 5.4(a)-(i) têm-se os resultados obtidos com a ativação das redes treinadas com 26 padrões de bordas bipolares, para um limiar de variância igual a 50. A utilização de padrões bipolares, objetiva a comparação visual com os resultados obtidos com as redes treinadas com padrões binários.

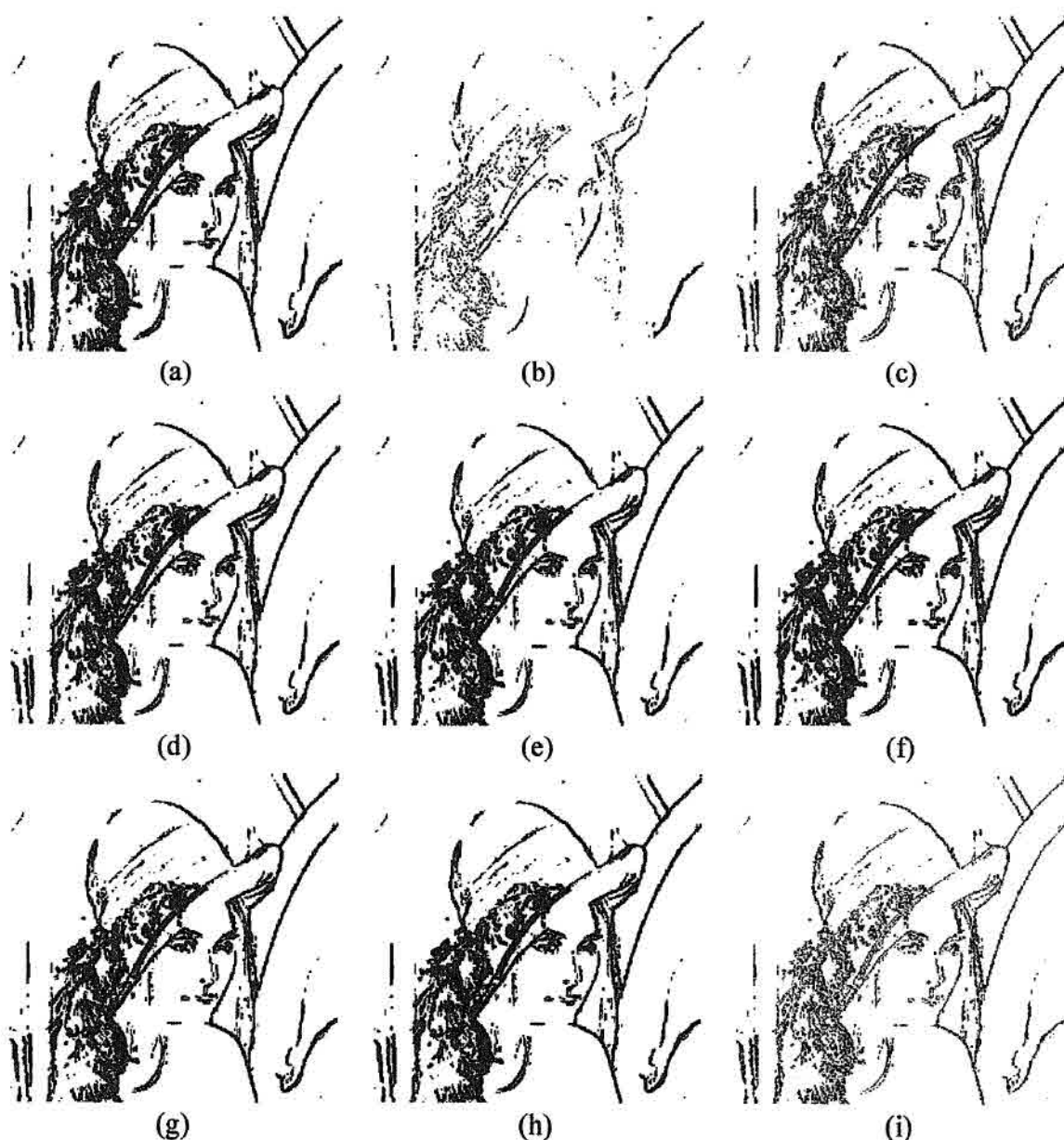


FIGURA 5.4 – Resultado das Redes Neurais treinadas com 26 padrões bipolares e com limiar da variância igual a 50 para a imagem Lena NC: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.

Pode-se observar através das Figuras 5.4(a)-(i) que o uso de padrões bipolares não resultou em mudanças significativas nas imagens resultantes, quando comparados aos resultados das Figuras 5.2(a)-(i). Como não houve mudanças significativas nos resultados, os testes realizados com os 14 e 26 padrões bipolares não são apresentados.

Para verificar a influência e a importância do limiar de variância, foram realizados experimentos com diferentes valores de limiar de variância. A Figura 5.5 mostra esses resultados para imagem da Lena em níveis de cinza usando a rede LVQ 1.

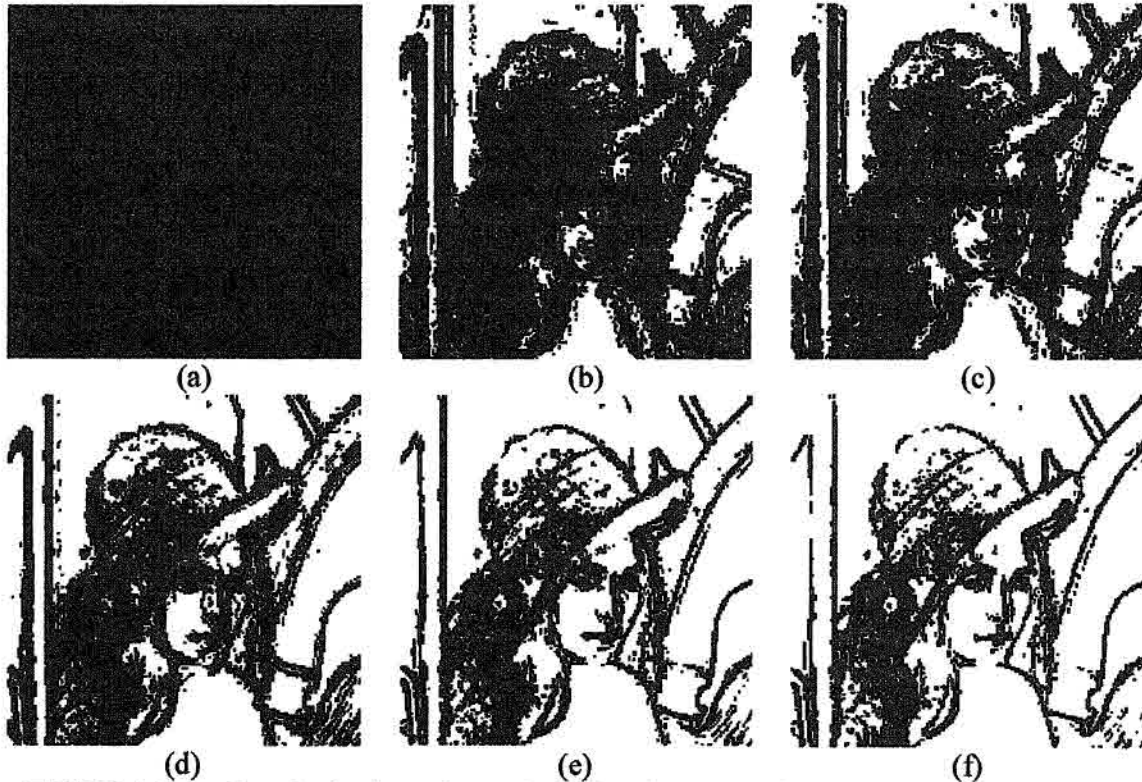


FIGURA 5.5 – Resultado da variação do limiar de Variância: (a) LV = 1, (b) LV = 2, (c) LV = 3, (d) LV = 5, (e) LV = 15, (f) LV = 25.

Podemos observar que valores altos de limiar evitam o processamento em janelas com pouco contraste, e com isso obtêm-se resultados melhores. É possível verificar que a escolha do limiar tem influência sobre o resultado final. Quando a imagem é exposta a um limiar mais baixo, surgem bordas mais densas e com isso mais ruídos.

5.1.2 Resultado para Imagem Bloco em Níveis de Cinza

As Figuras 5.6(a)-(i) apresentam os resultados obtidos pelas redes neurais BAM, ART1, ART2, RPMC, RBF, LVQ1, LVQ2, FAM e Maxnet, respectivamente, treinadas com os 14 padrões de bordas. O valor do limiar igual a 100 foi encontrado de forma empírica. Pode-se observar que a rede ART1 apresentou falhas em algumas bordas quando comparada às outras redes. A análise visual mostra que outras redes tiveram um desempenho qualitativo melhor, detectando bordas mais definidas (cheias).



FIGURA 5.6 – Resultado das Redes Neurais treinadas com 14 padrões binários e com limiar da variância igual a 100 para a imagem bloco em níveis de cinza: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.

As Figuras 5.7 (a)-(i) mostram os resultados das ativações das redes neurais, treinadas com 26 padrões binários.

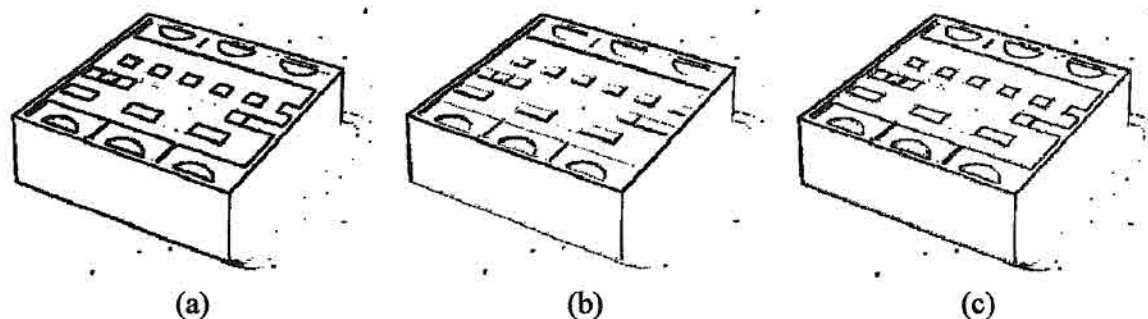


FIGURA 5.7 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 100: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.

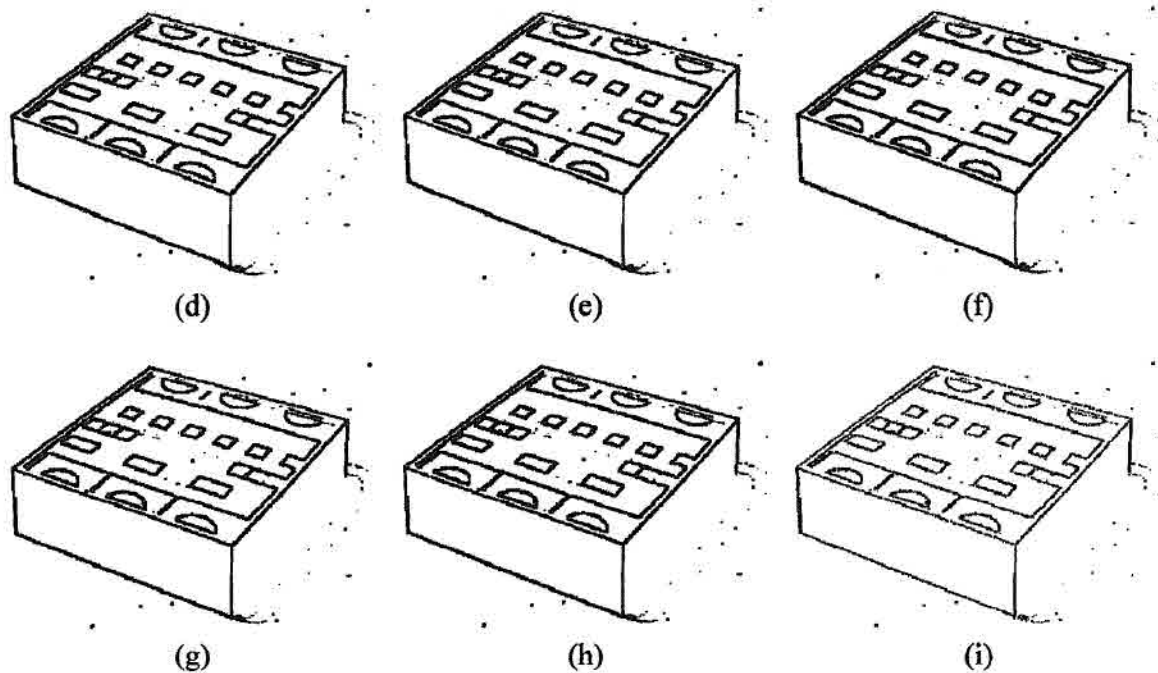


FIGURA 5.7 – Conclusão.

Comparando-se os resultados das Figuras 5.6(a)-(i) e 5.7(a)-(i) pode-se observar que o número de padrões apresentados à rede não produziu modificações visíveis nas imagens de bordas obtidas na maioria das redes. A rede ART1, treinada com 26 padrões, apresentou uma densidade maior de bordas. Nas ativações das redes treinadas com 26 padrões de bordas foi utilizado o limiar de variância igual a 100, com o mesmo objetivo de evitar processamento em janelas com pouco contraste.

Novos experimentos foram realizados para um valor do limiar de variância igual a 50.

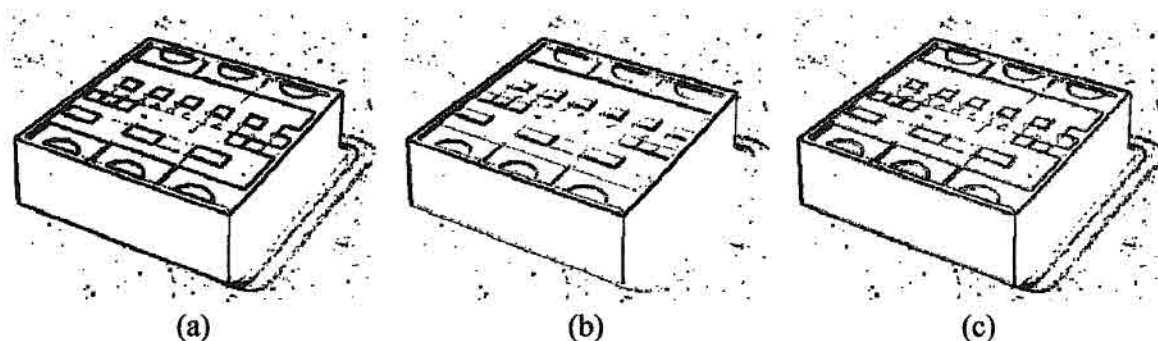


FIGURA 5.8 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 50 para a imagem bloco NC: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.

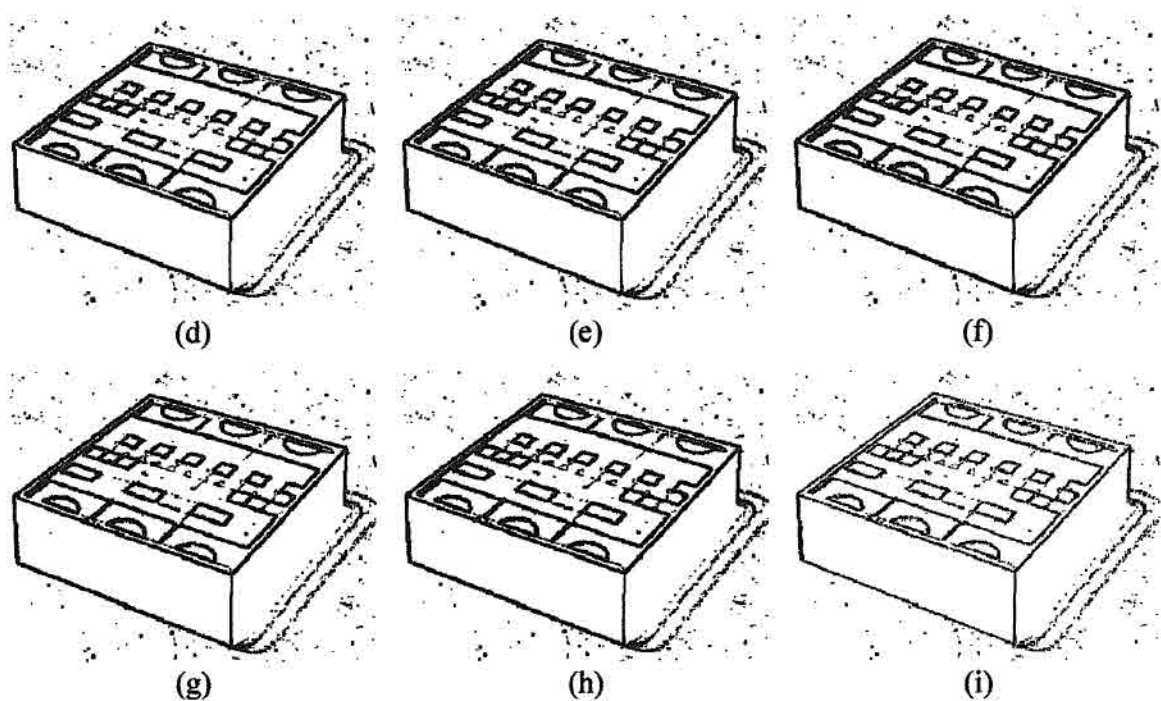


FIGURA 5.8 – Conclusão.

Nas Figuras 5.8(a)-(i) pode-se observar que há influência no resultado com a mudança do limiar de variância, resultando em um aumento no número de elementos de bordas encontrados.

5.1.3 Resultado para Imagem Pista em Níveis de Cinza

As Figuras 5.9(a)-(i) apresentam os resultados obtidos pelas redes neurais BAM, ART1, ART2, RPMC, RBF, LVQ1, LVQ2, FAM e Maxnet, respectivamente, treinadas com 14 padrões. O valor do limiar de variância utilizado é igual a 100. Também pode-se observar que a rede ART1 apresentou falhas em algumas bordas.

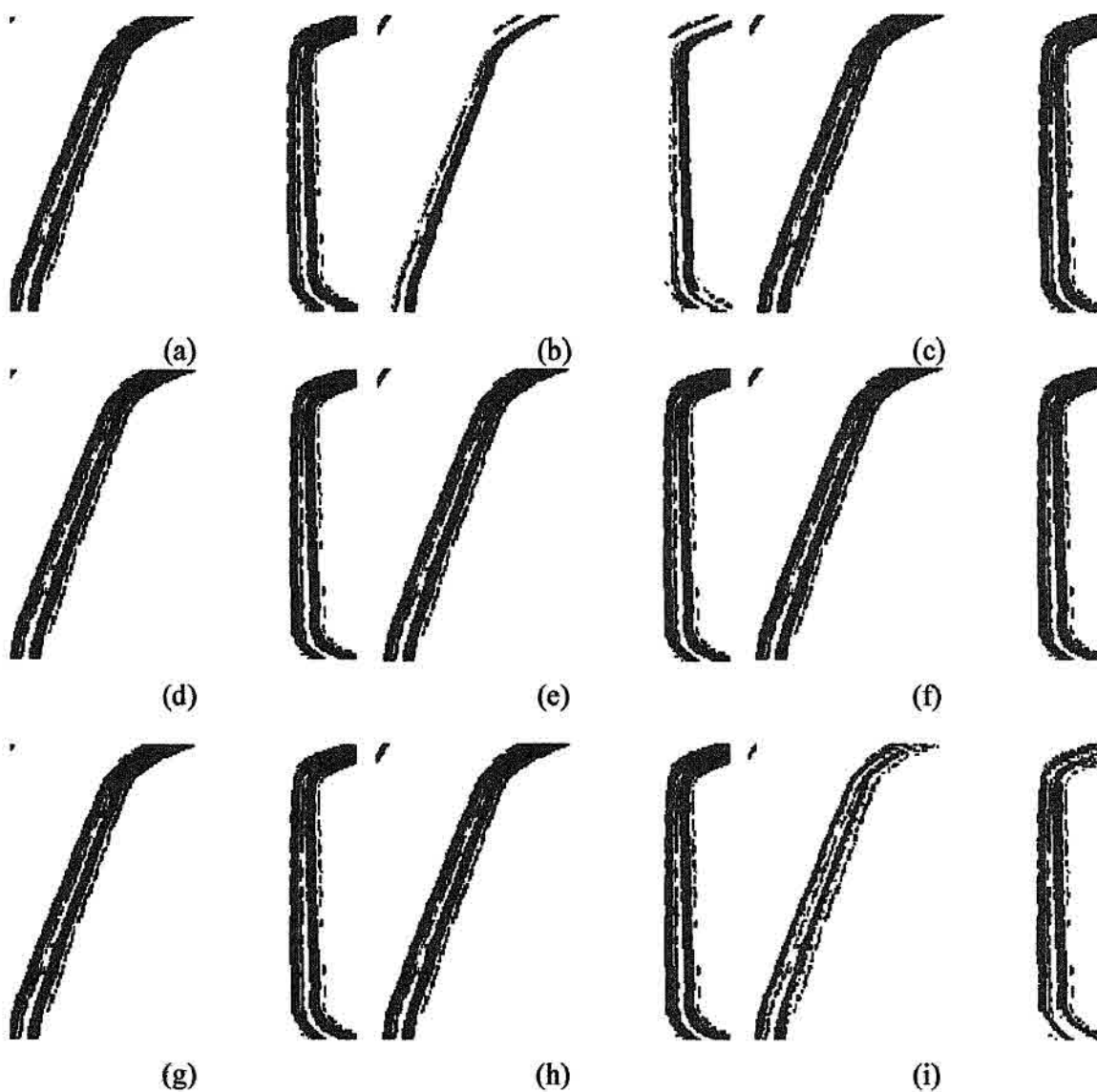


FIGURA 5.9 – Resultado das Redes Neurais treinadas com 14 padrões binários e com limiar da variância igual a 100: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.

Também como nas outras imagens, vê-se uma comparação visual com as imagens originais mostra que as redes localizaram as bordas mais salientes nas imagens. As Figuras 5.10 (a)-(i) mostram os resultados das ativações das redes neurais, treinadas com 26 padrões binários.

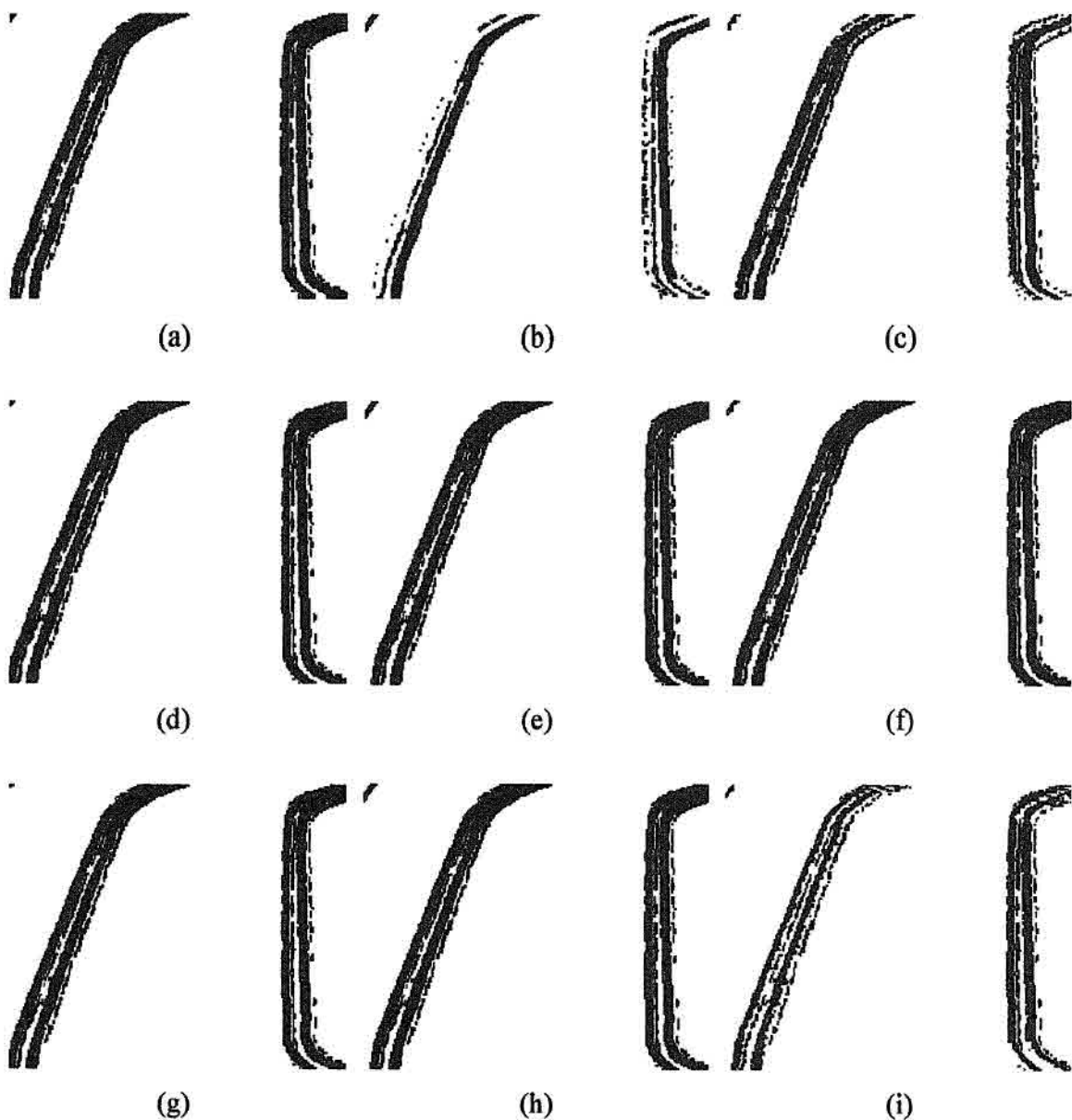


FIGURA 5.10 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 100 para a imagem pista em níveis de cinza: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.

Comparando-se os resultados das Figuras 5.9(a)-(i) observar-se que o número de padrões apresentados à rede não produziu modificações significativas nas imagens de bordas obtidas na maioria das redes. Também nas ativações das redes treinadas com 26 padrões de bordas, utilizo-se o limiar de variância igual a 100.

Os resultados mostrados nas Figuras 5.9 e 5.10, foram conseguidos utilizando-se um valor do limiar de variância igual a 50.

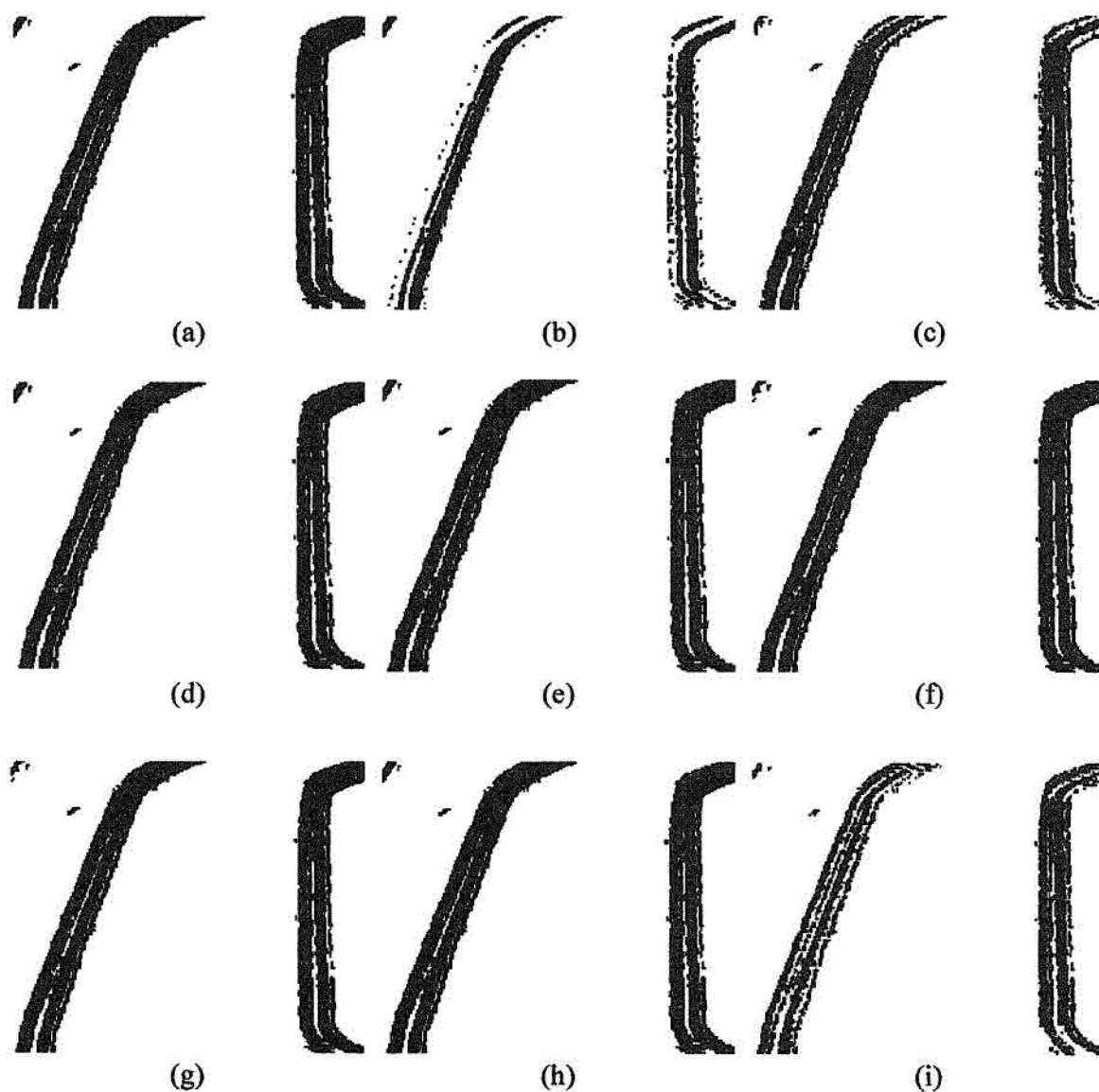


FIGURA 5.11 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 50 para a imagem pista NC: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.

Das Figuras 5.11(a)-(i) observa-se que há influência nos resultados com a mudança no parâmetro limiar de variância, resultando em um aumento no número de elementos de bordas encontrados. Como já foi observado anteriormente, o valor do limiar adequado apresenta uma melhora no resultado da rede. Apesar do bom desempenho, a escolha do limiar é uma tarefa difícil, pois depende da natureza das imagens.

5.1.4 Resultado para Imagem do SPOT em Níveis de Cinza

As Figuras 5.12(a)-(i) apresentam os resultados obtidos pelas redes neurais BAM, ART1, ART2, RPMC, RBF, LVQ1, LVQ2, FAM e Maxnet, respectivamente, treinadas com 14 padrões. Utilizando um limiar de variância igual a 100, portanto informação de contraste significativo. Pode-se observar que a rede ART1 apresentou falhas em algumas bordas é possível ver que as redes não tiveram um bom desempenho sobre a imagem utilizada. As redes conseguiram detectar apenas bordas bem salientes.

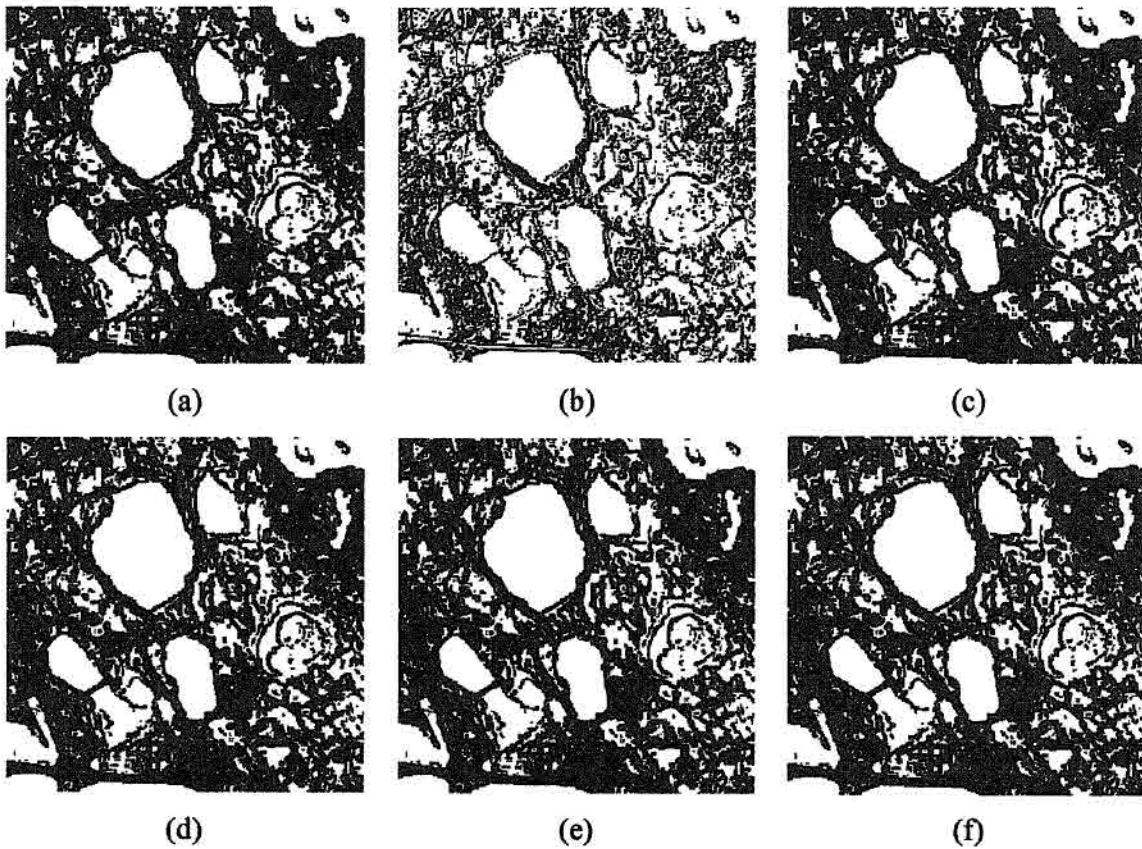


FIGURA 5.12 – Resultado das Redes Neurais treinadas com 14 padrões binários e com limiar da variância igual a 100: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.

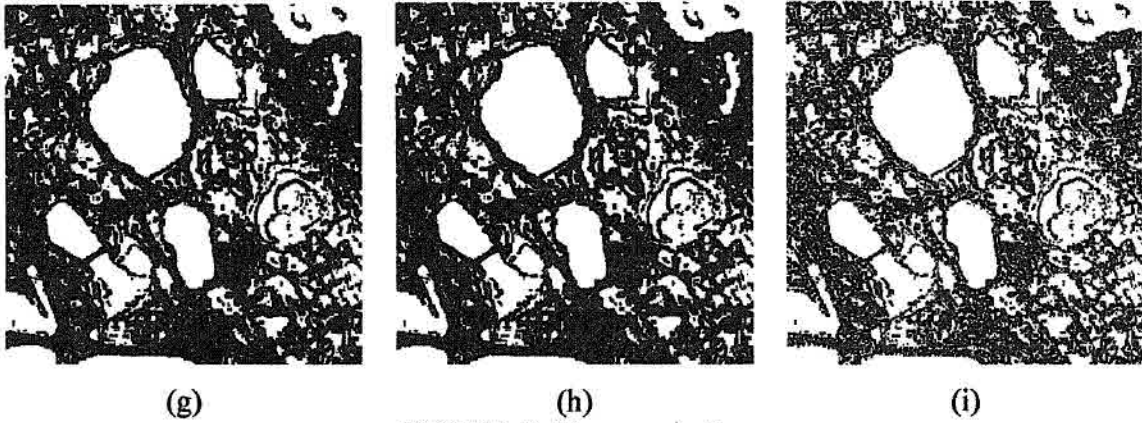


FIGURA 5.12 – conclusão.

As Figuras 5.13 (a)-(i) mostram os resultados das ativações das redes neurais, treinadas com 26 padrões binários.

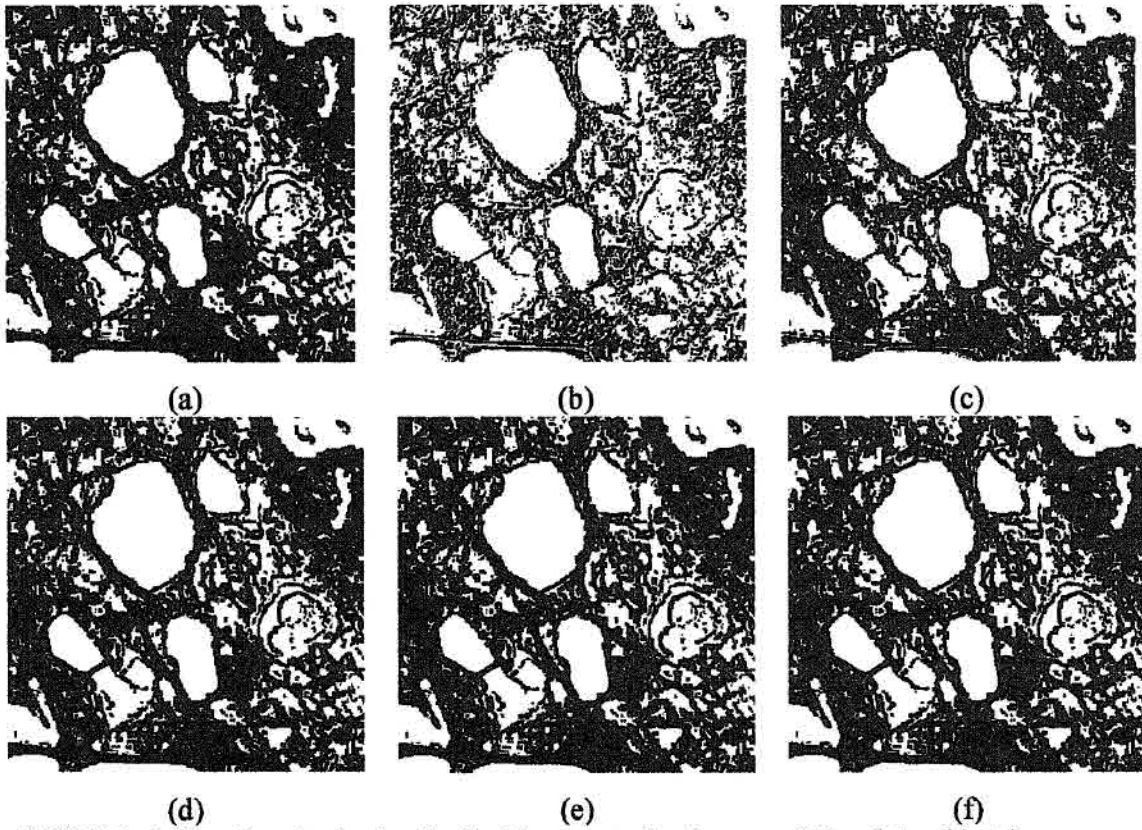


FIGURA 5.13 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 100 para a imagem Spot em níveis de cinza: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.

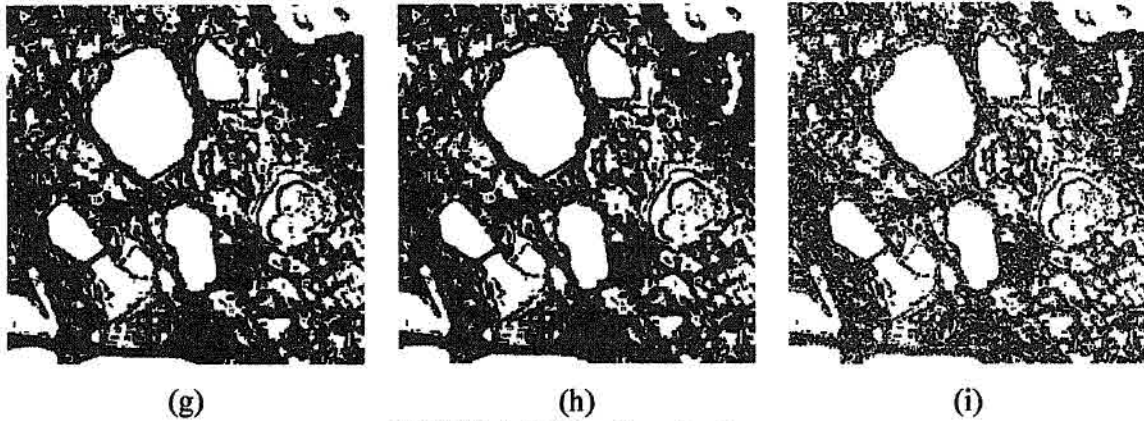


FIGURA 5.13 – Conclusão.

Pode-se observar que o número de padrões apresentados à rede não produziu modificações visíveis nas imagens de bordas obtidas na maioria das redes. Nas ativações das redes treinadas com 26 padrões de bordas foi utilizado o limiar de variância igual a 100.

Como as alterações no número de padrões não influenciam os resultados mostrados nas Figuras 5.12 e 5.13, as redes foram ativadas considerando um limiar de variância igual 50.

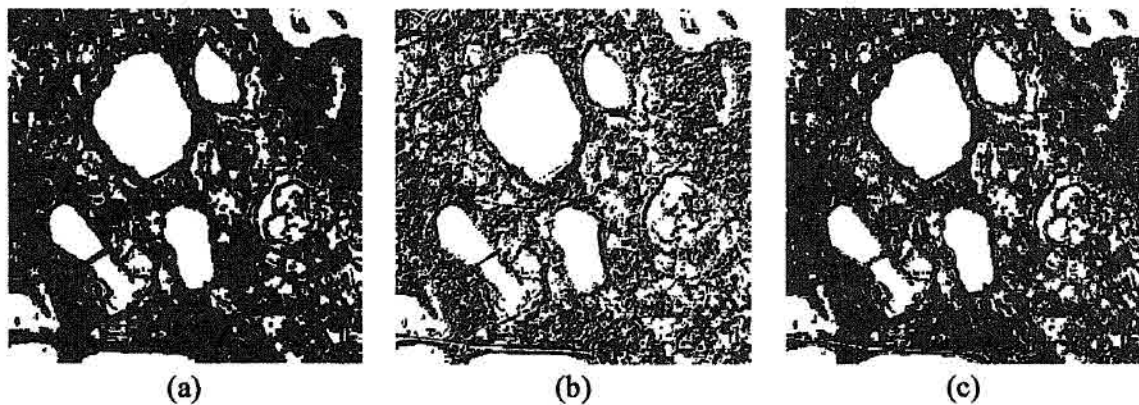


FIGURA 5.14 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 50 para a imagem spot NC: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.

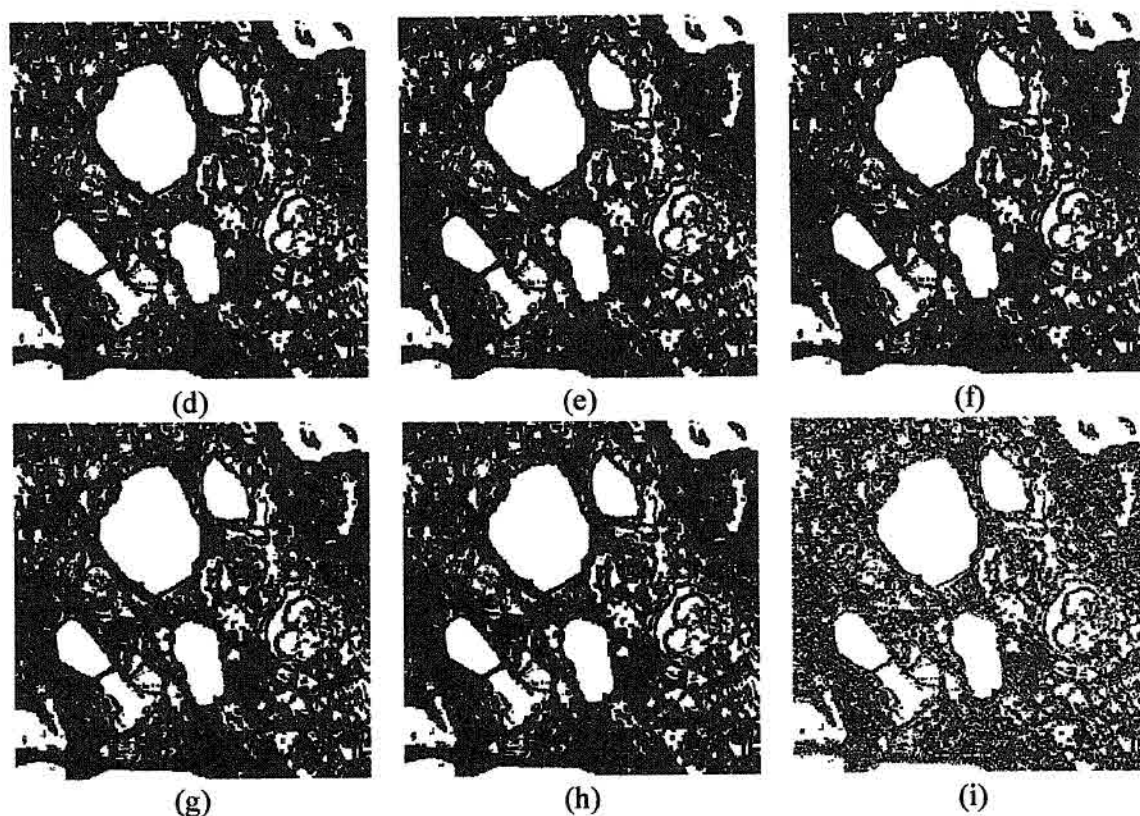


FIGURA 5.14 – Conclusão.

Nas Figuras 5.14(a)-(i) pode-se observar que os resultados foram bem influenciados pelo limiar de variância, resultando em um aumento no número de elementos de bordas encontrados, quando comparados aos resultados das Figuras 5.12(a)-(i). Nesse experimento podemos observar que o resultado para imagem de satélite não é satisfatório. Por se tratar de uma imagem de satélite com componentes de dimensões pequenas (devido à escala), muitas bordas foram detectadas levando a um resultado denso. Os resultados obtidos mostram que a escolha do limiar é importante e depende da classe de imagens utilizadas.

5.1.5 Resultado para Imagem Lena RGB

Além das imagens em nível de cinza, os algoritmos de redes foram ativados utilizando algumas imagens RGB apresentadas no capítulo 4.

As Figuras 5.15(a)-(i) mostram os resultados das redes com os 26 padrões de bordas binários e limiar de variância igual a 100. Pode-se observar que, em imagens RGB, as

bordas detectadas parecem mais contínuas. Isso deve-se ao fato do operador ser aplicado separadamente em cada componente (R, G, B). Na visualização, a imagem de bordas é tratada com uma imagem RGB.

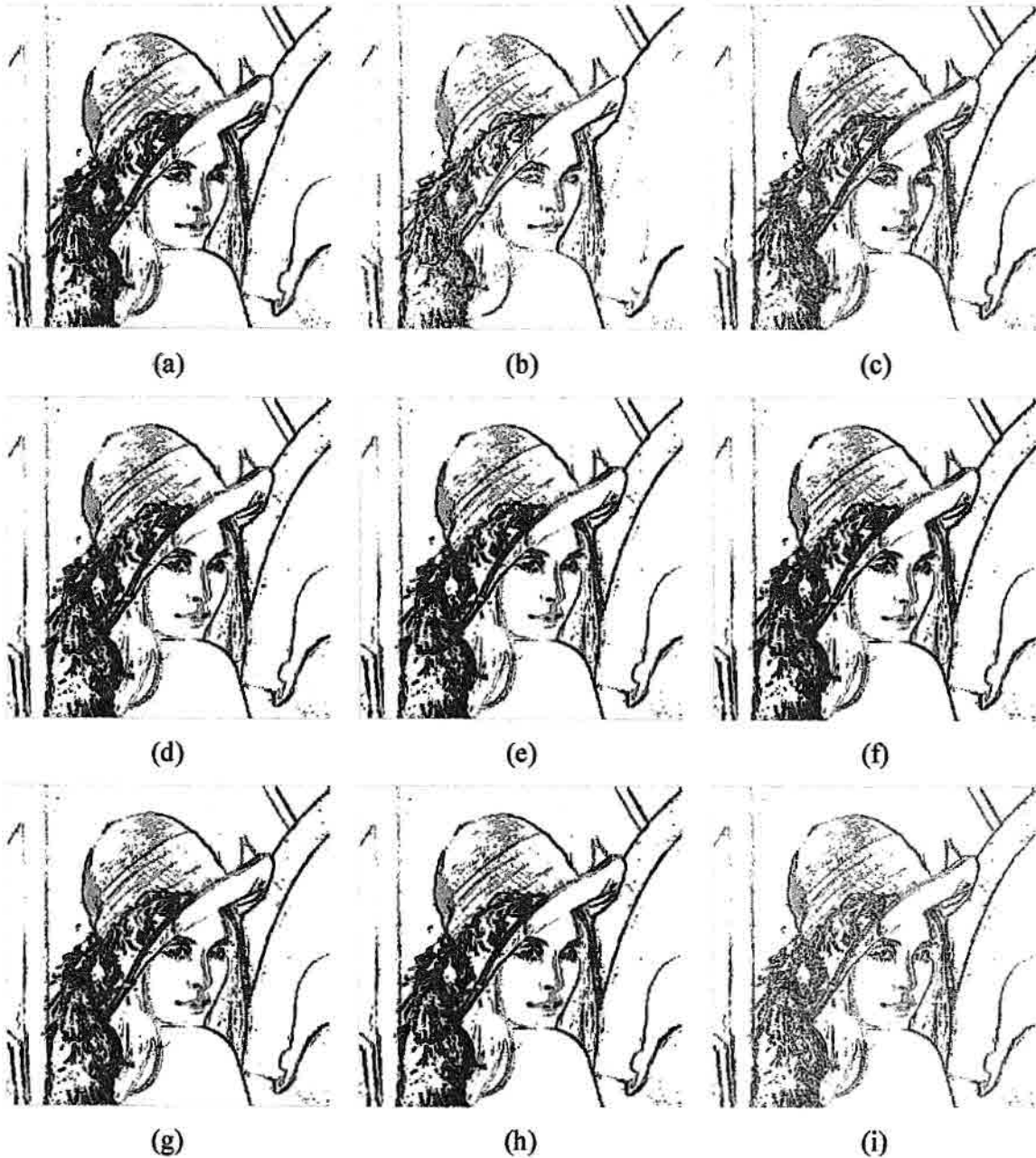


FIGURA 5.15 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 100 para a imagem Lena RGB: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.

Nas Figuras 5.16(a)-(i) são apresentados os resultados das redes treinadas com 26 padrões de bordas binários e com o limiar 50.



FIGURA 5.16 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 50 para imagem Lena RGB: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.

É possível verificar que a escolha do limiar influencia no resultado final. Quando a imagem RGB é exposta a um limiar mais baixo, surgem bordas mais densas.

5.1.6 Resultado para Imagem Peppers RGB

As Figuras 5.17(a)-(i) apresentam os resultados obtidos pelas redes neurais treinadas com 26 padrões de bordas binárias e com o limiar de variância igual a 100. Pode-se observar que a rede ART1 apresentou também falhas em algumas bordas quando comparada às outras redes que tiveram um desempenho melhor, detectando bordas mais definidas (cheias).

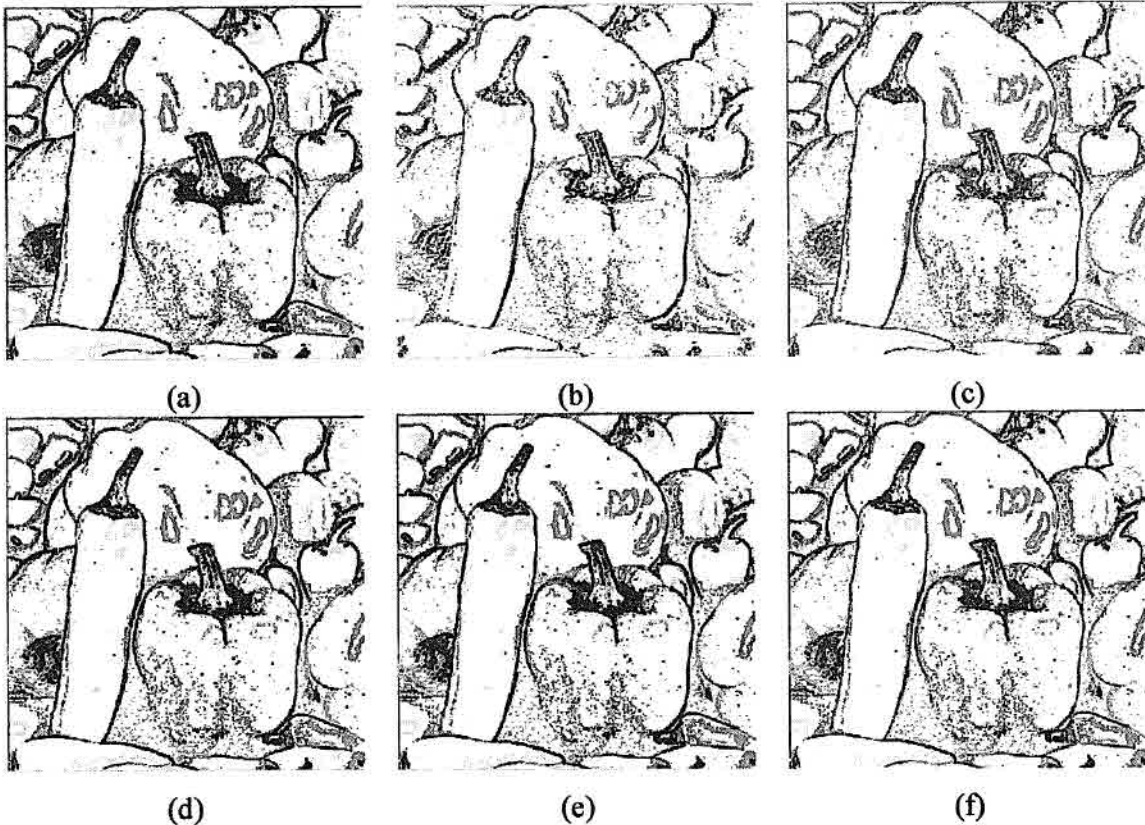


FIGURA 5.17 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 100 para a imagem peppers RGB: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.

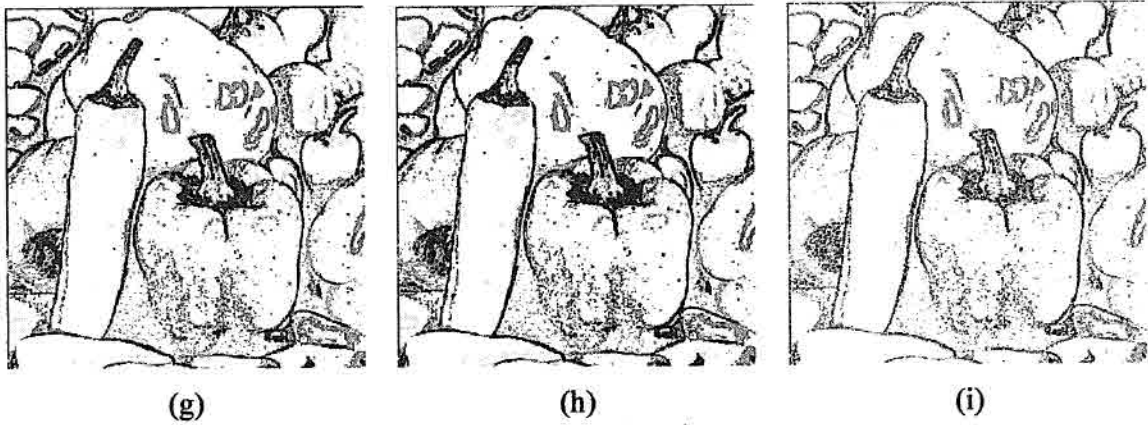


FIGURA 5.17 – Conclusão.

Nas Figuras 5.18(a)-(i) são apresentados os resultados das ativações das redes treinadas com 26 padrões de bordas binários e com o limiar de variância igual a 50. A análise qualitativa das imagens mostra as redes tiveram resultados semelhantes.

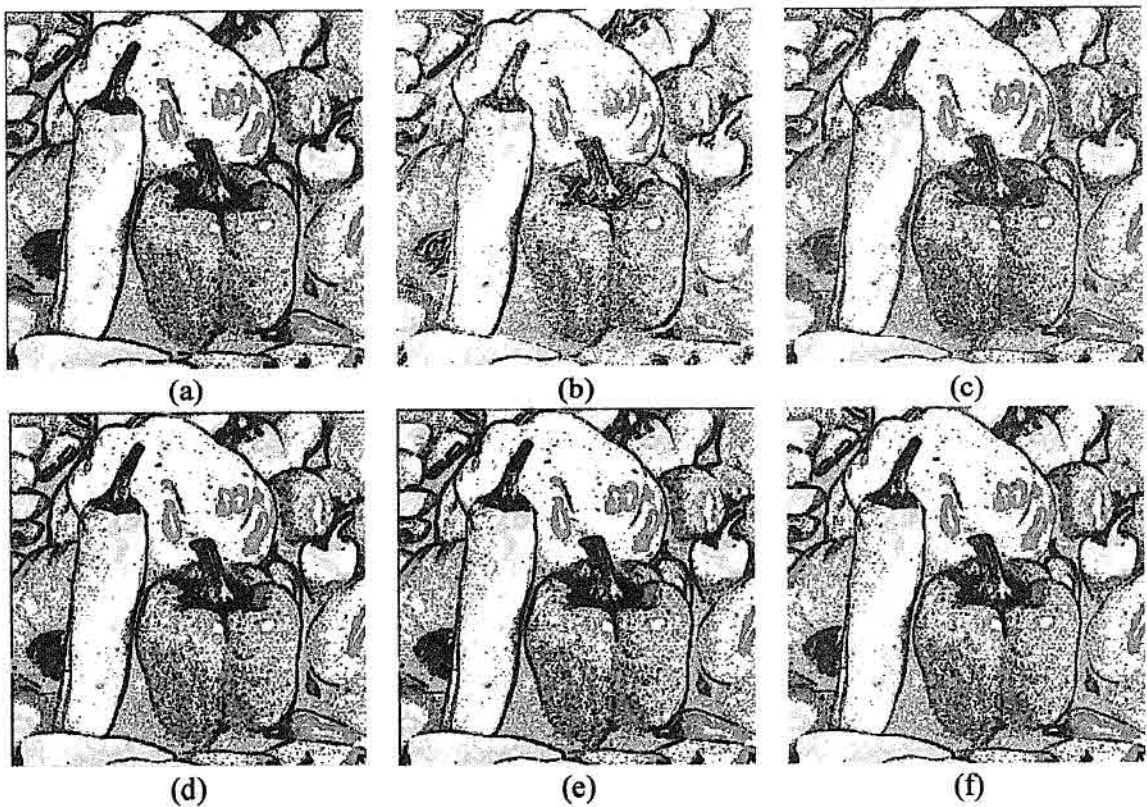


FIGURA 5.18 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 50 para imagem peppers RGB: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.

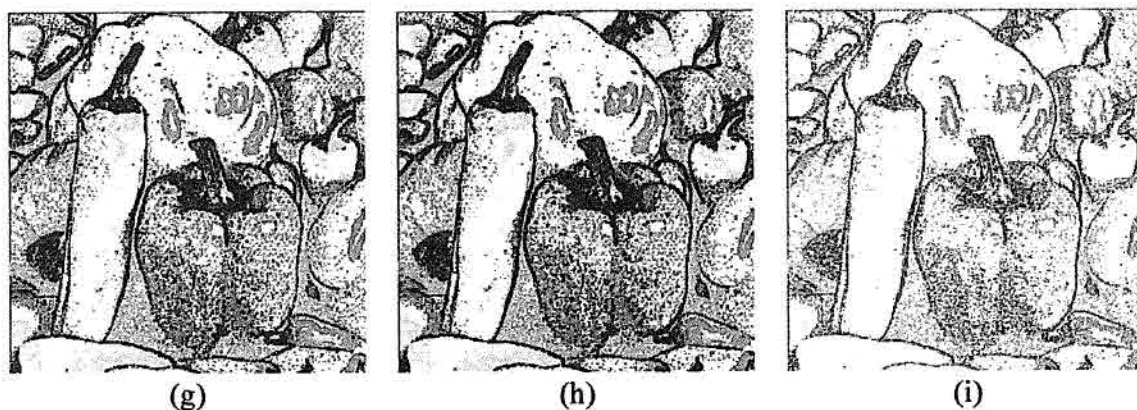


FIGURA 5.18 – Conclusão.

5.1.7 Resultado para Imagem Bloco RGB

Nas Figuras 5.19(a)-(i) e Figuras 5.20(a)-(i) são mostrados os resultados das ativações das redes, sobre a imagem Bloco RGB, com os 26 padrões de bordas binários, para limiares de variância de 100 e 50, respectivamente.

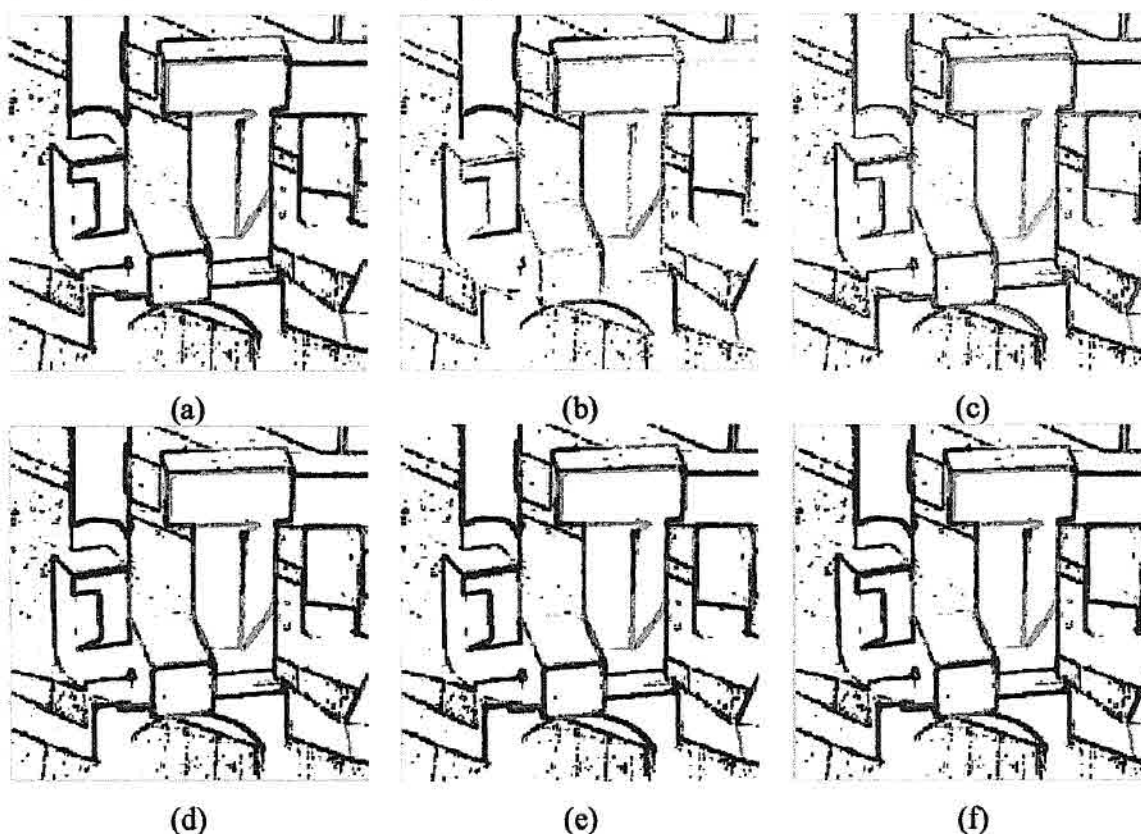


FIGURA 5.19 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 100 para a imagem blocos RGB: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.

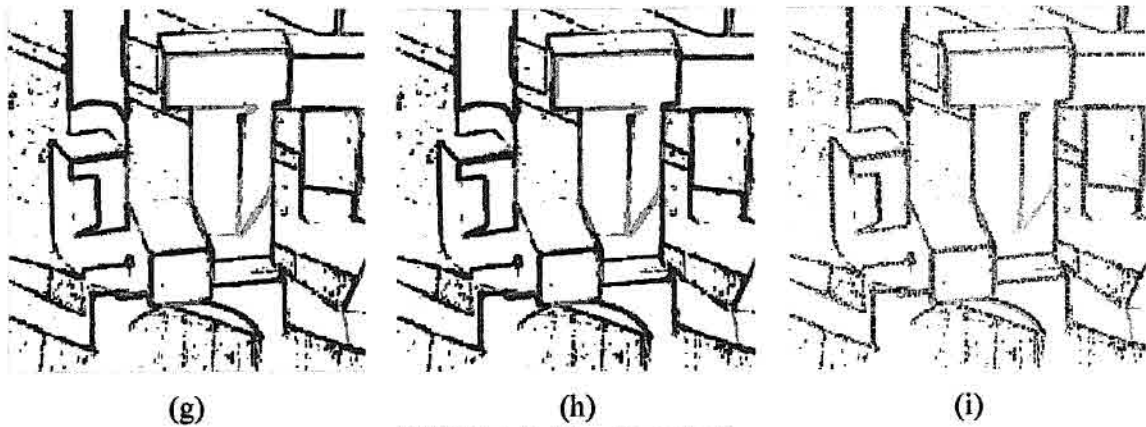


FIGURA 5.19 – Conclusão.

Através das Figuras 5.19(a)-(i), verifica-se que os resultados de todas as redes são semelhantes. As Figuras 5.20(a)-(i) apresentam os resultados das ativações das redes treinadas com 26 padrões de bordas binários, considerando um limiar de variância igual a 50.

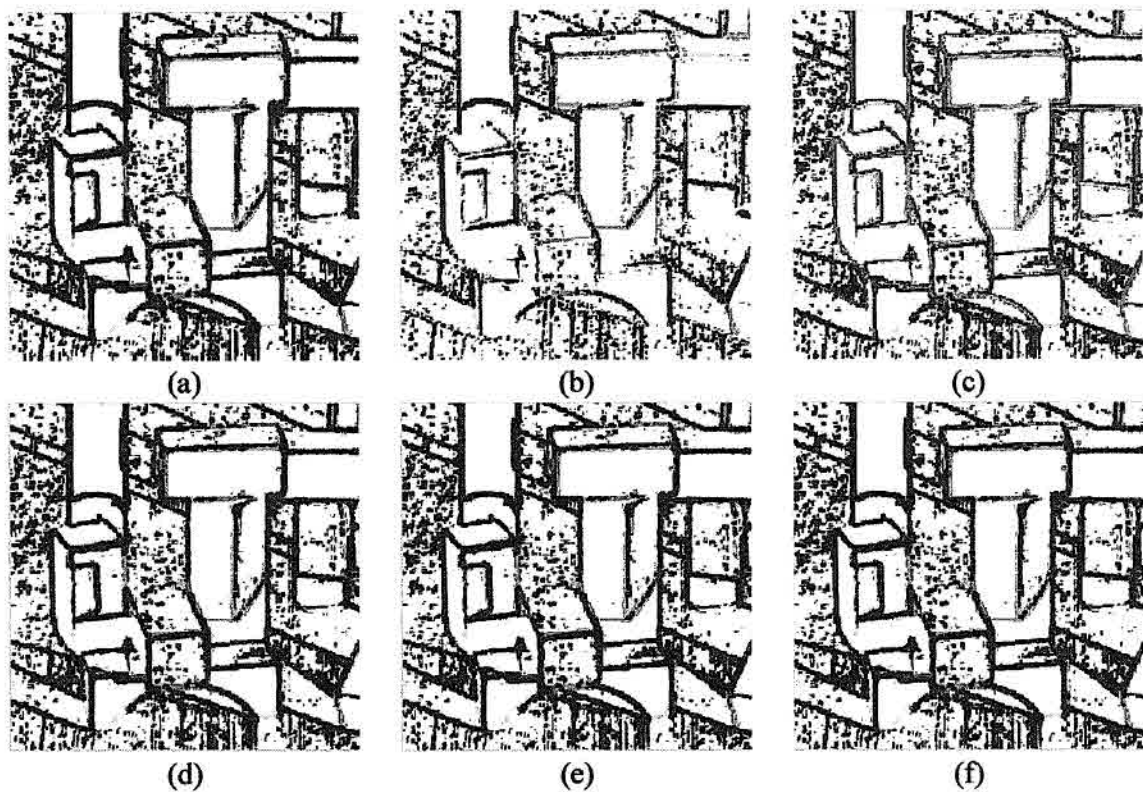


FIGURA 5.20 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 50 para imagem bloco RGB: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.

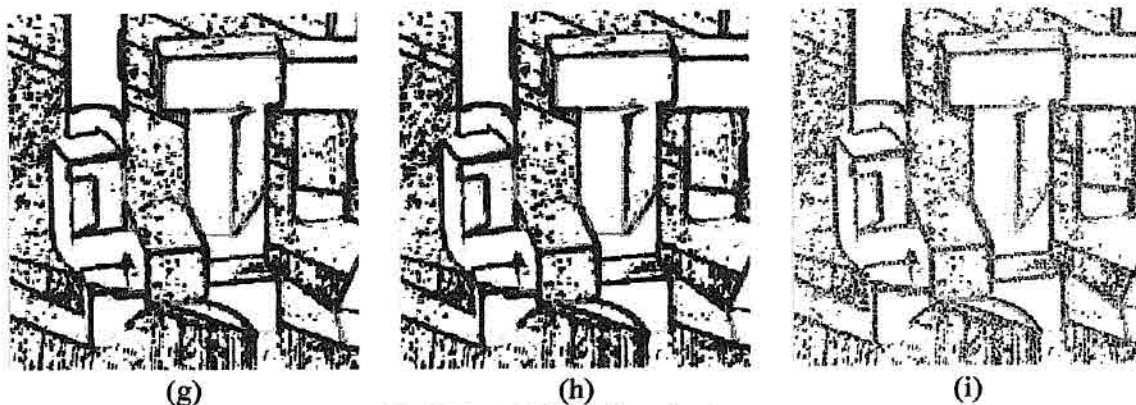


FIGURA 5.20 – Conclusão.

As Figuras 5.20(a)-(i) apresentam resultados com bordas bem definidas, mas com bastante ruído, mostrando a influência do limiar de variância.

5.1.8 Resultado para Imagem Casa RGB

Os resultados das ativações das redes treinadas com 26 padrões de bordas binários, para um limiar de variância igual a 100 são apresentados nas Figuras 5.21(a)-(i). Uma análise visual mostra que as redes detectaram as bordas mais salientes.

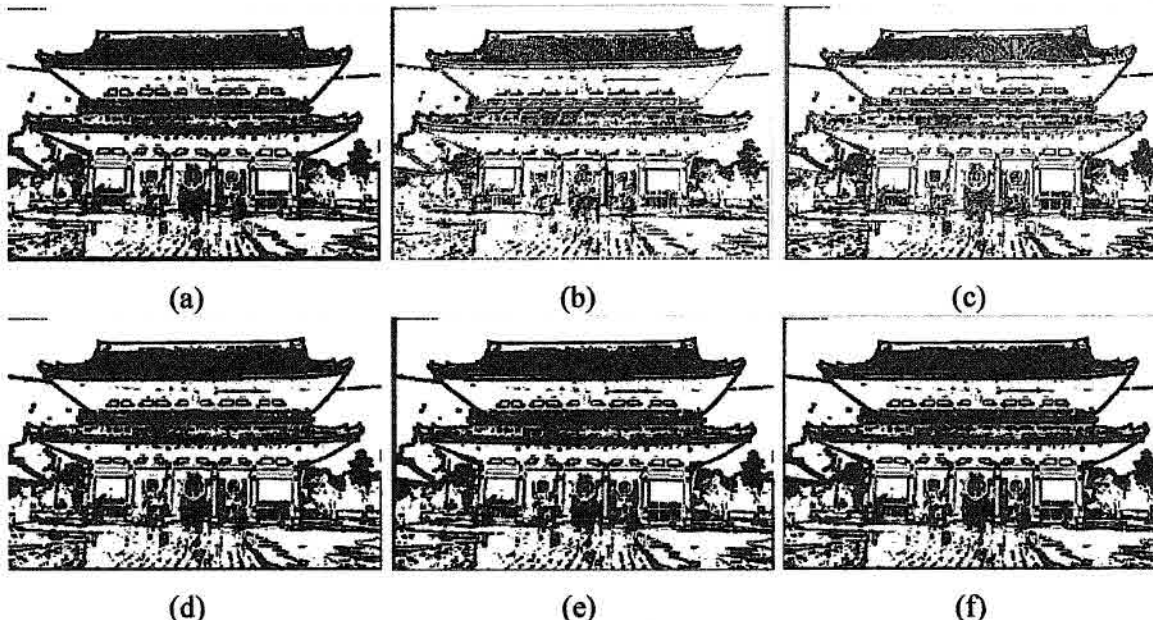


FIGURA 5.21 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 100 para a imagem casa RGB: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.

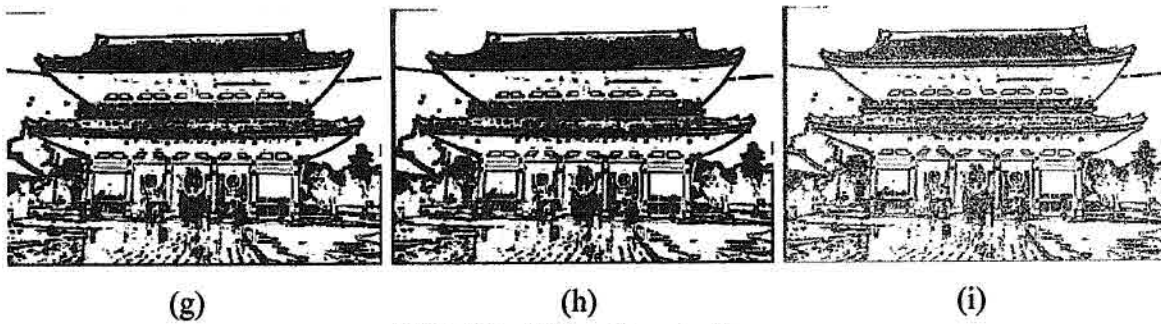


FIGURA 5.21 – Conclusão.

As Figuras 5.22(a)-(i), apresentam os resultados obtidos considerando um limiar de variância igual a 50.

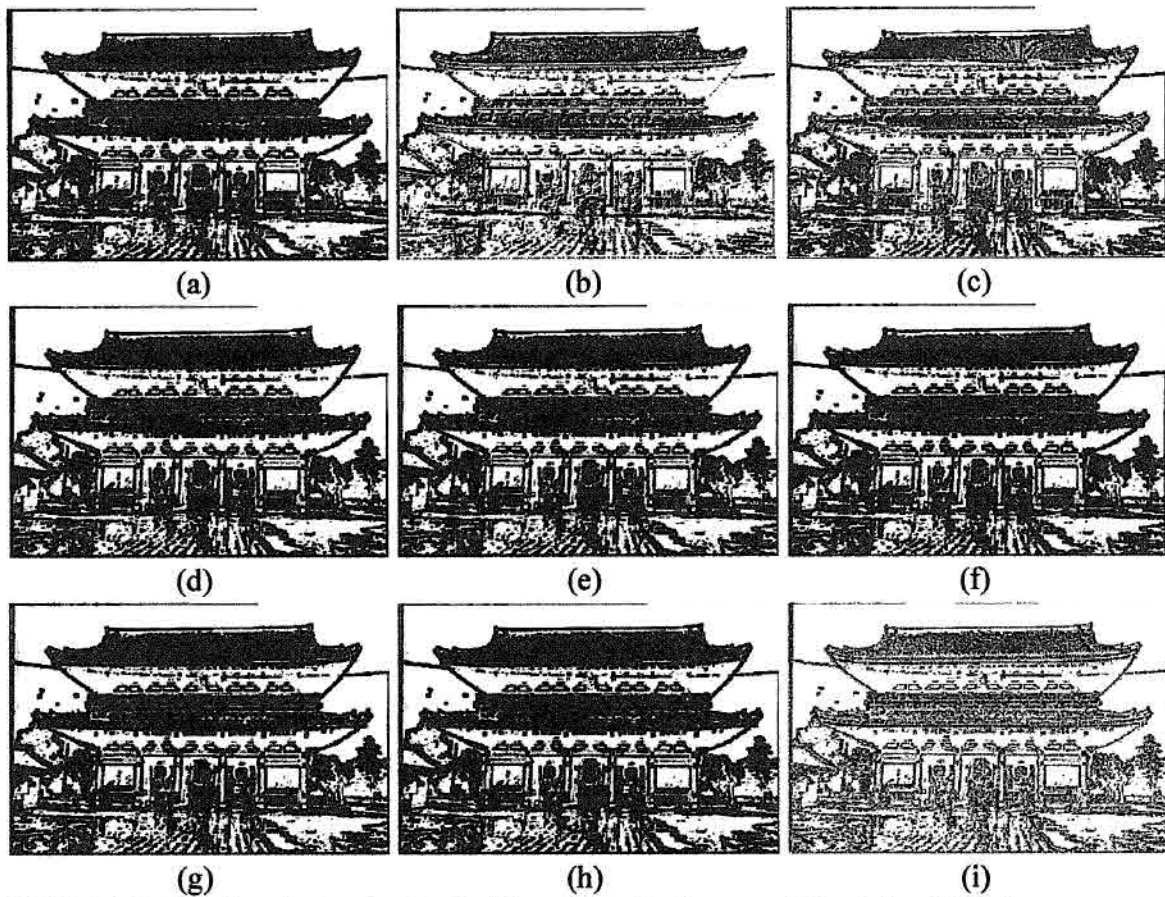


FIGURA 5.22 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 100 para imagem casa RGB: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.

5.1.9 Resultado para Imagem Pista Real RGB

Nas Figuras 5.23(a)-(i) e Figuras 5.24(a)-(i) são mostrados os resultados das ativações das redes treinadas com 26 padrões binários sobre a imagem Pista RGB, para limiares de variância de 100 e 50, respectivamente.

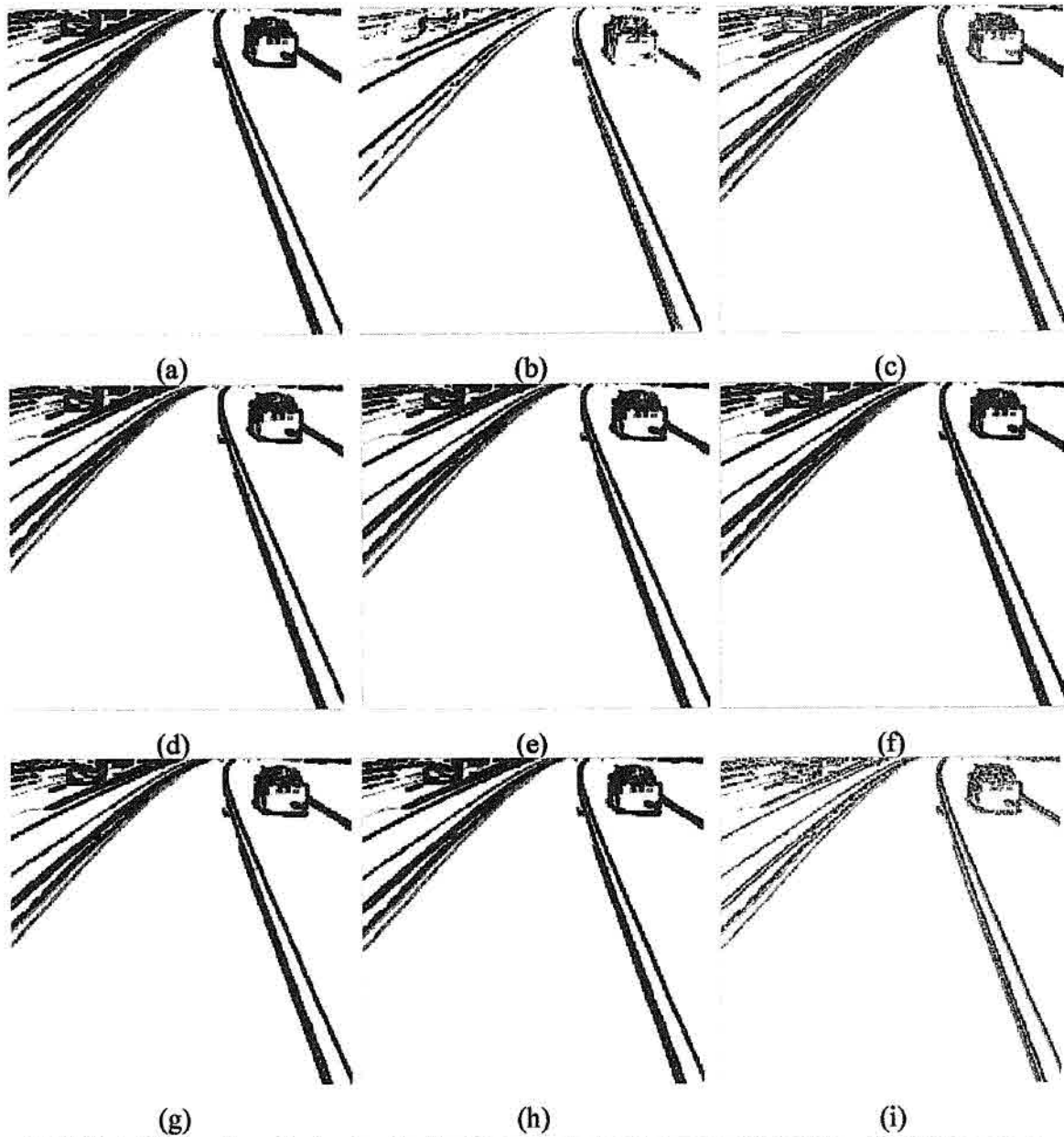


FIGURA 5.23 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 100 para a imagem pista real RGB: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.

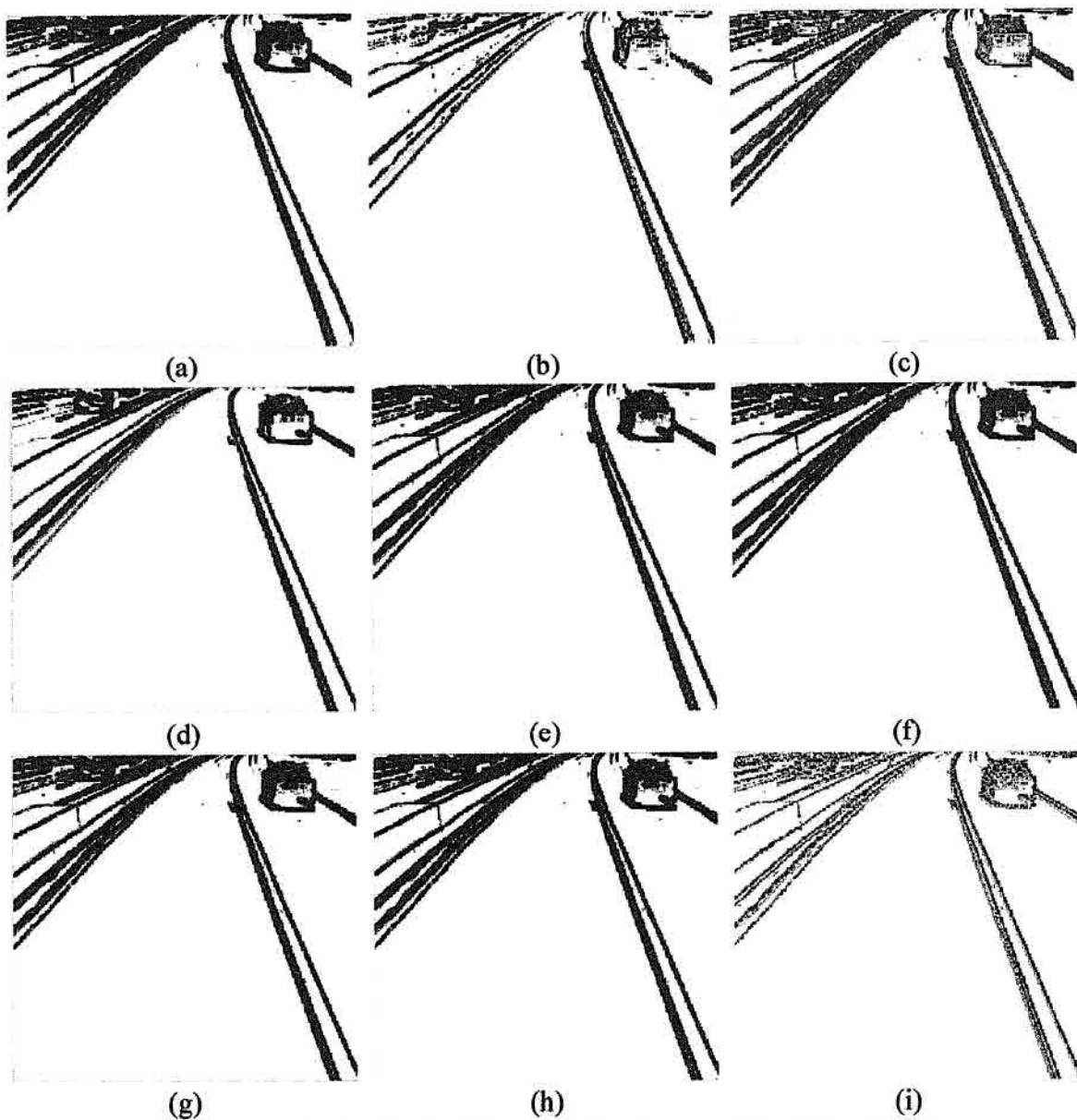


FIGURA 5.24 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 50 para imagem da pista RGB: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.

As Figuras 5.24(a)-(i) apresentam resultados com bordas mais definidas, mostrando a influência do limiar de variância.

5.1.10 Resultado para Imagem Satélite RGB

Nas Figuras 5.25(a)-(i) e Figuras 5.26(a)-(i) são mostrados os resultados das ativações das redes treinadas com 26 padrões de bordas binários sobre a imagem Satélite RGB para limiares de variância de 100 e 50, respectivamente.

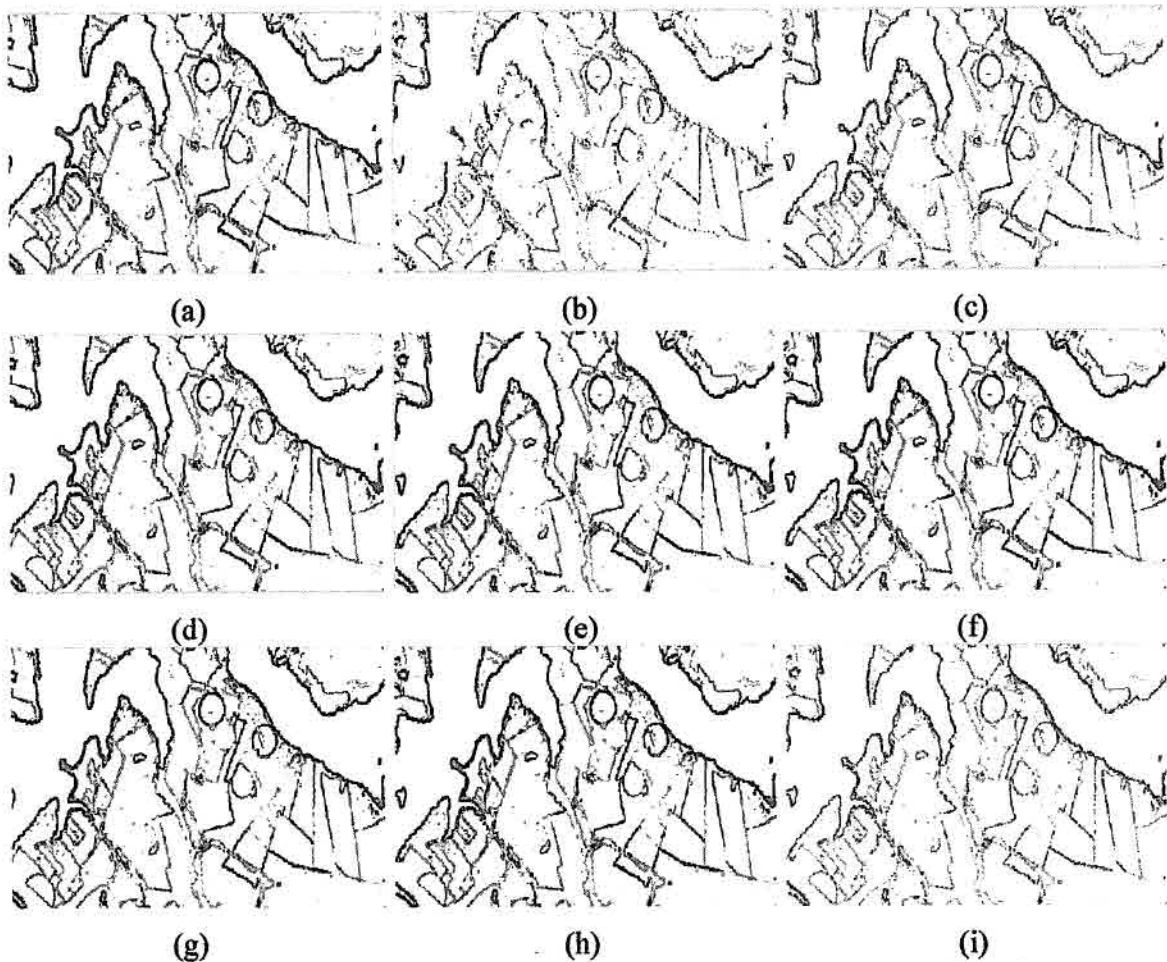


FIGURA 5.25 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 100 para a imagem satélite RGB: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.

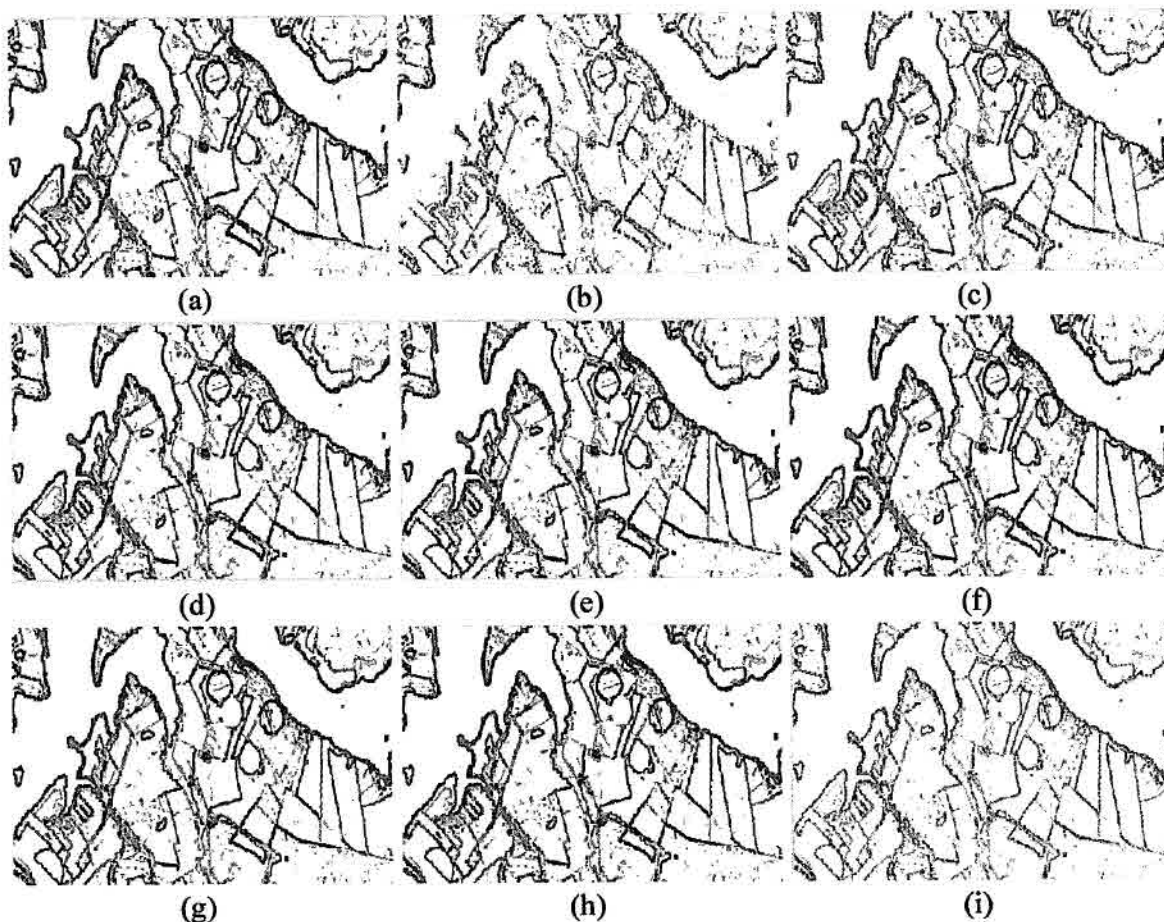


FIGURA 5.26 – Resultado das Redes Neurais treinadas com 26 padrões binários e com limiar da variância igual a 50 para imagem satélite RGB: (a) BAM, (b) ART1, (c) ART2, (d) RPMC, (e) RBF, (f) LVQ1, (g) LVQ2, (h) FAM e (i) Maxnet.

Dos resultados apresentados com a alteração do valor do limiar de variância é possível notar uma influencia no número de pontos de bordas detectados. Esta influencia foi constatada em todos os experimentos com imagens em níveis de cinza e RGB. A escolha do limiar adequado é uma tarefa difícil, pois depende da natureza da imagem.

5.1.11 Operador de Canny

O operador de Canny implementado nesta dissertação foi baseado no algoritmo de Parker (1997). O desvio padrão σ é um parâmetro do detector de Canny que quando alterado aumenta ou diminui a densidade das bordas detectadas. Nas Figuras 5.27(a)-(d) são apresentados os resultados da aplicação do operador de Canny sobre as imagens em níveis de cinza, utilizando um desvio padrão igual a 0.5.

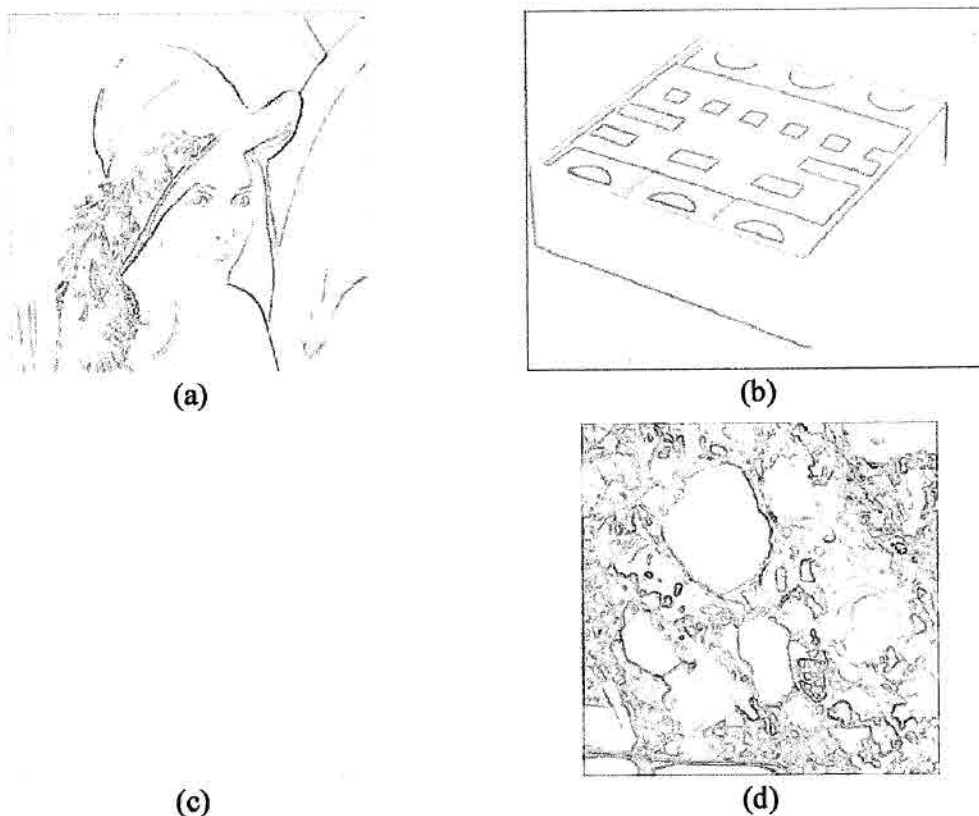


FIGURA 5.27 – Resultado do Operador Canny com $\sigma = 0.5$: (a) Lena, (b) Blocos, (c) Pista, (d) Spot_pan.

Das Figuras 5.27(a)-(d) pode-se observar que o operador de Canny consegue detectar bordas com diferentes intensidades. Nos experimentos com este operador, não se utilizou limiar de variância, por isso nos resultados se verifica a detecção de bordas em áreas com pouca informação de contraste.

Uma comparação visual com os resultados das redes neurais mostra que as redes apresentam resultados satisfatórios, quando comparados com o operador Canny. Uma diferença que pode ser notada está na intensidade das bordas detectadas.

Nas Figuras 5.28(a)-(f) são apresentados os resultados da aplicação do operador de Canny em imagens em RGB, utilizando o mesmo valor de desvio padrão (0.5).

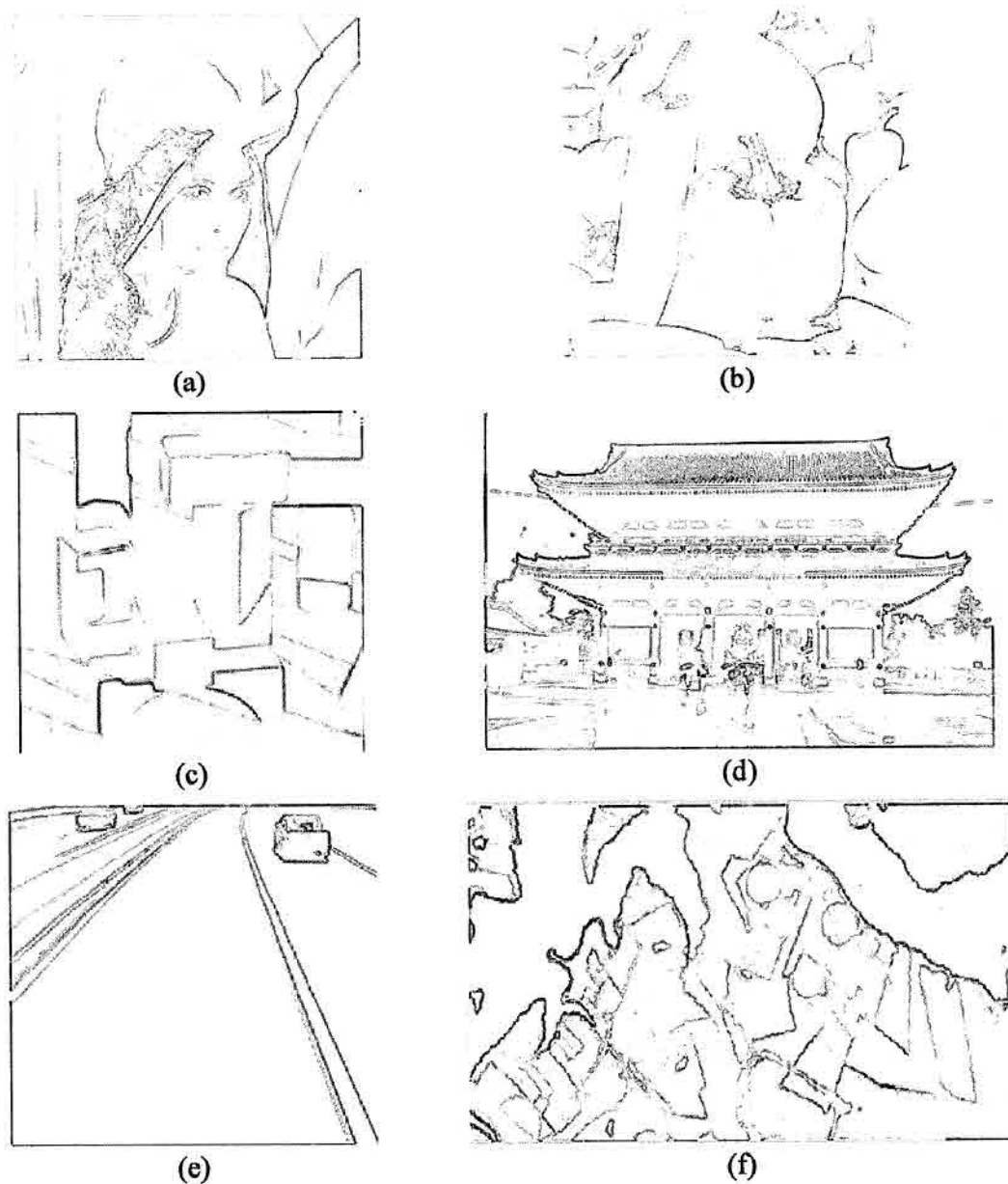


FIGURA 5.28 – Resultado do Operador Canny com $\sigma = 0.5$: (a) Lena, (b) Peppers, (c) Blocos, (d) Casa, (e) Pista, (f) Plantação.

A Figura 5.29(a)-(f) apresenta o resultado do operador de Canny com o desvio padrão igual a 1, sobre as imagens RGB.

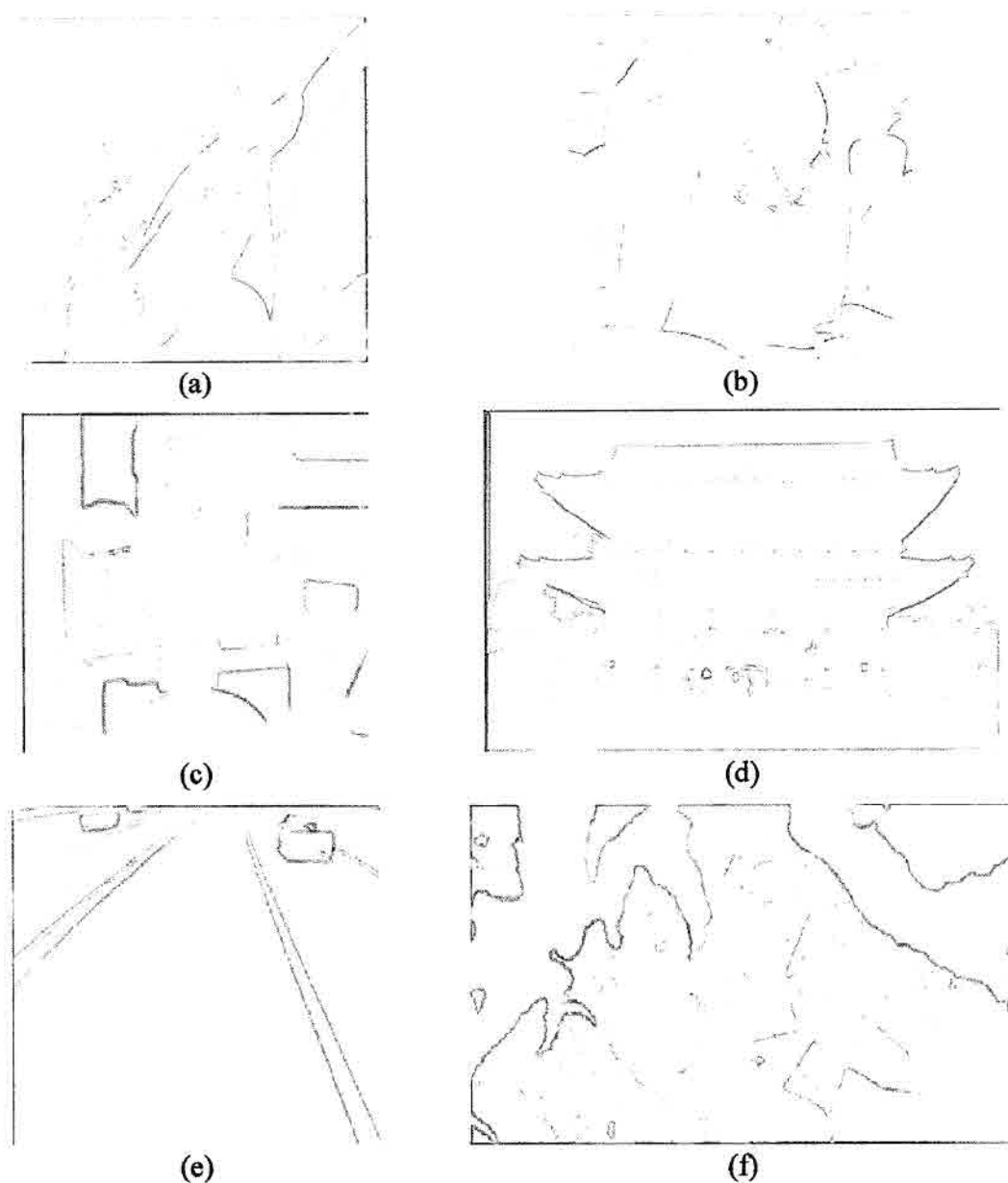


FIGURA 5.29 – Resultado do Operador Canny com $\sigma = 1.0$: (a) Lena, (b) Peppers, (c) Blocos, (d) Casa, (e) Pista, (f) Plantação.

Pode-se observar que com o aumento do desvio padrão, detalhes da imagem foram perdidos, pois as bordas de menor intensidade foram descartadas. É possível concluir que a escolha do parâmetro σ é uma tarefa difícil, pois também pode depender do tipo de imagem que se está processando.

5.1.12 Operador Nebuloso

Uma outra implementação é o operador nebuloso baseado no trabalho de Cho e Cho (1994), que utiliza um método automático de limiarização usando lógica nebulosa. O método proposto detecta a associação da transição com a variação da intensidade, considerando um brilho local. As medidas de entrada do operador nebuloso são as médias locais (brilho) e a variância (contraste). A saída do operador nebuloso é a medida do limiar que muda em função da imagem (detalhe desse algoritmo pode ser encontrado em Cho e Cho, 1994).

Nas Figuras 5.30(a)-(d) são apresentados os resultados da aplicação do operador nebuloso para imagens em níveis de cinza.

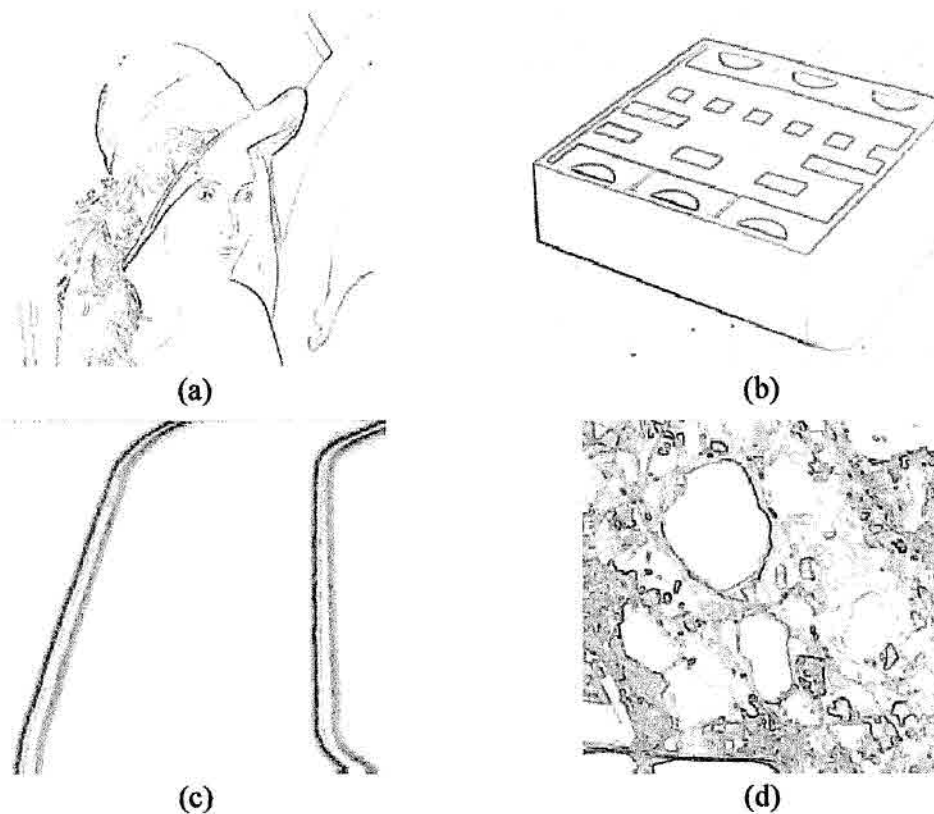


FIGURA 5.30 – Resultado do Operador Fuzzy: (a) Lena, (b) Blocos, (c) Pista, (d) Spot_pan.

Observa-se que, assim como o operador de Canny, o operador nebuloso detecta bordas com diferentes intensidades.

Em Cho e Cho (1994) são utilizadas apenas imagens em níveis de cinza. Nesta dissertação foram feitas pequenas adaptações para se usar o algoritmo nebuloso em imagens RGB. As Figuras 5.31(a)-(f) apresentam esses resultados.

Assim como no operador de Canny, também não se usou limiar de variância para o operador nebuloso.

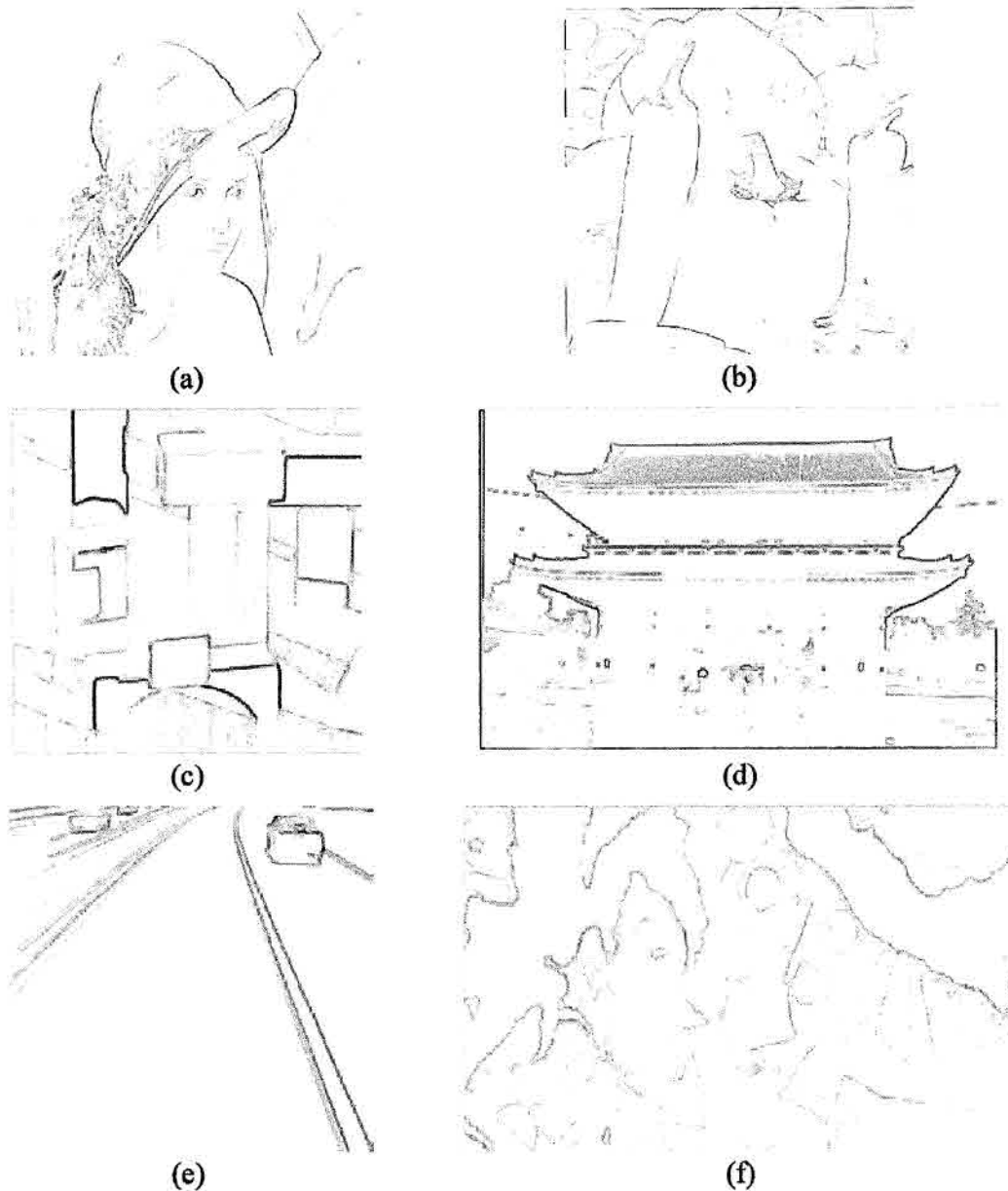


FIGURA 5.31 – Resultado do Operador Nebuloso: (a) Lena, (b) Peppers, (c) Blocos, (d) Casa, (e) Pista, (f) Plantação.

5.2 ANÁLISE QUANTITATIVA

Nesta dissertação, são considerados dois aspectos dos algoritmos: o tempo de processamento sobre as imagens e a qualidade das imagens de bordas geradas.

A comparação entre os resultados foi feita qualitativamente, através de uma análise visual, quantitativamente através da comparação das imagens de bordas geradas por cada algoritmo, com as imagens de bordas tomadas como verdade terrestre (*ground truth*), usando diferentes técnicas. Os trabalhos de Peli e Malah (1982) e Shin et al. (2001) apresentam diferentes abordagens que podem ser usadas para medir a qualidade das imagens de bordas obtidas. Um método simples, citado em Peli e Malah (1982), é o cálculo do percentual de pontos de bordas encontradas em relação aos pontos de bordas reais, o que requer o uso da verdade terrestre (*ground truth*).

Nesta dissertação, a verdade terrestre das imagens utilizadas nos experimentos não está disponível. Para estabelecer padrões para fins de comparação, consideram-se as afirmações em Shin et al. (2001) que estabelece o operador de Canny como o detector de bordas padrão. Com isso, as imagens originais usadas na ativação dos operadores neurais, também são apresentadas ao operador de Canny, cuja implementação segue o algoritmo de Parker (1997).

Para estabelecer uma medida de desempenho dos operadores, as imagens obtidas com os operadores neurais são comparadas com as imagens geradas pelo operador de Canny utilizando duas medidas diferentes:

- O percentual de pontos de bordas corretos gerados por cada operador neural, como sugerido por Peli e Malah (1982), tomando a imagem de bordas gerada pelo operador de Canny como padrão;

$$P_c = \frac{\text{n.º de pontos de borda encontrados}}{\text{n.º de pontos de bordas gerados pelo operador Canny}} \quad (5.1)$$

- O erro quadrático médio entre a imagem de bordas gerada por cada operador neural e a imagem de bordas gerada pelo operador de Canny.

$$eqm = \frac{1}{nl * np} \sum_x \sum_y [gb_{Canny}(x,y) - gb_{Neural}(x,y)]^2 \quad (5.2)$$

onde nl é o máximo de linhas, np é o número de pixel por linha, x e y são as coordenadas do pixel e da linha consideradas.

Entretanto, como observado nos resultados apresentados neste capítulo as imagens de bordas geradas pelos operadores neurais possuem intensidades binárias ou numéricas. O operador de Canny, por sua vez, gera imagens com diferentes intensidades (gradiente), proporcionais às informações de bordas calculadas em cada ponto. Assim, faz-se necessário tornar as imagens compatíveis, antes de aplicar um método de comparação. Para isso, as imagens de bordas geradas pelo operador de Canny são binarizadas. A binarização das imagens nesta dissertação utiliza o método de limiarização unimodal proposto por Rosin (2001).

O método considera que a imagem em análise apresenta um histograma unimodal, ou seja, assume-se que a imagem apresenta uma população dominante produzindo um pico principal localizado na extremidade mais baixa do histograma, em relação à segunda população com níveis de cinza mais altos. Esta segunda extremidade pode ou não produzir um pico discernível, mas necessita estar razoavelmente bem separada do pico alto para evitar confusão. A Figura 5.32 apresenta um exemplo do histograma citado e uma visão geral do método.

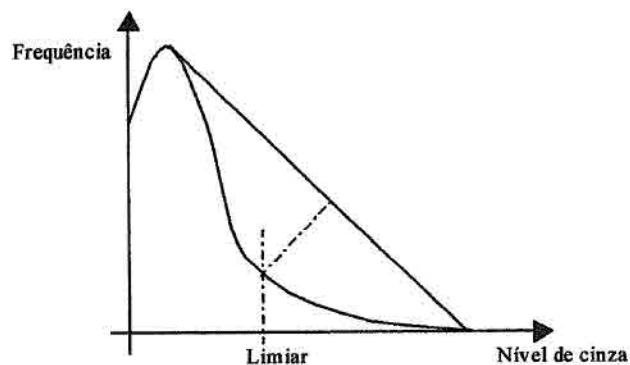


FIGURA 5.32 – Procedimento para cálculo de limiar a partir de histograma.
FONTE: Adaptado de Rosin (2001 p.2084).

Uma linha reta é construída a partir da extremidade do pico é do histograma, até o primeiro nível de cinza com frequência zero, no lado direito do ponto de pico (Figura 5.33). Considerando o histograma descrito por uma função $H(nc)$, onde nc representa o nível de cinza, calcula-se a distância entre a reta encontrada e todos os pontos $(nc, H(nc))$ entre o pico e o primeiro de ponto de frequência zero. O ponto $(nc, H(nc))$ para o qual a distância é máxima determina o valor do limiar como sendo nc . A eficiência do método é mostrada em Rosin (2001). Nas Figuras 5.33 (a)-(d) tem-se as imagens de bordas geradas pelo operador de Canny após a aplicação do método de limiarização descrito anteriormente.

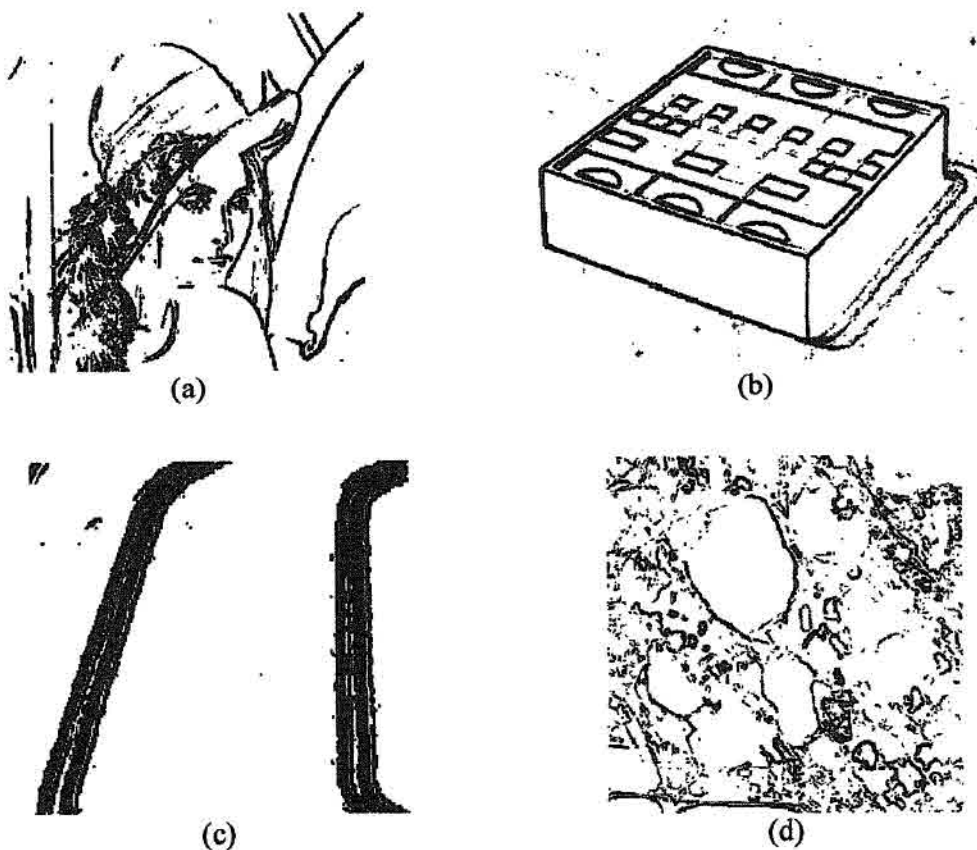


FIGURA 5.33 – Resultados da aplicação da limiarização sobre as imagens geradas pelo operador de Canny com $\sigma = 0.5$, sobre imagens em níveis de cinza: (a)Lena, Limiar=12; (b) Blocos, Limiar=10; (c) Pista, Limiar=8; (d)Spot_pan, Limiar=33.

O cálculo de limiarização sobre a imagem de bordas obtida com o operador de Canny, com o método descrito, é uma tentativa de gerar uma imagem binária apenas para as bordas mais significativas. Entendendo que há subjetividade no conceito de borda

significativa, pode-se existir outras abordagens que produzam resultados mais satisfatórios.

As imagens das Figuras 5.33(a)-(d) são tomadas como referências, uma vez que foram geradas pelo operador de Canny, tomado como operador padrão para detecção de bordas (Shin et al. 2001).

No algoritmo de Parker (1997), a limiarização da imagem de borda é feita através da limiarização por histerese. Histerese é um método que utiliza dois limiares, um alto e um baixo e todo pixel situado acima do limiar alto é considerado borda e o pixel situado abaixo do limiar baixo é rejeitado. Pontos situados entre os dois limites são aceitos se estiverem relacionados com os pixels que tenham como vizinhos pixels com limiar alto. Para verificar o resultado da limiarização pelo método Rosin (2001), utilizou-se o algoritmo de Ma (1996), que usa limiarização por histerese para efeito de comparação.

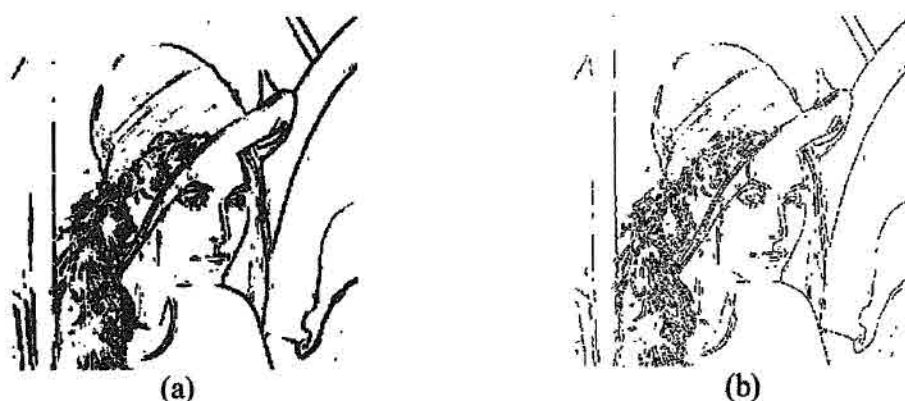


FIGURA 5.34 – Imagem da Lena em níveis de cinza Limiarizada: (a) Método de Rosin e (b) Método de Histerese.

Observa-se da Figura 5.34 que o método de Rosin apresentou bons resultados. Considerando a facilidade de implementação e o custo computacional baixo, optou-se, nesta dissertação, pelo uso do algoritmo de Rosin.

A Tabela 5.1 apresenta o percentual de pontos de bordas encontrados pelo operador de Canny, em relação às dimensões da imagem; o percentual de pontos de bordas encontrados pelos operadores neurais, em relação às dimensões das imagens; o erro quadrático médio entre as imagens de bordas geradas; o tipo de rede e as imagens

utilizadas. As imagens geradas pelos operadores neurais consideravam um limiar de variância igual a 100.

TABELA 5.1 - DIFERENÇAS ENTRE AS IMAGENS DE BORDAS DE CANNY E OS OPERADORES NEURAIIS COM LIMIAR 100.

Número de pontos Canny	Número de pontos Rede	Percentual de Pontos de Bordas Canny, em relação às dimensões da imagem original.	Percentual de Pontos de Bordas - Rede em relação às dimensões da imagem original.	Erro quadrático médio entre as imagens de bordas de Canny e da Rede	Tipo de Rede	Imagem
39560	54237	0,15	0,21	0,06	BAM	Lena
39560	21076	0,15	0,08	0,09	ART1	Lena
39560	37387	0,15	0,14	0,04	ART2	Lena
39560	37387	0,15	0,14	0,04	BP	Lena
39560	37387	0,15	0,14	0,04	RBF	Lena
39560	37387	0,15	0,14	0,04	LVQ1	Lena
39560	37379	0,15	0,14	0,04	LVQ2	Lena
39560	37387	0,15	0,14	0,04	FAM	Lena
39560	37387	0,15	0,14	0,04	MAXNET	Lena
14055	14286	0,11	0,12	0,08	BAM	Blocos
14055	6877	0,11	0,06	0,06	ART1	Blocos
14055	10722	0,11	0,09	0,03	ART2	Blocos
14055	10722	0,11	0,09	0,03	BP	Blocos
14055	10722	0,11	0,09	0,03	RBF	Blocos
14055	10722	0,11	0,09	0,03	LVQ1	Blocos
14055	10722	0,11	0,09	0,03	LVQ2	Blocos
14055	10722	0,11	0,09	0,03	FAM	Blocos
14055	10722	0,11	0,09	0,03	MAXNET	Blocos
6163	2801	0,19	0,08	0,13	ART1	Pista
6163	5970	0,19	0,18	0,02	BAM	Pista
6163	4843	0,19	0,15	0,04	ART2	Pista
6163	4843	0,19	0,15	0,04	BP	Pista
6163	4843	0,19	0,15	0,04	RBF	Pista
6163	4843	0,19	0,15	0,04	LVQ1	Pista
6163	4843	0,19	0,15	0,04	LVQ2	Pista
6163	4843	0,19	0,15	0,04	FAM	Pista
6163	4843	0,19	0,15	0,04	MAXNET	Pista
12976	61343	0,14	0,65	0,51	BAM	Spot-pan
12976	28544	0,14	0,30	0,30	ART1	Spot-pan
12976	51149	0,14	0,54	0,40	ART2	Spot-pan
12976	51149	0,14	0,54	0,40	BP	Spot-pan
12976	51149	0,14	0,54	0,40	RBF	Spot-pan
12976	51149	0,14	0,54	0,40	LVQ1	Spot-pan
12976	51140	0,14	0,54	0,40	LVQ2	Spot-pan
12976	51149	0,14	0,54	0,40	FAM	Spot-pan
12976	51149	0,14	0,54	0,40	MAXNET	Spot-pan

Através da Tabela 5.1, podemos observar que algumas redes neurais tiveram o mesmo desempenho. Isso aconteceu devido ao modelo de rede utilizado. Em alguns modelos as saídas estão associadas aos tipos de bordas. Em outros modelos a saída da rede é o próprio elemento de borda.

A Figura 5.35 apresenta os resultados das ativações das redes com saídas geradas pelos tipos de bordas e o histograma com os tipos de bordas encontrados. Cada cor da imagem de borda está relacionada com um tipo de borda (lembrando que foram utilizados 26 padrões no treinamento das redes, apresentados na Figura 4.2 e Figura 4.3).

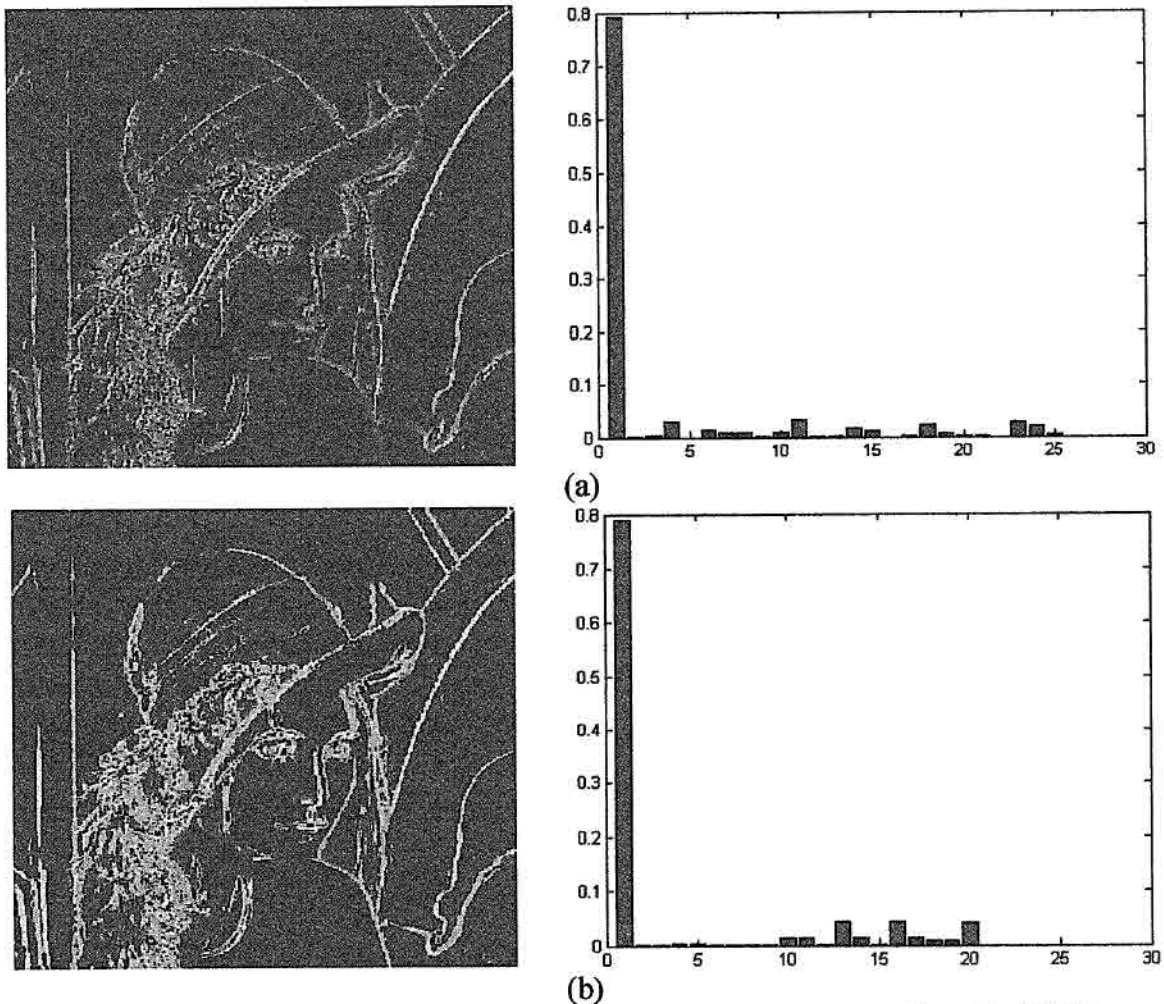
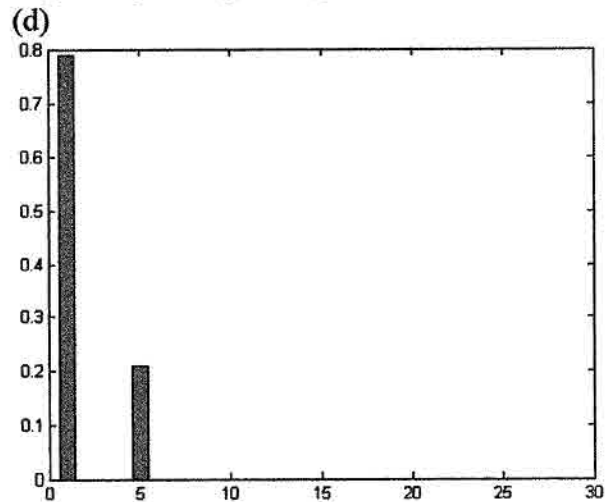
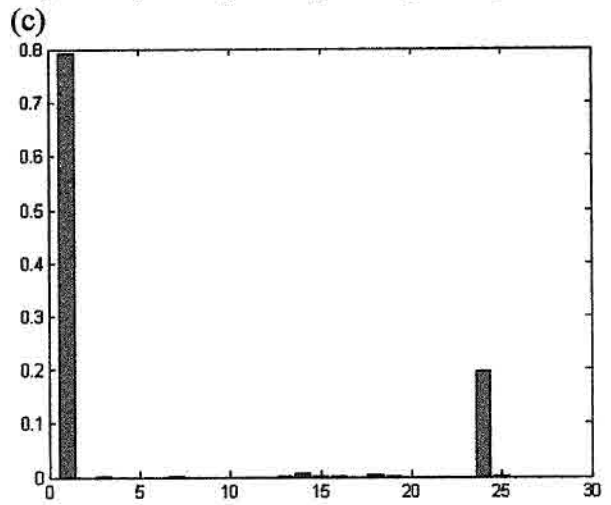
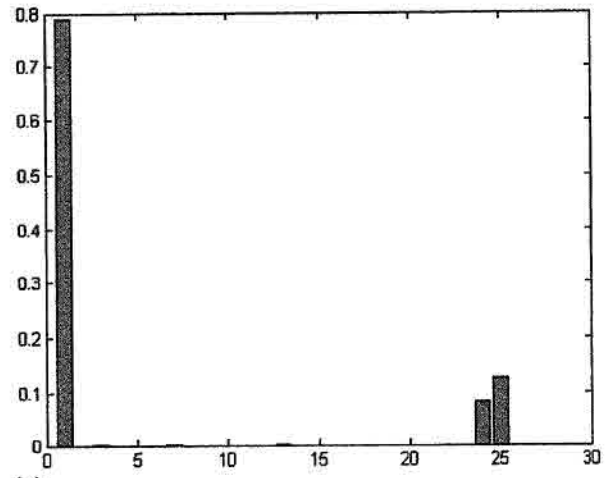


FIGURA 5.35 - Resultados das redes com os tipos de bordas encontrados: (a) ART 1 e seu histograma, (b) ART 2 e seu histograma, (c) LVQ 1 e seu histograma, (d) LVQ 2 e seu histograma, (e) FAM e seu histograma, (f) MAXNET e seu histograma.



(e)
FIGURA 5.35 – Continua.

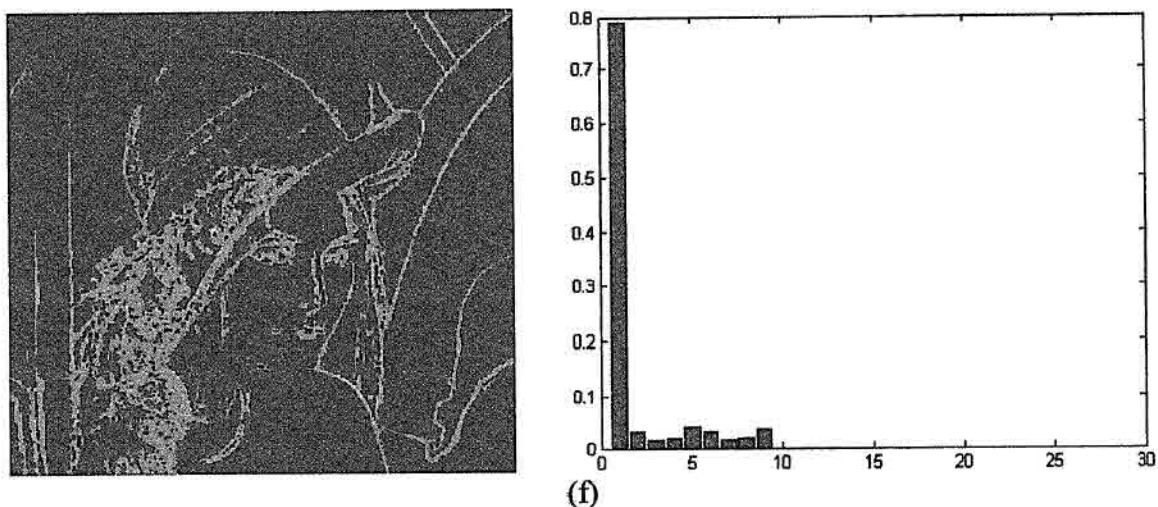


FIGURA 5.35 - Conclusão.

Podemos observar na Figura 5.35(a)-(f), os tipos de bordas encontrados em cada rede. Os histogramas das imagens de bordas mostram a quantidade de vezes que o tipo de borda aparece na imagem. Os valores de não picos são pontos de bordas, e o pico é pontos considerados não bordas.

A Tabela 5.2 apresenta as mesmas diferenças, para os operadores neurais aplicadas sobre as mesmas imagens, mas considerando um limiar de variância igual a 50.

TABELA 5.2 - DIFERENÇAS ENTRE AS IMAGENS DE BORDAS DE CANNY E OS OPERADORES NEURAIS COM LIMIAR 50.

Número de pontos Canny	Número de pontos Rede	Percentual de Pontos de Bordas Canny, em relação às dimensões da imagem original.	Percentual de Pontos de Bordas Rede em relação às dimensões da imagem original.	Erro quadrático médio entre as imagens de bordas de Canny e da Rede	Tipo de Rede	Imagem
39560	54237	0,15	0,21	0,06	BAM	Lena
39560	31337	0,15	0,12	0,11	ART1	Lena
39560	54240	0,15	0,21	0,06	ART2	Lena
39560	54240	0,15	0,21	0,06	BP	Lena
39560	37387	0,15	0,15	0,04	RBF	Lena
39560	54240	0,15	0,21	0,06	LVQ1	Lena
39560	54206	0,15	0,21	0,06	LVQ2	Lena
39560	54240	0,15	0,21	0,06	FAM	Lena
39560	54240	0,15	0,21	0,06	MAXNET	Lena
14055	14286	0,11	0,12	0,03	BAM	Blocos
14055	8832	0,11	0,07	0,06	ART1	Blocos
14055	14286	0,11	0,12	0,03	ART2	Blocos
14055	14286	0,11	0,12	0,03	BP	Blocos
14055	10722	0,11	0,09	0,03	RBF	Blocos

14055	14286	0,11	0,12	0,03	LVQ1	Blocos
14055	14277	0,11	0,12	0,03	LVQ2	Blocos
14055	14286	0,11	0,12	0,03	FAM	Blocos
14055	14286	0,11	0,12	0,03	MAXNET	Blocos
6163	5970	0,19	0,18	0,02	ART1	Pista
6163	3465	0,19	0,10	0,12	BAM	Pista
6163	5970	0,19	0,18	0,02	ART2	Pista
6163	5970	0,19	0,18	0,02	BP	Pista
6163	4843	0,19	0,15	0,04	RBF	Pista
6163	5970	0,19	0,18	0,02	LVQ1	Pista
6163	5967	0,19	0,18	0,02	LVQ2	Pista
6163	5970	0,19	0,18	0,02	FAM	Pista
6163	5970	0,19	0,18	0,02	MAXNET	Pista
12976	61347	0,14	0,66	0,51	BAM	Spot-pan
12976	61343	0,14	0,66	0,51	ART1	Spot-pan
12976	35066	0,14	0,38	0,37	ART2	Spot-pan
12976	61347	0,14	0,66	0,51	BP	Spot-pan
12976	61347	0,14	0,66	0,51	RBF	Spot-pan
12976	51149	0,14	0,54	0,40	LVQ1	Spot-pan
12976	61326	0,14	0,65	0,51	LVQ2	Spot-pan
12976	61347	0,14	0,65	0,51	FAM	Spot-pan
12976	61347	0,14	0,65	0,51	MAXNET	Spot-pan

Podemos observar que a Tabela 5.2 apresenta resultados semelhantes ao da tabela 5.1, ou seja, algumas redes neurais obtiverem o mesmo desempenho, comparando a constatação da pouca influencia do limiar de variância.

A Tabela 5.3 apresenta o tempo de processamento de cada algoritmo para um limiar de variância igual 100.

TABELA 5.3 – TEMPO DE PROCESSAMENTO DE CADA ALGORITMO SOBRE AS DIFERENTES IMAGENS, UTILIZANDO UM LIMIAIR DE VARIÂNCIA DE 100 PARA AS REDES NEURAIIS (EM SEGUNDOS).

Imagem	Lena	Blocos	Pista	Spot pan
Canny	15,8	7,4	2,1430	5,6
Fuzzy	280,0	133,9	111,2	101,2
BAM	39,9	17,3	14,9	18,9
ART1	55,4	25,5	20,5	20,0
ART2	36,6	14,4	4,6	25,2
LVQ1	27,1	11,7	3,3	12,8
LVQ2	27,6	11,8	3,4	12,5
RBF	34,0	13,6	5,7	21,9
BP	29,3	12,3	3,5	14,8
FAM	32,90	13,3	4,3	20,5
Maxnet	194,2	37,2	17,3	164,7

Através da Tabela 5.3, pode-se observar que o algoritmo de Canny é mais rápido. Porém o algoritmo implementado não utilizou nenhum tipo de limiarização. A limiarização foi realizada após a medida do tempo. O algoritmo baseado em lógica nebulosa foi o mais demorado, seguido da rede Maxnet. A rede LVQ1 foi a rede que obteve o menor tempo.

A análise das Tabelas 5.1, 5.2 e 5.3 mostra que, no geral as redes neurais mostraram-se mais lentas do que o operador de Canny, nos experimentos realizados. Possivelmente isso está associado à forma de implementação dos algoritmos. Todas as implementações foram realizadas em ambiente interpretado (MatLab); de forma geral a rede LVQ2 se aproximou mais dos resultados gerados pelo operador de Canny para todas as imagens, considerando um limiar de variância de 100.

CAPÍTULO 6

NAVEGAÇÃO AUTÔNOMA UTILIZANDO REDES NEURAIIS ARTIFICIAIS

Este capítulo mostra a implementação de dois modelos de redes neurais para navegação autônoma, utilizadas nesta dissertação. São apresentados os resultados obtidos e uma revisão bibliográfica relacionada à tarefa de navegação autônoma, utilizando redes neurais artificiais.

6.1 REDES NEURAIIS EM CONTROLE DA NAVEGAÇÃO AUTÔNOMA

Navegação autônoma é um tema muito estudado e pesquisado na área de Inteligência Artificial (Kosaka et al, 1995; Pan et al, 1995; Sargent et al, 1997). Diferentes paradigmas vêm sendo utilizados para abordar o problema, usando diferentes sensores para recuperar a informação do ambiente para uma navegação segura e eficiente.

A navegação autônoma está relacionada com a habilidade de um veículo mover-se sem intervenção humana até alcançar (ou não) o seu objetivo, em um ambiente no qual nenhuma informação a priori está disponível. O objeto móvel é guiado por informações on-line adquiridas enquanto a navegação acontece. Tais tarefas requerem diferentes habilidades para executar ações tais como: alcançar uma certa localização; reagir a situações indeterminadas e imprevistas em tempo real, como por exemplo, um obstáculo inesperado; construir, usar e manter um mapa do ambiente; determinar a posição do objeto móvel na região do mapa, criar planos de navegação com objetivos específicos; e adaptar mudanças no ambiente. A navegação autônoma é uma área multidisciplinar, que integra conhecimentos de várias áreas.

Em geral, os sistemas implementados usam um conjunto de sensores para extrair informações da posição do objeto, do caminho a seguir, e/ou de objetos existentes no ambiente, assim exigindo conhecimento e uso de tecnologias diferentes. Entretanto, quando estamos tentando desenvolver um sistema de navegação autônoma semelhante ao comportamento humano, sob um paradigma de inteligência artificial, há uma

necessidade de focalizar o uso das informações extraídas das imagens, que se constituem na principal fonte de informação, para tarefas de navegação humana em ambientes do mundo real.

De uma forma geral, existem várias aplicações de RNA e técnicas inteligentes na robótica. A utilização de técnicas computacionais não convencionais, em especial as redes neurais artificiais, vem proporcionando avanços significativos no desenvolvimento de sistemas de controle em relação a outras metodologias utilizadas. A característica mais atrativa em seu uso está na sua capacidade de aprendizado, o que a torna capaz de lidar com novas situações de forma segura e robusta.

A Tabela 6.1 apresenta diversos trabalhos na literatura que utilizam imagens e redes neurais para navegação autônoma.

TABELA 6.1 – NAVEGAÇÃO AUTÔNOMA POR RNA

Autores	Ano	RNA	Sensor	Descrição
Huntsberger e Rose	2000	Mapas característicos auto-organizáveis nebulosos (FSOFM)	Câmeras	- Utiliza o sistema neuro-fuzzy para o controle do robô baseado no comportamento para superfícies planetárias.
Mitchell e Keating	2000	Rede de Hopfield Perceptron de Múltiplas Camadas	Ultra-som	- Utiliza duas redes, uma permite que o robô mova-se no ambiente, desviando dos obstáculos, e a outra determina a posição do robô e indica o caminho correto.
Cao et al.	1999	Neuro-fuzzy com algoritmo backpropagation	Câmera	- Utiliza um sistema neuro fuzzy para controlar o veículo através dos ângulos extraídos do ambiente.
Parasnis et al.	1998	Perceptron de Múltiplas Camadas	Câmera	- Compara as redes neurais com a lógica nebulosa. Ambos controla o robô através dos cálculos dos ângulos extraídos do sensor.
Barreto et al.	1997	Maxnet e Selectron	Infravermelhos ultra-som e câmeras	- Utiliza os dois modelos competitivos para resolver o problema do planejamento de trajetória, através do algoritmo de busca, procura o ponto cujo valor calculado pela função heurística é o menor de todos.
Bajula	1996	Perceptron de Múltiplas Camadas	Câmeras	- As imagens da pista são apresentadas para rede como entrada e a direção correta como a saída desejada.
Jochem et al.	1995	Perceptron de Múltiplas Camadas	Câmera Virtual	- A rede neural é utilizada na geração de comandos de direção para o robô através das características obtidas pelas imagens da pista contendo apenas uma única faixa.
Pormealeu.	1995			
Jochem et al.	1995	Incremental Networks Switching	Câmera	- Requer 2 treinamentos, utilizando as imagens da pista, um para controlar o veículo e o outro para reconhecer a pista.

A Tabela 6.1 tem o objetivo de mostrar que o uso das redes neurais na tarefa de navegação autônoma tem sido explorado com o decorrer dos anos. Além disso, apresenta um resumo sobre os trabalhos encontrados na literatura mostrando o tipo de rede e o tipo de sensor utilizado nos trabalhos. Podemos observar que a rede Perceptron de múltiplas camadas é a mais utilizada.

Castro et al. (2001) é um trabalho que utiliza técnicas inteligentes para navegação autônoma, e que serviu de motivação principal para o desenvolvimento desta dissertação. Castro et al. (2001) apresenta o desenvolvimento de um modelo computacional de um sistema de navegação autônoma adaptável, baseado em imagens usando lógica nebulosa. O robô move-se pelo ambiente do qual as imagens fornecem as informações necessárias para navegação. O sistema de lógica nebulosa determina a direção a ser seguida e a velocidade a ser empregada com base nas informações de pista. O sistema, automaticamente, se adapta ao ambiente e corrige a sua trajetória e velocidade baseada nas informações providas pelas imagens. O sistema usa uma abordagem baseada em sensor para navegação autônoma, e apresenta um comportamento análogo a um humano ao navegar em um ambiente. Experimentos foram realizados com um veículo controlado remotamente a partir de um módulo de controle adicional adaptado ao controle remoto.

Nesta dissertação apresenta-se o desenvolvimento de um modelo computacional de um sistema de navegação autônoma adaptável baseado em imagens usando redes neurais artificiais, como continuidade dos trabalhos de Castro et al (2001).

6.2 IMPLEMENTAÇÃO DO CONTROLADOR NEURAL

O sistema de navegação autônoma desenvolvido em Castro et al. (2001) foi concebido para o deslocamento de um robô em um ambiente de pista (rodovia). Os resultados conseguidos e relatados foram obtidos com um protótipo de robô navegando em um pequeno pedaço de pista construída especialmente para os experimentos.

As informações visuais adquiridas estão associadas as características da pista como as faixas laterais que indicam os limites da pista. A extração dessas informações da pista é

realizada por um algoritmo que implementa uma operação de gradiente, permitindo a busca de elementos de bordas na imagem. O operador gradiente busca pelos pontos da imagem localizados na região de alto contraste entre as faixas laterais e a pista. Para tornar o sistema veloz, são localizados pontos em apenas algumas linhas da imagem. Estes pontos, quando conectados, definem segmentos de bordas, associados a região de fronteira entre as faixas laterais e a pista.

Os segmentos determinados por cada dois elementos de bordas encontrados, determinam segmentos cujas orientações são medidas em relação à linha perpendicular ao plano da imagem, que é paralelo ao eixo dianteiro do robô (Figuras 6.2).

São estimados as orientações de 5 segmentos de retas em ambas as faixas laterais (Figura 6.2), das quais se estimam as orientações médias usadas como entrada para a tomada de decisão de navegação, ou seja, quanto a direção a seguir e a velocidade a ser empregada pelo robô.

O operador gradiente usado na extração das características da pista apresenta sensibilidade a ruídos nas imagens, provenientes da transmissão por rádio frequência (VHF) das imagens captadas pela câmera a bordo do veículo em navegação. Esse problema é a principal motivação para a pesquisa por operadores neurais para a detecção de bordas desenvolvida nesta dissertação, explorando as características intrínsecas das redes neurais artificiais. O uso das redes neurais na detecção de bordas como proposto nesta dissertação, está descrito no capítulo 4.

Na Figura 6.1, DE e DD representam os ângulos médios das faixas lateral esquerda e direita, respectivamente. Na saída do sistema neural D e V representam a direção à ser seguida e a velocidade a ser empregada pelo robô, respectivamente.

Na pesquisa desenvolvida nesta dissertação, o controle da navegação autônoma foi implementado utilizando-se dois modelos de redes neurais com aprendizagem supervisionada, treinadas para simular os processos envolvidos na navegação realizada por um humano. O desempenho das redes neurais na navegação é comparado com o

sistema baseado em lógica nebulosa (Castro et al. 2001), cuja parametrização é usada como base para o treinamento da rede neural.

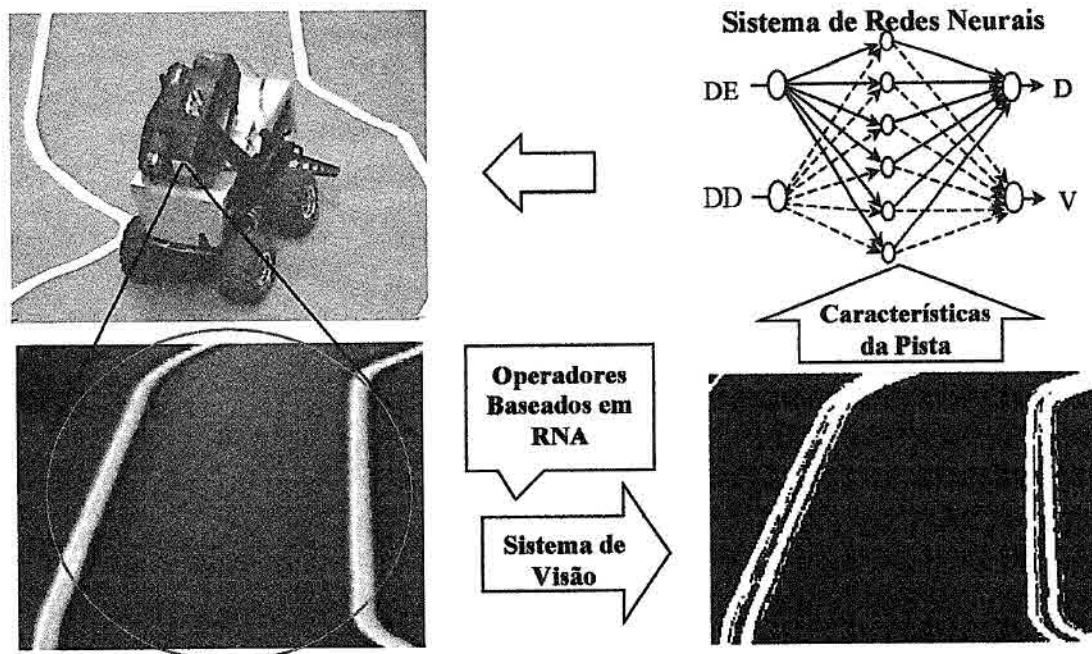


FIGURA 6.1 – Sistema de Navegação Autônoma

6.2.1 Extração das Características da Pista

Sabe-se que a visão humana é sensível à informação de contraste da imagem (Marr, 1982) que permite a extração de contornos dos objetos que as formam. Quando em navegação, o sistema visual humano recebe diferentes estímulos do ambiente. Supondo um humano em um veículo, navegando em uma rodovia. Durante a navegação alguns sentidos são utilizados para executar a tarefa de manter o controle do veículo dentro dos limites da pista.

Uma informação importante para manter a estabilidade da navegação é a direção das faixas laterais existentes na sinalização horizontal da pista. O alto contraste entre a pista (asfalto) e as cores das faixas laterais permite a extração de informações importantes para a tomada de decisão sobre a direção a seguir a cada instante de navegação. Em geral, um motorista que segue as direções das faixas, em determinadas condições de

pista, tem um nível de segurança maior durante a navegação, especialmente quando se guia no período noturno.

O sistema de controle de navegação autônoma, baseado em lógica nebulosa (Castro et al. (2001)) foi concebido para ter características semelhantes ao comportamento de um humano enquanto em navegação. O sistema modela um motorista comum, ou seja, um humano com um grau de instrução suficiente para leitura, escrita e entendimento da sinalização de trânsito, mas com capacidade de raciocínio lógico que permita-o guiar. Durante a navegação considera-se que a tomada de decisão baseia-se na posição atual do veículo e na informação (visual) extraída da pista. O motorista segue proporcionalmente as direções da estrada dadas pelas faixas laterais em relação à linha de visão do motorista, considerada como sendo perpendicular à linha de direção a ser seguida. É natural que o motorista vire para esquerda se ambas as faixas forem para esquerda, ou vire para direita se ambas as faixas forem para direita, podendo realizar curvas suaves, ou seja, com mudança suave na direção. Há situações, entretanto, nas quais o motorista precisa virar muito para a esquerda ou para direita, quando as faixas formam ângulos, à esquerda ou à direita, muitos fechados. Assim, a intensidade da mudança de direção depende da intensidade dos ângulos formados pelas faixas laterais da pista com uma linha perpendicular à linha de visão do motorista (linha branca central na Figura 6.2). A direção das faixas também influencia a velocidade a ser imprimida pelo veículo quando em uma curva mais acentuada ou não na pista.

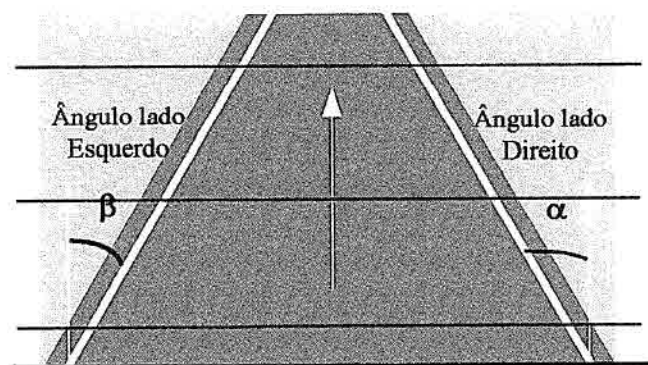


FIGURA 6.2 – Ângulos Formados pelas Faixas Laterais com a Linha Perpendicular à Linha de Visão do Motorista.

É importante notar a incerteza intrínseca no processo, devido à imprecisão do sistema humano na extração das direções das faixas. Em geral, uma olhada rápida na pista é suficiente para ele se decidir, alterando a velocidade e tomando a direção da pista que o mantenha dentro das faixas laterais (ou seja, da pista). Além disso, um motorista humano pode se adaptar a novas situações enquanto executa as ações de navegação. A Figura 6.3 mostra um exemplo de uma situação de estrada no mundo real que um motorista pode enfrentar.



FIGURA 6.3 – Exemplo de Imagem da Estrada na Visão de um Motorista

Deste modo, é possível concluir que dirigir pode ser uma tarefa complexa para a qual os humanos se adaptam com um treinamento permanente e complexo.

O desenvolvimento de sistemas de navegação robótica autônoma utilizando imagens requer operadores apropriados para extração das características da pista. Para os resultados apresentados nesta dissertação emprega-se uma rede neural do tipo LVQ2, (veja a discussão no capítulo 5), que apresentou o melhor desempenho em tempo de processamento, comparando-se com o operador de Canny. A rede LVQ2 foi aplicada ao longo de algumas linhas da imagem para detectar a o ponto de maior contraste local entre a pista e a faixa lateral da esquerda e da direita.

As direções das faixas da pista são tomadas como o valor médio de todos os ângulos medidos. Isto é, possivelmente similar ao modo com que os seres humanos deduzem as direções das faixas enquanto dirige. Como não dispõe de um sistema sofisticado para computar as direções exatas (ângulos), um humano usa a direção aproximada deduzida pela incerteza existente.

6.2.2 Controle da Navegação por Redes Neurais Artificiais

Na implementação do controle de navegação são testados dois modelos de rede neurais: a rede perceptron de múltiplas camadas e a rede de funções de base radial, ambas treinadas de forma supervisionada.

Os dados de treinamento consistem dos resultados obtidos com o sistema nebuloso de Castro et al. (2001), quando apresentado a todas as combinações possíveis dos ângulos da direita e da esquerda, dentro do intervalo de validade das definições do sistema, ou seja, os ângulos variam no intervalo discreto fechado $(-30,30)$. Os resultados obtidos foram usados para o treinamento das redes neurais. Portanto, ambas as redes, depois de treinadas, devem se comportar como o sistema nebuloso desenvolvido em Castro et al. (2001).

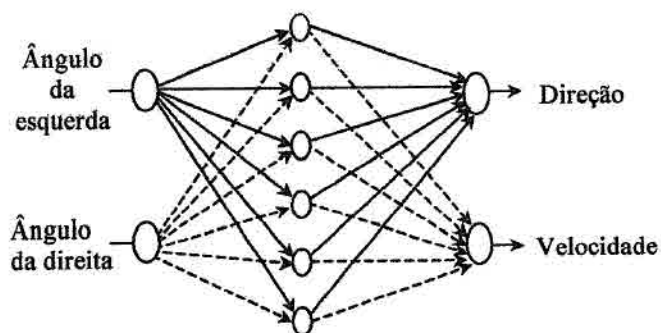


FIGURA 6.4 – Sistema Neural

Na Figura 6.4 tem-se um esboço da arquitetura das redes neurais utilizadas no controle da navegação autônoma. Os resultados obtidos com as duas redes neurais são mostrados na seção seguinte.

6.3 ANÁLISE DOS RESULTADOS DO CONTROLADOR NEURAL

Para a aplicação das redes neurais em navegação autônoma, foi realizado um experimento usando um veículo (robô) controlado remotamente, equipado com uma câmera conectada a um equipamento de vídeo que emite sinais de rádio frequência (RF) capturados por um equipamento de TV, que por sua vez está conectado a uma placa captadora de imagens. Esses dispositivos alimentam um sistema de visão computacional

para extração das características da pista. A Figura 6.5 mostra o veículo usado nos experimentos. A Figura 6.6 mostra um exemplo da janela da cena obtida pela câmera e a imagem da direção das faixas laterais extraídas.

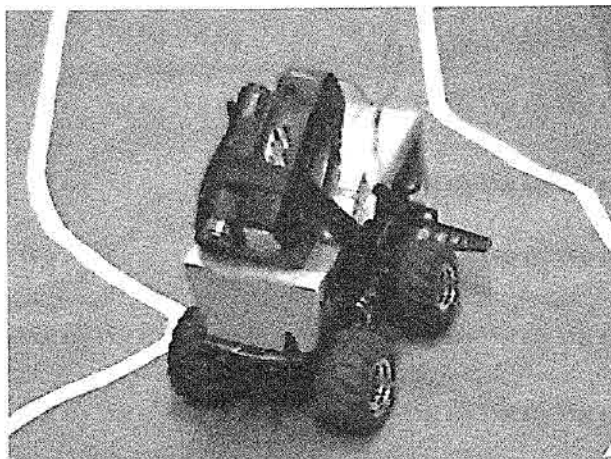


FIGURA 6.5 – Veículo usado nos experimentos

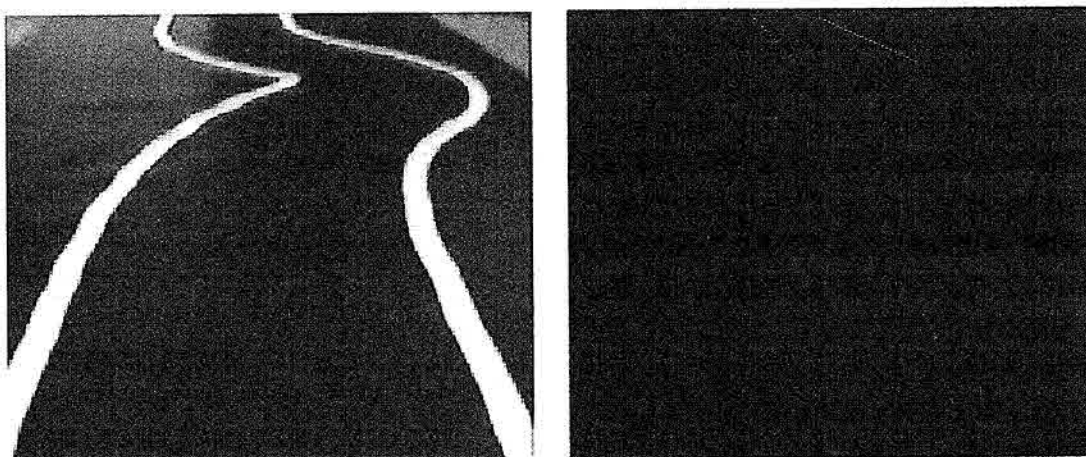


FIGURA 6.6 – (a) Imagem da pista pela câmera; (b) Imagem com informação de bordas das faixas.

Em algumas imagens de bordas pode-se notar que pontos de bordas são confusos para os operadores utilizados, dificultando um bom desempenho do sistema neural e/ou nebuloso. A Figura 6.7 mostra um exemplo da imagem de borda extraída, apresentando pontos confusos.



FIGURA 6.7 – Imagem de borda com alguns pontos confusos.

Para realizar os testes comparativos entre os sistemas neurais e nebulosos, foram feitas seqüências de imagens da pista, com a câmera a bordo do robô. Esta seqüência de imagens foi apresentada ao sistema nebuloso de Castro et al. (2001) e depois apresentada aos sistemas neurais treinados. Foram treinadas duas redes neurais uma RPMC com 5 neurônios na camada escondida e uma RBF com 25 unidades na camada escondida do tipo Gaussiano, ou seja, utilizando função Gaussiana descrita na Equação (2.13).

Os ângulos médios resultantes da esquerda e da direita, bem como o ângulo e a velocidade de saída, computados da imagem na Figura 6.8(b), são mostrados na Tabela 6.2.

TABELA 6.2 – EXEMPLOS DE ÂNGULOS EXTRAÍDOS DA IMAGEM.

	Lógica Fuzzy	Redes Neurais
Ângulo da esquerda	19	19
Ângulo da direita	-20	-20
Direção	0	1.2
Velocidade	6	6.4

A Tabela 6.3 mostra o erro quadrático médio do ângulo, o erro quadrático médio da Velocidade e o tempo médio de processamento dos controladores neurais em relação ao desempenho do controlador nebuloso. Através dessa tabela, podemos observar que a rede RPMC obteve o menor tempo de processamento.

TABELA 6.3 – ERRO QUADRÁTICO MÉDIO EM 29 SEGUNDOS DE NAVEGAÇÃO.

	RPMC	RBF	Fuzzy
Erro quadrático médio do ângulo	18.8082	18.9225	-
Erro quadrático médio da Velocidade	0.7284	1.7988	-
Tempo médio de processamento	0.4674	1.2187	0.4706

A Figura 6.8 mostra um exemplo de situação de pista não prevista no treinamento da rede neural, nem nas regras do sistema de controle baseado em lógica nebulosa. Situações deste tipo precisam ser tratadas através da incorporação de novas regras no sistema.

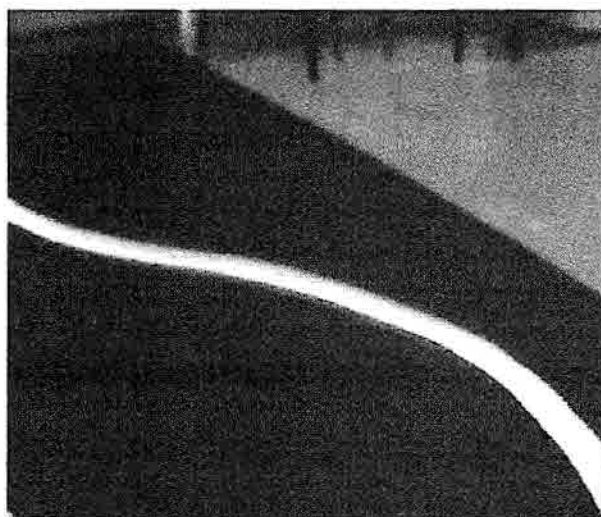


FIGURA 6.8 – Imagem da Pista com alguns pontos confusos.

CAPÍTULO 7

CONCLUSÃO

Nesta dissertação, são apresentados resultados de pesquisa, estudos e experimentações do uso de RNA diferentes na detecção de bordas em imagens e no controle de um veículo em navegação autônoma. Os estudos e experimentos foram realizados procurando mostrar a eficiência das redes neurais como operadores para detecção de bordas, comparando-as com algoritmos de detecção de bordas considerados padrões na literatura de visão computacional. Foram experimentados nove modelos de RNA que utilizam paradigmas de aprendizagem supervisionada e não supervisionada.

Os padrões utilizados nos treinamentos das redes neurais são padrões considerados como possíveis elementos de bordas, partindo-se do princípio que uma linha de borda é formada por pequenos elementos de bordas. Os padrões são apresentados às redes para o treinamento que após serem treinadas podem ser utilizadas para recuperar informação de bordas em imagens. No total foram utilizados 26 padrões de treinamento, definidos em uma janela de tamanho 3x3, codificados em função do tipo de rede neural, como padrões binários (0 e 1) ou bipolares (-1 e 1). A escolha desses padrões e a operação básica dos neurônios, que compõem alguns modelos de redes utilizadas, mostram que as redes neurais comportam-se como filtros diferenciais, semelhantes aos operadores de bordas baseados em gradientes, como por exemplo, Sobel, Roberts e Canny.

Foi observado, que o número de padrões utilizados no treinamento influencia o desempenho em apenas um modelo de rede, a ART1. Outro fator importante é a escolha do parâmetro limiar de variância que evita processamento excessivo; a escolha do limiar influencia os resultados. Observou-se que a escolha de um limiar inadequado pode implicar em perdas de informação de bordas. A análise comparativa visual com o operador de Canny, tomado como operador padrão (Shin et al. 2001), apresenta resultados semelhantes com as redes apresentando bordas mais definidas na intensidade, por terem saídas binárias.

A análise quantitativa envolveu a limiarização das imagens de bordas obtidas pelo operador de Canny pelo método de Rosin (2001). A idéia da limiarização é ter uma imagem com bordas significativas, dispensando as bordas com baixa intensidade que, possivelmente, não são detectadas pelos operadores neurais, dependendo do limiar de variância escolhido.

Quantitativamente, foram comparados os números de pontos de bordas encontrados por cada operador, o percentual em relação às dimensões da imagem, o erro quadrático médio entre as imagens de bordas produzidas pelos operadores neurais e aquelas produzidas pelo operador de Canny.

Um operador baseado em lógica nebulosa (Cho e Cho, 1994) foi implementado, mas devido ao tempo de processamento ser extremamente mais elevado do que os operadores de redes neurais, não foi considerado para efeito de comparação.

As Tabelas 5.1, 5.2 e 5.3 formam uma base de informações que podem ser usadas para medir o desempenho das redes. O mais importante na análise comparativa entre os resultados foi à concepção da metodologia a ser empregada, pois trata-se de uma metodologia simples, empregando resultados e métodos já testados na literatura, mas arranjados de uma forma lógica que pode levar a estudos de comparação dos algoritmos. Na análise feita, pode-se concluir que o algoritmo LVQ2, no geral, mostrou resultados mais compatíveis com aqueles obtidos pelo operador de Canny.

Nesta dissertação também foi descrita a implementação de duas RNA para um sistema de navegação autônoma para robôs móveis. A interação com o ambiente é realizada através de um sistema de visão que fornece imagens do ambiente das quais são extraídas as informações para navegação. O desempenho das redes neurais utilizadas é comparado com o desempenho do sistema baseado em lógica nebulosa desenvolvido por (Castro et al. 2001) para a mesma tarefa. O sistema baseado em lógica nebulosa determina a direção a ser seguida e a velocidade a ser empregada para cada ponto da pista, em função da informação da direção das faixas laterais de uma pista extraída das imagens.

O controle neural da navegação autônoma foi implementado com dois modelos de redes neurais com aprendizagem supervisionada, treinadas para simular os processos envolvidos na navegação realizada por um humano, usando a parametrização do sistema de Castro et al. (2001) como base para o treinamento.

Os algoritmos de redes neurais mencionados foram experimentados dotando um robô (veículo controlado remotamente) que move-se autonomamente por um ambiente do qual as imagens fornecem as informações necessárias para a tomada de decisão para a navegação. O sistema automaticamente se adaptou ao ambiente e corrigiu a sua trajetória com base nas informações fornecidas pelas imagens.

Os experimentos realizados focalizaram somente a correção da trajetória e a velocidade. As situações inesperadas como obstáculos na pista não foram tratadas. Entretanto, para evitar obstáculos podem ser utilizados pela verificação de alterações das condições da pista após a extração das faixas, utilizando operadores de cálculo de contraste da imagem.

Os resultados encontrados mostram-se promissores, com as redes neurais apresentando desempenho aproximado a um método padrão em visão computacional, na tarefa de detecção de bordas. Ressalte-se que cada modelo de rede neural exigiu adequação dos dados de treinamento devido às suas características de aprendizagem. Assim, conclui-se que as redes neurais podem ser efetivamente usadas como ferramentas para detecção de bordas e como um controlador neural para um sistema de navegação autônoma. Os resultados obtidos mostram o potencial das redes neurais para ambos os problemas. Entretanto, dada à existência de um número crescente de modelos de redes neurais, é possível concluir que muita pesquisa ainda precisa ser realizada para explorar todo o potencial dos sistemas de redes neurais artificiais.

As principais contribuições apresentadas nessa dissertação são:

- Desenvolvimento de uma metodologia para a comparação de resultados dos operadores padrão e das redes neurais artificiais;

- Desenvolvimento de dois sistemas baseados em redes neurais para o controle da navegação autônoma.
- Integração dos sistemas neurais experimentados nesta dissertação (detecção de bordas e controle), para concepção de um modelo computacional adaptável para um sistema de navegação autônoma por imagens, onde o veículo decide o replanejamento de navegação baseado na informação visual.

A principal vantagem em se utilizar redes neurais para as tarefas investigadas nesta dissertação é a possibilidade de implementação em hardware programável, permitindo assim a exploração do paralelismo intrínseco das redes neurais artificiais em aplicações em tempo real.

Como continuidade das pesquisas realizadas até o presente, algumas possibilidades de trabalho que resultaram da realização dos experimentos são:

1. Extensão da metodologia desenvolvida para ambientes multirobôs, configurados segundo uma abordagem multiagentes (múltiplos agentes) inteligentes, organizados em colônias (agrupamentos de robôs com características semelhantes);
2. O uso de neurocomputadores baseados em FPGA (Field Programmable Gate Arrays) para implementação dos modelos neurais, de forma que possam ser embarcados em veículos ou robôs dotados com capacidade de navegação;
3. Estudo de outros métodos de recuperação de informação em imagens, como, por exemplo, informação de 3D, que permita o cálculo da distância de possíveis obstáculos no ambiente que impliquem necessariamente em uma tomada de decisão de navegação.

REFERÊNCIAS BIBLIOGRÁFICAS

- Ahmed, M.N. e Farag, A.A. Two-Stage Neural Network for Volume Segmentation of Medical Images, **Pattern Recognition Letters**, v. 18, n.11-13, p. 1143-1151, November 1997.
- Awcock, G. W. e Thomas, R. **Applied Image Processing**. McGraw-Hill, 1996
- Baluja, S. Evolution of an artificial neural network based autonomous land vehicle controller. **IEEE Transactions on Systems, Man and Cybernetics, Part B**, v.26, n.3, p. 450 - 463, June 1996.
- Barreto, G.A.; Araújo, A.F.R. e Rosa, M.O. Algoritmo de Busca Heurística Usando Redes Neurais Competitivas para Planejamento Ótimo de Trajetória de um Robô Móvel. In: III Congresso Brasileiro de Redes Neurais - CBRN 1997. **Anais**. Florianópolis, SC. , 1997, p. 408-413.
- Basak, J. Chandra, B. e Majumder, D.D. On Edge and Line Linking with Connectionist Models. **IEEE Transactions on Systems, Man and Cybernetics**, v.24, n.3, p. 413-428, 1994.
- Beale, R. e Jackson, T. **Neural Computing: An Introduction**. Institute of Physics Publishing Bristol and Philadelphia: Institute of Physics Publishing, 1990. 240 p.
- Bitterncourt, J.R. Osório, F.S. Adaptive Filters for Image Processing Based on Artificial Neural Networks, **XIII Brazilian Symposium on Computer Graphics and Image Processing**, p. 17-20, December 2000.
- Blanz, W. E. e Gish, S. L. A Connectionist Classifier Architecture Applied to Image Segmentation. In: **Proceedings 10th International Conference on Pattern Recognition**, v. 2, p. 272-277, 1990.

- Blume, M. e Ballard, D.R. Image Annotation Based on Learning Vector Quantization and Localized Haar Wavelet Transform Features. **SPIE Applications and Science of Neural Networks Conference**, p. 181-190, February 1997.
- Braga, A. P.; Carvalho, A.P.L.F. e Ludermir, T.B. **Redes Neurais Artificiais Teorias e Aplicações**. Rio de Janeiro: LTC, 2000. p. 262
- Campadelli, P.; Medici, D. e Schettini, R. Color Image Segmentation Using Hopfield Networks. **Image and Vision Computing**, v. 15, p. 161-166, July 1997.
- Canny, J. F. A Computational Approach to Edge Detection. **IEEE Trans. Pattern Analysis and Machine Intelligence**, v. PAMI, n. 8, p. 679-698, 1986.
- Cao, J.; Liao, X. e Hall, E. Reactive Navigation for Autonomous Guided Vehicle Using the Neuro-fuzzy Techniques. **Proceedings of Spie Technical Conference** [online]. v. 3837, p. 1-10, September 1999. <<http://robotics.uc.edu/pdfdocs/spiejn/jin99>>
- Carpenter, G.A. e Grossberg, S. ART2: Self-Organization of Stable Category Recognition Codes for Analog Input Patterns. **Applied Optics**, v.26, n. 23, p. 4919-4930, 1987.
- Casasent, D. e Psaltis, D. Position, Rotation, and Scale Invariant Optical Correlation. **Applied Optics**, v. 15, n. 7, p. 1795--1799, July 1976.
- Castro, A. P. A.; Silva, J. D. S. e Simoni, P. O. Image Based Autonomous Navigation Fuzzy Logic Control. **IJCNN International Joint Conference on Neural Networks**, v.1, p. 2200-2205, 2001.
- Castro, A.P.A. e Silva, J.D.S. Navegação Autônoma por Imagens com Controle por Redes Neurais. **SBA Controle & Automação**, 2003. (Submetido)
- Chai, M. I. B.; Chai, A. e Sullivan, P. Boundary Detection of Retinoblastoma Tumors with Neural Networks. **Computerized Medical Imaging and Graphics**, v. 25, p. 257-264, 2001.

- Chen, K. e Wang, D. A Dynamically Coupled Neural Oscillator Network for Image Segmentation. **Neural Networks**, v. 15, n. 3, p. 423-439, April 2002.
- Cho, S.M. e Cho, J.H. Thresholding for Edge Detection Using Fuzzy Reasoning Techniques. **IEEE International Conference on Communication Systems (ICCS'94)**, v.3, p. 1121-1124, 1994.
- Cohen, M. A. e Grosseberg, S. Absolute Stability of Global Pattern Formation and Parallel Memory Storage by Competitive Neural Networks. **IEEE Transactions on Neural Networks**, v.2, p. 302-309, 1980.
- Cortes, C. e Hertz, J.A. A Network System for Image Segmentation. In **Proceedings International Joint Conference on Neural Networks**, v.1, p. 121-125, 1989.
- Davis, D.N.; Linying S. e Sharp B. Neural Networks for X-Ray Image Segmentation. **ICEIS 1999**, p. 264-271, July 1999.
- Davis, I. **A Modular Neural Network Approach to Autonomous Navigation**. Robotics Institute. 1-154 p. (Philosophy in Robotics). These - Carnegie Mellon University, June 1996.
- Deriche, R. Fast Algorithms for low level Visio. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v.12, n. 1, p. 78-87, 1990.
- Ding, L. e Goshtasby, A. On the Canny edge detector. **Pattern Recognition**, v. 34, n. 3, p. 721-725, March 2001.
- Duda, R.O. e Hart, P.E. **Pattern Classification and Scene Analysis**. New York: Wiley, 1973. p.
- Fausett, L. **Fundamentals of Neural Networks: Architectures, Algorithms, and Applications**. New Jersey: Prentice Hall, 1994. 461 p.

- Fedorov, D. **Sistema Semi-Automático de Registro e Mosaico de Imagens**. São José dos Campos. 131p. Dissertação (Mestrado em Computação Aplicada) - Instituto Nacional de Pesquisas Espaciais, 2002.
- Filho, L. e Neto, H.V. **Processamento Digital de Imagens**. Rio de Janeiro: Brasport, 1999. 409 p.
- Fonseca, L.M.G. **Notas de Aula de Processamento Digital de Imagens do Curso Pós Graduação em Computação Aplicada**. (Instituto Nacional de Pesquisas Espaciais - INPE, Junho 2000).
- Gleriane, J.M. **Classificação Espectro-Temporal de Culturas Agrícolas Utilizando Redes Neurais**. São José dos Campos. Tese (Doutorado em Sensoriamento Remoto) - Instituto Nacional de Pesquisas Espaciais, Em Andamento.
- Goltsev, A. An Assembly Neural Network for texture Segmentation. **Neural Network**, v. 9, n. 4, p. 643-653, June 1996.
- Gonzalez, R.C. e Woods, R.E. **Processamento de Imagens Digitais**. Editora Edgard Blucher Ltda, 2000 509 p.
- Grosseberg, S. How Does a Brain Build a Cognitive Code? **Psychological Review**, v.87, p. 1-51, 1987.
- Haines, K. e Hecht-Nielsen, R. A BAM with increased Information Storage Capacity. **IEEE International Conference on Neural Networks**, p.181-190, 1988.
- Haykin, S. **Redes Neurais: Princípios e Prática**. Porto Alegre: Bookman, 2001. 900 p.
- Heath, M.; Sarkar, S.; Sanocki, T. e Bowyer, K. Comparison of Edge Detection: A Methodology and Initial Study. **Computer Vision and Image Understanding** [online]. v. 69, p. 38-54, January 1998. <<http://citeseer.nj.nec.com/225984.html>>.

- Hsieh, J.W.; Ko, M.T.; Liao, H.Y.M. e Fan, K.C. A New Wavelet-Based Edge Detector Via Constrained Optimization. **Image and Vision Computing**, v.15, n. 7, p. 511-527, July 1997.
- Huntsberger, T. e Rose, J. Behavior Based Control for Autonomous Mobile Robot. In **Proceedings ROBOTICS** [online], p. 299-305, March 2000. <<http://citeseer.nj.nec.com/article/huntsberger00behaviorbased.html>>.
- Jain, A.K. **Fundamentals of Digital Image Processing**. New Jersey: Prentice-Hall Information and System Sciences Series, June 1989. 569 p.
- Jochem, T. M. Pormerleau, D.A. e Thorpe, C.E. Vision-Based Neural Network Road and Intersection Detection and Traversal. **IEEE Conference on Intelligent Robots and Systems**, v. 3, p. 344-349, August 1995.
- Jochem, T.; Pormerleau, D. e Thorpe, C. Vision Guided Lane transition. **IEEE Symposium on Intelligent Vehicles**, v.25-26, p. 30-35, September 1995.
- Jorgensen, T.M.; Christensen, S.S. e Andersen, A.W. Detecting Danger Labels with RAM-Based Neural Networks. **Pattern Recognition Letters**, v.17, n. 4, p.399-412, April 1996.
- Kohonen, T. Improved Versions of learning vector Quantization. In **Proceeding of IJCNN International Joint Conference on Neural Networks**, v.1, n. 17-21, p. 545--550, June 1990.
- Kohonen, T. Self-Organized Formation of Topologically Correct Feature Maps. **Biological Cybernetics**, v.43, p. 59-69, 1982.
- Kohonen, T. Hynninen, J., Kangas, J., Laaksonen, J. e Torkkola, K. **LVQ_PAK: The Learning Vector Quantization Program Package** - Finland: Helsinki University of Technology, Laboratory of Computer and Information Science, 1996. 1-26 p. (Technical Report A30 - FIN-02150 Espoo)

- Kohonen, T.; Barna, G. e Chrisley, R. Statistical Pattern Recognition with Neural Network: Benchmarking Studies. In **Proceeding IEEE International Conference On Neural Networks**, v. 1, p. 61-68, 1988.
- Kosaka, A. e Pan, J. Purdue Experiments in Model-Based Vision for Hallway Navigation. **Proceeding of Workshop on Vision for Robots in IROS'95**, p. 87-96, 1995.
- Kosko, B. Adaptive Bidirectional Associative Memories, **Applied Optics**, v. 26, n. 23, p. 4947-4959, December 1987.
- Kosko, B. Bidirectional Associative Memories, **IEEE Transactions On Systems, Man and Cybernetics**, v. 18, n. 1, p. 49-60, February 1988.
- Kosko, B. **Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence**. New Jersey: Prentice-Hall, Inc., 1992. 449 p.
- Kulkarni, A.D. **Computer Vision and Fuzzy Neural Systems**. New Jersey: Prentice Hall, 2001. 509 p.
- Li, W.; Lu, G. e Wang, Y. Recognizing White Line Markings for Vision-Guided Vehicle Navigation by Fuzzy Reasoning. **Pattern Recognition Letters**, v. 18, n. 8, p. 771-780, August 1997.
- Lin, W.C.; Tsao, E.C.K. e Chen, C.T. Constraint Satisfaction Neural Networks for Image Segmentation. **Pattern Recognition**, v.25, n. 7, p. 679-693, 1992.
- Lindblad, T.; Becanovic, V.; Lindsey, C.S. e Szekely, G. Intelligent Detectors Modeled from the Cat's Eye. **Nuclear Instruments and Methods in Physics Research**, v. A, n.389, p. 245-250, 1997.
- Ma, Yi. Canny Edge Detector [online], April 1996.
<<http://robotics.eecs.berkeley.edu/~mayi/imgproc/cacode.html>>.

- Marr, D. e Hildreth, E. Theory of Edge Detection. **Proceeding of the Royal Society**, v. B, n. 207, p. 187-217, 1980.
- Mascarenhas, N.D.A. e Velasco, F.R.D. **Processamento Digital de Imagens**. IV Escola de Computação, 1984.
- McIvor, A.M. Edge Linking and Junction Preservation. **Digital Image Computing: Techniques and Applications** [online], p. 72-779, December 1993. <http://www.is.irl.cri.nz/pubdoc/1993/amm-dicta93.ps>.
- Mitchell, R.J e Keating, D.A.; Neural Network Control of a Simple Mobile Robot. In: **Concepts for Neural Networks: a Survey**: Eds Landau, L.J. and Taylor, J.G., Springer Verlag, 1997. p. 95-108.
- Nalwa, V.S. **A Guided Tour of Computer Vision**, Addison-Wesley Publishing Company, June 1993. 361 p.
- Neves, S.C.M. Pelaes, E.G. Estudo e Implementação de Técnicas de Segmentação de Imagens. **Revista Virtual de Iniciação Científica** [on-line]. v. 1, n. 2, p. 1-11, July 2001. <<http://www.ufpa.br/revistaic>>.
- Pan, J; Pack, D., Kosaka, A. e Kak, A.C. FUZZY-NAV: A Vision-Based Robot Navigation Architecture Using Fuzzy Inference for Uncertainty Reasoning. **Proceedings of the WCNN**, v. 2, p. 602-607, July 1995.
- Parasnis, S; Velidandla, S; Hall, E.L. e Anand, S. **A Comparison of Optical Modeling and Neural Networks for Robot Guidance**, v. 3522, 1998. Proceeding Of the Intelligent Robots and Computer Vision XVII.
- Parker, J.R. **Algorithms for Image Processing and Computer Vision**. New York: Wiley Computer Publishing, 1997. 417 p.
- Peli, T. e Malah, D. A Study of Edge Detection Algorithms. **Computer Graphics and Image Processing**, v. 20, p. 1-21, 1982.

- Pomerleau, D.A. Efficient Training of Artificial Neural Networks for Autonomous Navigation. **Neural Computation**, v.3, n. 1, p. 88-97, 1991.
- Pomerleau, D.A. Neural Network Based Vision for Precise Control of a Walking Robot. **Machine Learning**, v. 15, n. 2, p. 125-135, 1994.
- Pormealeu, D.A. Neural Network Vision for Robot Driving In: **The Handbook of Brain Theory and Neural Networks**. Cambridge (MA): M.I.T. Press: M. Arbib, 1995. p. 1761.
- Powell, M. J. D. Radial Basic Functions for Multivariable Interpolation: A review, **IMA Conference on Algorithms for the Approximation of Functions and Data**, p. 143-167, Shrivvenham, Inglaterra, 1985.
- Pratt, W.K. **Digital Image Processing**. New York: Wiley Computer Publishing, 1991. 698 p.
- Prewer, D. **Connectionist Pyramid Powered Perceptual Organization: Visual Grouping with Hierarchical Structures of Neural Networks**. Department of Computer Science: The University of Melbourne, July 1995. 1-39 p.
- Prewitt, J.M.S. **Object Enhancement and Extraction**, In B. S. Lipkin and A. Rosenfeld, editors, **Picture Processing and Psychopictorics**, Academic Press, New York, 1970.
- Rangsanseri, Y.; Thitimajshima, P. e Kanotai, S. Multispectral Image Segmentation Using ART1/ART2 Neural Networks. In: **22nd. Asian Conference on Remote Sensing**, Singapore, April 2001.
- Reddick, W.E.; Glass, J.O. e Cook, E.N. Automated Segmentation and Classification of Multispectral Magnetic Resonance Images of Brain Using Artificial Neural Networks. **IEEE Transactions on Medical Imaging**, v. 16, n. 6, p. 911-918, December 1997.
- Ribeiro, F.F.S. e Lima, A.C.C. Detecção de Volume de Tráfego de Veículos Proporcionada por Visão Computacional Via Redes Neurais. **Proceedings of the IV**

Brazilian Conference on Neural Networks - IV Congresso Brasileiro de Redes Neurais, p. 96-101, July. 1999.

Ridder, D.; Hoekstra, A. e Duin, R.P.W. Feature Extraction in Shared Weights Neural Networks. In: **Proceedings of the Second Annual Conference of the Advanced School for Computing and Imaging**, p. 289-294, July 1996.

Roberts, L.G. **Machine Perception of Three-Dimensional Solids**, In J. P. Tippet et al. editors, **Optical and Electro optical Information Processing**, MIT Press, Cambridge, MA, 1965.

Rosin, P. Unimodal Thresholding. **Pattern Recognition** , v. 34, n. 11, p. 2083-2096, November 2001.

Sargent, R.; Bailey, B., Witty, C. e Wright, A. Dynamic Object Capture Using Fast Vision Tracking. **AI Magazine**, v.18, n.1, p. 65-72, 1997.

Shin, M.C.; Goldgof, D.B.; Bowyer, K.B. e Nikiforou, S. Comparison of Edge Detection Algorithms Using a Structure from Motion Task. **IEEE Transactions On Systems, Man, and Cybernetics - Part B** , v. 31, n. 4, p. 589-601, August 2001.

Silva, J.D.S. Narciso, M. e Rodrigues, V. Associação de Padrões com Memórias Associativas Bidirecionais. In: **Anais.** : Instituto Nacional de Pesquisas Espaciais, Abril 1992.

Silva, J.D.S. **Notas de Aula do Curso de Neurocomputação do Curso de Pós Graduação em Computação Aplicada.** (Instituto Nacional de Pesquisas Espaciais - INPE, 2000).

Simpson, P. K. **Artificial Neural Systems: Foundations, Paradigms, Applications, and Implementations.** San Diego: Pergamon Press, 1990. 209 p.

Srinivasan, S. e Kanal, L. Qualitative Landmark Recognition Using Visual Cues. **Pattern Recognition Letters**, v. 18, n.14, p. 1405-1414, December 1997.

- Tsang, W.H. e Tsang, P.W.M. Suppression of False Edge Due to Specula Reflection in Color Images. **Pattern Recognition Letters**, v.18, n. 2, p. 165-171, February 1997.
- Verikas, A; Malmqvist, K. e Bergman, L. Color Image Segmentation by Modular Neural Network. **Pattern Recognition Letters**, v.18, n.2, p. 173-185, February 1997.
- Wong, H.S.; Caelli, T. e Guan, L. A Model-Based Neural Networks for Edge Characterization. **Pattern Recognition**, v. 33, n. 3, p. 427-444, March 2000.
- Young, S.S.; Scott, P.D. e Nasrabadi, M. Object Recognition Using Multilayer Hopfield Neural Network. **IEEE Transactions on Image Processing**, v.6, n. 3, p. 357-372, March 1997.