

JOSEPH KALIL KHOURY JUNIOR

**DESENVOLVIMENTO E AVALIAÇÃO DE UM SISTEMA DE VISÃO
ARTIFICIAL PARA CLASSIFICAÇÃO DE MADEIRA SERRADA DE
EUCALIPTO**

Tese apresentada à Universidade
Federal de Viçosa, como parte das
exigências do Programa de Pós-graduação
em Engenharia Agrícola, para obtenção do
título de *Doctor Scientiae*

VIÇOSA
MINAS GERAIS – BRASIL

2004

**Ficha catalográfica preparada pela Seção de Catalogação e
Classificação da Biblioteca Central da UFV**

T

K45d
2004

Khoury Junior, Joseph Kalil, 1966-
Desenvolvimento e avaliação de um sistema de visão
artificial para classificação de madeira serrada de euca-
lipto / Joseph Kalil Khoury Junior. – Viçosa : UFV, 2004.
xii, 101f. : il. ; 29cm.

Inclui apêndices.

Orientador: Francisco de Assis de Carvalho Pinto.
Tese (doutorado) - Universidade Federal de Viçosa.

Inclui bibliografia.

1. Processamento de imagens - Técnicas digitais. 2.
Madeira serrada - Classificação. 3. Visão por computador.
4. Sistemas de reconhecimento de padrões. 5. Redes neu-
rais (computação). 6. Análise multivariada. 7. Eucalipto.
I. Universidade Federal de Viçosa. II. Título.

CDD 20.ed. 621.367

JOSEPH KALIL KHOURY JUNIOR

**DESENVOLVIMENTO E AVALIAÇÃO DE UM SISTEMA DE VISÃO
ARTIFICIAL PARA CLASSIFICAÇÃO DE MADEIRA SERRADA DE
EUCALIPTO**

Tese apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-graduação em Engenharia Agrícola, para obtenção do título de *Doctor Scientiae*

APROVADA: 29 de Março de 2004.

Prof. José Tarcísio da Silva Oliveira

Prof. Paulo Marcos de Barros Monteiro

Prof. Ricardo Marius Della Lucia
(Conselheiro)

Prof. Nerilson Terra Santos
(Conselheiro)

Prof. Francisco de Assis de Carvalho Pinto
(Orientador)

'O importante não é apenas alcançar um novo horizonte, mas também as amizades e experiências adquiridas ao longo da caminhada'

À minha esposa Denise pelo apoio em todos os momentos.
Aos meus filhos, Viviane e Fuad, pelos momentos de alegria.
Aos meus pais, que me educaram.

AGRADECIMENTO

A Deus, por tudo.

À Universidade Federal de Viçosa e ao Departamento de Engenharia Agrícola, pela oportunidade de realização do curso e pelos ensinamentos.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pela concessão da bolsa de estudos.

À Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG), pelo financiamento do projeto.

Ao Engenheiro Carlos Gilberto Marques, pela concessão da visita à serraria da Aracruz Produtos de Madeira S.A.

Ao Professor Francisco de Assis de Carvalho Pinto, pela orientação, pela dedicação e apoio à realização deste trabalho e, sobretudo, pela amizade.

Aos Professores, Daniel Marçal de Queiroz, Ricardo Marius Della Lucia, Nerilson Terra Santos e Paulo César Corrêa, pelo apoio, pelas críticas e sugestões no decorrer do trabalho.

Aos colegas da pós-graduação do Laboratório de Projetos de Máquinas e Visão Artificial, Darly, Varella, Cristiano e Enrique, pelas calorosas discussões técnicas.

Aos colegas da pós-graduação Leidy, Roberto, Natanael, Alisson, Jean, José Marcelo, Diogo, Elton, Gilton, João Paulo, Geraldo Magela, Sebastião Eudes, Denilson, Donizette, Luciano, Anderson, José Luiz, Elder, Murilo, Leonardo, Alcir, Renato, Carlos Viliotti, Jorge e Andrea pelo apoio e amizade.

Aos colegas da pós-graduação e funcionários do Laboratório de Propriedades Físicas e Mecânicas da Madeira, Cristovão, Daniel e Cássia, José Tarcisio, Oswaldo e Geraldo, pelo auxílio e amizade.

Aos estagiários, Eduardo, Alexandre, Antonio, Rodrigo e Igor, o apoio durante a condução do trabalho.

Aos funcionários do Laboratório de Mecanização Agrícola Marcelo, Danilo, Wantuil e Geraldo pela amizade e serviços prestados.

Aos funcionários do Departamento de Engenharia Agrícola, Álvaro, Edna, Marcos, Zé Mauro, Galinari, Maria Célia, Joel, Evaristo e Juvercino, pela amizade e serviços prestados.

A todos os professores do Departamento de Engenharia Agrícola

À minha esposa Denise Andrade Khoury e sua família, em especial a seus pais Maria Halfa e Lacyr Andrade, pelo estímulo, pela compreensão e apoio nos momentos de dificuldade.

A meus pais Joseph e Maria, aos meus irmãos Rogéria, Rozani, Amarildo e Leonardo, e a minha Tia Tereza, pelo apoio e bom convívio.

Meus sinceros agradecimentos a todos que, direta ou indiretamente, colaboraram para a realização deste trabalho.

BIOGRAFIA

JOSEPH KALIL KHOURY JUNIOR, filho de Joseph Kalil El-Khoury e Maria da Conceição Brumano Kalil, nasceu em 25 de janeiro de 1966, São Miguel do Anta, Estado de Minas Gerais.

Em 1985, iniciou o curso de Engenharia Agrícola na Universidade Federal de Viçosa, graduando-se em agosto de 1990.

Trabalhou na iniciativa privada e, por ultimo, como professor em curso de nível médio durante o período de 1986 a 1996.

Em 1997, iniciou o Curso de Mestrado em Engenharia Agrícola, área de concentração em Mecanização Agrícola, na Universidade Federal de Viçosa, defendendo tese em fevereiro de 2000.

Em março de 2000, iniciou o Curso de Doutorado em Engenharia Agrícola, área de concentração em Mecanização Agrícola, na Universidade Federal de Viçosa, defendendo tese em março de 2004.

Em março de 2004, foi contratado como professor substituto, no Curso de Engenharia Agrícola, Universidade Federal de Viçosa.

ÍNDICE

RESUMO.....	ix
ABSTRACT	xi
INTRODUÇÃO.....	1
REFERÊNCIAS BIBLIOGRÁFICAS	7
ANÁLISE DISCRIMINANTE PARAMÉTRICA PARA RECONHECIMENTO DE DEFEITOS EM MADEIRA SERRADA DE EUCALIPTO UTILIZANDO IMAGENS DIGITAIS.....	9
RESUMO.....	9
PARAMETRIC DISCRIMINANT ANALYSIS FOR RECOGNIZING THE DEFECTS IN EUCALYPTUS SAWN LUMBER USING THE DIGITAL IMAGES	9
ABSTRACT	9
INTRODUÇÃO.....	10
MATERIAL E MÉTODOS.....	12
RESULTADOS E DISCUSSÃO	17
CONCLUSÕES.....	24
REFERÊNCIAS BIBLIOGRÁFICAS	25
DESENVOLVIMENTO DE REDES NEURAS PARA RECONHECIMENTO DE DEFEITOS EM IMAGENS DE MADEIRA SERRADA DE EUCALIPTO.....	27
RESUMO.....	27
DEVELOPING THE NEURAL NETWORKS FOR RECOGNITION OF DEFECTS IN THE DIGITAL IMAGES OF EUCALYPTUS SAWN LUMBER.....	27
ABSTRACT	27
INTRODUÇÃO.....	28

MATERIAL E MÉTODOS.....	31
RESULTADOS E DISCUSSÃO	40
CONCLUSÕES.....	51
REFERÊNCIAS BIBLIOGRÁFICAS	52
CLASSIFICAÇÃO DE MADEIRA SERRADA DE EUCALIPTO POR VISÃO ARTIFICIAL	55
RESUMO.....	55
CLASSIFICATION OF EUCALYPTUS SAWN LUMBER BY MACHINE VISION SYSTEM	55
ABSTRACT	55
INTRODUÇÃO.....	56
MATERIAL E MÉTODOS.....	58
RESULTADOS E DISCUSSÃO	64
CONCLUSÕES.....	72
REFERÊNCIAS BIBLIOGRÁFICAS	73
CONCLUSÕES GERAIS	75
APÊNDICES	77
Apêndice A.....	78
Código-fonte dos algoritmos implementados no MATLAB e SAS utilizados no capítulo 1	78
Apêndice B	84
Código-fonte dos algoritmos implementados no MATLAB utilizados no capítulo 2	84
Apêndice C	92
Código-fonte dos algoritmos implementados no MATLAB utilizados no capítulo 3	92

RESUMO

KHOURY JUNIOR, Joseph Kalil, D.S., Universidade Federal de Viçosa, março de 2004. **Desenvolvimento e avaliação de um sistema de visão artificial para classificação de madeira serrada de eucalipto.** Orientador: Francisco de Assis de Carvalho Pinto. Conselheiros: Daniel Marçal de Queiroz, Nerilson Terra Santos e Ricardo Marius Della Lúcia.

Na seleção de madeira serrada para fabricação de móveis e utilização em projeto de interiores, a aparência é o requisito de maior importância. Classificando e selecionando-se as peças, pode-se conseguir um produto com a aparência desejada. Dessa forma, a indústria de madeira serrada tem dispensado especial atenção às etapas de classificação e seleção de madeira serrada. A etapa de classificação visual é considerada uma atividade estressante, em razão da exigência de concentração para identificar e mensurar defeitos nas peças de madeira e classificá-las por tempo prolongado. Os sistemas de classificação, que utilizam imagens digitais no processo de automatização são conhecidos como sistemas de visão artificial. No presente trabalho, o objetivou-se desenvolver um algoritmo com base em processamento de imagens digitais para classificação de madeira serrada de eucalipto, para aplicação em um sistema de visão artificial. Para o desenvolvimento e testes dos algoritmos, foram coletadas imagens digitais coloridas de peças de madeira serrada de eucalipto, secas ao ar e previamente aplainadas. As amostras foram escolhidas de forma que estariam presentes, além de madeiras limpas, os defeitos visuais que afetavam a qualidade da peça de madeira serrada. Características obtidas de imagens de defeitos da madeira foram utilizadas para desenvolver e testar dois tipos de classificadores: um

classificador estatístico e outro com base em redes neurais. Um algoritmo foi implementado, para classificação de madeira serrada com base na norma ABNT - NBR 14806 (Madeira serrada de eucalipto – Requisitos), que estabelece os requisitos exigidos para madeira serrada de eucalipto, proveniente de florestas plantadas. As características utilizadas como descritores dos defeitos e madeira limpa foram percentis de blocos da imagem das bandas do vermelho, verde e azul. Foram testados dois tamanhos de blocos da imagem 64x64 pixels e 32x32 pixels. Os resultados das exatidões globais, encontrados nestes dois tamanhos de blocos para os classificadores estatísticos, foram de 81% e 76%, enquanto para as redes neurais foram de 83% e 77%, respectivamente. O classificador estatístico apresentou melhor identificação em algumas classes específicas de defeitos. O algoritmo para classificação das faces das peças de madeira serrada foi implementado, utilizando-se o classificador estatístico e blocos de 32x32 pixels para identificar os tipos de defeito e madeira limpa. A exatidão global foi de 65%, na classificação de cada face larga da peça independentemente. Entretanto, a exatidão foi de 90% na classificação da madeira serrada, em que ambas as faces foram avaliadas simultaneamente. A ocorrência de erros proveniente da classificação não foi devida à identificação de madeira limpa, mas, principalmente, à confusão entre alguns tipos de defeitos.

ABSTRACT

KHOURY JUNIOR, Joseph Kalil, D.S., Universidade Federal de Viçosa, March 2004.
Developing and evaluating an machine vision system for grading of the eucalyptus sawn lumber. Adviser: Francisco de Assis de Carvalho Pinto.
Committee members: Daniel Marçal de Queiroz, Nerilson Terra Santos and Ricardo Marius Della Lúcia.

In selecting the sawn lumber for production of furniture manufacture and utilization in interiors design, a very important requirement is the appearance. The grading and selection of the boards are fundamental stages for getting a product with the desired appearance. So, the sawn lumber industry has giving a special attention to these stages. The visual classification stage is considered as a stressful activity, because its requires special concentration in identifying and measuring the defects in the boards, as well as to classify them for an extended time. The grading systems using digital images in the automation process are known as machine vision systems. This study aimed at the development of an algorithm based on the digital image processing for grading the eucalyptus sawn lumber to be applied on a machine vision system. Some colored digital images of the air-dried and flattened eucalyptus boards were collected in order to develop and test the algorithms. The board samples were chosen in such a way that would show clear wood as well as the visual defects affecting the quality of the board. The characteristics obtained from the lumber defect were used to developing and testing two classifier types: a statistical one and the artificial neural network. Finally, an algorithm was implemented for the board classification based on the Brazilian norm ABNT - NBR 14806 (Eucalyptus sawn Wood– Requirements) that

establishes the requirements for eucalyptus sawn lumber from the planted forests. The characteristics used as descriptors of the defects and clear wood were the percentiles of the red, green and blue bands of the image blocks. Two block sizes of the image (64x64 and 32x32) pixels were tested. The overall accuracy found in these two block sizes for the statistical classifiers were 81% and 76%, whereas the neural networks reached 83% and 77%, respectively. And the statistical classifier showed a better identification for some specific defect classes. The algorithm for grading the face of the sawn lumber boards was implemented, by using the statistical classifier and the boards (32x32 to pixels) to identify both the defect types and clear wood. The overall accuracy was 65%, when classifying independently each wide face of the board. However, an accuracy of 90% was attained, when classifying the sawn lumber in which both faces were simultaneously evaluated. The occurrence of the errors upon classification were not due to the identification of clear wood, but rather to the embarrassment among some kinds of defects.

INTRODUÇÃO

Atualmente, há uma crescente preocupação na indústria madeireira no sentido de utilizar matéria-prima de forma sustentável, pois, os consumidores finais preferem produtos que não causam danos ao meio ambiente, evitando o corte de árvores de florestas nativas. As florestas de eucalipto, as mais cultivadas atualmente, são utilizadas, principalmente para fins energéticos, estruturais, produção de celulose e, recentemente, para indústrias de móveis e uso em projetos de interiores.

Quando a madeira serrada é utilizada para esses últimos fins, a aparência é o fator que mais influi na qualidade do produto. Entretanto, as madeiras de eucalipto, principalmente aquelas provenientes de florestas sem manejo adequado, nem sempre apresentam as aparências demandadas, tão facilmente conseguidas nas madeiras centenárias de origem nativa. Para adequá-las, são necessários tratamentos mais especializados, que abragem desde os tratos silviculturais até a seleção final da madeira serrada.

A etapa de classificação e seleção das peças é muito importante, pois, é quando se consegue um produto com qualidade superior. Esta etapa é, geralmente, realizada por classificadores humanos, sendo um dos pontos de limitação do aumento da produção na indústria de madeira serrada. Acrescenta-se que a inspeção visual é uma atividade monótona e estressante.

HUBER et al. (1985) avaliaram o desempenho de seis classificadores de tábuas de Carvalho Vermelho, em três indústrias. Na metodologia para pontuação, foram considerados o reconhecimento, a identificação e localização dos defeitos em tábuas,

obtendo-se um desempenho médio de 68%. Os autores concluíram que um sistema automatizado para esta tarefa não precisaria, necessariamente, ser perfeito. Qualquer melhoria seria aceitável, pois, o desempenho dos classificadores foi relativamente baixo. Além disto, os equipamentos não são afetados por fadiga e outros fatores humanos, que podem diminuir o rendimento.

A cor da madeira é um fator que vem sendo estudado há décadas, como parâmetro à avaliação de sua qualidade. WEBB et al.(1964) e SULLIVAN (1967), por meio de características da cor obtidas pelas curvas espectrais na faixa do visível, estudaram o desbotamento em madeiras devido à ação de luz ultravioleta e exposição à umidade.

LEBOW et al. (1996) estudaram as características das curvas espectrais, também na faixa do visível, de oito tipos de defeitos em chapas de madeira de Abeto, utilizando-se análise de componentes principais. Os resultados mostraram que, apenas, dois componentes principais explicavam 97% da variação das curvas espectrais dos defeitos. Ao classificarem as amostras, utilizando função discriminante quadrática com esses dois componentes principais, a exatidão foi de 80%. Entretanto, quando utilizaram sete componentes principais a exatidão foi de 99%.

Uma das opções para automatização do processo de seleção e classificação de madeira serrada consiste em substituir por um sistema artificial as etapas de classificação realizado, visualmente, pelo ser humano. Nestes sistemas, a cena é adquirida, processada e uma decisão é tomada. Um sistema de inspeção automatizada por visão artificial pode operar em alta velocidade, ininterruptamente, e prover qualidade superior a classificadores humanos (GONZALEZ & WOODS, 1992).

As câmeras RGB com fotossensores sensíveis aos comprimentos de onda, na faixa de 400 a 1000nm, constituem um dos métodos de aquisição de imagens mais estudados por pesquisadores para classificação de superfícies de madeiras (SZYMANI et al., 1981; KLINE et al., 1998; HANEY et al., 1994; LU, 1997; KAUPPINEN, 1999).

A cor é adquirida por três sensores, ajustados aos comprimentos de ondas referentes às cores primárias: vermelho, verde e azul. Deste modo, a cor adquirida por essas câmeras é representada por um sistema de coordenadas tridimensional. Outros

espaços de cor podem ser criados, a partir de transformações lineares e, ou não lineares do espaço RGB como YIQ, rgb, escala de cinza, HSI, XYZ, $L^*a^*b^*$ e etc. Entretanto, pesquisas têm demonstrado que o espaço RGB tem melhor desempenho na classificação de madeiras.

LU (1997) concluiu que, na classificação da cor de peças de madeira para confecção de painéis colados, o espaço RGB foi o mais apropriado porque o contraste entre áreas escuras e claras da imagem é importante para percepção humana. Em outros espaços como XYZ e rgb (RGB normalizado), essas características são minimizadas. Mesmo outros espaços de cor, como HSI, que mensuram essas características e descorrelacionam a variação da iluminação, não justificou o seu emprego devido à perda, em tempo computacional, para transformação do espaço de cor.

KAUPPINEN (1999), ao testar um algoritmo de classificação de defeitos em madeira, transformou o espaço de cor RGB para outros sete e concluiu que a melhor classificação foi obtida com o espaço RGB. Outra informação importante foi a baixa diferença na classificação comparado com a cor transformada para escala de cinza, evidenciando, assim, a possibilidade de utilização de câmeras monocromáticas.

No processo de inspeção automatizado de madeira, devem ser alcançados os seguintes requisitos: desempenho igual ou superior ao de classificadores humanos; operação em tempo real; insensibilidade ao meio industrial (variação da iluminação, vibrações e presença de pó); possibilidade de mudança nos parâmetros de classificação; possibilidade de identificar os defeitos, assim como quantificar o tamanho, tipo e localização; e obtenção de informação estatística dos defeitos para avaliação da frequência do problema (SZYMANI et al., 1981; PHAM & ALCOCK, 1998). O sistema deve ser robusto, ter custo compatível com os benefícios alcançados, ser isento de radiações nocivas ao operador e de fácil montagem e operação.

Um sistema de visão artificial para inspeção de superfícies é constituído das seguintes etapas: aquisição, processamento e análise de imagens (PHAM & ALCOCK, 1998). Na primeira etapa, utilizam-se diversos tipos de sensores, sendo os sensores ópticos os mais utilizados. A etapa de processamento da imagem tem como objetivo a segmentação (realce) e reconhecimento (identificação e medição) dos objetos de

interesse (KLINE et al., 1992). Uma das técnicas para a segmentação desses objetos utiliza as características que melhor distinguem uma classe da outra. Na última etapa, a informação é transformada de sua forma de imagem digital para expressões, que afetam a classificação da peça da madeira, do tipo “peça da classe A”, “peça descartada”, etc.

Segundo PHAM e ALCOCK (1998), quando implementados na indústria, sistemas de visão artificial não apresentam o mesmo desempenho da etapa de desenvolvimento, degradando a exatidão da classificação. Geralmente, este fato ocorre em razão de os algoritmos desenvolvidos para esses sistemas não se ajustarem às variações e imprevistos, que ocorrem no meio industrial. No entanto, esses sistemas têm se mostrado superiores à baixa exatidão de classificação, realizada por classificadores humanos. Os autores concluíram que a maior dificuldade, encontrada nesse tipo de tecnologia, refere-se ao desenvolvimento de algoritmos para a etapa de processamento de imagens.

SCHMOLDT et al. (1997) identificaram defeitos em toras de madeira, por meio de tomografia computadorizada, em uma serraria. Em espécie de folhosas, o algoritmo foi capaz de identificar, distintamente, com 95% de exatidão: casca, nós, podridão, trincas e madeira limpa.

QUIN et al. (1998) desenvolveram um sistema, com base em câmera infravermelha, para identificar nós em madeiras de pinos aquecidas previamente. Apesar do sistema identificar, eficientemente, os nós na madeira, o período necessário para aquecê-la inviabilizou o uso desse sensor em linhas de produção.

KOIVO e KIM. (1989) desenvolveram um algoritmo de sistema de visão artificial, para classificar nove classes de defeitos em tábuas de Carvalho Vermelho. Imagens foram adquiridas por câmera monocromática com 255 níveis de cinza. Inicialmente, os pesquisadores verificaram que funções discriminantes lineares, em um espaço bidimensional de características convencionais (média e desvio-padrão dos níveis de cinza) separaram somente três classes. Assim, os autores propuseram outras cinco características, obtidas a partir da modelagem da textura de sub-blocos da imagem. Utilizando-se uma árvore de decisão com os três componentes principais deste vetor de características, foi possível diferenciar nove classes de interesse (casca,

madeira limpa, podridão, furo, nós, inclusão de mineral, racha, mancha, esmoado). A exatidão da classificação foi de 97,2 %.

HANEY et al. (1994) construíram um protótipo equipado com sistema de visão artificial para separação em 4 classes por semelhança de cor de madeiras de lenho inicial e lenho tardio. Na aquisição das imagens, utilizou-se câmera RGB e iluminação controlada de lâmpadas incandescentes halógenas. Foi desenvolvida uma rede neural para classificar cada pixel da imagem em função dos seus valores nas bandas Vm, Vd e Az. O sistema desenvolvido foi capaz de trabalhar, com eficiência, a uma velocidade da esteira transportadora de 110 metros por minuto.

LU e TAN (1998) desenvolveram um algoritmo para classificação da grã da madeira (direção, tamanho, arranjo e aparência das fibras ou partículas da madeira) em tábuas de Nogueira. As imagens digitais das peças foram adquiridas no formato RGB e transformadas para escala de cinza, pois, a cor tem pouca caracterização das fibras de madeira. No desenvolvimento do algoritmo o maior esforço ocorreu na segmentação das linhas das fibras do fundo. Depois de segmentadas, os formatos das linhas foram extraídos como características. O classificador empregado foi o de distância mínima e a exatidão do algoritmo foi de 97,4%.

PACKIANATHER e DRAKE (2000) desenvolveram uma árvore de decisão, com base em redes neurais. Alimentando essas redes neurais com 11 características, obtidas em imagens monocromáticas de madeira de Videiro, distinguiram-se 13 classes: casca, madeira limpa, linha colorida, grã da madeira, descoloração, furos, nós de acícula, madeira lanosa, nós sadios, rachas, listas e furo de inseto. A exatidão do algoritmo foi de 96%.

PHILIPP e RATH (2002) avaliaram diferentes espaços de cores, função discriminante linear, função discriminante quadrática, transformação canônica, $l_1l_2l_3$, HSI, HSV e LAB, pela transformação a partir do espaço RGB, os quais foram comparados com base no erro de classificação, ao separar solo e planta em imagens digitais. A função discriminante logarítmica foi a que obteve menor erro de classificação, 2%. Entretanto, para trabalho em tempo real, foi a de maior exigência do sistema computacional, em comparação com os métodos de limiarização pelo histograma.

A definição da técnica utilizada em cada componente de um sistema de visão artificial é, altamente, dependente da cena (objeto e iluminação). Acrescenta-se que cada indústria utiliza seus próprios parâmetros de classificação da madeira serrada. A norma NBR 14806 (ABNT, 2002a) estabelece os requisitos para a madeira serrada de eucalipto, proveniente de florestas plantadas, enquanto a norma NBR 12551 (ABNT, 2002b) estabelece as especificações dos defeitos de madeira serrada de coníferas e folhosas, provenientes de florestas plantadas e nativas para uso geral. Tais normas facilitam a definição dos parâmetros de classificação de madeira serrada.

No presente trabalho, a hipótese é que a classificação de madeira serrada de eucalipto pode ser eficientemente realizada, utilizando-se técnicas de processamento e análise de imagens. Desta maneira, disponibilizará para as indústrias uma tecnologia que lhes permita usufruir dos benefícios da automação, unindo-se esforços para o desenvolvimento de técnicas de processamento de imagens, que incluam a base de conhecimento de classificação de madeiras de eucalipto.

No presente trabalho, foi desenvolvido um algoritmo com base em processamento de imagens digitais para classificação de madeira serrada de eucalipto, para aplicação em um sistema de visão artificial. No primeiro capítulo avaliou-se, por meio de técnicas de análise multivariada, a capacidade de discriminar defeitos em madeira serrada de eucalipto, utilizando as características de percentis de imagens coloridas. No segundo capítulo, foram desenvolvidas e testadas redes neurais artificiais para discriminar defeitos em madeira serrada de eucalipto, utilizando as características de percentis de câmeras coloridas. No terceiro capítulo, foi desenvolvido e testado um algoritmo para um sistema de visão artificial, para classificação de madeira serrada de eucalipto pela aparência, de acordo com os requisitos das classes de qualidade mencionados na norma ABNT - NBR 14806.

REFERÊNCIAS BIBLIOGRÁFICAS

- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS – **ABNT**. Madeira serrada de eucalipto – Requisitos, NBR 14806. Rio de Janeiro, 2002a. 11 p.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS – **ABNT**. Madeira serrada – Terminologia, NBR 12551. Rio de Janeiro, 2002b. 10 p.
- GONZALEZ, R.C.; WOODS, R.E. **Processamento de imagens digitais**. São Paulo: Editora Edgard Blücher Ltda, 1992, 509p.
- HANEY, L.; PRECETTI, C.; GIBSON, H.; KRUTZ, G.; CASSENS, D. Color matching of wood with a real-time machine vision system. **ASAE Paper No. 943579**, ASAE Meeting Presentation, St Joseph, Michigan, 1994, 15 p.
- KAUPPINEN, H., **Development of a color machine vision method for wood surface inspection**. Oulu, Finlandia: Department of Electrical Engineering and Infotech Oulu, - University of Oulu, Dissertation, 138p, 1999.(<http://herkules oulu.fi/issn03553213/>)
- HUBER, H. A.; McMILLIN, C. W.; McKINNEY, J. P.; Lumber defect detection abilities of furniture rough mill employees. **Forest Products Journal**, 35(11/12), p. 79-82, 1985.
- KLINE, D. E.; WIDOYOKO, A.; WIEDENBECK, J. K.; ARAMAN, P.A. Performance of color camera machine vision in automated furniture rough mill systems. **Forest Products Journal**, 48(3), p. 38-45, 1998.
- KLINE, D. E.; HOU, Y. J.; CONNERS, R. W.; SCHOLMOLDT, D. L.; ARAMAN, P.A. **Lumber scanning system for surface defect detection**, ASAE Paper No. 923582. St. Joseph, Mich.: ASAE, 1992.

- KOIVO, A. J.; KIM, C. W.; Automatic classification of surface defects on red oak boards. **Forest Products Journal**, 39(9), p. 22-30, 1989.
- LEBOW, P. K.; BRUNNER, C. C.; MARISTANY, A. G.; BUTLER, D.A. Classification of wood surface features by spectral reflectance. **Wood and Fiber Science**, 28(1), p. 74-90, 1996.
- LU, Q. **A real-time system for color sorting edge-glued panel parts**. Blacksburg, Virginia: Faculty of the Virginia Polytechnic Institute and State University, Dissertation, 119 p, 1997.
- LU, W.; TAN, J. Wood grain pattern characterization and classification. **ASAE Paper No. 98-3045**, ASAE Meeting Presentation, University of Missouri-Columbia, Orlando, Florida, 11 p, 1998.
- PACKIANATHER, M. S.; DRAKE, P. R. Neural networks for classifying images of wood veneer part 2. **International Journal of advanced manufacturing Technology**, 16(6), p.424-433, 2000.
- PHAM, D.T.; ALCOCK, R.J. Automated grading and defect detection: a review. **Forest Products Journal**, 48(4), p34-42, 1998.
- PHILIPP, I.; RATH, T. Improving plant discrimination in image processing by use of different colour space transformations. **Computers and Electronics in Agricultural**, 35, p-1-15, 2002.
- QUIN JR., F.; STEELE, P. H.; SHMULSKY, R. Locating knots in wood with an infrared detector system. **Forest Products Journal**, 48(10), p.80-84, 1998.
- SCHMOLDT, D. L.; LI, P.; ABBOTT, A. L. Machine vision using artificial neural networks with local 3D neighborhoods. **Computers and electronics in agriculture**, 16, p.255-271, 1997.
- SULLIVAN, J. D. Spectrophotometry and wood color. **Forest Products Journal**, 17(7), p. 43-48, 1967.
- SZYMANI, R.; MCDONALD, K. A. Defect detection in lumber: state of the art. **Forest Products Journal**, 31(11), p. 34-44, 1981.
- WEBB, D. A.; SULLIVAN, J. D. Surface effect of light & water on wood, **Forest Products Journal**. 14(1), p. 531-534, 1964.

ANÁLISE DISCRIMINANTE PARAMÉTRICA PARA RECONHECIMENTO DE DEFEITOS EM MADEIRA SERRADA DE EUCALIPTO UTILIZANDO IMAGENS DIGITAIS

RESUMO

Neste trabalho, foram realizadas as análises discriminantes linear e quadrática para classificação de defeitos e madeira limpa em imagens digitais de madeira serrada de eucalipto. As características de percentis do histograma das bandas do vermelho, verde e azul, retiradas de dois tamanhos de blocos de imagens, foram utilizadas para desenvolvimento e teste das funções discriminantes. Foram utilizados 492 blocos, contendo os 12 defeitos e madeiras limpas estudados, retiradas das imagens de 40 peças de madeira serrada aleatoriamente amostradas. As características foram analisadas com seus valores originais, escores dos componentes principais e escores das variáveis canônicas. Os menores erros globais de classificação foram 19% e 24% para funções discriminantes lineares com os escores das variáveis canônicas para tamanho de bloco de 64x64 e 32x32 pixels, respectivamente. Tendo em vista a magnitude destes erros, as características de percentis foram consideradas adequadas para discriminar defeitos e madeira limpa em imagens digitais.

Palavras-chave: Processamento de imagem, reconhecimento de padrões, classificação de tábuas.

PARAMETRIC DISCRIMINANT ANALYSIS FOR RECOGNIZING THE DEFECTS IN EUCALYPTUS SAWN LUMBER USING THE DIGITAL IMAGES

ABSTRACT

The linear and the quadratic discriminant analyses were performed to classify the defects and clear wood in digital images of the eucalyptus sawn lumber. The characteristics of the histogram percentiles of the red, green and blue bands, taken from two sizes of the image blocks, were used for developing and testing the

discriminant functions. Four hundred ninety two blocks containing those twelve defects and clear lumber taken from the images of 40 sawn boards that were randomly sampled. The characteristics were analyzed, by using their original values, main component scores and canonical variable scores. The least overall errors upon classification were 19% and 24% for linear discriminant functions with the scores of the canonical variables for a block size of 64x64 and 32x32 pixels, respectively. Taking into account the magnitudes of those errors, the percentile characteristics were considered as appropriate ones to discriminate the defects and clear wood in the digital images.

Key words: Image processing, pattern recognition, board grading

INTRODUÇÃO

Na seleção de madeira serrada para fabricação de móveis e utilização em projeto de interiores, a aparência é um importante requisito. Classificando e selecionando as madeiras serradas, pode-se conseguir um produto com a aparência desejada, mesmo em madeiras oriundas de florestas plantadas sem manejo adequado. Desta forma, a indústria de madeira serrada tem dispensado especial atenção às etapas de classificação e seleção de madeira serrada. Estas etapas são, geralmente, realizadas por classificadores humanos. A etapa de classificação é considerada uma atividade estressante, em razão da exigência de concentração para identificar e mensurar defeitos das madeiras e classificá-las por tempo prolongado.

Os sistemas que utilizam as informações, obtidas por meio da energia eletromagnética refletida pelos objetos, para automatizar processos são conhecidos como Sistema de Visão Artificial (SVA). Um SVA para inspeção de superfícies consiste das seguintes etapas: aquisição, processamento e análise de imagens (PHAM & ALCOCK, 1998).

Na primeira etapa, são utilizados diversos tipos de sensores para aquisição de imagens, sendo os sensores ópticos os mais utilizados. A etapa de processamento da imagem tem como objetivo a segmentação (realce) e reconhecimento (identificação e mensuração) dos objetos de interesse (KLINE et al., 1992). Na etapa de análise, a

informação é transformada de sua forma de imagem digital para expressões, que afetam a classificação da madeira serrada: peça da classe A, peça descartada, etc.

Um SVA, quando implementado na indústria, não apresenta o mesmo desempenho da etapa de desenvolvimento, degradando a exatidão da classificação. Geralmente, este fato ocorre devido aos algoritmos desenvolvidos para esses sistemas não se ajustarem às variações e imprevistos, que ocorrem no meio industrial. No entanto, esses sistemas têm se mostrado superiores à baixa exatidão de classificadores humanos (HUBER et al., 1985). Neste tipo de tecnologia, a maior dificuldade refere-se ao desenvolvimento de algoritmos na etapa de processamento das imagens, por estarem diretamente relacionados ao erro do sistema, (PHAM & ALCOCK, 1998; KLINE et al., 2001).

As câmeras coloridas, com fotosensores sensíveis nas bandas do vermelho (Vm), verde (Vd) e azul (Az) são um dos métodos de aquisição de imagens mais estudados por pesquisadores na classificação de superfícies de madeiras (SZYMANI & MCDONALD 1981; KLINE et al., 1998; HANEY et al., 1994; LU, 1997; KAUPPINEN, 1999). Estas pesquisas têm demonstrado que o espaço de cor Vm, Vd e Az tem melhor desempenho na classificação de madeiras, quando comparado com câmeras monocromáticas.

LEBOW et al. (1996) estudaram as características das curvas espectrais, adquiridas por espectroradiômetro na faixa do visível, de oito tipos de defeitos em tábuas de abeto. Esses autores realizaram as análises discriminante linear e quadrática nos dados originais das reflectâncias com resolução espectral de 10nm e nos escores dos componentes principais. A função discriminante quadrática apresentou melhores resultados do que a linear, tanto para os dados originais quanto para os escores dos componentes, apresentando erros de classificação global dos defeitos entorno de 3%. Os autores recomendaram a investigação do uso de câmeras coloridas para classificação de tábuas em serrarias, devido aos bons resultados encontrados para as características da reflectância.

KOIVO e KIM (1989) desenvolveram um algoritmo, para classificar nove classes de defeitos em tábuas de Carvalho Vermelho. Imagens foram adquiridas por câmera monocromática com 255 níveis de cinza. Inicialmente, os pesquisadores

verificaram que as funções discriminantes lineares, utilizando-se média e desvio-padrão dos níveis de cinza, separaram somente três classes. Assim, os autores propuseram outras cinco características, obtidas a partir da modelagem da textura de blocos da imagem. Utilizando uma árvore de decisão com os três componentes principais deste vetor de características, foi possível diferenciar as nove classes de interesse (casca, madeira limpa, podridão, furo, nós, inclusão mineral, rachadura, mancha e esmoado). O erro de classificação foi de 2,8 %.

As características de textura melhoram a exatidão de classificação. Entretanto, características texturais são de aquisição complexa, tornando-se inviáveis na utilização de SVA, que objetivam classificação em tempo real.

KAUPPINEN (1999) desenvolveu um classificador k-NN para seis tipos de defeitos de tábuas de pinos, utilizando características de percentis do histograma das bandas Vm, Vd e Az. O erro de classificação foi de 34%. O alto nível de erro foi atribuído à dificuldade em definir algumas classes de defeitos da amostra tais como, os tipos de nós; o nó preto, por exemplo, é uma subcategoria do nó seco. Concluiu que os percentis são simples de calcular e possuem discriminações satisfatórias, permitindo implementação em tempo real.

O objetivo deste trabalho foi avaliar, por meio de técnicas de análise multivariada, a capacidade de discriminar defeitos em madeira serrada de eucalipto, utilizando-se as características de percentis de imagens coloridas.

MATERIAL E MÉTODOS

Foram amostradas 40 peças de madeira serrada de eucalipto secas ao ar e previamente aplainadas, provenientes do Laboratório de Propriedades Físicas e Mecânicas da Madeira da UFV. As amostras foram escolhidas de forma que estariam presentes, além de madeiras limpas de coloração mais clara ou mais escura, com ou sem alburno, os defeitos visuais mencionados na norma NBR 12551 (ABNT, 2002b): (1) bolsa de goma, (2) esmoado, (3) fendilhado, (4) furos de inseto, (5) medula, (6) nó cariado, (7) nó sadio, (8) nó sadio corrompido, (9) nó solto encaixado, (10) nó solto oco, (11) podridão e (12) racha.

Para aquisição das imagens, foi utilizada a câmera colorida Duncantech, modelo MS3100, conectada a um computador por meio de uma placa de aquisição de vídeo National Instruments, modelo PCI 1424. A altura da câmera foi de 0,9 m da superfície da face larga da peça da madeira e a resolução espacial foi de 0,24 x 0,24 mm por pixel. O sistema foi iluminado por duas lâmpadas halógenas de 300 w, instaladas em refletores com difusor de luz. O iluminamento na superfície da face larga da peça de madeira foi de 2500 lux.

As imagens das madeiras serradas, com 1392 pixels na horizontal e 1039 pixels na vertical, foram adquiridas em suas duas faces largas. Feita a varredura das faces largas da madeira, obtiveram-se 594 imagens. Destas, foram cortados blocos da imagem que representaram as classes de defeito e madeira limpa, obtendo-se a seguinte amostra: 41 de madeira limpa de alburno, 48 de madeira limpa clara, 24 de madeira limpa clara áspera, 74 de madeira limpa escura, 25 de bolsa de goma, 9 de esmoado, 16 de fendilhado, 34 de furos de inseto, 54 de medula, 46 de nó cariado, 34 de nó sadio, 29 de nó sadio corrompido, 28 de nó solto encaixado, 9 de nó solto oco, 5 de podridão e 16 de racha.

Foram testados dois tamanhos de blocos de imagem, de 64x64 e 32x32 pixels. Na Figura 1, são mostrados exemplos de blocos de imagem, que representam as classes de defeitos e madeira limpa estudadas.

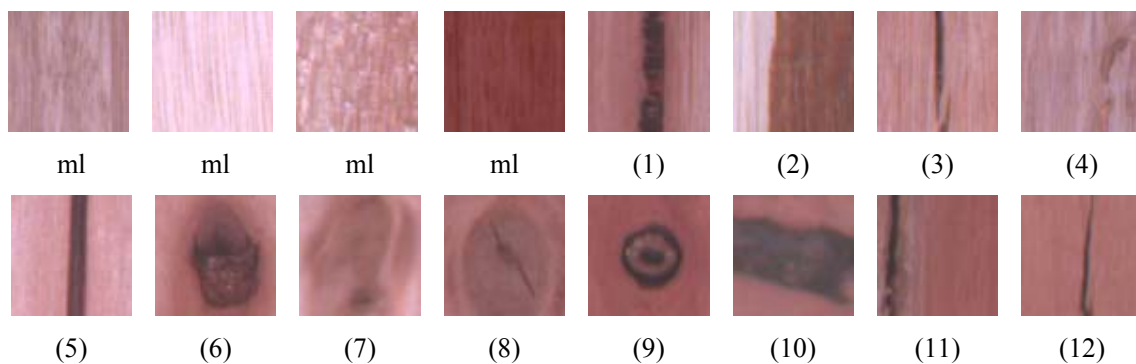


Figura 1- Exemplo de blocos de imagem de 64x64 pixels de (ml) madeira limpa, (1) bolsa de goma, (2) esmoado, (3) fendilhado, (4) furos de inseto, (5) medula, (6) nó cariado, (7) nó sadio, (8) nó sadio corrompido, (9) nó solto encaixado, (10) nó solto oco, (11) podridão e (12) racha.

Características das imagens

As características retiradas de cada bloco da imagem foram 12 percentis das bandas Vm, Vd e Az, construindo, assim, um vetor de 36 características. Utilizaram-se os percentis de 2, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90 e 95 %. A diferença entre os percentis 2 e 95% foi utilizada, para normalizar o vetor

$$C_{Bd}^Y = \frac{P_{Bd}^Y}{P_{Bd}^{95\%} - P_{Bd}^{2\%}} :$$

em que, C_{Bd}^Y = característica normalizada do percentil y na banda Bd; P_{Bd}^y = nível de cinza da banda Bd do percentil y; y = 2, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 95 %; e Bd = bandas Vm, Vd e Az.

A partir do vetor de características originais, foi realizado o teste de médias T^2 de Hotelling, a fim de verificar se a separação entre as classes de interesse foi significativa, ao nível de significância de 0,01. A separação significativa não implica, necessariamente, em boa classificação. A eficácia de um método de classificação pode ser avaliada independentemente de qualquer teste de separação. Por outro lado, se a separação não for significativa, a procura por uma regra de classificação seria, provavelmente, infrutífera (REGAZZI, 2000).

Após o teste de médias, foi realizado o descarte de características pelo método passo-a-passo (stepwise). Neste método, o poder das características em discriminar as classes foi avaliado pelo critério de Wilks Lambda, utilizando-se um nível de significância de 0,15. O procedimento STEPDISC do programa computacional SAS foi utilizado para esta tarefa (SAS, 1999).

As características, selecionadas no descarte, formaram um vetor de variáveis que foi assumido com distribuição normal multivariada, para o desenvolvimento dos classificadores estatísticos.

Classificadores estatísticos

Os classificadores estatísticos são derivados do teorema de Bayes. Nesse teorema, determina-se a probabilidade *a posteriori*, $P(w_j|X)$, de que um indivíduo X seja classificado na classe w_j . Esta probabilidade *a posteriori* é obtida para cada uma das classes w_j , por meio do produto da probabilidade *a priori* de ocorrência da classe,

$P(w_j)$, pela probabilidade condicional dada as características X dentro da classe w_j , $P(X|w_j)$. O indivíduo é classificado na classe que apresentar maior valor de $P(w_j|X)$.

O desenvolvimento dos classificadores, também denominado fase de treinamento, pode ser realizado de duas maneiras, ou seja, considerando-se uma covariância para cada classe ou que todas as classes possuem covariância comum (JOHNSON & WICHERN, 1998). A probabilidade $P(w_j|X)$, quando se considera uma covariância para cada classe, é obtida pela função discriminante quadrática:

$$D_j(X) = -\frac{1}{2}(X - \bar{X}_j)^T (\hat{\Sigma}_j)^{-1} (X - \bar{X}_j) - \frac{1}{2} \ln |\hat{\Sigma}_j| + \ln [P(w_j)]$$

em que, $D_j(X)$ = valor da função discriminante quadrática da classe w_j dado X ; X = vetor de característica de um indivíduo; \bar{X}_j = vetor de média da classe j ; $\hat{\Sigma}_j$ = estimativa da matriz da covariância da classe j ; $|\hat{\Sigma}_j|$ = determinante da estimativa da matriz da covariância da classe j ; e $P(w_j)$ = probabilidade *a priori* de ocorrência da classe j .

Quando se considera a covariância comum para as classes, que é obtida pela média das covariâncias das classes, a função discriminante reduz-se a seguinte combinação linear:

$$D_j(X) = -\frac{1}{2} \bar{X}_j^T (\hat{\Sigma}_c)^{-1} \bar{X}_j + X^T (\hat{\Sigma}_c)^{-1} \bar{X}_j + \ln P(w_j)$$

em que, $D_j(X)$ = valor da função discriminante linear da classe w_j dado X ; $\hat{\Sigma}_c$ = matriz de covariância comum das classes w_j 's.

Sendo assim, dos valores encontrados de $D_1(X)$, $D_2(X)$, ..., $D_k(X)$ das k classes, o maior valor dos $D_k(X)$'s será onde o indivíduo X terá maior probabilidade de pertencer. Os classificadores foram testados, baseando-se nas funções discriminantes quadrática e linear.

As estimativas dos parâmetros das funções discriminantes ($\bar{X}_j, \hat{\Sigma}_j$) foram adquiridas dos vetores de características das amostras dos blocos de imagem. A probabilidade *a priori* $P(w_j)$ foi estimada pela frequência de ocorrência das classes na amostra. O procedimento DISCRIM do programa SAS foi utilizado para obtenção das funções discriminantes (SAS, 1999).

Neste trabalho, foram testadas funções discriminantes com as características originais e também com os escores dos componentes principais e com os escores das variáveis canônicas. A análise de componentes principais e as variáveis canônicas são técnicas que encontram combinações lineares dessas características que podem carregar grande parte da variabilidade das características originais com um número reduzido de variáveis descorrelacionadas (KHATTREE & NAIK, 2000).

Na análise de componentes principais, obtém-se um outro grupo de variáveis que são ordenadas de tal forma que as primeiras variáveis, ou primeiros componentes principais, carregam as maiores variâncias dos dados. Essas combinações são encontradas, a partir da diagonalização da matriz de covariância ou correlação. Com isso, é possível reduzir a dimensionalidade das variáveis utilizando-se somente os primeiros componentes principais. Neste trabalho, os classificadores foram desenvolvidos e testados com os escores dos componentes principais da matriz de correlação das médias das classes (JOHNSON & WICHERN, 1998). O número de componentes principais, utilizados no desenvolvimento das funções discriminantes, foi definido pela variância acumulada de, no mínimo, 90% dos escores das médias entre as classes.

Na análise de variáveis canônicas, obtém-se um outro grupo de variáveis que são ordenadas de tal forma que as primeiras variáveis canônicas carregam as maiores variâncias entre classes, considerando-se a variância dentro das classes. Essas combinações são encontradas, a partir da diagonalização de uma matriz que tem informação da variância entre e dentro das classes, formada pelo produto da matriz de soma de quadrado de tratamento e da inversa da matriz de covariância comum. Assim, é possível reduzir a dimensionalidade das variáveis, utilizando somente as primeiras canônicas. Neste trabalho, os classificadores foram desenvolvidos e testados com os escores das variáveis canônicas (JOHNSON & WICHERN, 1998). O número de variáveis canônicas, utilizadas no desenvolvimento das funções discriminantes, foi definido pela variância acumulada de, no mínimo, 90% de seus escores.

Os erros de classificação das funções discriminantes foram realizados por validação cruzada, deixando uma amostra de fora (leave-one-out). A partir das n observações da amostra, foram realizados n treinamentos e n testes de validação. Para

cada treinamento de tamanho (n-1) observações, uma observação fica de fora para teste de validação (SAS/STAT, 2003), de forma que utilizará toda a amostra para treinamento e para teste das funções discriminantes. Assim, o erro de classificação foi calculado pelo número de observações classificadas, incorretamente, nas observações de teste, em relação ao número de observações da amostra. A opção “crosvalidate” do procedimento Discrim executa esta análise (SAS, 1999).

RESULTADOS E DISCUSSÃO

No Quadro 1 são apresentados os valores de F_0 calculado para o T^2 de Hotelling dos vetores de médias das classes de defeitos e madeira limpa estudadas, para dois tamanhos de blocos.

As diferenças entre as médias das classes 1, 6, 9 e das classes 7 e 8 não foram significativas ($p>0,01$). Segundo a norma de classificação de madeira serrada de eucalipto (ABNT, 2002a), os defeitos 1, 6, 9 e 10 são igualmente quantificados e pontuados para definição das classes de qualidade das peças de madeira serrada, assim como os defeitos 7 e 8. Desta maneira, pode-se agrupar estes defeitos sem causar prejuízo à classificação final da peça.

Quadro 1 – Valores de F_0 calculado para o T^2 de Hotelling para 12 classes de defeitos e madeira limpa para blocos de imagens de 64x64 pixels acima da diagonal principal e abaixo para blocos de 32x32 pixels

classes	1	2	3	4	5	6	7	8	9	10	11	12	ml
1	-	5,32	5,05	5,69	3,39	1,18^{ns}	3,96	2,74	1,18^{ns}	2,61	3,72	6,76	14,69
2	3,51	-	7,05	5,42	6,31	5,14	4,41	4,55	5,77	3,66	6,00	7,52	6,87
3	8,23	7,62	-	4,58	4,25	6,44	5,04	3,95	4,96	5,46	3,67	2,33	12,06
4	6,81	6,61	5,96	-	5,48	5,72	2,91	2,99	5,17	3,98	4,05	4,67	5,14
5	3,63	5,36	6,47	4,58	-	5,94	3,25	2,90	4,15	3,97	5,04	6,61	19,69
6	2,5	3,28	10,09	7,89	5,86	-	4,12	2,83	1,49^{ns}	1,99	4,15	7,74	21,01
7	5,5	4,62	7,57	4,00	2,76	4,96	-	0,73^{ns}	3,92	3,23	4,28	5,33	9,41
8	4,11	4,82	4,29	2,12	1,86	4,26	1,65^{ns}	-	2,91	3,01	3,94	4,46	9,89
9	1,41^{ns}	4,18	7,09	5,07	2,53	1,83	3,65	2,28	-	2,39	3,93	6,00	16,16
10	3,68	4,26	7,47	5,7	4,73	2,36	4,37	4,17	3,13	-	3,98	5,45	6,99
11	2,88	3,6	3,75	2,43	2,41	3,31	2,54	2,43	2,63	3,60	-	3,89	5,70
12	7,91	6,34	3,83	6,02	6,04	9,34	6,59	3,98	6,9	7,05	3,20	-	11,12
ml	12,34	6,97	12,4	7,55	11,77	17,47	6,67	7,73	10,33	6,92	2,28	10,23	-

1 = bolsa de goma; 2 = esmoado; 3 = fendilhado; 4 = furos de inseto; 5 = medula; 6 = nó cariado; 7 = nó sadio; 8 = nó sadio corrompido; 9 = nó solto encaixado; 10 = nó solto oco; 11 = podridão; 12 = racha e ml = madeira limpa.

Classes i de uma linha e j de uma coluna não diferem ($p>0,01$) se o valor de $F_0(i,j)$ é seguido de ns.

No Quadro 2, são apresentados os resultados do teste T^2 de Hotelling dos vetores de médias após o agrupamento das classes 1, 6, 9 e 10, bem como das classes 7 e 8 para dois tamanhos de blocos. Os resultados das diferenças entre as médias dessas novas classes foram significativos ($p < 0,01$). Portanto, o desenvolvimento de um classificador para discriminar essas classes pode ser viável.

Quadro 2 - Valores de F_0 calculado para o teste T^2 de Hotelling para 8 classes de defeitos e madeira limpa para blocos de imagens de 64x64 pixels acima da diagonal principal e abaixo para blocos de 32x32 pixels

classes	1/6/9/10	2	3	4	5	7/8	11	12	ml
1/6/9/10	-	5,91	7,05	7,83	7,25	6,97	4,31	8,71	40,30
2	3,92	-	6,89	5,41	6,21	4,92	5,98	7,43	6,90
3	10,56	7,32	-	4,47	4,22	5,20	3,69	2,32	11,84
4	9,45	6,47	5,89	-	5,39	3,74	4,06	4,65	5,14
5	5,98	5,21	6,29	4,54	-	4,11	5,08	6,60	19,47
7/8	7,75	4,98	6,59	3,55	2,52	-	4,39	5,74	16,35
11	3,15	3,54	3,7	2,43	2,43	2,55	-	3,91	5,69
12	10,09	6,14	3,85	6,02	5,96	5,94	3,18	-	11,02
ml	30,11	6,85	11,89	7,35	11,57	11,38	2,27	9,91	-

1 = bolsa de goma; 2 = esmoado; 3 = fendilhado; 4 = furos de inseto; 5 = medula; 6 = nó cariado; 7 = nó sadio; 8 = nó sadio corrompido; 9 = nó solto encaixado; 10 = nó solto oco; 11 = podridão; 12 = racha e ml = madeira limpa.

Classes i de uma linha e j de uma coluna não diferem ($p > 0,01$) se o valor de $F_0(i,j)$ é seguido de ns.

Os resultados do descarte de características pelo método passo-a-passo são apresentados no Quadro 3. Para características retiradas de blocos de 64x64 pixels, ocorreu redução de 36 para 21 variáveis. Para blocos de 32x32 pixels, resultou uma redução de 36 para 18 variáveis. Estes resultados demonstram a dificuldade de se utilizar uma câmera com apenas uma banda espectral na classificação de madeiras, pois, mesmo após o descarte, características de todas as bandas são necessárias para discriminar as classes.

Quadro 3 - Características selecionadas pelo método passo a passo de blocos da imagem de 64x64 e 32x32 pixels

Blocos	Características
64x64	$C_{Vm}^2, C_{Vm}^{10}, C_{Vm}^{30}, C_{Vm}^{60}, C_{Vm}^{70}, C_{Vm}^{90}, C_{Vd}^5, C_{Vd}^{10}, C_{Vd}^{40}, C_{Vd}^{60}, C_{Vd}^{70}, C_{Vd}^{95}, C_{Az}^5, C_{Az}^{10}, C_{Az}^{30}, C_{Az}^{40}, C_{Az}^{50}, C_{Az}^{60}, C_{Az}^{70}, C_{Az}^{90}, C_{Az}^{95}$.
32x32	$C_{Vm}^5, C_{Vm}^{10}, C_{Vm}^{20}, C_{Vm}^{30}, C_{Vm}^{95}, C_{Vd}^5, C_{Vd}^{20}, C_{Vd}^{30}, C_{Vd}^{60}, C_{Vd}^{70}, C_{Vd}^{80}, C_{Az}^2, C_{Az}^5, C_{Az}^{10}, C_{Az}^{20}, C_{Az}^{40}, C_{Az}^{60}, C_{Az}^{70}$.

Classificadores estatísticos

Os resultados dos erros de classificação da validação cruzada, para os blocos de imagens de 64x64 pixels, são apresentados nos Quadros 4 e 5, respectivamente, para as funções discriminantes lineares e quadráticas. Os escores das variáveis canônicas tenderam a apresentar menores erros de classificação, comparativamente às variáveis originais e com os escores dos componentes principais. Observou-se uma tendência de menores erros globais para as funções discriminantes quadráticas, quando foram utilizados componentes principais. Entretanto, as funções lineares com variáveis canônicas não apresentaram erro de 100% em nenhuma das classes, indicando um maior poder de generalização do classificador linear sobre o quadrático. Este problema também pode ter ocorrido devido ao número de observações relativamente pequeno de algumas classes (2, 3, 11 e 12), que tenderam a apresentar maiores erros nas funções discriminantes quadráticas.

O menor erro global encontrado foi de 19%, tanto para a função discriminante linear quanto para a quadrática com variáveis canônicas. Porém, com os problemas apresentados em algumas classes com as funções quadráticas, conclui-se que, para tamanhos de blocos de imagens 64x64, as funções lineares foram as mais adequadas.

Quadro 4- Erro de classificação, em percentagem, de cada classe e global para função discriminante linear com blocos de imagens de 64x64 pixels

Número de variáveis ^a	Classes									
	1/6/9/10	2	3	4	5	7/8	11	12	ml	Erro Global
Originalis										
21 (100,0)	21	44	56	50	30	27	60	69	6	23
Componentes principais										
5 (100,0)	23	67	50	88	50	25	100	94	10	31
4 (99,8)	12	100	44	91	52	57	100	94	11	33
3 (99,5)	3	89	100	85	100	35	100	100	11	35
2(98,7)	2	100	100	94	100	38	100	100	11	36
1(96,5)	1	100	100	100	100	49	100	100	7	37
Variáveis canônicas										
8 can (100)	19	22	50	41	27	25	20	44	6	19
7 can (98,9)	18	22	50	50	26	25	20	50	6	20
6 can (96,8)	17	22	56	68	26	25	20	44	5	20
5 can (93,7)	16	22	56	71	26	30	40	38	5	21

1-bolsa de goma ; 2-esmoado; 3-fendilhado; 4-furos de inseto; 5-medula; 6-nó cariado; 7-nó sadio; 8-nó sadio corrompido;9-nó solto encaixado; 10-nó solto oco; 11-podridão; 12-racha e ml- madeira limpa.

a - entre parêntesis é apresentada a variância acumulada.

A contribuição de uma variável canônica ou de um componente principal altera, relativamente, pouco a variância acumulada (Quadros 6 e 7). No entanto, o efeito no erro de algumas classes chega a cair pela metade. Assim, para discriminar alguns defeitos, foi necessária a utilização de variáveis que carregassem mais de 90% da variância acumulada.

Quadro 5- Erro de classificação, em percentagem, de cada classe e global para função discriminante quadrática com blocos de imagens de 64x64 pixels

Número de variáveis ^a	Classes									Erro global
	1/6/9/10	2	3	4	5	7/8	11	12	ml	
Originais										
21 (100,0)	10	100	100	82	26	24	100	100	4	25
Componentes principais										
5 (100,0)	4	67	44	62	37	21	100	88	5	20
4 (99,8)	4	67	38	59	67	30	100	94	10	26
3 (99,5)	2	33	100	53	80	27	100	100	19	32
2(98,7)	2	89	100	82	81	21	100	100	12	31
1(96,5)	3	100	100	100	100	22	100	100	6	33
Variáveis canônicas										
8 can (100)	13	100	56	44	30	32	100	63	3	21
7 can (98,9)	14	78	50	47	30	32	100	63	3	21
6 can (96,8)	12	67	38	59	30	29	100	63	3	20
5 can (93,7)	10	44	25	56	26	30	100	56	3	19

1-bolsa de goma ; 2-esmoado; 3-fendilhado; 4-furos de inseto; 5-medula; 6-nó cariado; 7-nó sadio; 8-nó sadio corrompido;9-nó solto encaixado; 10-nó solto oco; 11-podridão; 12-racha e ml- madeira limpa.

a - entre parêntesis é apresentada a variância acumulada.

O Quadro 6 apresenta a matriz de confusão da classificação, realizada por função discriminante linear a partir de 8 canônicas. Os defeitos 3 (fendilhado) e 12 (racha) foram confundidos. Estes defeitos são semelhantes, sendo o fendilhado uma racha superficial (ABNT, 2002b). Este problema poderia ser minimizado ao acrescentar na regra de decisão a localização do defeito, pois, a racha ocorre nas extremidades das peças. A classificação do defeito 4 (furo de inseto) também apresentou, relativamente, erro elevado. Provavelmente, isto ocorreu devido a resolução dos pixels da imagem não ter sido suficiente para identificação dos furos. Furos de inseto não são frequentes em toras de madeiras selecionadas para serraria.

Quadro 6 - Matriz de confusão para função discriminante linear com 8 variáveis canônicas para blocos de imagens de 64x64 pixels

Classes	Referência									
	1/6/9/ 10	2	3	4	5	7/8	11	12	ml	
1/6/9/10	88	0	1	0	4	7	0	2	1	
2	5	7	0	0	0	0	0	0	0	
3	3	0	8	0	3	1	0	0	0	
4	0	0	1	20	0	0	0	1	3	
5	5	0	1	1	38	4	0	1	1	
7/8	4	1	0	1	6	47	0	1	6	
11	0	0	0	0	0	0	4	0	0	
12	3	0	5	1	0	1	0	9	0	
ml	0	1	0	11	3	3	1	2	176	
Total	108	9	16	34	54	63	5	16	187	492
Erro (%)	19	22	50	41	30	25	20	44	6	19

1-bolsa de goma; 2-esmoado; 3-fendilhado; 4-furos de inseto; 5-medula; 6-nó cariado; 7-nó sadio; 8-nó sadio corrompido; 9-nó solto encaixado; 10-nó solto oco; 11-podridão; 12-racha e ml -madeira limpa.

Os resultados dos erros de classificação da validação cruzada, para os blocos de imagens de 32x32 pixels, são apresentados nos Quadros 7 e 8, respectivamente para as funções discriminantes lineares e quadráticas. Os classificadores para blocos 32x32 pixels apresentaram tendências semelhantes aos dos blocos 64x64 pixels, porém com tendência a erros de classificação mais altos. Menores dimensões de blocos, entretanto, possibilitam uma melhor caracterização da dimensão e formato dos defeitos.

Quadro 7- Erro de classificação, em percentagem, de cada classe e global para função discriminante linear com blocos de imagens de 32x32 pixels

Número de Variáveis ^a	Classes									
	1/6/9/10	2	3	4	5	7/8	11	12	ml	Global
Originais										
21 (100)	20	78	44	56	41	41	80	56	9	27
Componentes principais										
6 cp (100)	21	100	63	91	54	38	100	44	10	32
5 cp (99,9)	21	100	63	91	54	38	100	44	10	32
4 cp (99,9)	21	100	38	97	63	33	100	81	10	33
3 cp (99,5)	4	100	100	100	94	41	100	94	10	36
2 cp (99,0)	4	100	100	100	94	41	100	100	12	37
1 cp (97,9)	1	100	100	100	100	56	100	100	12	39
Variáveis canônicas										
8 can (100)	18	67	44	53	39	38	80	50	8	25
7 can (98,3)	18	56	44	53	37	33	80	50	9	24
6 can (96,3)	20	56	38	53	35	38	100	50	8	25
5 can (93,2)	14	56	38	53	41	60	100	50	7	26

1-bolsa de goma ; 2-esmoado; 3-fendilhado; 4-furos de inseto; 5-medula; 6-nó cariado; 7-nó sadio; 8-nó sadio corrompido;9-nó solto encaixado; 10-nó solto oco; 11-podridão; 12-racha e ml- madeira limpa.

A - entre parêntesis é apresentada a variância acumulada.

Quadro 8- Erro de classificação, em percentagem, de cada classe e global para função discriminante quadrática com blocos de imagens de 32x32 pixels

Número de Variáveis ^a	Classes									
	1/6/9/10	2	3	4	5	7/8	11	12	ml	Global
Originais										
21 (100)	18	100	100	56	52	37	100	100	5	29
Componentes principais										
5 cp (100)	9	89	44	56	65	46	100	69	7	28
4 cp (99,8)	9	78	31	59	63	44	100	81	11	29
3 cp (99,5)	7	56	25	88	63	62	100	88	13	33
2 cp (98,7)	7	67	38	97	65	63	100	100	17	37
1 cp (96,5)	6	89	81	97	67	56	100	100	10	35
Variáveis canônicas										
8 can (100)	14	100	50	59	44	41	100	56	5	25
7 can (98,3)	13	78	38	47	43	44	100	56	5	24
6 can (96,3)	15	67	31	50	50	40	100	50	5	24
5 can (93,2)	13	44	25	53	52	49	100	44	4	24

1-bolsa de goma ; 2-esmoado; 3-fendilhado; 4-furos de inseto; 5-medula; 6-nó cariado; 7-nó sadio; 8-nó sadio corrompido;9-nó solto encaixado; 10-nó solto oco; 11-podridão; 12-racha e ml- madeira limpa.

a - entre parêntesis é apresentada a variância acumulada.

Os erros apresentados pelas funções discriminantes lineares com os escores das variáveis canônicas foram considerados adequados, para discriminar defeitos e madeira limpa em imagens digitais. Esses erros indicam um desempenho melhor do que os dos classificadores humanos estudados por HUBER et al. (1985).

Os classificadores com base nos percentis apresentaram erros superiores aos de características texturais utilizadas por outros autores (KOIVO & KIM, 1989; PACKIANATHER & DRAKE, 2000). No entanto, os percentis são mais adequados para implementação em um sistema em tempo real devido à facilidade de extração.

CONCLUSÕES

Neste trabalho, foram realizadas análises discriminante linear e quadrática de defeitos e madeira limpa em imagens de madeira serrada de eucalipto. As características de percentis foram extraídas do histograma das bandas do vermelho, verde e azul de blocos de imagens com 64x64 pixels (resolução espacial de 15,4x15,4 mm) e 32x32 pixels (resolução espacial de 7,7x7,7 mm).

As funções discriminantes com os escores das variáveis canônicas tenderam a apresentar menores erros de classificação do que com as variáveis originais e com os escores dos componentes principais.

As funções discriminantes quadráticas tenderam a apresentar erros de classificação global menores do que as funções lineares, porém a função linear tendeu a apresentar uma melhor distribuição de erros entre as classes de defeitos.

Características retiradas de blocos de 64x64 pixels tenderam a apresentar menores percentuais de erros de classificação do que em blocos de 32x32 pixels.

Os menores erros globais de classificação foram 19% e 24% para funções discriminantes lineares com variáveis canônicas para tamanho de bloco de 64x64 e 32x32 pixels, respectivamente. Tendo em vista a magnitude desses erros, as características de percentis foram consideradas adequadas para discriminar defeitos e madeira limpa em imagens digitais.

REFERÊNCIAS BIBLIOGRÁFICAS

- ARACRUZ PRODUTOS DE MADEIRA S.A. Descrição do produto. Nova Viçosa, Ba, 2000. 9p.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS – **ABNT**. Madeira serrada de eucalipto – Requisitos, NBR 14806. Rio de Janeiro, 2002a. 11 p.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS – **ABNT**. Madeira serrada – Terminologia, NBR 12551. Rio de Janeiro, 2002b. 10 p.
- HANEY, L.; PRECETTI, C.; GIBSON, H.; KRUTZ, G.; CASSENS, D. Color matching of wood with a real-time machine vision system. **ASAE Paper No. 943579**, ASAE Meeting Presentation, St. Joseph, Michigan, 1994, 15 p.
- HUBER, H. A.; McMILLIN, C. W.; McKINNEY, J. P. Lumber defect detection abilities of furniture rough mill employees. **Forest Products Journal**, 35(11/12), p. 79-82, 1985.
- JOHNSON, R. A.; WICHERN, D. W. R. **Applied Multivariate statistical analysis**, 4. ed. ,Rio de Janeiro: Prentice-Hall, 1998, 816 p.
- KAUPPINEN, H.; **Development of a color machine vision method for wood surface inspection**. Oulu, Finlandia,: Department of Electrical Engineering and Infotech Oulu, University of Oulu, Dissertation, 138p, 1999. <http://herkules.oulu.fi/isbn9514254244/> . 11 nov. 2003.
- KHATTREE, R.; NAIK, D. N. **Multivariate data reduction and discrimination with SASR software**. Cary, NC: SAS Institute Inc., 2000. 558p.
- KLINE, D. E.; WIDOYOKO, A.; WIEDENBECK, J. K.; ARAMAN, P.A. Performance of color camera machine vision in automated furniture rough mil systems, **Forest Products Journal**, 48(3), p. 38-45, 1998.
- KLINE, D. E.; HOU, Y. J.; CONNERS, R. W.; SCHOLMOLDT, D. L.; ARAMAN, P.A. **Lumber scanning system for surface defect detection**, ASAE Paper No. 923582.St. Loseph, Mich.:ASAE, 1992.
- KLINE, D. E.; ARAMAN, P. A.; SURAK, C. Evaluation of an automated hardwood lumber grading system. Proceedings, ScanTech, **The Ninth International Conference on Scanning Technology and Process Optimization for the Wood Industry**. 141-151, 2001.

- KOIVO, A. J.; KIM, C. W. Automatic classification of surface defects on red oak boards, **Forest Products Journal**, 39(9), p. 22-30, 1989.
- LEBOW, P. K.; BRUNNER, C. C.; MARISTANY, A. G.; BUTLER, D.A. Classification of wood surface features by spectral reflectance, **Wood and Fiber Science**, 28(1), p. 74-90, 1996.
- LU, Q. **A real-time system for color sorting edge-glued panel parts**. Blacksburg, Virginia: Faculty of the Virginia Polytechnic Institute and State University, Dissertation, 119 p, 1997.
- PACKIANATHER, M. S., DRAKE, P. R., Neural networks for classifying images of wood veneer part 2. **International Journal of advanced manufacturing Technology**, 16(6), p.424-433, 2000.
- PHAM, D.T.; ALCOCK, R.J. Automated grading and defect detection: a review. **Forest Products Journal**, 48(4), p34-42, 1998.
- REGAZZI, A. J. **Análise multivariada**, notas de aula INF 766, Departamento de Informática da Universidade Federal de Viçosa, vol. 2, 2000.
- SZYMANI, R.; MCDONALD, K. A. Defect detection in lumber: state of the art, **Forest Products Journal**, 31(11), p. 34-44, 1981.
- SAS Institute, 1999. **SAS OnlineDoc, Version 8**, SAS Institute Inc., Cary, NC, USA.
- SAS/STAT. **The Discrim Procedure Overview**. <http://www.id.unizh.ch/software/unix/statmath/sas/sasdoc/stat/chap25/index.htm>. 30 nov. 2003.

DESENVOLVIMENTO DE REDES NEURAIIS PARA RECONHECIMENTO DE DEFEITOS EM IMAGENS DE MADEIRA SERRADA DE EUCALIPTO

RESUMO

As redes neurais artificiais (RNA) vêm sendo utilizadas em sistemas de visão artificial, para identificação de classes de padrões em imagens. Neste trabalho, o objetivo foi desenvolver e testar RNAs para identificar defeitos em madeira serrada de eucalipto, utilizando-se as características de percentis das bandas do vermelho, verde e azul de câmeras coloridas. Foram testadas diferentes arquiteturas e métodos de treinamento de RNAs. A arquitetura escolhida apresentou 14 neurônios nas duas camadas escondidas. A utilização de uma árvore de RNAs não apresentou vantagens significativas, em relação a uma única RNA. Os classificadores por redes neurais artificiais foram considerados satisfatórios para identificação dos defeitos de madeira serrada de eucalipto, apresentando exatidões globais de 83,1 e 76,6% , dependendo do tamanho de blocos da imagem.

Palavras-chave: processamento de imagem, visão artificial, classificação de madeira.

DEVELOPING THE NEURAL NETWORKS FOR RECOGNITION OF DEFECTS IN THE DIGITAL IMAGES OF EUCALYPTUS SAWN LUMBER

ABSTRACT

The artificial neural networks (ANN) have been used in the machine vision systems for identifying the pattern classes based by images. The objective of this study was to develop and test the RNAs in order to identify the defects in eucalyptus sawn lumber, using the percentile characteristics of the red, green and blue bands in the colored cameras. Different architectures and RNAs' training methods were tested. The chosen architecture showed 14 neurons in both hidden layers. The use of the ANN decision tree did not present any significant advantages, compared with a single RNA. The neural artificial network classifiers showed to be satisfactory for the identification

of defects in eucalyptus sawn lumber, because they presented overall accuracy of 83.1 and 76.6%, depending on the size of image..

Keywords: image processing, machine vision, lumber grading.

INTRODUÇÃO

A indústria de madeira serrada tem dispensado especial atenção às etapas de classificação e seleção de peças de madeira. Estas etapas são, geralmente, realizadas por classificadores humanos. A etapa de classificação é considerada uma atividade estressante, em razão da dificuldade dos classificadores em realizar as tarefas de identificação dos tipos de defeitos, medi-los e graduar a peça durante tempo prolongado.

Um sistema de visão artificial automatiza um processo, baseado nas informações adquiridas de imagens digitais. O sucesso deste sistema está no desenvolvimento de algoritmos de processamento e análise das imagens, capaz de gerar o sinal adequado para os atuadores eletromecânicos. Nesses algoritmos, geralmente, são incorporadas técnicas de inteligência artificial, sendo as redes neurais artificiais (RNAs) uma das técnicas mais utilizadas (HANEY et al.,1994; PACKIANATHER & DRAKE, 2000; SARIGUL et al.,2003).

Em comparação com classificadores estatísticos, que determinam planos lineares ou quadráticos, o maior benefício de uma RNA é sua capacidade de determinar planos não-lineares de separação das classes (GONZALEZ & WOODS, 1992). No desenvolvimento de uma RNA, não há necessidade de assumir um tipo de função de distribuição para os dados, bem como existe a possibilidade de implementá-la em um sistema de processamento paralelo (PINTO et al., 1999).

HANEY et al. (1994) construíram um protótipo equipado com sistema de visão artificial, para separação em 4 classes baseada na semelhança de cor de madeiras de lenho inicial e lenho tardio. Na aquisição de imagens, utilizou-se a câmera RGB e iluminação controlada, obtida com lâmpadas incandescentes do tipo halógena. Foi desenvolvida uma RNA, para classificar cada pixel da imagem em função dos seus valores nas bandas vermelho (Vm), verde (Vd) e azul (Az). O sistema desenvolvido

foi capaz de trabalhar a uma velocidade da esteira transportadora de 110 metros por minuto.

PACKIANATHER & DRAKE (2000) desenvolveram um classificador de defeitos de madeira de vidoeiro, a partir de imagens monocromáticas. Uma árvore de decisão com base em RNAs foi desenvolvida com nove características de estatísticas do histograma e três de texturas da imagem binarizada. Distinguiram-se 13 classes: casca, madeira limpa, linha colorida, grã-ondulada, descoloração, furos, nós de acícula, madeira grosseira, nós sadios, rachas, listas e furo de inseto. O erro de classificação do algoritmo foi de 4%.

KAUPPINEN (1999) avaliou os defeitos de tábuas de pinos por um classificador de k-vizinhos mais próximo, utilizando-se características de percentis do histograma das bandas Vm, Vd e Az. Na análise de exatidão em uma amostra de 2834 observações de seis tipos de defeitos, ele obteve um erro de classificação de 34%. Concluiu-se que as características de percentis são simples de calcular, possuem boa capacidade discriminante de defeitos de madeira e são convenientes para implementação em tempo real. O alto nível de erro foi atribuído à dificuldade em definir algumas classes de defeitos da amostra como, por exemplo, o nó preto é uma subcategoria do nó seco.

Uma RNA é composta por um conjunto de elementos computacionais (neurônio artificial), que relacionam os valores de saída e entrada por meio da seguinte expressão:

$$y^{ij} = f\left(\sum_{i'=1}^n y^{i'(j-1)} w_{i'}^{ij} + b^{ij}\right)$$

em que, y^{ij} é a saída do neurônio i da camada j ; n é o número de neurônios da camada anterior; $y^{i'(j-1)}$ é a saída do neurônio i' da camada anterior; $w_{i'}^{ij}$ é o peso sináptico do neurônio i da camada j , ativado pelo neurônio i' da camada anterior; b^{ij} é a compensação do neurônio i da camada j ; f é a função de ativação do neurônio i .

O desenvolvimento de uma RNA consiste em determinar sua arquitetura, isto é, os números de camadas e neurônios em cada camada, bem como ajustar os seus parâmetros w 's e b 's, fase esta conhecida como treinamento. A arquitetura varia de acordo com a complexidade do problema e não pode ser definida *a priori* ao

treinamento. Um dos algoritmos mais utilizados para treinamento de uma RNA é o de retropropagação de erro, (HAYKIN,1999; MATLAB, 2000), em que o ajuste dos parâmetros é realizado por método iterativo de minimização do erro quadrático médio (EQM) entre os valores calculados e aqueles esperados na saída.

No desenvolvimento de uma RNA, deve-se sempre ter em mente o poder de generalização da mesma, pois, caso contrário, a rede pode apresentar baixos erros com a amostra de treinamento e altos níveis de erro com valores não-utilizados no ajuste dos parâmetros. Esta 'memorização' da amostra de treinamento pode ocorrer devido a um número elevado de iterações ou a uma arquitetura superdimensionada.

Para o desenvolvimento de uma RNA, uma amostra da população é coletada. Esta amostra é dividida em duas partes: uma para treinamento e a outra para teste, em que se avalia o erro de classificação da RNA. A técnica de 'parada mais cedo' utiliza parte da amostra de treinamento para interromper a atualização dos parâmetros livres e, assim, evitar a 'memorização'. Nesta técnica, a parada ocorre quando se detecta um aumento do EQM na amostra utilizada para interromper o treinamento (HAYKIN, 1999). O número de iterações utilizadas para confirmar a tendência a aumento do EQM é dependente do problema (PRECHELT, 1998). Uma desvantagem desta técnica é quando se tem uma amostra para desenvolvimento de uma RNA com número reduzido de observações, pois, com a 'parada mais cedo', reduzirá ainda mais o número de observações para treinamento e teste.

Uma das maneiras utilizadas para melhorar a estimativa do erro de classificação, sem diminuir o tamanho da amostra de treinamento, é a validação cruzada (PRECHELT, 1998; HAYKIN, 1999). Neste método, a amostra é dividida em n partes, mas somente uma das partes é reservada para teste. As outras $(n-1)$ partes restantes serão utilizadas no treinamento. Para uma mesma arquitetura, realiza-se o treinamento n vezes e o erro de classificação é avaliado pela soma dos erros encontrados nas observações, quando estas ficaram de fora do treinamento.

A escolha da arquitetura, do método de parada do treinamento, assim como o teste são meios para que o ajuste de uma RNA possa generalizar-se com maior eficiência.

Neste trabalho, o objetivo foi desenvolver e testar RNAs para discriminar defeitos em madeiras serradas de eucalipto, utilizando-se as características de percentis das bandas do vermelho, verde e azul de câmeras coloridas.

MATERIAL E MÉTODOS

Foram amostradas 40 peças de madeira serrada de eucalipto secas ao ar e previamente aplainadas, provenientes do Laboratório de Propriedades Físicas e Mecânicas da Madeira da UFV. As amostras foram escolhidas de forma que estariam presentes, além de madeiras limpas de coloração mais clara ou mais escura, com ou sem alburno, os defeitos visuais mencionados na norma NBR 14806 (ABNT, 2002): (1) bolsa de goma, (2) esmoado, (3) fendilhado, (4) furos de inseto, (5) medula, (6) nó cariado, (7) nó sadio, (8) nó sadio corrompido, (9) nó solto encaixado, (10) nó solto oco, (11) podridão e (12) racha.

Para aquisição das imagens, foi utilizada a câmera colorida Duncantech, modelo MS3100, conectada a um computador por meio de uma placa de aquisição de vídeo National Instruments, modelo PCI 1424. A altura da câmera foi de 0,9 m da superfície da peça da madeira e a resolução espacial foi de 0,24 x 0,24 mm por pixel. O sistema foi iluminado com duas lâmpadas halógenas de 300 w instaladas em refletores com difusor de luz. Na superfície da face larga da madeira, o iluminamento foi de 2500 lux.

As imagens das madeiras serradas, com 1392 pixels na horizontal e 1039 pixels na vertical, foram adquiridas nas suas duas faces largas. Feita a varredura das faces largas da madeira serrada, obtiveram-se 594 imagens; destas, foram cortados blocos que representaram as classes de defeito e madeira limpa, obtendo-se a seguinte amostra: 41 de madeira limpa de alburno, 48 de madeira limpa clara, 24 de madeira limpa clara áspera, 74 de madeira limpa escura, 25 de bolsa de goma, 9 de esmoado, 16 de fendilhado, 34 de furos de inseto, 54 de medula, 46 de nó cariado, 34 de nó sadio, 29 de nó sadio corrompido, 28 de nó solto encaixado, 9 de nó solto oco, 5 de podridão e 16 de racha.

Foram testados dois tamanhos de blocos de imagens de 64x64 e 32x32 pixels. Na Figura 1, são mostrados exemplos de blocos de imagens que representam as classes de defeitos e madeira limpa estudadas.

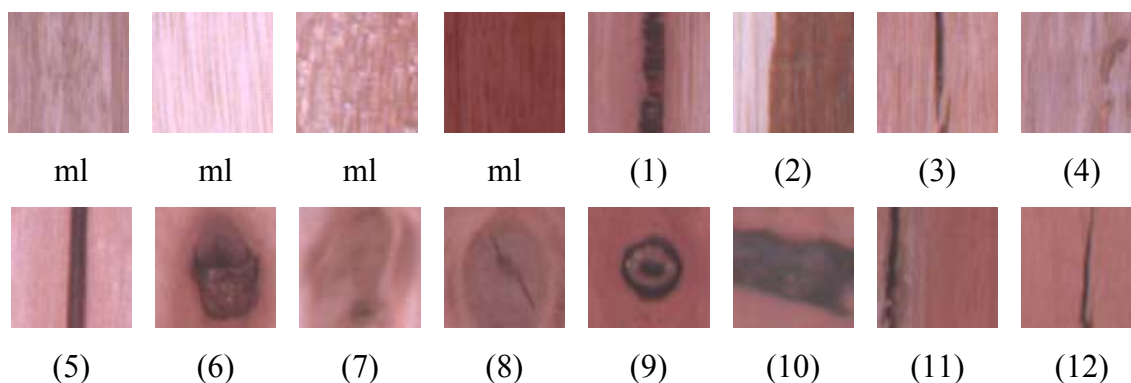


Figura 2- Exemplo de blocos de imagens de 64x64 pixels de (ml) madeira limpa, (1) bolsa de goma, (2) esmoado, (3) fendilhado, (4) furos de inseto, (5) medula, (6) nó cariado, (7) nó sadio, (8) nó sadio corrompido, (9) nó solto encaixado, (10) nó solto oco, (11) podridão e (12) racha.

Características das imagens

As características retiradas de cada bloco de imagens foram 12 percentis das banda Vm, Vd e Az, construindo, assim, um vetor de 36 características. Utilizaram-se os percentis de 2, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90 e 95 %. A diferença entre os percentis 2 e 95% foi utilizada para normalizar o vetor:

$$C_{Bd}^Y = \frac{P_{Bd}^Y}{P_{Bd}^{95\%} - P_{Bd}^{2\%}}$$

em que C_{Bd}^y = característica normalizada do percentil y na banda Bd; P_{Bd}^y = nível de cinza da banda Bd do percentil y; y = 2, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 95 %; e Bd = bandas Vm, Vd e Az.

A partir do vetor de características originais, foi realizado o teste de médias T^2 de Hotelling, para verificar se a separação entre as classes de interesse foi significativa, ao nível de significância de 0,01. A separação significativa não implica, necessariamente, em boa classificação. A eficácia de um método de classificação pode

ser avaliada, independentemente de qualquer teste de separação. Por outro lado, se a separação não for significativa, a procura de uma regra de classificação seria, provavelmente infrutífera (REGAZZI, 2000).

Após o teste de médias, foi realizado o descarte de características pelo método passo-a-passo (stepwise). Neste método, o poder das características em indentificar as classes foi avaliado pelo critério de Wilks Lambda, usando-se um nível de significância de 0,15. O procedimento STEPDISC do programa computacional SAS foi utilizado para esta tarefa (SAS, 1999). As características escolhidas pelo método passo a passo definiram a dimensão do vetor de variáveis para treinamento e teste das RNAs.

Neste trabalho, foram desenvolvidos dois tipos de classificadores com base em RNAs, sendo o primeiro com base em uma RNA única e o segundo com base em uma árvore de decisão composta de RNAs. Para uma RNA única, foram testados dois métodos de treinamento e teste. No primeiro, o treinamento foi por 'parada mais cedo' e o teste foi por validação cruzada, dividindo-se a amostra em quatro partes iguais. Devido a agilidade de treinamento e teste, este método, foi utilizado para determinar as melhores arquiteturas das RNAs. No primeiro método, uma parte da amostra é reservada para parada das iterações e outra parte para teste, reduzindo-se o tamanho da amostra para treinamento e podendo comprometer a representatividade da amostra de algumas classes.

Com a finalidade de reduzir o problema de tamanho da amostra, uma outra RNA única foi desenvolvida, por um segundo método. Neste, limitou-se o número de iterações no treinamento e o teste foi por validação cruzada, pelo método 'deixando um de fora'. Desta forma, tem-se a vantagem de perder apenas uma observação para o treinamento. No entanto, para acessar o erro de classificação, é necessário realizar um número de treinamentos igual ao tamanho da amostra. Por esta razão, foram utilizadas somente as arquiteturas escolhidas pelo método anterior.

O segundo classificador foi baseado em uma árvore de decisão. Na hipótese de que RNAs específicas para discriminar cada defeito, nas ramificações da árvore, possa apresentar resultados de classificação superiores ao de uma RNA única.

Classificador por redes neurais artificiais únicas

As arquiteturas das RNAs propostas foram do tipo $n_i-n_1-n_2-n_o$, isto é, n_i é o número de variáveis da camada de entrada, n_1 e n_2 são números de neurônios nas camadas escondidas e n_o número de neurônios na camada de saída. O número de variáveis da camada de entrada de uma RNA correspondeu à dimensão do vetor de características, enquanto o número de neurônios da camada de saída foi igual ao número de classes de defeitos e madeira limpa a serem discriminados. As funções de ativação, utilizadas nos neurônios das camadas escondidas, foram tangente hiperbólica sigmóide, que apresentam resultados entre menos um e mais um, ao passo que, as funções da camada de saída foram logística sigmóide, que apresentam resultados entre zero e um. Desta maneira, uma vez apresentado à RNA um vetor de características de entrada, este era classificado para a classe de defeito ou madeira limpa, cujo neurônio da camada de saída tivesse valor mais próximo de um (Figura 3).

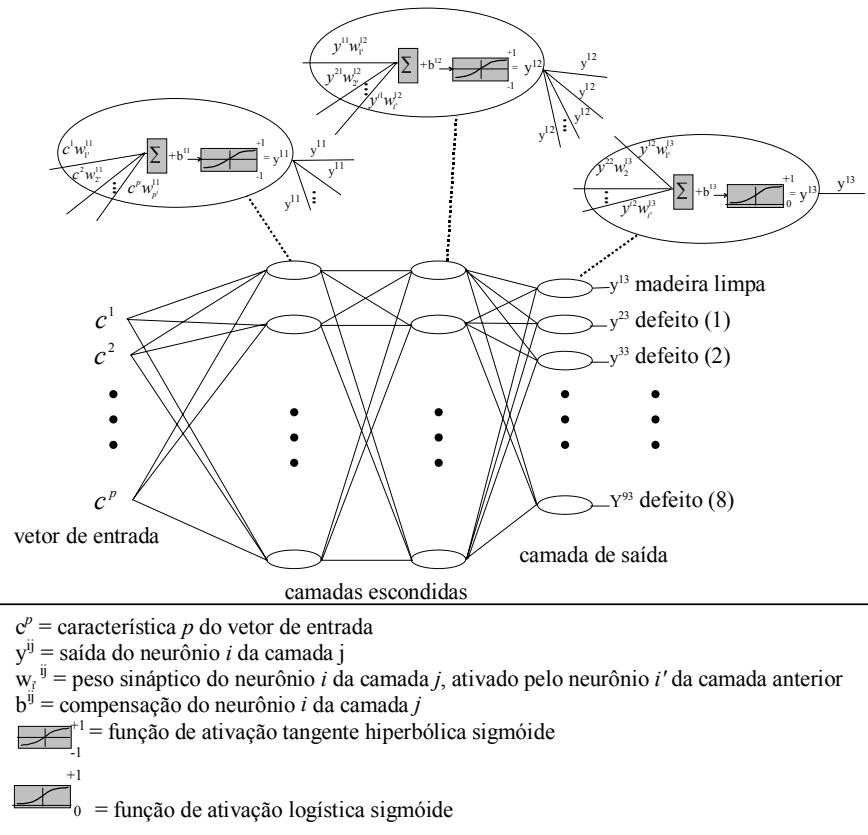


Figura 3- Esquema da arquitetura de uma rede neural artificial.

As arquiteturas das redes testadas foram 28 combinações de n1 e n2: 26-26; 26-24; 26-22; 26-20; 26-18; 26-16; 26-14; 24-24; 24-22; 24-20; 24-18; 24-16; 24-14; 22-22; 22-20; 22-18; 22-16; 22-14; 20-20; 20-18; 20-16; 20-14; 18-18; 18-16; 18-14; 16-16; 16-14; 14-14.

As redes foram treinadas, utilizando-se o pacote de ferramentas do programa computacional Matlab^(R) (MathWorks, Inc.). O algoritmo de retropropagação do erro foi utilizado, para minimização do erro quadrático médio (EQM) entre a resposta obtida de uma RNA e a resposta esperada. A minimização é realizada por método iterativo, em que os parâmetros livres da RNA (w's e b's) são ajustados por meio de um gradiente, que dá a direção de decrescimento do EQM para cada iteração.

Existem diversos métodos para encontrar este gradiente. A escolha do método influi principalmente no tempo de treinamento, que é função do número de iterações para convergência e do esforço no processamento computacional desse gradiente. Os métodos padrões de minimização, que utilizam diferenciação dos parâmetros para cálculo do gradiente, em ordem crescente do esforço computacional, são: os do gradiente, os que utilizam derivada de primeira ordem; os de gradiente conjugado, que utilizam derivadas de primeira ordem conjugadas ou ortogonais; os quase-newton ou secante, que utilizam uma aproximação da inversa da matriz hessiana e o de Newton, que utiliza a inversa da matriz hessiana, composta de segundas derivadas (MATEUS & LUNA, 1986).

Neste trabalho, foi utilizado o método de minimização do erro 'uma etapa secante', que utiliza a função "TRAINOSS" do programa Matlab^(R). Este é uma variante do método quase-newton, em que a matriz hessiana é utilizada sem a necessidade de calcular sua inversa, sendo este o maior motivo de seu melhor desempenho, colocando-o entre os métodos do gradiente conjugado e quase-newton quanto ao esforço computacional (MATLAB, 2000).

A parada do treinamento foi através da técnica de 'parada mais cedo', em que se utiliza uma parte da amostra não vista no treinamento, a fim de parar as atualizações dos parâmetros livres da RNA. Desta maneira, o EQM era calculado nesta parte separada da amostra para cada iteração. Após detectado um aumento do EQM, os parâmetros da rede eram, ainda, atualizados por mais um número pré-definido de

iterações, e se dentro deste limite pré-definido o EQM voltasse a decrescer o treinamento seguiria normalmente, caso contrario ele pararia. Neste trabalho foram testadas cinco, 100, 200, 400 , 600 e 800 atualizações após a detecção do aumento do EQM antes que fosse realizada a parada do treinamento. Tendo em vista que no início do treinamento os parâmetros livres são gerados, aleatoriamente, e que esses valores iniciais podem influenciar o resultado final do treinamento, cada arquitetura foi treinada cinco vezes. Dentre essas cinco RNAs treinadas, escolhia-se aquela que apresentava menor erro de classificação na amostra de teste.

Para acessar o erro de classificação de cada arquitetura testada, aplicou-se a técnica de validação cruzada. A amostra foi dividida em quatro partes (A1, A2, A3 e A4) de tal maneira que cada parte tivesse observações de todas as classes. Duas partes foram utilizadas para treinamento, sendo uma para parada das iterações e outra para teste. Para cada arquitetura, realizavam-se quatro treinamentos e quatro testes (Quadro 1). O erro de classificação foi determinado, somando-se os erros obtidos com as amostras de teste.

Quadro 1 - Disposição das amostras utilizadas para treinamento, parada e teste de validação das RNAs

	Treinamento	Parada	Teste
Rede 1	A2, A3	A1	A4
Rede 2	A1, A4	A2	A3
Rede 3	A1, A4	A3	A2
Rede 4	A2, A3	A4	A1

Outro método de treinamento e teste de RNAs, utilizado neste trabalho foi o 'deixando um de fora'. Este método é uma variante da validação cruzada, sendo utilizado quando a amostra tem número reduzido de observações (HAYKIN, 1999). Para cada treinamento, fica somente uma observação de fora para teste. Assim, consegue-se um maior número de observações na amostra de treinamento. As arquiteturas das RNAs que apresentaram melhores resultados pelo método de treinamento por 'parada mais cedo' foram utilizadas no treinamento e teste neste

método. Os números de iterações testados nos treinamentos foram 500, 1000, 2000, 5000 e 8000. Tendo em vista que, no início do treinamento, os parâmetros livres são gerados aleatoriamente e que esses valores iniciais podem influenciar o resultado final do treinamento, cada arquitetura foi treinada três vezes. Entre essas três RNAs treinadas, escolhia-se aquela que apresentava menor erro de classificação na observação de teste.

Arquitetura e treinamento de uma árvore de decisão de redes neurais artificiais

Foi desenvolvida uma árvore de decisão com redes específicas, para os defeitos que apresentassem níveis de erro superiores a 50%, quando utilizou-se com RNA única. As redes foram desenvolvidas por 'parada mais cedo' e validação cruzada. A amostra foi dividida em quatro partes, sendo duas partes utilizadas para treinamento, uma para parada das iterações e a outra para teste da árvore. A arquitetura da rede foi a mesma selecionada, quando utilizou-se RNA única.

Avaliação dos classificadores

Partindo-se da amostra de validação, os resultados da classificação são organizados em uma matriz quadrada de dimensão igual ao número de classes, denominada matriz de erro ou confusão (CONGALTON, 1991). Nesta matriz, os resultados da classificação são colocados nas colunas. Na diagonal principal, estão os números de observações que têm classificação de acordo com sua referência. No restante da coluna, estão os números de observações omitidas e classificadas como outras classes, denominados erro de omissão ou exclusão. Ao dividir o número de observações omitidas pelo total de observações daquela classe, ou seja, pelo total da coluna, tem-se o erro de omissão por classe. A soma desses erros, em todas as colunas, dividido pelo total de amostras é o erro global, ao passo que, na linha, a soma dos valores fora da diagonal principal é o erro de comissão ou inclusão naquela classe, que é calculado ao ser dividido pelo total na linha (Quadro 2). O erro de omissão (E_O) está relacionado com a 'exatidão do produtor' ($1-E_O$), pois o produtor do classificador tem interesse em conhecer como ocorreu o acerto da classificação dos objetos. O erro de

comissão (E_C) está relacionado com a 'exatidão do usuário' ($1-E_C$), pois, é a confiabilidade que o usuário tem quanto aos resultados da classificação.

Quadro 2. Exemplo de uma matriz de erros (CONGALTON, 1991).

Dados de referência						
Classes	A	B	C	D	Total linha $x_{i\oplus}$	Erro de Comissão (%)
A	65	4	22	24	115	43
B	6	81	5	8	100	19
C	0	11	85	19	115	26
D	4	7	3	90	104	13
Total coluna $x_{\oplus i}$	75	103	115	141	Total	Erro
Erro de omissão (%)	13	21	26	36	geral 434	Global (%) 26

Uma maneira simples de avaliar um classificador é por meio do erro global. Entretanto, este erro independe dos erros de omissão e comissão, que são importantes na avaliação de um classificador. Cohen (1960), citado por ROSENFELD e FITZPATRICK-LINS (1986), desenvolveu um coeficiente de concordância denominado kappa. CONGALTON e MEAD (1983) afirmam que esta é uma técnica de análise estatística multivariada discreta, que pode ser utilizada, pois, os dados em uma matriz de erros são discretos e multinomialmente distribuídos. De acordo com os autores, esta estatística é uma medida do acerto real menos o acerto por acaso. De acordo com ROSENFELD e FITZPATRICK-LINS (1986) o coeficiente kappa pode ser definido de acordo com a equação

$$\hat{K} = \frac{P_o - P_c}{1 - P_c}$$

em que, \hat{K} = estimativa do coeficiente Kappa; P_o = proporção de unidades corretamente classificadas; e P_c = proporção esperada de acerto por acaso.

Conforme HUDSON e RAMM (1987), ao computar os totais marginais na matriz de erro, o coeficiente kappa pode ser estimado por

$$\hat{K} = \frac{n \sum_{i=1}^c x_{ii} - \sum_{i=1}^c x_{i\oplus} x_{\oplus i}}{n^2 - \sum_{i=1}^c x_{i\oplus} x_{\oplus i}}$$

em que, x_{ii} = valor na linha i e coluna i da matriz de erros; $x_{i\oplus}$ = total da linha i ; $x_{\oplus i}$ = total da coluna i ; n = número total da amostra; e c = número total de classes.

O kappa é igual a zero, quando o acerto obtido pelo classificador é igual ao acerto por acaso. Valores positivos de kappa ocorrem, quando o acerto for maior que o acaso, enquanto os negativos ocorrem quando o acerto é pior do que uma classificação por acaso. O maior valor de kappa (+1,00) ocorre, quando existe uma classificação perfeita (Cohen, 1960 citado ROSENFELD e FITZPATRICK-LINS, 1986). O coeficiente kappa pode ser estimado para cada classe individual, empregando-se a equação (Bishop et al., 1975, citado por ROSENFELD e FITZPATRICK-LINS, 1986):

$$\hat{K}_{CL} = \frac{nx_{ii} - x_{i\oplus} x_{\oplus i}}{nx_{i\oplus} - x_{i\oplus} x_{\oplus i}}$$

em que, K_{CL} = coeficiente kappa condicional na linha.

Para o coeficiente kappa individual na coluna, MASCARO & FERREIRA, (2003), sugerem a equação

$$\hat{K}_{CC} = \frac{nx_{ii} - x_{i\oplus} x_{\oplus i}}{nx_{\oplus i} - x_{i\oplus} x_{\oplus i}}$$

em que, K_{CC} = coeficiente kappa condicional na coluna.

Para testar a significância entre dois coeficientes kappa independentes, a um determinado nível de significância, CONGALTON e MEAD, (1983), sugerem a equação

$$Z = \frac{\hat{K}_1 - \hat{K}_2}{\sqrt{\text{var}(\hat{K}_1) + \text{var}(\hat{K}_2)}}$$

em que, Z é padronizado e normalmente distribuído, e a variância pode ser determinada de acordo com (Bishop et al., 1975, citados por HUDSON e RAMM, 1987)

$$\text{var}(\hat{K}) = \frac{1}{n} \left[\frac{\theta_1(1-\theta_1)}{(1-\theta_2)^2} + \frac{2(1-\theta_1)(2\theta_1\theta_2 - \theta_3)}{(1-\theta_2)^3} + \frac{(1-\theta_1)^2(\theta_4 - 4\theta_2^2)}{(1-\theta_2)^4} \right]$$

em que, $\theta_1 = \frac{1}{n} \sum_{i=1}^c x_{ii}$; $\theta_2 = \frac{1}{n^2} \sum_{i=1}^c x_{i\oplus} x_{\oplus i}$; $\theta_3 = \frac{1}{n^2} \sum_{i=1}^c x_{ii} (x_{i\oplus} + x_{\oplus i})$; e

$$\theta_4 = \frac{1}{n^3} \sum_{i=1}^c \sum_{j=1}^c x_{ij} (x_{j\oplus} + x_{\oplus j})^2.$$

As arquiteturas das redes, as técnicas de treinamento e o tamanho de blocos da imagem são avaliados, utilizando-se o coeficiente kappa, enquanto o nível de significância dos testes estatísticos foi de 5%.

RESULTADOS E DISCUSSÃO

O Quadro 3 apresenta os valores de F_0 , calculado para o T^2 de Hotelling dos vetores de médias das classes de defeitos e madeira limpa estudadas, para dois tamanhos de blocos. As diferenças entre as médias das classes 1, 6, 9 e das classes 7 e 8 não foram significativas ($p > 0,01$). Segundo a norma de classificação de madeira serrada de eucalipto (ABNT, 2002), os defeitos 1, 6, 9 e 10 são igualmente quantificados e pontuados para definição das classes de qualidade das peças, assim como os defeitos 7 e 8. Deste modo, pode-se agrupar estes defeitos sem causar prejuízo na classificação final da peça.

Quadro 3 – Valores de F_0 calculado para o T^2 de Hotelling para 12 classes de defeitos e madeira limpa para blocos de imagens de 64x64 pixels acima da diagonal principal e abaixo para blocos de 32x32 pixels

Classes	1	2	3	4	5	6	7	8	9	10	11	12	ml
1	-	5,32	5,05	5,69	3,39	1,18^{ns}	3,96	2,74	1,18^{ns}	2,61	3,72	6,76	14,69
2	3,51	-	7,05	5,42	6,31	5,14	4,41	4,55	5,77	3,66	6,00	7,52	6,87
3	8,23	7,62	-	4,58	4,25	6,44	5,04	3,95	4,96	5,46	3,67	2,33	12,06
4	6,81	6,61	5,96	-	5,48	5,72	2,91	2,99	5,17	3,98	4,05	4,67	5,14
5	3,63	5,36	6,47	4,58	-	5,94	3,25	2,90	4,15	3,97	5,04	6,61	19,69
6	2,5	3,28	10,09	7,89	5,86	-	4,12	2,83	1,49^{ns}	1,99	4,15	7,74	21,01
7	5,5	4,62	7,57	4,00	2,76	4,96	-	0,73^{ns}	3,92	3,23	4,28	5,33	9,41
8	4,11	4,82	4,29	2,12	1,86	4,26	1,65^{ns}	-	2,91	3,01	3,94	4,46	9,89
9	1,41^{ns}	4,18	7,09	5,07	2,53	1,83	3,65	2,28	-	2,39	3,93	6,00	16,16
10	3,68	4,26	7,47	5,7	4,73	2,36	4,37	4,17	3,13	-	3,98	5,45	6,99
11	2,88	3,6	3,75	2,43	2,41	3,31	2,54	2,43	2,63	3,60	-	3,89	5,70
12	7,91	6,34	3,83	6,02	6,04	9,34	6,59	3,98	6,9	7,05	3,20	-	11,12
ml	12,34	6,97	12,4	7,55	11,77	17,47	6,67	7,73	10,33	6,92	2,28	10,23	-

1 = bolsa de goma; 2 = esmoado; 3 = fendilhado; 4 = furos de inseto; 5 = medula; 6 = nó cariado; 7 = nó sadio; 8 = nó sadio corrompido; 9 = nó solto encaixado; 10 = nó solto oco; 11 = podridão; 12 = racha e ml = madeira limpa.

Classes i de uma linha e j de uma coluna não diferem ($p > 0,01$) se o valor de $F_0(i,j)$ é seguido de ns.

O Quadro 4 apresenta os resultados do teste T^2 de Hotelling dos vetores de médias após o agrupamento das classes 1, 6, 9 e 10, bem como das classes 7 e 8 para dois tamanhos de blocos. Os resultados das diferenças entre as médias destas novas classes foram significativos ($p < 0,01$). Portanto, o desenvolvimento de um classificador para discriminar essas classes pode ser viável.

Quadro 4 - Valores de F_0 calculado para o teste T^2 de Hotelling para 8 classes de defeitos e madeira limpa para blocos de imagens de 64x64 pixels acima da diagonal principal e abaixo para blocos de 32x32 pixels

Classes	1/6/9/10	2	3	4	5	7/8	11	12	ml
1/6/9/10	-	5,91	7,05	7,83	7,25	6,97	4,31	8,71	40,30
2	3,92	-	6,89	5,41	6,21	4,92	5,98	7,43	6,90
3	10,56	7,32	-	4,47	4,22	5,20	3,69	2,32	11,84
4	9,45	6,47	5,89	-	5,39	3,74	4,06	4,65	5,14
5	5,98	5,21	6,29	4,54	-	4,11	5,08	6,60	19,47
7/8	7,75	4,98	6,59	3,55	2,52	-	4,39	5,74	16,35
11	3,15	3,54	3,7	2,43	2,43	2,55	-	3,91	5,69
12	10,09	6,14	3,85	6,02	5,96	5,94	3,18	-	11,02
ml	30,11	6,85	11,89	7,35	11,57	11,38	2,27	9,91	-

1 = bolsa de goma; 2 = esmoado; 3 = fendilhado; 4 = furos de inseto; 5 = medula; 6 = nó cariado; 7 = nó sadio; 8 = nó sadio corrompido; 9 = nó solto encaixado; 10 = nó solto oco; 11 = podridão; 12 = racha e ml = madeira limpa.

Classes i de uma linha e j de uma coluna não diferem ($p > 0,01$) se o valor de $F_0(i,j)$ é seguido de ns.

Os resultados do descarte de características pelo método passo-a-passo são apresentados no Quadro 5. Para características retiradas de blocos de 64x64 pixels, ocorreu redução de 36 para 21 variáveis. Para blocos de 32x32 pixels, resultou uma redução de 36 para 18 variáveis. Estes resultados demonstram a dificuldade de se utilizar uma câmera com apenas uma banda espectral na classificação de madeiras, pois, mesmo após o descarte, as características de todas as bandas são necessárias para discriminar as classes.

Quadro 5 - Características selecionadas pelo método passo-a-passo de blocos da imagem de 64x64 e 32x32 pixels

Blocos	Características
64x64	$C_{Vm}^2, C_{Vm}^{10}, C_{Vm}^{30}, C_{Vm}^{60}, C_{Vm}^{70}, C_{Vm}^{90}, C_{Vd}^5, C_{Vd}^{10}, C_{Vd}^{40}, C_{Vd}^{60}, C_{Vd}^{70}, C_{Vd}^{95}, C_{Az}^5, C_{Az}^{10}, C_{Az}^{30}, C_{Az}^{40}, C_{Az}^{50}, C_{Az}^{60}, C_{Az}^{70}, C_{Az}^{90}, C_{Az}^{95}$.
32x32	$C_{Vm}^5, C_{Vm}^{10}, C_{Vm}^{20}, C_{Vm}^{30}, C_{Vm}^{95}, C_{Vd}^5, C_{Vd}^{20}, C_{Vd}^{30}, C_{Vd}^{60}, C_{Vd}^{70}, C_{Vd}^{80}, C_{Az}^2, C_{Az}^5, C_{Az}^{10}, C_{Az}^{20}, C_{Az}^{40}, C_{Az}^{60}, C_{Az}^{70}$.

Classificadores de Redes Neurais

Os Quadros 6, 7, 8 e 9 apresentam os resultados da comparação do tamanho de arquiteturas e métodos de treinamento de RNAs para características de blocos de 64x64 pixels.

No Quadro 8, são apresentados os coeficientes kappa global, variando-se o número de neurônios nas camadas escondidas. Somente foram significativas as diferenças do coeficiente kappa das arquiteturas 21-24-18-9, 21-24-16-9 e 21-20-18-9 com a arquitetura 21-20-16-9. Entretanto, os resultados mostraram que não houve diferenças significativas entre as redes de maior e as de menor número de neurônios escondidos. Por exemplo, a diferença do kappa entre as arquiteturas 21-14-14-9 e 21-26-26-9 não foi significativa.

Quadro 6 - Coeficiente kappa global, em percentagem, das redes neurais com arquiteturas 21-n₁-n₂-9*, para blocos de imagens de 64x64 pixels

n ₁	n ₂						
	26	24	22	20	18	16	14
26	63,7 a b	60,4 a b	61,2 a b	60,5 a b	61,5 a b	64,2 a b	60,0 a b
24		63,5 a b	61,1 a b	61,2 a b	59,1 b	59,6 b	60,8 a b
22			62,7 a b	60,3 a b	63,2 a b	61,7 a b	61,1 a b
20				62,3 a b	59,1 b	65,3 a	60,3 a b
18					60,1 a b	60,1 a b	60,1 a b
16						60,4 a b	62,1 a b
14							63,6 a b

* = número de neurônios nas camadas de entrada, intermediárias e saída, respectivamente. Coeficientes kappa seguidos de mesma letra, não diferem entre si pelo teste Z, ao nível de 5 %.

No Quadro 7, são apresentados os coeficientes kappa globais dos diferentes números de iterações do treinamento, testados com base na técnica 'parada mais cedo', para diferentes arquiteturas de RNA. Somente cinco iterações não foram suficientes para encontrar o melhor ajuste da RNA. No entanto, de 100 a 800 iterações não houve diferenças significativas do valor de kappa. Portanto, 100 iterações são suficientes para qualquer uma das arquiteturas testadas.

Quadro 7 - Coeficientes kappa globais, em percentagem, dos diferentes números de iterações testados segundo a técnica 'parada mais cedo', para arquiteturas de redes 21-n₁-n₂-9* com blocos de imagens de 64x64 pixels

Número de iterações	n1-n2			
	14-14	16-16	18-18	20-20
5	63,6 B a	60,4 B a	60,1 B a	62,3 B a
100	73,8 A a	74,8 A a	72,4 A a	74,4 A a
200	73,8 A a	73,3 A a	73,2 A a	73,4 A a
400	74,6 A a	72,9 A a	74,9 A a	76,0 A a
600	74,4 A a	75,1 A a	75,0 A a	75,3 A a
800	73,2 A a	75,1 A a	75,4 A a	75,2 A a

* = número de neurônios nas camadas de entrada, intermediárias e saída, respectivamente.

Coeficientes kappa seguidos de mesma letra maiúscula na vertical ou minúscula na horizontal, não diferem entre si pelo teste Z, ao nível de 5 % de probabilidade.

Visto que não houve diferenças significativas quanto ao efeito das arquiteturas e que arquiteturas mais simples possuem maior capacidade de generalização e demandam menor esforço computacional, o número de neurônios escolhidos para ambas as camadas escondidas foi 14.

No Quadro 8, são apresentados os resultados do coeficiente kappa global, variando-se o número de iterações no método 'deixando um de fora'. Mil iterações foram suficientes para obter o ajuste da RNA, pois, o mesmo não diferiu significativamente do maior coeficiente kappa obtido, quando o treinamento foi realizado com 2000 iterações.

Quadro 8 - Coeficiente kappa global, em percentagem, da arquitetura de rede neural 21-14-14-9 treinada pelo método “deixando um de fora”, para blocos de imagens de 64x64 pixels

N.º de iterações	Coeficiente kappa global
500	71,2 b
1000	75,2 a b
2000	77,8 a
5000	75,4 a b
8000	76,1 a b

Coeficientes kappa global seguidos de mesma letra, não diferem entre si pelo teste Z, ao nível de 5 %.

No Quadro 9, são apresentados os coeficientes kappa comparando ambos os métodos de treinamento. Comparações múltiplas do coeficiente kappa global não apresentaram diferenças significativas. Entretanto os resultados do kappa por classe nos métodos de treinamento 'deixando um de fora' e árvore de decisão tenderam a resultados superiores nas classes de defeitos 2, 3, 11 e 12, em relação ao método de 'parada mais cedo'. Isto indica que, em tais classes de defeito, o número de observações não foi suficientemente representativo para o ajuste da rede por meio deste método.

Quadro 9 - Coeficiente kappa global e kappa por classe nas colunas, em percentagem, da rede única com dois métodos de treinamento e da árvore de decisão, para blocos de imagens de 64x64 pixels

Classes	1/6/9/10	2	3	4	5	7/8	11	12	ml	Global ^a
-----Rede Única (21-14-14-9) [*] , Parada mais cedo -----										
	92,7	0,0	35,5	26,7	65,0	83,3	0,0	12,1	96,1	73,2
N.º de iterações -----Rede Única (21-14-14-9) [*] , Deixando um de fora-----										
2000	95,2	44,0	48,3	58,5	77,0	78,3	0,0	5,5	91,0	77,8
-----Árvore de decisão (21-14-14-x ^{**}), Parada mais cedo-----										
	96,3	44,0	35,9	12,0	68,6	77,9	-0,4	23,9	90,7	72,7

1-bolsa de goma; 2-esmoado; 3-fendilhado; 4-furos de inseto; 5-medula; 6-nó cariado; 7-nó sadio; 8-nó sadio corrompido; 9-nó solto encaixado; 10-nó solto oco; 11-podridão; 12-racha e ml - madeira limpa.

* = Arquitetura de RNA escolhida para rede única, com 21 neurônios na camada de entrada, 14 nas intermediárias e nove na saída.

** = número de neurônios de saída nas ramificações da árvore, x=2 nas cinco ramificações da árvore para discriminar as classes específicas de defeitos; x=4 na última ramificação da árvore para discriminar as quatro classes restantes.

^a= comparações múltiplas de coeficientes kappa global pelo teste Z, não diferiram entre si, ao nível de 5 %

Os Quadros 10, 11, 12 e 13 apresentam os resultados da comparação do tamanho de arquiteturas e métodos de treinamento de RNAs para características de blocos de 32x32 pixels.

Não se obteve diferenças significativas dos coeficientes kappa em comparações múltiplas dos tamanhos de arquitetura (Quadro 10). Assim, analogamente ao observado para blocos de 64x64 pixels, o número de neurônios escolhidos para as camadas escondidas foi 14.

Quadro 10 - Coeficiente kappa global, em percentagem, das redes neurais com arquiteturas 21-n₁-n₂-9*, para blocos de imagens de 32x32 pixels^a

n ₁	n ₂						
	26	24	22	20	18	16	14
26	60,2	61,6	59,4	60,0	59,4	60,1	59,7
24		59,9	57,9	59,3	61,6	57,3	59,7
22			60,1	59,4	59,6	60,0	57,8
20				58,6	60,4	58,8	58,3
18					58,4	60,8	58,2
16						58,3	56,5
14							58,2

* = número de neurônios nas camadas de entrada, intermediárias e saída, respectivamente.

^a = comparações múltiplas de coeficientes kappa pelo teste Z, não diferiram entre si, ao nível de 5 %.

Conforme se observa no Quadro 11, ao variar o número de iterações para 'parada mais cedo' do treinamento, verificou-se que o número de 100 iterações não diferiu, estatisticamente, dos melhores resultados para as arquiteturas testadas. Portanto, 100 iterações são suficientes para este tamanho de bloco, independentemente da variação da arquitetura.

Quadro 11 - Coeficientes kappa globais, em percentagem, dos diferentes números de iterações testados segundo o método de 'parada mais cedo', para arquiteturas de redes 21-n₁-n₂-9* com blocos de imagens de 32x32 pixels

Número de iterações	n ₁ -n ₂							
	14-14		16-16		18-18		20-20	
5	58,2	B a	58,3	B a	58,4	B a	58,6	B a
100	62,9	A B a	64,2	A a	64,4	A a	62,2	A B a
200	63,8	A B a	64,8	A a	65,7	A a	64,3	A B a
400	63,6	A B a	64,9	A a	65,2	A a	64,7	A a
600	65,0	A a	65,8	A a	64,9	A a	64,5	A a
800	65,1	A a	65,8	A a	64,9	A a	65,7	A a

* = número de neurônios nas camadas de entrada, intermediárias e saída, respectivamente.

Coeficientes kappa seguidos de mesma letra maiúscula na vertical ou minúscula na horizontal, não diferem entre si pelo teste Z, ao nível de 5 % de probabilidade.

No Quadro 12, comparando a variação no número de iterações pelo método de treinamento 'deixando um de fora', não foram obtidas diferenças significativas do

coeficiente kappa. Ou seja, para os números de iterações testados, não houve nem 'memorização' da amostra com 8000 iterações, nem falta de ajuste da RNA com apenas 500 iterações.

Quadro 12 - Coeficiente kappa global, em percentagem, da arquitetura de rede neural 21-14-14-9 treinada pelo método “deixando um de fora”, para blocos de imagens de 32x32 pixels

Número de iterações	Coeficiente kappa global ^a
500	64.6
1000	66.2
2000	68.8
5000	69.3
8000	66.1

^a = comparações múltiplas de coeficientes kappa pelo teste Z, não diferiram entre si, ao nível de 5 %.

No Quadro 13, são apresentados os coeficientes kappa, comparando-se os métodos de treinamento. Comparações múltiplas do coeficiente kappa global dos métodos de treinamento não apresentaram diferenças significativas. Os resultados do kappa por classe nos métodos de treinamento 'deixando um de fora' e árvore de decisão, entretanto, tenderam a resultados superiores nas classes de defeitos 2 e 12 em comparação com o método 'parada mais cedo'. Isto indica que, nestas classes de defeito, o número de observações não foi, suficientemente, representativo para ajuste da rede por meio deste método de treinamento.

Quadro 13 - Coeficiente kappa global e condicional na coluna, em percentagem, para rede única com dois métodos de treinamento e árvore de decisão, para blocos de imagens de 32x32 pixels

Classes	1/6/9/10	2	3	4	5	7/8	11	12	ml	Global ^a
-----Rede Única (18-14-14-9) [*] , Parada mais cedo-----										
	91,5	-0,2	42,3	10,9	49,8	54,4	0,0	-0,2	96,1	65,1
-----Rede Única (18-14-14-9) [*] , Deixando um de fora-----										
N.º de iterações	90,5	54,9	48,1	40,7	56,8	53,1	0,0	28,8	88,1	69,3
5000										
-----Árvore de decisão (18-14-14-x ^{**}), Parada mais cedo-----										
	90,4	32,5	55,2	32,1	52,6	56,0	0,0	36,5	92,4	69,4

1-bolsa de goma; 2-esmoado; 3-fendilhado; 4-furos de inseto; 5-medula; 6-nó cariado; 7-nó sadio; 8-nó sadio corrompido; 9-nó solto encaixado; 10-nó solto oco; 11-podridão; 12-racha e ml - madeira limpa.

* = Arquitetura de RNA neural escolhida para rede única, com 18 neurônios na camada de entrada, 14 nas intermediárias e nove na saída.

** = número de neurônios de saída nas ramificações da árvore, x=2 nas cinco ramificações da árvore para discriminar as classes específicas de defeitos e; x=4 na última ramificação da árvore para discriminar as quatro classes restantes.

^a = comparações múltiplas de coeficientes kappa pelo teste Z, não diferiram entre si, ao nível de 5 % de probabilidade.

No Quadro 14 são comparados os dois tamanhos de blocos nos melhores resultados encontrados nos métodos de treinamento estudados. O método 'deixando um de fora', para blocos de 64x64 pixels, apresentou melhor resultado que os métodos de treinamento realizados para os blocos de 32x32 pixels. Porém, os métodos 'deixando um de fora' e 'árvore de decisão' dos blocos de 32x32 pixels não diferiram dos métodos 'parada mais cedo' e 'árvore de decisão' dos blocos de 64x64 pixels.

Apesar do melhor resultado para bloco de 64x64 pixels, não se deve descartar a utilização de blocos menores em um classificador, pois, estes apresentam uma melhor resolução espacial na imagem classificada. Conseqüentemente, os formatos dos defeitos serão mais bem representados e quantificados. Portanto, o tamanho de bloco 32x32 pixels deve ser, também, utilizado na identificação de defeitos de madeira.

Quadro 14 - Coeficiente kappa para dois tamanhos de blocos de imagem e diferentes métodos de treinamento

Classes	1/6/9/10	2	3	4	5	7/8	11	12	ml	Global
Blocos 64x64										
Parada mais cedo	92,7	0,0	35,5	26,7	65,0	83,3	0,0	12,1	96,1	73,2 a b
Deixando um de fora	95,2	44,0	48,3	58,5	77,0	78,3	0,0	5,5	91,0	77,8 a
Árvore de decisão	96,3	44,0	35,9	12,0	68,6	77,9	-0,4	23,9	90,7	72,7 a b
Blocos 32x32										
Parada mais cedo	91,5	-0,2	42,3	10,9	49,8	54,4	0,0	-0,2	96,1	65,1 c
Deixando um de fora	90,5	54,9	48,1	40,7	56,8	53,1	0,0	28,8	88,1	69,3 b c
Árvore de decisão	90,4	32,5	55,2	32,1	52,6	56,0	0,0	36,5	92,4	69,4 b c

1-bolsa de goma; 2-esmoado; 3-fendilhado; 4-furos de inseto; 5-medula; 6-nó cariado; 7-nó sadio; 8-nó sadio corrompido; 9-nó solto encaixado; 10-nó solto oco; 11-podridão; 12-racha e ml - madeira limpa.

Coeficientes kappa global seguidos de mesma letra, não diferem entre si pelo teste Z, ao nível de 5 % de probabilidade.

Nos Quadros 15 e 16, são apresentadas as matrizes de erro para os melhores resultados, para blocos de 64x64 e 32x32 pixels, respectivamente. Os maiores erros ocorreram nas classes de defeitos com menor número de observações na amostra. Como são realmente menos frequentes nas madeiras, esses defeitos teriam um peso menor no desempenho final de um sistema de classificação automática de defeitos. Os defeitos furo de inseto e podridão apresentaram os piores resultados, por terem sido incluídos na classe de madeira limpa. Isto poderia classificar uma peça de quinta classe como de primeira (ABNT, 2002). Os defeitos fendilhado e racha também tiveram altos níveis de erro, porém foram confundidos entre si, embora a localização destes são distintas na peça. Tal fato poderia ajudar na distinção entre eles. O defeito nó sadio foi confundido com madeira limpa, o que afetaria pouco na classificação final, pois este defeito afeta somente a primeira classe de qualidade da peça (ABNT, 2002) e algumas serrarias nem o consideram como defeito (ARACRUZ, 2000).

Quadro 15 - Matriz de confusão para rede neural única com treinamento 'deixando um de fora' com 2000 iterações, blocos de imagens de 64x64 pixels

Classes	Dados de referência									Total	Erro* comissão
	1/6/9/10	2	3	4	5	7/8	11	12	ml		
1/6/9/10	104	2	1	0	0	3	1	2	0	113	8,0
2	0	4	0	0	0	0	0	0	0	4	0,0
3	1	0	8	0	0	0	2	5	0	16	50,0
4	0	0	2	21	2	1	0	5	8	39	46,2
5	1	0	3	2	43	5	0	2	0	56	23,2
7/8	2	1	1	0	3	51	0	0	2	60	15,0
11	0	0	0	0	0	0	0	0	0	0	0,0
12	0	0	1	0	1	1	0	1	0	4	75,0
ml	0	2	0	11	5	2	2	1	177	200	11,5
Total	108	9	16	34	54	63	5	16	187	Erro global 16,9	
Erro omissão	3,7	55,6	50,0	38,2	20,4	19,0	100,0	93,8	5,3	Exatidão global 83,1	

1-bolsa de goma; 2-esmoado; 3-fêndilhado; 4-furos de inseto; 5-medula; 6-nó cariado; 7-nó sadio; 8-nó sadio corrompido; 9-nó solto encaixado; 10-nó solto oco; 11-podridão; 12-racha e ml -madeira limpa.

* = valores de erro, kappa e exatidão em percentagem.

Para confirmar se as características de percentis não foram eficientes na identificação dos defeitos que apresentaram altos níveis de erro, deve-se coletar um maior número de observações. No entanto, a exatidão dos classificadores de 83,1 e 76,6%, respectivamente para os blocos de 64x64 e 32x32 pixels, foi considerada satisfatória para que fossem utilizados na identificação de defeitos de madeira. Os defeitos de maior frequência e importância na pontuação das classes de qualidade das peças de madeira serrada obtiveram baixos níveis de erro comparativamente à classificação humana (HUBER et al., 1985).

Quadro 16 - Matriz de confusão para rede neural única com treinamento "deixando um de fora" para 5000 iterações, blocos de imagens de 32x32 pixels

		Dados de referência									Total	Erro* comissão
Classes	1/6/9/10	2	3	4	5	7/8	11	12	ml			
1/6/9/10	100	3	0	0	1	4	0	1	0	109	8,3	
2	1	5	0	0	0	0	0	0	1	7	28,6	
3	0	0	8	2	1	2	0	5	0	18	55,6	
4	0	0	2	15	3	3	1	0	4	28	46,4	
5	1	0	0	5	33	8	0	1	1	49	32,7	
7/8	5	1	0	1	7	37	0	2	6	59	37,3	
11	0	0	0	0	0	0	0	0	0	0	0,0	
12	0	0	6	2	1	1	1	5	1	17	70,6	
ml	1	0	0	9	8	8	3	2	174	205	15,1	
Total	108	9	16	34	54	63	5	16	187	Erro global 23,4		
Erro omissão	7,4	44,4	50,0	55,9	38,9	41,3	100,0	68,8	7,0	Exatidão global 76,6		

1-bolsa de goma; 2-esmoado; 3-fendilhado; 4-furos de inseto; 5-medula; 6-nó cariado; 7-nó sadio; 8-nó sadio corrompido; 9-nó solto encaixado; 10-nó solto oco; 11-podridão; 12-racha e ml -madeira limpa.

* = valores de erro, kappa e exatidão em percentagem.

CONCLUSÕES

As arquiteturas escolhidas para desenvolvimento das redes neurais foram aquelas com 14 neurônios nas camadas escondidas, para os dois tamanhos de blocos testados.

No método de treinamento por 'parada mais cedo' com cinco iterações, obteve-se menor coeficiente kappa, em comparação com 100 a 800 iterações, para os dois tamanhos de blocos.

No método de treinamento 'deixando um de fora', o número de 2000 iterações obteve coeficiente kappa superior, quando comparado com 500 iterações para blocos de 64x64 pixels e para blocos de 32x32 pixels, não existiram diferenças significativas.

Os métodos de treinamento avaliados em cada tamanho de bloco, não apresentaram diferenças significativas quanto ao coeficiente kappa. Porém, existiu uma tendência de melhor coeficiente kappa individual nas classes de menor frequência, quando se utilizaram utilizados os métodos 'deixando um de fora' e 'árvore de decisão'.

Comparando os dois tamanhos de blocos da imagem, o método de treinamento 'deixando um de fora' do tamanho de bloco de 64x64 pixels apresentou coeficiente kappa superior aos demais métodos de treinamento do tamanho de bloco de 32x32 pixels.

Classificadores por redes neurais artificiais utilizando-se características de percentis das bandas vermelho, verde e azul foram considerados satisfatórios na identificação dos defeitos de madeira serrada de eucalipto obtendo exatidão global de 83,1 e 76,6% para os blocos de 64x64 e 32x32 pixels, respectivamente.

REFERÊNCIAS BIBLIOGRÁFICAS

- ARACRUZ. Descrição do produto. Aracruz Produtos de Madeira S.A., Nova Viçosa, BA, 2000. 9p.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS – **ABNT**. Madeira serrada de eucalipto – Requisitos, NBR 14806. Rio de Janeiro, 2002. 11 p.
- CONGALTON, R.G. A review of assessing the accuracy of classifications of remotely sensed data. *Remote sensing of environment* n. 37 p. 35-46, 1991.
- CONGALTON, R.G.; MEAD, R.A. A quantitative method to test for consistency and correctness in photointerpretation. *Photogrammetric Engineering & Remote Sensing*, v.49 n.1 p.69-74, 1983.
- GONZALEZ, R.C.; WOODS, R.E. **Processamento de imagens digitais**. São Paulo: Editora Edgard Blücher Ltda, 1992, 509p.
- HANEY, L.; PRECETTI, C.; GIBSON, H.; KRUTZ, G.; CASSENS, D. Color matching of wood with a real-time machine vision system. **ASAE Paper No. 943579**, ASAE Meeting Presentation, St Joseph, Michigan, 1994, 15 p.
- HAYKIN, S. **Neural Networks: a comprehensive foundation**. 2 ed. New Jersey: Prentice Hall, 1999, 842 p.
- HUBER, H. A.; McMILLIN, C. W.; McKINNEY, J. P. Lumber defect detection abilities of furniture rough mill employees, **Forest Products Journal**, 35(11/12), p. 79-82, 1985.

- HUDSON, W.D.; RAMM, C.W. Correct formulation of the kappa coefficient of agreement. *Photogrammetric Engineering & Remote Sensing*, v.53 n.4 p.421-422, 1987.
- KAUPPINEN, H. **Development of a color machine vision method for wood surface inspection**. Oulu, Finlandia: Department of Electrical Engineering and Infotech Oulu, - University of Oulu, Dissertation, 138p, 1999.(<http://herkules.oulu.fi/issn03553213/>)
- KOIVO, A. J.; KIM, C. W. Automatic classification of surface defects on red oak boards, **Forest Products Journal**, 39(9), p. 22-30, 1989.
- MASCARO, S. A.; FERREIRA, M. C. análise comparativa entre algoritmos de classificação digital de imagem com base na exatidão do mapeamento do uso e cobertura do solo: um exemplo na área de influência do reservatório de Jurumirim - SP. In: SIMPÓSIO BRASILEIRO DE SENSORIAMENTO REMOTO, XI, 2003, Belo Horizonte. Anais ... Belo Horizonte: INPE, 2003. p. 1365 - 1372.
- MATEUS, G.R.; LUNA, H.P.L. **Programação não linear**. Editora Gráfica Formato Ltda. Belo Horizonte, MG. 289 p. 1986.
- MATLAB User's Guide, The Math Works Inc., 2000
- PACKIANATHER, M. S.; DRAKE, P. R. Neural networks for classifying images of wood veneer part 2. *International Journal of advanced manufacturing Technology*, 16(6), p.424-433, 2000.
- PRECHELT, L. Automatic early stopping using cross validation: quantifying the criteria. **Neural Networks** , 11, p 761-767, 1998.
- PINTO, F. A. C; REID, J. F.; ZHANG, Q.; NOGUCHI, N. Guidance parameter determination using artificial neural network classifier, **ASAE Paper No. 993004**, ASAE Meeting Presentation, University UILU, Toronto, Canada , 14 p, 1999.
- REGAZZI, A. J. **Análise multivariada**, notas de aula INF 766, Departamento de Informática da Universidade Federal de Viçosa, vol. 2, 2000.
- ROSENFELD, G.H.; FITZPATRICK-LINS, K. A coefficient of agreement as a measure of thematic classification accuracy. *Photogrammetric Engineering & Remote Sensing*, v.52 n.2 p.223-227, 1986.

SARIGUL, E.; ABBOTT, A. L.; SCHMOLDT, D. L. Rule-driven defect detection in CT images of hardwood logs, **Computers and electronics in agriculture**, 19, p.101-119, 2003.

SAS Institute. SAS Online Doc. Version 8, SAS Institute Inc., Cary, USA, 1999.

CLASSIFICAÇÃO DE MADEIRA SERRADA DE EUCALIPTO POR VISÃO ARTIFICIAL

RESUMO

A automatização de processos pode ser realizada por meio de um sistema de visão artificial que consiste das seguintes etapas: aquisição, processamento e análise de imagens. Neste trabalho, foi desenvolvido e testado um algoritmo de um sistema de visão artificial para classificação de madeira serrada de eucalipto pela aparência de acordo com a norma brasileira NBR 14806. Um algoritmo foi implementado para identificação de blocos da imagem com defeitos e madeiras limpa, mensuração das dimensões dos defeitos e classificação das peças de madeira. O algoritmo foi testado em imagens de 20 faces largas de madeira serrada de eucalipto e apresentou exatidão global de 65% na classificação de cada peça, independentemente. Entretanto, a exatidão foi de 90% na classificação da peça, quando ambas as faces foram avaliadas simultaneamente. Os erros causados na classificação não foram devido à identificação de madeira limpa, mas principalmente na confusão entre alguns tipos de defeitos.

Palavras-chave: processamento de imagem, classificação de madeira.

CLASSIFICATION OF EUCALYPTUS SAWN LUMBER BY MACHINE VISION SYSTEM

ABSTRACT

Process automation may be accomplished by a vision machine system consisting of the following stages: acquisition, processing and image analysis. An study was carried out to develop and test an algorithm of a machine vision system for classification of the eucalyptus sawn lumber based on appearance, according to the Brazilian norm NBR 14806. The algorithm was implemented for: identifying the image blocks of defects and clear wood, dimension measurements and board grading. The algorithm was tested on 20-face images of eucalyptus sawn lumber, and showed an overall accuracy of 65% in classification of each board face. However, an accuracy

of 90% was attained in classification of the board, when both faces were simultaneously evaluated. The occurrence of errors in classification was not due to the identification of clear wood, but mainly because the embarrassment among some kinds of defects.

INTRODUÇÃO

Na seleção de madeira serrada para fabricação de móveis e utilização em projeto de interiores, a aparência é um importante requisito. Classificando e selecionando-se as peças, pode-se conseguir um produto com a aparência desejada, mesmo em madeiras oriundas de florestas plantadas sem manejo adequado. Desta forma, a indústria de madeira serrada tem dispensado especial atenção às etapas de classificação e seleção dessas peças. Nessas etapas, geralmente, o procedimento é realizado por classificadores humanos. A etapa de classificação é considerada uma atividade estressante, em razão da exigência de concentração para identificar e mensurar defeitos das madeiras e classificá-las por tempo prolongado.

Os sistemas que utilizam as informações, adquiridas por meio da energia eletromagnética refletida pelos objetos, para automatizar processos são conhecidos como sistema de visão artificial (SVA). Um SVA para inspeção de superfícies consiste das seguintes etapas: aquisição, processamento e análise de imagens (PHAM & ALCOCK, 1998). Na primeira etapa, são utilizados diversos tipos de sensores, sendo os sensores ópticos os mais utilizados. Na etapa de processamento da imagem, o objetivo é a segmentação (realce) e o reconhecimento (identificação e mensuração) dos objetos de interesse (KLINE et al., 1992). Na etapa de análise, a informação é transformada de imagem digital para expressões, que afetam a classificação da madeira serrada em: peça da classe A, peça descartada, dentre outros.

Quando implementado na indústria, o SVA não apresenta o mesmo desempenho da etapa de desenvolvimento, degradando a exatidão da classificação. Em geral, este fato, ocorre por que os algoritmos desenvolvidos para estes sistemas não se ajustam às variações e imprevistos, que ocorrem no meio industrial. No entanto, esses sistemas têm se mostrado superior à fraca exatidão da classificação realizada visualmente.

HUBER et al. (1985) pesquisaram o desempenho de seis classificadores humanos, em três serrarias. Concluíram que o acerto médio foi de 68%, nas etapas de reconhecimento, identificação e localização de defeitos em tábuas de carvalho vermelho. Provavelmente, um SVA é capaz de executar esta tarefa, mesmo que não seja perfeito, com potencial para obter resultados melhores, além de não ser afetado pela fadiga ou outros fatores humanos, que podem afetar o rendimento.

Avaliando sistemas automatizados por visão artificial, para classificação de madeira serrada, pesquisadores concluíram, que os erros cometidos na etapa de processamento de imagens estão, diretamente, relacionados à exatidão final do sistema (PHAM & ALCOCK, 1998; KLINE et al., 2001).

As câmeras coloridas, com fotossensores sensíveis nas bandas do vermelho (Vm), verde (Vd) e azul (Az), são um dos métodos de aquisição de imagens mais estudados por pesquisadores, em classificação de superfícies de madeiras (SZYMANI & McDONALD, 1981; KLINE et al., 1998; HANEY et al., 1994; LU, 1997; KAUPPINEN, 1999). Essas pesquisas têm demonstrado que, ao utilizar o espaço de cor Vm, Vd e Az, obtém-se um melhor desempenho na classificação de madeiras, em comparação com as câmeras monocromáticas.

KAUPPINEN (1999) desenvolveu um classificador de k vizinhos mais próximos, k-NN, para seis tipos de defeitos de tábuas de pinos, utilizando características de percentis do histograma das bandas Vm, Vd e Az. Quando o programa foi implementado em um protótipo, o erro de classificação foi de 34%. O alto nível de erro foi atribuído à dificuldade em definir algumas classes de defeitos da amostra de treinamento do classificador. Como exemplo, cita-se o nó preto que é uma subcategoria do nó seco.

Nos capítulos anteriores do presente trabalho, foram desenvolvidos classificadores para identificação de defeitos em madeiras serradas de eucalipto, mencionadas na norma NBR 14806 (ABNT, 2002), utilizando-se imagens digitais. Os menores erros globais na identificação dos defeitos e madeira limpa foram, em média, de 19%. Os maiores erros foram atribuídos às classes de defeitos, que apresentaram números menores de observações na amostra utilizada para definição dos padrões. Na prática, tais defeitos como o furo de inseto, esmoado e podridão não são muito

freqüentes em madeira serrada, quando extraídas de árvores destinadas à produção de materiais sólidos de madeira. Os resultados indicaram que os classificadores desenvolvidos podem ser empregados na identificação de defeitos, para classificação de madeiras serradas com base em padrões de qualidade.

Neste trabalho, objetivou-se desenvolver e testar um algoritmo para um sistema de visão artificial para classificação de madeira serrada de eucalipto pela aparência de acordo com as classes de qualidade mencionadas na norma NBR 14806 (ABNT, 2002).

MATERIAL E MÉTODOS

Foram amostradas dez peças de madeira serrada de eucalipto secas ao ar e previamente aplainadas, provenientes do Laboratório de Propriedades Físicas e Mecânicas da Madeira da UFV.

Para aquisição das imagens, foi utilizada a câmera colorida Duncantech, modelo MS3100, conectada a um computador por meio de uma placa de aquisição de vídeo National Instruments, modelo PCI 1424. A altura da câmera foi de 0,9 m da superfície da face larga da peça de madeira serrada, sendo a resolução espacial foi de 0,24 x 0,24 mm por pixel. O sistema foi iluminado com duas lâmpadas halógenas de 300 w, instaladas em refletores com difusor de luz. O iluminamento na superfície da madeira serrada foi de 2500 lux.

O algoritmo, utilizado neste trabalho, foi desenvolvido no Laboratório de Projeto de Máquinas e Visão Artificial do departamento de Engenharia Agrícola da Universidade Federal de Viçosa e implementado na plataforma do MatLab®, com o pacote de ferramentas para processamento de imagens.

O objetivo final do desenvolvimento desse algoritmo foi transformar a informação de uma imagem digital, imagem das faces largas de madeira serrada, em uma informação de dados tabelados das quantidades de defeitos, em relação à madeira limpa, que servirão para classificar a peça em cinco classes de qualidade ou descartada das exigências mínimas de qualidade da regra de classificação adotada.

O algoritmo foi desenvolvido, simulando-se a entrada de dados de uma imagem adquirida em uma esteira transportadora de fundo preto, contendo uma peça de

madeira serrada (Figura 1). O campo de visão da câmera foi de 0,33 m transversal e 0,25 m longitudinal ao deslocamento da esteira, sendo necessárias de 5 a 9 imagens para cobrir a extensão total das faces largas das madeiras serradas.



Figura 1 - Sistema de aquisição de imagens simulando uma esteira transportadora.

Na Figura 2 estão as etapas do algoritmo desenvolvido neste trabalho, desde a aquisição da imagem até a classificação final da peça. A seguir, são apresentadas as três fases do algoritmo: pré-processamento, processamento e análise da imagem.

As três primeiras etapas formaram a fase de pré-processamento, cujo objetivo era formar uma imagem da peça de madeira, subdividida em blocos, para a etapa de processamento. Inicialmente, as imagens são agrupadas até cobrir toda a extensão da face larga da peça, formando a 'imagem de madeira e fundo'. Em seguida, o fundo da imagem é retirado por meio de um algoritmo, desenvolvido para identificar as bordas da peça. Os seguintes critérios foram adotados, para presença de madeira: colunas na imagem que apresentasse mais de 200 pixels, com valores de nível de cinza da imagem monocromática superior a 50 e linhas que apresentassem mais de 35 pixels com valores de nível de cinza da imagem monocromática superior a 50, sendo estes os limites longitudinais e transversais entre a madeira e fundo. Finalmente, ajustou-se a imagem em blocos de 32x32 pixels, para retirada de características de percentis, que representassem cada bloco no classificador de classes de defeitos ou madeira limpa. Este tamanho de bloco foi escolhido por representar melhor o formato dos defeitos do que o bloco de 64x64 pixels, embora neste último tamanho de bloco a exatidão global

na identificação de defeitos, utilizando características de percentis, foi em média 5% superior (Capítulo 1).

As etapas de quatro a oito formaram a fase de processamento da imagem, cujo objetivo era formar uma imagem com os defeitos e madeira limpa identificados. Para classificar cada bloco da imagem, foi utilizada a função discriminante linear com escores das cinco primeiras variáveis canônicas. Esta função apresentou erro de classificação global de 26% para blocos de 32x32 pixels (Capítulo 1). As probabilidades *a priori* das classes de defeitos e madeira limpa foram alteradas na função discriminante, pois, as amostras de madeira limpa em relação aos defeitos não foram às mesmas das condições de desenvolvimento das funções discriminantes. A probabilidade de madeira limpa foi alterada de 38% para 90%, e os outros 10% restantes distribuídos entre os defeitos, proporcionalmente às suas ocorrências, observadas na amostra das peças de madeira serrada.

Nas etapas cinco e seis, inicialmente, a imagem classificada por tipos de defeitos foi transformada em imagem binária: zero para madeira limpa e um para defeitos. Nas imagens binárias, foram aplicadas um algoritmo de identificação, rotulação e agrupamento dos blocos de valor um, em que cada agrupamento foi considerado um objeto (defeito). A rotina "BWLABEL" do MATLAB foi utilizada nessa etapa, sendo esta a implementação do algoritmo desenvolvido por Haralick & Shapiro (1992), citado por MATLAB (2000). Em seguida, foram eliminados os objetos que tinham dimensões inferiores a três blocos, sendo os mesmos assumidos como sendo pequenas irregularidades ou sujeiras da madeira limpa.

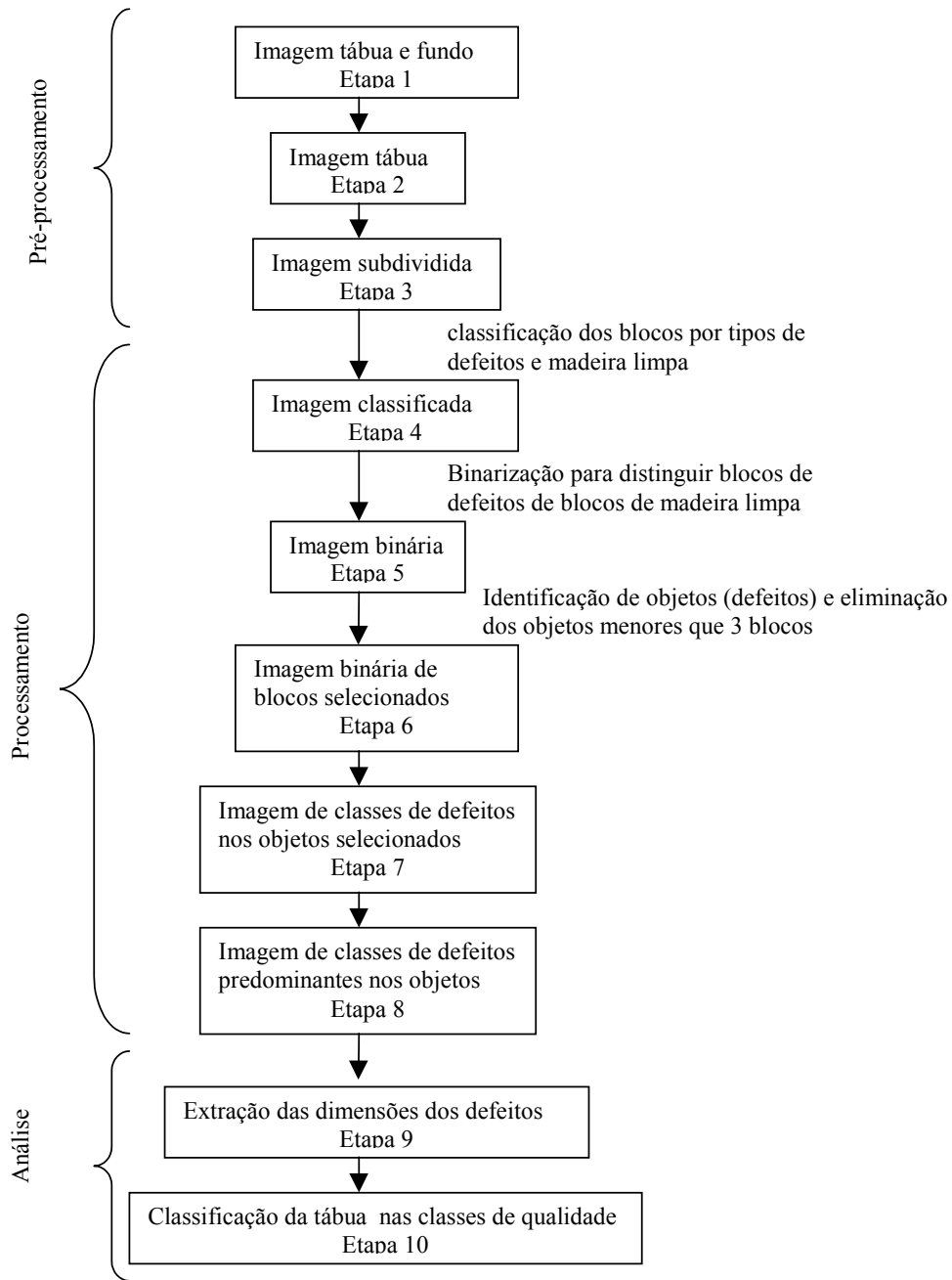


Figura 2 - Etapas do algoritmo de classificação de madeira serrada de eucalipto

Nas etapas sete e oito, realizou-se uma reclassificação dos objetos selecionados na etapa anterior. Assumindo-se que um objeto continha um mesmo tipo de defeito, este era classificado pela predominância da classe de defeito de seus blocos.

As etapas nove e dez formaram a fase de análise da imagem. Na etapa nove, foram extraídas as medidas das dimensões dos objetos, necessárias à classificação da

peça com a rotina 'IMFEATURE' (MATLAB, 2000). Com base na norma de classificação adotada, a medida de comprimento dos defeitos é paralela ao comprimento da peça, assim como a de largura dos defeitos é paralela à largura da peça. As medidas de comprimento, largura ou área dos defeitos são tomadas em relação às mesmas medidas totais da peça. Desta maneira, comprimentos totais de bolsa goma, nós cariados ou soltos, nós sadios e rachas foram representados em mm/m; os comprimentos e larguras totais de esmoados e fendilhados foram representados em cm/m; as áreas totais das podridões foram representadas em porcentagem, enquanto para os defeitos furos de inseto e medula, registrou-se a presença ou não na peça.

Finalmente, na última etapa do algoritmo, foram implementadas as regras da norma adotada para inclusão da peça na classe de qualidade da madeira serrada. Na Figura 3, são apresentadas as rotinas para cada tipo de defeito. As regras para inclusão em uma classe de qualidade de madeira são independentes por tipo de defeito. A face larga da peça foi classificada com base na regra de decisão do defeito que tiver a classe mais inferior.

Avaliação do algoritmo de classificação

As peças de madeira selecionadas foram classificadas, visualmente, em suas duas faces largas. Elaborou-se um relatório com os resultados da classificação por face e com a identificação dos defeitos encontrados em cada face. A classe da pior face é considerada a classe final da peça, segundo a norma adotada. No entanto, os resultados serão também apresentados por face.

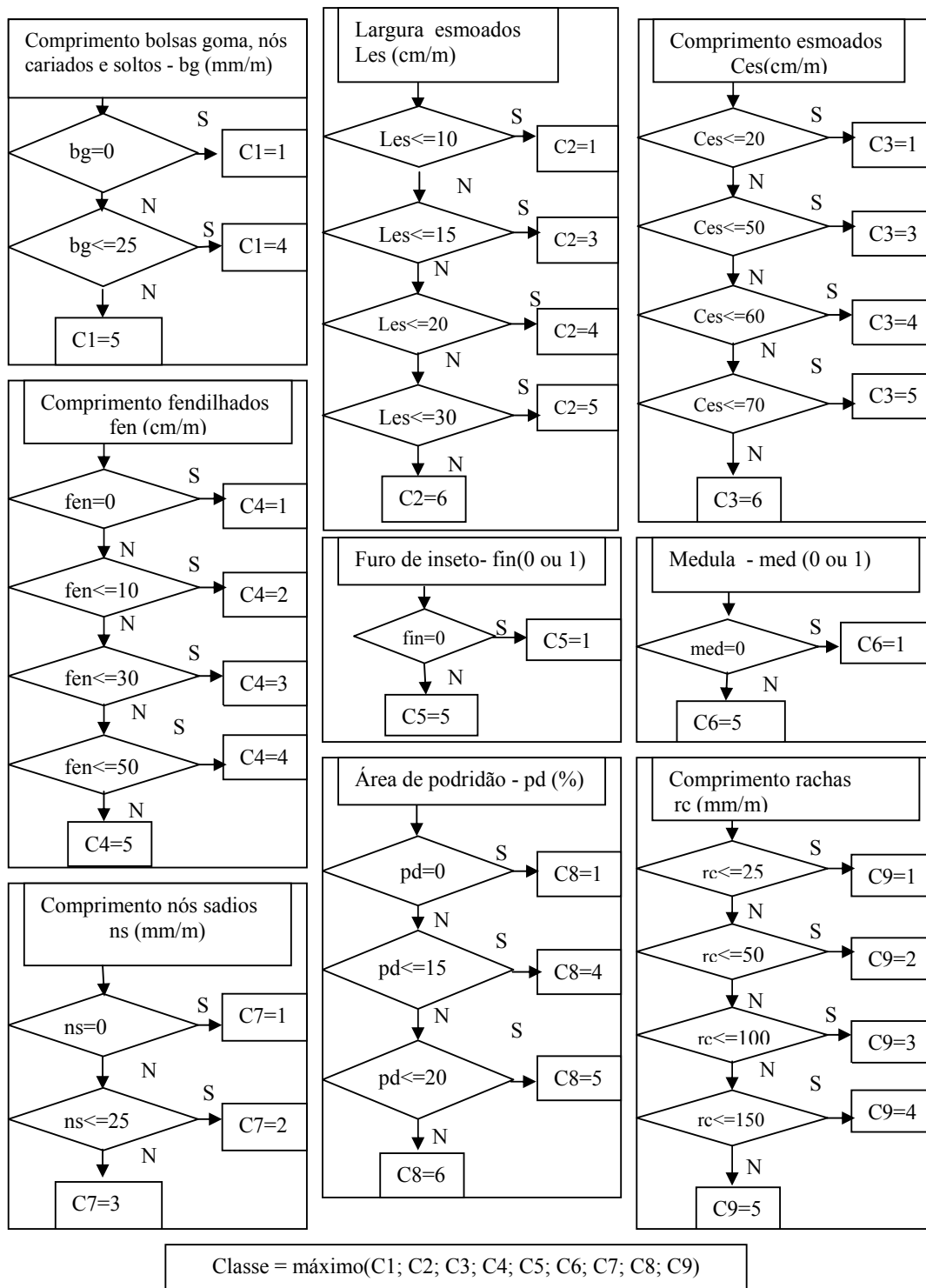


Figura 3 - Rotinas do algoritmo das regras de classificação de uma face da madeira.

Das próprias madeiras serradas, foram adquiridas suas imagens, sendo as mesmas classificadas pelo programa desenvolvido, para comparação com os resultados classificados visualmente. Os resultados da classificação foram organizados em uma matriz quadrada de dimensão igual ao número de classes das faces das peças e a classificação da peça considerando a sua pior face. Em uma outra matriz de resultados, foi avaliada a identificação das classes de defeitos nas peças.

Uma forma de apresentação dos resultados é por meio da matriz de erro ou confusão, (CONGALTON, 1991). Nesta matriz, os resultados da classificação são colocados nas colunas, sendo que, na diagonal principal, estão os números de observações que tiverem classificação de acordo com sua referência. No restante da coluna, está o número de observações omitidas e classificadas como outras classes, o qual é denominado erro de omissão ou exclusão. Ao dividir esse erro pelo total de observações daquela classe, ou seja, pelo total da coluna, tem-se o erro de omissão por classe. A soma desses erros dividido pelo total de amostras é o erro global.

Nas linhas da matriz de erro, a soma dos valores fora da diagonal principal é o erro de comissão ou inclusão naquela classe, que é calculado ao dividi-lo pelo total na linha. O erro de omissão (E_O) está relacionado com a 'exatidão do produtor' ($1-E_O$), porque o produtor do classificador tem interesse em conhecer como foi feito o acerto da classificação dos objetos. O erro de comissão (E_C) está relacionado à 'exatidão do usuário' ($1-E_C$), pois, é a confiabilidade que o usuário tem daqueles resultados da classificação.

RESULTADOS E DISCUSSÃO

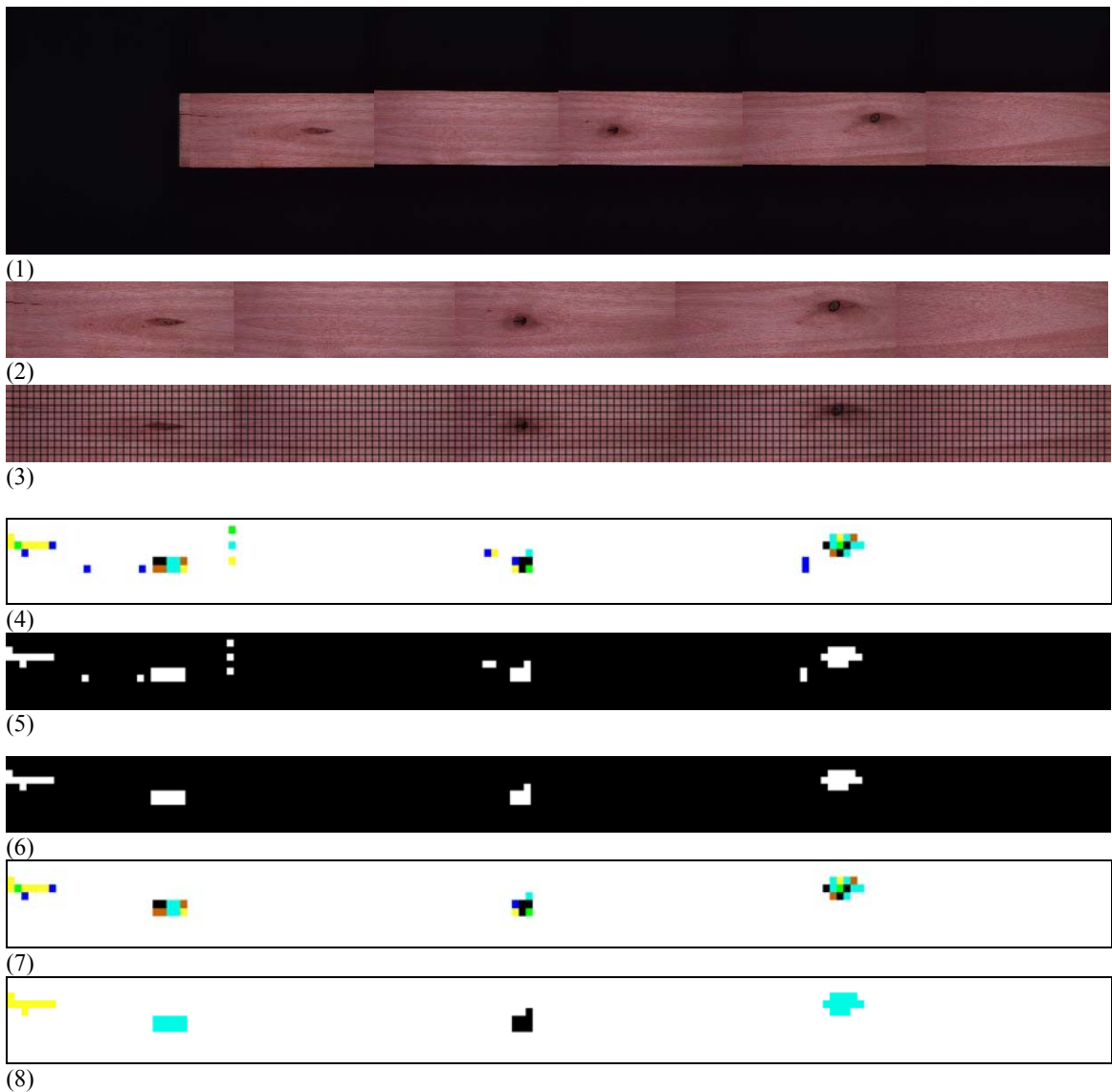
A Figura 4 exemplifica todas as etapas de classificação de uma face de madeira de eucalipto. Inicialmente, retira-se o fundo da imagem. Com isso, reduziu-se a imagem às dimensões da peça (etapa 2), possibilitando, assim, que um menor número de dados sejam processados nas etapas subsequentes.

Na etapa seguinte, a imagem foi ajustada a um número inteiro de blocos de 32x32 pixels, podendo, nesta etapa, perder no máximo 7,4 mm nas dimensões totais da peça,. Isto ocorrerá quando alguma das dimensões da face larga da peça não for múltipla de 32 pixels. Na etapa quatro, realizou-se a classificação dos blocos,

utilizando as funções discriminantes por meio das características de percentis do histograma. Apesar de nesta etapa ter havido boa distinção entre madeira limpa e defeitos, observou-se a ocorrência de ruídos. Esses ruídos foram blocos de madeiras limpas identificados como defeitos, que estavam desconectados e apresentavam pequenas sujeiras, manchas, variação da grã da madeira e variação da intensidade luminosa entre uma imagem e outra. Observou-se ainda, nesta etapa, uma confusão na classificação entre defeitos.

Nas etapas cinco e seis, os objetivos consistiram em agrupar os blocos conectados, utilizando o algoritmo componentes conectados, e eliminar os objetos que continham menos de três blocos. Nestas etapas, conseguiu-se eliminar aqueles blocos de madeira limpa classificados como defeitos e que, na verdade, eram ruídos, pois, os blocos que eram defeitos e foram classificados como defeitos estavam agrupados em mais de três blocos. Desta maneira, melhorou-se o erro de comissão (inclusão) da madeira limpa, evitando classificar uma parte limpa de defeitos como defeituosa.

Nas etapas sete e oito, após selecionados, os objetos maiores foram classificados, novamente, pela predominância do defeito contido em seu interior, no qual minimizou-se o problema da confusão entre defeitos. No exemplo da Figura 4, da esquerda para direita, observa-se que os defeitos racha, nó sadio e nó cariado foram classificados corretamente. Entretanto, o nó solto encontrado na extremidade direita da peça apresentava coloração clara, em seu interior, motivo provável de ter sido confundido com outras classes de defeitos, nos quais predominou, neste objeto, o nó sadio.



(8)

Legenda									
Defeitos	1,6,9,10	2	3	4	5	7,8	11	12	ml

1 = bolsa de goma; 2 = esmoado; 3 = fendilhado; 4 = furos de inseto; 5 = medula; 6 = nó cariado; 7 = nó sadio; 8 = nó sadio corrompido; 9 = nó solto encaixado; 10 = nó solto oco; 11 = podridão; 12 = racha e ml = madeira limpa.

Figura 4 - Imagem peça de madeira e fundo (1); imagem peça (2); imagem subdividida (3); imagem classificada (4); imagem binária (5); imagem binária de blocos selecionados (6); imagem de classes de defeitos nos objetos selecionados (7); imagem de classes de defeitos predominantes nos objetos (8).

No Quadro 1, são apresentadas as medidas dos defeitos apresentados na Figura 4. Neste exemplo, o algoritmo subestimou o resultado, classificando esta face como quarta classe, sendo ela, na realidade, da quinta classe. O erro na classificação da peça ocorreu devido à confusão na identificação de um nó solto como nó sadio. Como resultado, a medida do total da classe dos nós soltos foi subestimada, enquanto as medidas de nós sadios foram superestimada. Desta maneira, como o valor total do grupo de defeitos 1,6, 9 e 10 presentes na peça foram de 18,03mm/m, medido pelo programa computacional, o programa classificou-a como quarta classe. Na classificação visual, na verdade, os defeitos tinham, na verdade 30,73 mm/m, sendo este valor superior ao limite da quarta classe (25 mm/m), classificando-a como de quinta classe.

Quadro 1 – Resultado das medidas dos defeitos e da classificação da face larga da madeira do exemplo apresentado na Figura 4.

Medidas	L.* m	C.** m	1,6,9,10 mm/m	L. 2 cm/m	C. 2 Cm/m	3 cm/m	4 0 ou 1	5 0 ou 1	7 e 8 mm/m	11 %	12 mm/m	Classe face tábua
Manual	0,10	1,30	30,76	0,00	0,00	0,00	0,0	0,0	38,46	0,0	53,85	5
Imagem	0,10	1,28	18,03	0,00	0,00	0,00	0,0	0,0	66,11	0,0	42,07	4

* = largura da peça; ** = comprimento da peça; 1 = bolsa de goma; 2 = esmoado; 3 = fendilhado; 4 = furos de inseto; 5 = medula; 6 = nó cariado; 7 = nó sadio; 8 = nó sadio corrompido; 9 = nó solto encaixado; 10 = nó solto oco; 11 = podridão; e 12 = racha.

O Quadro 2 apresenta os resultados da matriz de confusão entre a classificação visual e pelo programa desenvolvido das faces largas das peças. Observa-se que as faces classificadas como primeira classe foram todas classificadas corretamente, porém, nesta classe, foi incluída uma face classificada visualmente como classe cinco, por apresentar furo de inseto que não fora identificado pelo algoritmo. Embora este defeito tenha sido identificado na etapa quatro do algoritmo, na etapa seis ele foi considerado como ruído, por estar em objetos menores que três blocos. O erro global na classificação por face foi de 35% e as maiores confusões ocorreram entre as classes quatro e cinco, sendo menos grave do que no exemplo comentado anteriormente, pois, estão em classes de qualidade próximas. Esses erros foram atribuídos à dificuldade em

identificar alguns tipos de defeitos como furo de inseto e na confusão entre medula e nó sadio.

Na classificação das faces das peças de madeira, o maior erro observado foi incluir três faces da quarta classe como quinta. Este erro ocorreu, principalmente, na quantificação dos defeitos bolsas de goma, nós cariados e soltos. De acordo com as medidas realizadas visualmente, os limites foram bem definidos. No entanto, na classificação por imagens, as bordas desses defeitos foram identificadas como nós sadios, pois, apresentam similaridade com estes nós. Na etapa do algoritmo, em que a classificação do objeto baseou-se na predominância dos defeitos, superestimaram-se as medidas desses defeitos. Com isto, a peça foi incluída em uma classe, imediatamente, inferior.

Quadro 2 - Matriz de confusão da classificação visual e pelo programa desenvolvido das faces das peças de madeira serrada

Classes	Dados de referência					Total	Erro de comissão (%)
	1	2	3	4	5		
1	3				1	4	25
2						0	0
3					1	1	100
4				1	2	3	67
5				3	9	12	25
Total	3	0	0	4	13	20	Exatidão global 65%
Erro de omissão (%)	0	0	0	75	31		

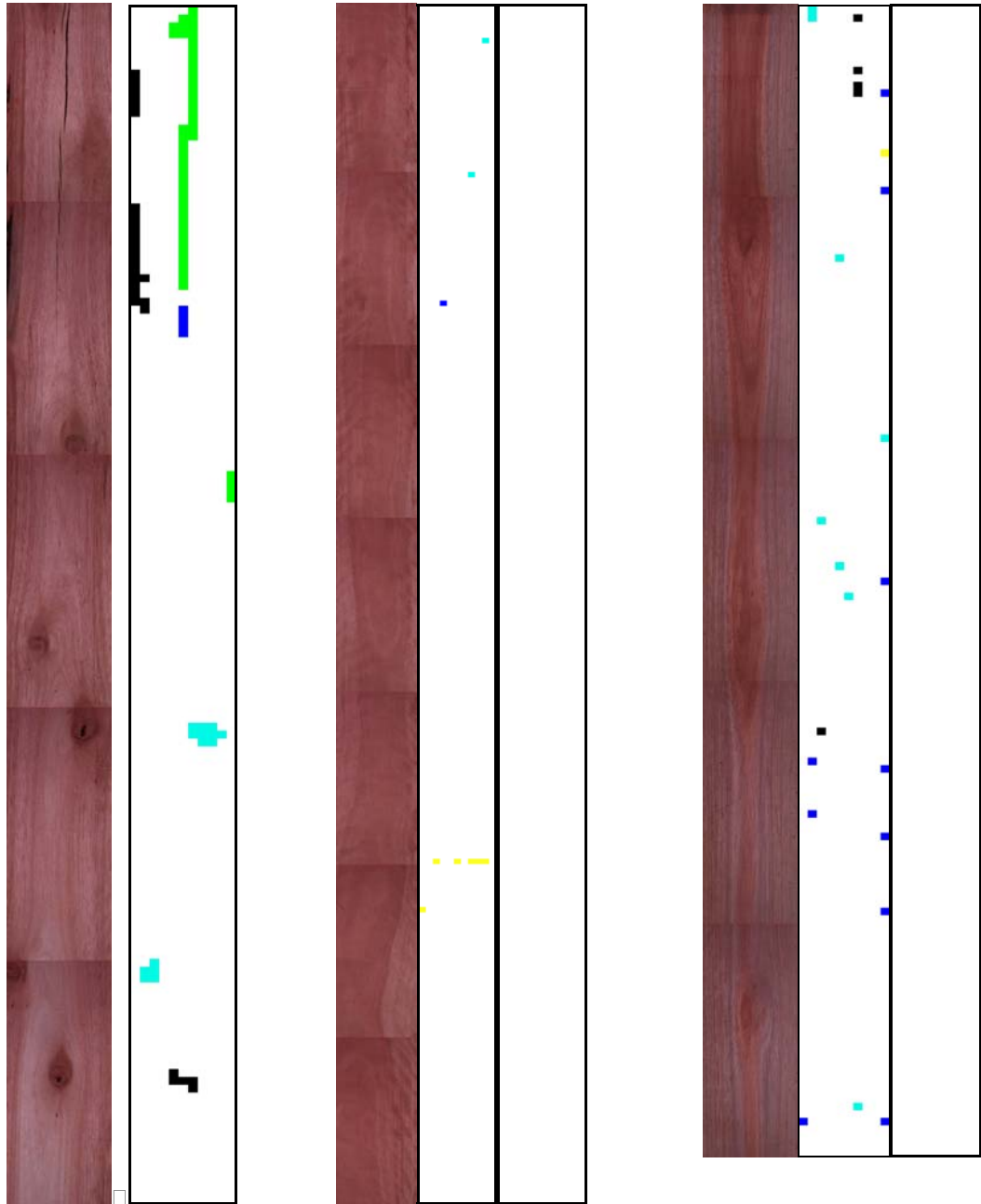
O Quadro 3 apresenta a classificação da peça, baseada em sua pior face, conforme propõe a norma (ABNT, 2002). O erro na classificação final reduziu para 10%. Esta redução foi, provavelmente, devido à tendência de os mesmos defeitos estarem presentes em ambas as faces, aumentando a probabilidade de acerto do algoritmo. Por exemplo, nós e rachas estarão, normalmente, presentes nas duas faces, pois, atravessam a face estreita da peça, assim como os outros defeitos são característicos da posição à qual a peça de madeira foi extraída na tora, ou por injurias

causadas na mesma, que afetam, geralmente, as duas faces. Desta maneira, quando erroneamente classificava-se uma face para uma classe superior devido à não identificação de um defeito, o algoritmo teria outra chance de identificar o mesmo defeito na outra face. Conseqüentemente, acertava a classe desta segunda face. Como a peça era classificada, baseando-se em sua pior face, acertava-se a classe da peça.

Quadro 3 - Matriz de confusão da classificação visual e pelo programa desenvolvido para cada peça de madeira serrada

Classes	Dados de referência					Total	Erro de comissão (%)
	1	2	3	4	5		
1	1					1	0
2						0	0
3						0	0
4						0	0
5				1	8	9	11
Total	1	0	0	1	8	10	Exatidão global 90%
Erro de omissão (%)	0	0	0	100	0		

A Figura 5 ilustra a classificação de três faces das peças de madeira. Na etapa do algoritmo de 'classificação de blocos', muitos blocos de madeira limpa foram classificados como defeitos, o que mostra a necessidade de implementação das etapas de eliminação de ruídos, em caso contrário, comprometeria a exatidão de classificação final. No caso particular apresentado em (C), entretanto, esta rotina não apresentou vantagem, pois, o defeito furos de inseto foi eliminado. Considerando-se que este defeito não é comum em madeiras serradas, num resultado final não haveria prejuízos na exatidão do sistema.



Manual = 5ª classe (A)
 Algoritmo = 5ª classe

Manual = 1ª classe (B)
 Algoritmo = 1ª classe

Manual = 5ª classe (C)
 Algoritmo = 1ª classe

Legenda									
Defeitos	1,6,9,10	2	3	4	5	7,8	11	12	ml

1 = bolsa de goma; 2 = esmoado; 3 = fendilhado; 4 = furos de inseto; 5 = medula; 6 = nó cariado; 7 = nó sadio; 8 = nó sadio corrompido; 9 = nó solto encaixado; 10 = nó solto oco; 11 = podridão; e 12 = racha.

Figura 5- Imagem original, classificação por blocos e classificação final.

No Quadro 4, é apresentada a matriz de confusão na identificação dos blocos pelo algoritmo e visualmente. Observa-se que a madeira limpa obteve erros de omissão e comissão de 0,1 e 0,2 %, respectivamente, isto justifica a boa classificação de peças de primeira classe, realizada pelo algoritmo. A maior magnitude do erro de comissão, encontrada na madeira limpa, foi devida à inclusão de nó sadio nesta classe, que teria efeito somente na avaliação da primeira classe de qualidade, pois, nas demais classes, o nó sadio é permitido.

Os defeitos medula e furo de inseto tiveram erros elevados de omissão, o que vem a ser uma falha no algoritmo, pois, na classificação, leva-se em conta a presença ou não dos mesmos. Portanto, foram os defeitos que mais afetaram a exatidão do algoritmo.

O defeito esmoado também apresentou alto erro de omissão, confundindo-o com bolsa goma. Na distinção entre esmoado e bolsa goma, a implementação de alguma característica, quanto à localização ou formato dos mesmos, poderia melhorar o desempenho do algoritmo.

Quadro 4 - Matriz de confusão da classificação de tipos de defeitos e madeira limpa na imagem classificada e imagem original, considerando o número de blocos nos objetos como a unidade de medida

	Referência na imagem original								total	Erro Comissão (%)
	bolsa goma no cariado e solto	esmoado	fendi- lhado	Furo de Inseto	medu- la	nó sadio	racha	madeira limpa		
bolsa goma, nó cariado e solto	188	62	0	0	21	11	0	4	286	34.3
esmoado	0	0	0	0	0	0	0	0	0	0.0
fendilhado	15	0	40	0	0	0	0	0	55	27.3
furo de inseto	17	0	0	0	0	0	10	11	38	100.0
medula	0	0	0	0	6	4	0	5	15	60.0
Nó sadio	46	0	0	0	47	139	0	5	237	41.4
racha	18	0	0	0	0	0	16	6	40	60.0
madeira limpa	5	0	0	30	0	66	0	44188	44289	0.2
Total	289	62	40	30	74	220	26	44219	total	Exatidão
E. Omissão(%)	34.9	100.0	0.0	100.0	91.9	36.8	38.5	0.1	44960	99.1(%)

A norma, adotada como referência, pode ser considerada complexa para ser empregada, com rigor, em uma indústria de serraria com produção em série. O termo

'complexa' é empregado no sentido que um classificador visual, ou mesmo um sistema de visão automatizado, teria dificuldades em identificar e medir, com exatidão, tantos tipos de defeitos, sendo que alguns apresentam dimensões milimétricas, como é o caso do furo de inseto e fendilhado.

Para facilitar e viabilizar as regras de classificação da norma ABNT, os tipos de defeitos semelhantes deveriam ser agrupados e as medidas dos mesmos deveriam ser simplificadas como, por exemplo, em números de defeitos em vez de dimensões.

Defeitos como medula e furo de inseto poderiam ser eliminados, precocemente, na etapa de desdobramento da tora, pois, o primeiro é característico de centro da tora, enquanto o segundo, quando se manifesta, é mais comum nas partes de alburno.

Apenas com a identificação dos defeitos mais freqüentes na etapa de seleção final, a eficiência do sistema de visão artificial aumenta não só quanto ao tempo de processamento, mas também quanto à exatidão dos defeitos mais comuns. Atualmente, a simplificação na descrição das classes de qualidade vem sendo adotada em algumas indústrias de serraria, pois, não mencionam tantos tipos de defeito nem medidas milimétricas dos defeitos mais comuns (ARACRUZ, 2000).

CONCLUSÕES

Um algoritmo para um sistema de visão artificial foi desenvolvido e avaliado para classificação de madeiras serradas de eucalipto pela aparência, com base na norma de classificação de madeira serrada de eucalipto (ABNT, 2002).

O algoritmo desenvolvido apresentou exatidão global de 65% na classificação de cada face da peça, independentemente. Entretanto, a exatidão foi de 90% quanto à classificação da peça, em que ambas as faces foram avaliadas simultaneamente.

A ocorrência de erros na classificação final da madeira serrada não foi devida à identificação de madeira limpa, mas, principalmente à confusão entre alguns tipos de defeitos.

O programa precisa de melhorias, principalmente na identificação dos defeitos furo de inseto, medula e podridão.

O algoritmo desenvolvido foi considerado satisfatório, para aplicação em sistemas automatizados de classificação de madeira serrada de eucalipto.

REFERÊNCIAS BIBLIOGRÁFICAS

- ARACRUZ PRODUTOS DE MADEIRA S.A. Descrição do produto. Nova Viçosa, Ba, 2000. 9p.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS – **ABNT**. Madeira serrada de eucalipto – Requisitos, NBR 14806. Rio de Janeiro, 2002. 11 p.
- CONGALTON, R.G. A review of assessing the accuracy of classifications of remotely sensed data. *Remote sensing of environment* n. 37 p. 35-46, 1991.
- HANEY, L.; PRECETTI, C.; GIBSON, H.; KRUTZ, G.; CASSENS, D. Color matching of wood with a real-time machine vision system. **ASAE Paper No. 943579**, ASAE Meeting Presentation, St. Joseph, Michigan, 1994, 15 p.
- HUBER, H. A.; McMILLIN, C. W.; McKINNEY, J. P. Lumber defect detection abilities of furniture rough mill employees. **Forest Products Journal**, 35(11/12), p. 79-82, 1985.
- KAUPPINEN, H. **Development of a color machine vision method for wood surface inspection**. Oulu, Finlandia: Department of Electrical Engineering and Infotech Oulu, - University of Oulu, Dissertation, 138p, 1999.(<http://herkules.oulu.fi/issn03553213/>)
- KLINE, D. E.; WIDOYOKO, A.; WIEDENBECK, J. K.; ARAMAN, P.A. Performance of color camera machine vision in automated furniture rough mill systems. **Forest Products Journal**, 48(3), p. 38-45, 1998.
- KLINE, D. E.; HOU, Y. J.; CONNERS, R. W.; SCHOLMOLDT, D. L.; ARAMAN, P.A. Lumber scanning system for surface defect detection, **ASAE Paper No. 923582**. St. Loseph, Mich.: ASAE, 1992.
- KLINE, D. E.; ARAMAN, P. A.; SURAK, C. Evaluation of an automated hardwood lumber grading system. Proceedings, ScanTech, **The Ninth International Conference on Scanning Technology and Process Optimization for the Wood Industry**. 141-151, 2001.
- LU, Q. **A real-time system for color sorting edge-glued panel parts**. Blacksburg, Virginia: Faculty of the Virginia Polytechnic Institute and State University, Dissertation, 119 p, 1997.
- MATLAB User's Guide, The Math Works Inc., 2000

- PHAM, D.T.; ALCOCK, R.J. Automated grading and defect detection: a review.
Forest Products Journal, 48(4), p34-42, 1998.
- SZYMANI, R.; MCDONALD, K. A.; Defect detection in lumber: state of the art,
Forest Products Journal, 31(11), p. 34-44, 1981.

CONCLUSÕES GERAIS

O objetivo do presente trabalho foi desenvolver um algoritmo, baseado em processamento de imagens digitais para classificação de madeira serrada de eucalipto, para aplicação em um sistema de visão artificial.

Inicialmente, 36 características de percentis foram retiradas de blocos de imagens com defeitos e madeira limpa, sendo 12 de cada banda vermelha, verde e azul. Em seguida, procedeu-se à seleção e descarte das características de percentis, restando 21 e 18 características para os tamanhos de blocos de 64x64 pixels e 32x32 pixels, respectivamente. Estas tinham percentis de todas as três bandas, indicando a necessidade de câmeras coloridas para discriminação de defeitos.

Entre os classificadores estatísticos, testados na identificação de blocos de defeitos e madeira limpa, as melhores exatidões globais alcançadas foram 81% e 76% utilizando-se funções discriminantes lineares com escores de variáveis canônicas para tamanho de bloco de 64x64 e 32x32 pixels, respectivamente.

Dentre os classificadores baseados em redes neurais artificiais e testados para a identificação de blocos de defeitos e madeira limpa, as maiores exatidões globais alcançadas foram 83% e 77%. Os métodos de treinamento testados não apresentaram diferença significativa do coeficiente kappa, porém os métodos que disponibilizavam maior número de observações, na fase de treinamento, tenderam a apresentar melhor acerto nas classes de defeitos com menor número de observações.

Os classificadores estatísticos, ou redes neurais artificiais foram considerados satisfatórios na identificação dos defeitos de madeira serrada de eucalipto e podem fazer parte de um sistema de visão artificial para classificação de peças de madeira serrada de eucalipto.

Para identificar blocos da imagem de defeitos e madeira limpa, o classificador estatístico foi escolhido na implementação do algoritmo, que classifica a face da peça de madeira. Isto foi devido à facilidade em alterar as probabilidades *a priori* nas funções discriminantes, já que o valor da proporção de madeira limpa, em relação aos defeitos na função discriminante na fase de desenvolvimento, apresentou valores diferentes dos encontrados na face da peça. A probabilidade *a priori* de madeira limpa

foi alterada de 38% (proporção de madeira limpa na amostra de desenvolvimento do classificador) para 90% (estimativa da proporção de madeira limpa numa face larga de madeira serrada) e o tamanho do bloco da imagem escolhido foi de 32x32 pixels, pois, com blocos de 64x64 pixels poderia ocorrer dificuldades em mensurar alguns tipos de defeitos com a precisão exigida pela norma adotada.

Na avaliação da classificação do algoritmo, baseada na norma ABNT - NBR 14806 / 2002 (Madeira serrada de eucalipto – Requisitos), em 20 faces de 10 peças de madeira serrada, foi obtida uma exatidão global de 65% na avaliação de classificação das faces, independentemente. Entretanto, na classificação da peça, a exatidão foi de 90% quando ambas as faces foram avaliadas, simultaneamente.

Os erros ocorridos na classificação final das peças de madeira não foram devidos à identificação de madeira limpa, mas, principalmente à confusão entre alguns tipos de defeitos. O programa precisa de melhorias, principalmente na identificação dos defeitos furo de inseto, medula e podridão.

A norma de classificação adotada como referência neste trabalho foi considerada complexa para ser empregada, com rigor, em uma indústria de serraria com produção em série.

APÊNDICES

Apêndice A

Código-fonte dos algoritmos implementados no MATLAB e SAS utilizados no capítulo 1

Teste T^2 de Hotteling, implementado no MATLAB

Autor: Joseph Kalil Khoury Junior

```
%matriz de dados n observações (linha) e p características (colunas); [492x36]
def=dlmread('def_32x32.txt','t'); % 305 observações de defeitos
ml=dlmread('ml_32x32.txt','t'); % 186 observações de madeira limpa

%classes de defeitos e madeira limpa
bg=def(1:25,:); %bolsa goma
esm=def(26:34,:); % esmoado
fen=def(35:50,:); %fendilhado
fins=def(51:84,:); %furo de inseto
med=def(85:138,:); %medula
nc=def(139:184,:); %nó cariado
ns=def(185:218,:); %nó sadio
nsc=def(219:247,:); %nó sadio corrompido
nst_en=def(248:275,:); nó solto encaixado
nst_oco=def(276:284,:); % nó solto oco
pod=def(285:289,:); % podridão
rc=def(290:305,:); %racha
ml; maderalimpa

%matriz de dados 492x36 = tudo
tudo=[bg;esm;fen;fins;med;nc;ns;nsc;nst_en;nst_oco;pod;rc;ml];

% repetições de cada classe
sml=size(ml,1);
sbg=size(bg,1);
sesm=size(esm,1);
sfen=size(fen,1);
sfins=size(fins,1);
smed=size(med,1);
snc=size(nc,1);
sns=size(ns,1);
snsc=size(nsc,1);
snst_en=size(nst_en,1);
snst_oco=size(nst_oco,1);
spod=size(pod,1);
src=size(rc,1);
size_tudo=size(tudo,1);

%Numero de classes
Ntrat=13;

% vetor de repetições de cada classe[rT1 rT2 rT3 ...rTk]
repet=[sbg sesm sfen sfins smed snc sns snsc snst_en snst_oco spod src sml]

disp('intervalo de tratamentos na matriz tudo')
interv=[1 25 26 34 35 50 51 84 85 138 139 184 185 218 219 247 248 275 276 284 285 289 290 305 306 492]
p=size(tudo,2);
```

```

% soma de quadrado e produto totais
for i=1:p
    for j=1:p
        A(i,j)=tudo(:,i)'*tudo(:,j)-(sum(tudo(:,i))*sum(tudo(:,j)))/size(tudo,1);
    end
end

% soma de quadrado e produto de tratamento (classe)
for i=1:Ntrat
    T(i,:)=sum(tudo(interv(2*i-1):interv(2*i,:),:));
end

for i=1:p
    for j=1:p
        H(i,j)=((T(:,i)'./repet)*T(:,j))-(sum(T(:,i))*sum(T(:,j)))/size(tudo,1);
    end
end

E=A-H;

% graus de liberdade tratamento (classe)
q=Ntrat-1;
gltot=size(tudo,1)-1;
ne=gltot-q;

% Teste de Wilks Lambda
LAMBDA=det(E)/det(H+E)

% LAMBDA tab(alfa,p,q,ne);
% rejeita-se Ho(m1=m2=m3) se lambda calculado < lambda tabelado

% TRANSFORMAÇÃO DE WILKS PARA F
a=ne-(1/2*(p-q+1));
c=1/2*(p*q-2);
if p^2+q^2-5 > 0;
    b=sqrt((p^2*q^2-4)/(p^2+q^2-5));
else
    b=1;
end
FW1=((1/LAMBDA)^(1/b))-1;
FW2=(a*b-c)/(p*q);
FW=FW1*FW2 % F transformado de lambda de Wilks

% Ftabelado(alfa,p*q,a*b-c)
p_value=1-fcdf(FW,p*q,a*b-c)

% Matriz de medias
for i=1:Ntrat
    mt(i,:)=T(i,:)/repet(i);
end

% teste T2 de Hotelling
% Matriz do teste de Hotelling para HO=mi=mj

for i=1:Ntrat
    for j=1:Ntrat
        TH(i,j)=(repet(i)*repet(j))/(repet(i)+repet(j))*[mt(i,:)-mt(j,:)]'*inv(E/ne)*[mt(i,:)-mt(j,:)];
    end
end

```

```

%constante que se relaciona T2 Hotelling com a distribuição F
k_TF=(ne-p+1)/(ne*p);

%FT é o valor de T2 de hotteling transformado para F, se FT > F(p,n-p+1) rejeita-se Ho
for i=1:Ntrat
for j=1:Ntrat
    FT(i,j)=k_TF*TH(i,j);
end
end

%Matriz de p value do teste de medias de tratamentos
for i=1:Ntrat
for j=1:Ntrat
p_val(i,j)=1-fcdf(FT(i,j),p,ne-p+1);
end
end
sc=E/ne; % matriz de covariância comum
dlmwrite('p_val_13cl.txt',p_val,'t');
dlmwrite('F_o_13cl.txt',FT,'t');

```

Descarte de variáveis (características), implementado no SAS após agrupamento em 9 classes de defeitos e ml

```

proc stepdisc data=bloco_32_36_9 bcov pcov method=stepwise slstay=0.15 slentry=0.15;
class trat;
var x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13 x14 x15 x16 x17 x18 x19 x20 x21 x22 x23 x24 x25 x26 x27
x28 x29 x30 x31 x32 x33 x34 x35 x36;
run;
quit;
arquivo bloco_32_36_9.wk1 é exportado para o SAS
data = é uma matriz de 492 observações x 36 variáveis, [492x36]

```

```

trat  x1  x2  ...  x36
1    valor
1
...
1
...
9
9

```

Análise de componentes principais, implementado no MATLAB

%tudo é a matriz de dados com 492 observações e 18 características selecionadas pelo método passo a passo [492x18]

```
tudo=dlmread('def_ml_18var_32x32_9cl.txt','t');
```

```

%classes de defeitos e ml
bg_16910=tudo(1:108,:);
esm=tudo(109:117,:);
fen=tudo(118:133,:);
fins=tudo(134:167,:);
med=tudo(168:221,:);
ns_78=tudo(222:284,:);
pod=tudo(285:289,:);
rc=tudo(290:305,:);
ml=tudo(306:492,:);

```

```

% matriz de médias de tratamento (classes)
m_bg_16910=mean(tudo(1:108,:));
m_esm=mean(tudo(109:117,:));
m_fen=mean(tudo(118:133,:));
m_fins=mean(tudo(134:167,:));
m_med=mean(tudo(168:221,:));
m_ns_78=mean(tudo(222:284,:));
m_pod=mean(tudo(285:289,:));
m_rc=mean(tudo(290:305,:));
m_ml=mean(tudo(306:492,:));

% matriz das medias de 9 classes e 18 características [9 X 18 ]
b=[m_bg_16910;m_esm;m_fen;m_fins;m_med;m_ns_78;m_pod;m_rc;m_ml];

% matriz de correlação – correspondente da matriz de covariância com os dados padronizados pelo desvio padrão
corr=CORRCOEF(b);

% autovalores(v) e autovetores(d) da matriz de correlação
[vr,dr]=eigs(corr)

% seis primeiros componentes principais (pc)
pc1r=vr(:,1);
pc2r=vr(:,2);
pc3r=vr(:,3);
pc4r=vr(:,4);
pc5r=vr(:,5);
pc6r=vr(:,6);

% padronização das 18 variáveis pelo desvio padrão
for i=1:18
tudop(:,i)=(tudo(:,i))/ std(tudo(:,i));
end

% escores dos seis primeiros componentes principais após padronização das observações
Epcr1=tudop*pc1r;
Epcr2=tudop*pc2r;
Epcr3=tudop*pc3r;
Epcr4=tudop*pc4r;
Epcr5=tudop*pc5r;
Epcr6=tudop*pc6r;

% matriz com os escores das 492 observações e dos seis pc [492x6]
[Epcr1 Epcr1 Epcr1 Epcr1 Epcr1 Epcr1];

```

Análise de componentes principais, implementado no SAS

```

proc princomp data=bloco_32_m_18_9 out=ccp_cor_32;
var X2 X3 X4 X5 X12 X14 X16 X17 X20 X21 X22 X25 X26 X27 X28 X30 X32 X33;
run;

```

arquivo bloco_32_m_18_9.wk1 é exportado para o SAS
data - é uma matriz de 9 médias das classes e 18 variáveis, [9x18]
trat x2 x3 ... x33
1 valor
2
...
9

Variáveis canônicas, implementado no SAS

```
proc candisc data=bloco32_d_ml_18_9 out=can all;
  class trat;
  var X2 X3 X4 X5 X12 X14 X16 X17 X20 X21 X22 X25 X26 X27 X28 X30 X32 X33;
run;
```

arquivo bloco32_d_ml_18_9.wk1 é exportado para o SAS
data = é uma matriz de 492 observações e 18 variáveis, [492x18]

```
trat  x2  x3  ...  x33
1     valor
1
...
1
...
9
9
```

Classificação utilizando função discriminante linear com as variáveis originais, implementado no SAS

```
proc discrim data=bloco.d_ml_18_9 outstat=no_def_32 out=out outd=outd outcross=cross
  method=normal
  pool=yes
  crossvalidate;
  class trat;
  var X2 X3 X4 X5 X12 X14 X16 X17 X20 X21 X22 X25 X26 X27 X28 X30 X32 X33;
  priors proportional;
run;
```

```
quit;
data - é uma matriz de 492 observações e 18 variáveis, [492x18]
```

Observação: Para utilizar função discriminante quadrática trocar o termo “yes” por “no”

Classificação utilizando função discriminante linear com os escores dos seis primeiros componentes principais, implementado no SAS

```
proc discrim data=bloco32.E_cor_6_9 outstat=no_32_s_cor out=no_32_cor
  method=normal
  pool=yes
  crossvalidate;
  class trat;
  var pc1 pc2 pc3 pc4 pc5 pc6;
  priors proportional;
run;
```

```
quit;
```

data - é uma matriz de 492 observações e 6 componentes principais (valores dos escores dos 6 pc's), [492x6]

```
trat  pc1  pc2  ...  pc6
1     valor
1
...
1
...
9
9
```

Observação: Para reduzir o número de componentes principais, utilizar na linha “var” somente os pc's que desejar.

Classificação utilizando função discriminante linear com os escores das seis primeiras variáveis canônicas, implementado no SAS

```
proc discrim data=bloco_32.E_can outstat=no_32_s_can out=no_32_can
    method=normal
    pool=yes
    crossvalidate;
    class trat;
    var can1 can2 can3 can4 can5 can6 can7 can8;
    priors proportional;
run;
quit;
```

data é uma matriz de 492 observações e 8 variáveis canônicas (valores dos escores das 8 can's), [492x8]

```
trat  can1  can2  ...  can8
1     valor
1
...
1
...
9
9
```

Observação: Para reduzir o número de variáveis canônicas, utilizar na linha “var” somente as cans que desejar.

Apêndice B

Código-fonte dos algoritmos implementados no MATLAB utilizados no capítulo 2

Algoritmo para testar número de neurônios nas camadas escondidas n1 e n2 da RNA única

Treinamento por parada mais cedo e, teste por validação cruzada

A amostra foi dividida em 4 partes: Sendo sempre duas utilizadas para treinamento, uma para parada e a uma para teste

Autor: Eduardo Eiji Maeda

```
%matrizes utilizadas no treinamento
load matrizes
whos
p1  18x238    34272 double array % 50 % da amostra para treinamento (A1+A2)
p2  18x243    34992 double array
p3  18x254    36576 double array
p4  18x249    35856 double array
A1  18x119    17136 double array
A2  18x119    17136 double array
A3  18x124    17856 double array % 25 % da amostra para 'parada mais cedo'
A4  18x130    18720 double array % 25 % da amostra para teste
r1  9x119     8568 double array
r2  9x119     8568 double array
r3  9x124     8928 double array % resposta desejada da amostra A3
r4  9x130     9360 double array % resposta desejada da amostra A4
t1  9x238     17136 double array % resposta desejada da amostra p1
t2  9x243     17496 double array
t3  9x254     18288 double array
t4  9x249     17928 double array

zz=14 % número de neurônios na segunda camada, n2, primeira arquitetura testado

for k=0:7 % for para variar o número de neurônios na primeira camada, n1, fixando o valor de n2

    n1=(zz+(2*k));
    n2=zz;

    min_max=minmax(p1);
    ext4=0;

% treinamento e teste com a primeira parte da amostra (p1)

    for z=1:5
        net1=newff(min_max,[n1 n2 9],{'tansig','tansig','logsig'},'trainoss'); % Cria a RNA com os parâmetros
                                                desejados

        net1.trainparam.epochs=1000;
        net1.trainparam.show=25;

        VV.P=A3;
        VV.T=r3;
        [net1,tr1]=train(net1,p1,t1,[],[],VV); % treina a RNA

        trn4=sim(net1,A4); % testa a RNA

    end

end

H=[];
```

```

for i=1:130
    [a cL]=max(trn4(:,i));      % Identifica em qual classe de defeito(de 1 a 9) cada amostra foi
                                colocada
    for j=1:9
        if trn4(j,i)<a;
            trn4(j,i)=0;
        else
            trn4(j,i)=1;
        end
    end
    Ha=[cL];
    H=[H Ha];                  % cria vetor com as classes de cada amostra para ser montada a matriz
                                de confusão
end

%classes=[1 29, 30 32, 33 36, 37 45, 46 59, 60 76, 77 78, 79 82, 83 130]
confus1=
sum(H(1:29)==1) sum(H(1:29)==2) sum(H(1:29)==3) sum(H(1:29)==4) sum(H(1:29)==5) sum(H(1:29)==6)
sum(H(1:29)==7) sum(H(1:29)==8) sum(H(1:29)==9);
sum(H(30:32)==1) sum(H(30:32)==2) sum(H(30:32)==3) sum(H(30:32)==4) sum(H(30:32)==5)
sum(H(30:32)==6) sum(H(30:32)==7) sum(H(30:32)==8) sum(H(30:32)==9);
sum(H(33:36)==1) sum(H(33:36)==2) sum(H(33:36)==3) sum(H(33:36)==4) sum(H(33:36)==5)
sum(H(33:36)==6) sum(H(33:36)==7) sum(H(33:36)==8) sum(H(33:36)==9);
sum(H(37:45)==1) sum(H(37:45)==2) sum(H(37:45)==3) sum(H(37:45)==4) sum(H(37:45)==5)
sum(H(37:45)==6) sum(H(37:45)==7) sum(H(37:45)==8) sum(H(37:45)==9);
sum(H(46:59)==1) sum(H(46:59)==2) sum(H(46:59)==3) sum(H(46:59)==4) sum(H(46:59)==5)
sum(H(46:59)==6) sum(H(46:59)==7) sum(H(46:59)==8) sum(H(46:59)==9);
sum(H(60:76)==1) sum(H(60:76)==2) sum(H(60:76)==3) sum(H(60:76)==4) sum(H(60:76)==5)
sum(H(60:76)==6) sum(H(60:76)==7) sum(H(60:76)==8) sum(H(60:76)==9);
sum(H(77:78)==1) sum(H(77:78)==2) sum(H(77:78)==3) sum(H(77:78)==4) sum(H(77:78)==5)
sum(H(77:78)==6) sum(H(77:78)==7) sum(H(77:78)==8) sum(H(77:78)==9);
sum(H(79:82)==1) sum(H(79:82)==2) sum(H(79:82)==3) sum(H(79:82)==4) sum(H(79:82)==5)
sum(H(79:82)==6) sum(H(79:82)==7) sum(H(79:82)==8) sum(H(79:82)==9);
sum(H(83:130)==1) sum(H(83:130)==2) sum(H(83:130)==3) sum(H(83:130)==4) sum(H(83:130)==5)
sum(H(83:130)==6) sum(H(83:130)==7) sum(H(83:130)==8) sum(H(83:130)==9)]

```

Algoritmo da arvore de decisão para distinguir com uma RNA para cada defeitos (esmoado, fendilhado, furo de inseto, racha e podridão. E uma Rna para distinguir os restantes
Treinamento por parada mais cedo e, teste por validação cruzada
A amostra foi dividida em 4 partes: Sendo duas utilizadas para treinamento, uma para parada e a uma para teste

%treinamento da rede somente para distinguir esmoado

```

load mat_esm.mat ;
p1  18x238    34272 double array
p2  18x243    34992 double array
p3  18x254    36576 double array
p4  18x249    35856 double array
A1  18x119    17136 double array
A2  18x119    17136 double array
A3  18x124    17856 double array
A4  18x130    18720 double array
r1  2x119     1904 double array
r2  2x119     1904 double array
r3  2x124     1984 double array
r4  2x130     2080 double array
t1  2x238     3808 double array
t2  2x243     3888 double array

```

```
t3    2x254    4064 double array
t4    2x249    3984 double array
```

```
n1=16;
```

```
n2=16;
```

```
Max_fail=800; (Nº de iterações após a detecção do aumento do MSE)
```

```
min_max=minmax(p1);
```

```
ext_esm=0;
```

```
for z=1:5
```

```
    net1=newff(min_max,[n1 n2 2],{'tansig','tansig','logsig'},'trainoss'); %Cria RNA com os parâmetros
                                                desejados
```

```
    net1.trainparam.epochs=20000;
```

```
    net1.trainparam.show=500;
```

```
    net1.trainParam.max_fail= Max_fail;
```

```
    net1.trainParam.min_grad=1.0000e-010;
```

```
    VV.P=pp3;
```

```
    VV.T=r3;
```

```
    [net1,tr1]=train(net1,p1,t1,[],[],VV); %treina a RNA com a matriz p1
```

```
    trn=sim(net1,A4); % Testa a RNA com a matriz A4
```

```
    H1=[]; % Verifica se as amostra foram identificadas como esmoado ou outro defeito
```

```
    for i=1:130
```

```
        [a cL]=max(trn(:,i));
```

```
        Ha=[cL];
```

```
        H1=[H1 Ha];
```

```
    end
```

```
    trn_res1=H1(1:29);
```

```
    trn_res2=H1(33:130);
```

```
    trn_rest=[trn_res1 trn_res2];
```

```
    trn_esm=H1(30:32);
```

```
    acerto_esm=trn_esm==1;
```

```
    acerto_rest=trn_rest==2;
```

```
    ext_esm1=sum(acerto_esm)/size(acerto_esm,2);
```

```
    ext_rest1=sum(acerto_rest)/size(acerto_rest,2);
```

```
    exta=(sum(acerto_esm)+sum(acerto_rest))/size(trn,2); % Verifica a exatidão do teste
```

```
end
```

```
end
```

O mesmo processo foi feito para Fendilhado, Podridão, furo de inseto e Racha, e outra RNA foi criada para os 4 defeitos restantes.

Teste da árvore de decisão

```
load matrizes % carrega as Matrizes com as amostras
```

```
load redes_esm % carrega as RNAs criadas para cada defeito
```

```
load redes_fen
```

```
load redes_finse
```

```
load redes_rc
```

```
load redes_pod
```

```
load redes_4c
```

```

% treinamento e teste com a parte 1 da amostra

R=[];
for i=1:119
    x=A1(:,i:i);

    trn=sim(netb_esm,x); %Teste com a RNA para esmoado

    [a cL]=max(trn(:,1));
    if cL==1
        Ra=2;
        R=[R Ra];
    else
        trn=sim(netb_pod,x); %Teste com a RNA para podridão
        [a cL]=max(trn(:,1));
        if cL==1
            Ra=7;
            R=[R Ra];

        else
            trn=sim(netb_rc,x); %Teste com a RNA para racha
            [a cL]=max(trn(:,1));
            if cL==1
                Ra=8;
                R=[R Ra];

            else
                trn=sim(netb_fen,x); %Teste com a RNA para fendilhado
                [a cL]=max(trn(:,1));
                if cL==1
                    Ra=3;
                    R=[R Ra];

                else
                    trn=sim(netb_finse,x); %Teste com a RNA para furo de inseto
                    [a cL]=max(trn(:,1));
                    if cL==1
                        Ra=4;
                        R=[R Ra];

                    else
                        trn=sim(netb_4c,x); %Teste com a RNA para os 4 defeitos restantes
                        [a cL]=max(trn(:,1));
                        if cL==1
                            Ra=1;
                            R=[R Ra];

                        elseif cL==2
                            Ra=5;
                            R=[R Ra];

                        elseif cL==3
                            Ra=6;
                            R=[R Ra];

                        else cL==4;
                            Ra=9;
                            R=[R Ra];

                    end

                end
            end
        end
    end
end

```

```

end
end
end

end
end
end

```

```

%classes=[1 26, 27 28, 29 32,33 40, 41 53, 54 68, 69 69, 701 73, 74 119]
confus1=[sum(R(1:26)==1) sum(R(1:26)==2) sum(R(1:26)==3) sum(R(1:26)==4) sum(R(1:26)==5)
sum(R(1:26)==6) sum(R(1:26)==7) sum(R(1:26)==8) sum(R(1:26)==9)
sum(R(27:28)==1) sum(R(27:28)==2) sum(R(27:28)==3) sum(R(27:28)==4) sum(R(27:28)==5)
sum(R(27:28)==6) sum(R(27:28)==7) sum(R(27:28)==8) sum(R(27:28)==9)
sum(R(29:32)==1) sum(R(29:32)==2) sum(R(29:32)==3) sum(R(29:32)==4) sum(R(29:32)==5)
sum(R(29:32)==6) sum(R(29:32)==7) sum(R(29:32)==8) sum(R(29:32)==9)
sum(R(33:40)==1) sum(R(33:40)==2) sum(R(33:40)==3) sum(R(33:40)==4) sum(R(33:40)==5)
sum(R(33:40)==6) sum(R(33:40)==7) sum(R(33:40)==8) sum(R(33:40)==9)
sum(R(41:53)==1) sum(R(41:53)==2) sum(R(41:53)==3) sum(R(41:53)==4) sum(R(41:53)==5)
sum(R(41:53)==6) sum(R(41:53)==7) sum(R(41:53)==8) sum(R(41:53)==9)
sum(R(54:68)==1) sum(R(54:68)==2) sum(R(54:68)==3) sum(R(54:68)==4) sum(R(54:68)==5)
sum(R(54:68)==6) sum(R(54:68)==7) sum(R(54:68)==8) sum(R(54:68)==9)
sum(R(69:69)==1) sum(R(69:69)==2) sum(R(69:69)==3) sum(R(69:69)==4) sum(R(69:69)==5)
sum(R(69:69)==6) sum(R(69:69)==7) sum(R(69:69)==8) sum(R(69:69)==9)
sum(R(70:73)==1) sum(R(70:73)==2) sum(R(70:73)==3) sum(R(70:73)==4) sum(R(70:73)==5)
sum(R(70:73)==6) sum(R(70:73)==7) sum(R(70:73)==8) sum(R(70:73)==9)
sum(R(74:119)==1) sum(R(74:119)==2) sum(R(74:119)==3) sum(R(74:119)==4) sum(R(74:119)==5)
sum(R(74:119)==6) sum(R(74:119)==7) sum(R(74:119)==8) sum(R(74:119)==9)]

```

Algoritmo pelo método deixando um de fora para RNA unica

Treinamento por limite de iterações

A amostra foi dividida em n partes para teste

Autor: Joseph Kalil Khoury Junior

```

clear
dados=dlmread('def_ml_18var_32x32_9cl.txt','t');
dados =dados';

%matriz resposta para cada classe
t1=(repmat([1 0 0 0 0 0 0 0],108,1))';
t2=(repmat([0 1 0 0 0 0 0 0],9,1))';
t3=(repmat([0 0 1 0 0 0 0 0],16,1))';
t4=(repmat([0 0 0 1 0 0 0 0],34,1))';
t5=(repmat([0 0 0 0 1 0 0 0],54,1))';
t6=(repmat([0 0 0 0 0 1 0 0],63,1))';
t7=(repmat([0 0 0 0 0 0 1 0],5,1))';
t8=(repmat([0 0 0 0 0 0 0 1],16,1))';
t9=(repmat([0 0 0 0 0 0 0 0 1],187,1))';
%
número de neurônios nas camadas escondidas
n1=14;
n2=14;

% observação 1 para teste
x=dados(:,2:492);%dados para treinamento fora a observação 1
pp=dados(:,1:1);
min_max=minmax(x);
tti=[t1(:,2:108) t2 t3 t4 t5 t6 t7 t8 t9];%resposta desejada fora a observação 1

```

```

perfor_M=1;
for k=1:3%chutes
    net=newff(min_max,[n1 n2 9],{'tansig','tansig','logsig'},'trainoss');%cria a rede
    net.trainparam.epochs=1000;
    net.trainparam.show=500;
    net.trainParam.min_grad=1.0000e-010;
    [net,tr]=train(net,x,tti,[],[]);%treina a rede
    perfor=min(tr.perf);

    if perfor < perfor_M
        net_M=net;
        perfor_M=perfor;
    end
end
net=net_M;
tr=perfor_M;
trn=sim(net,pp);%simula a rede com uma observação que ficou de fora
[va cL]=max(trn);
R=[cL];
net_arq=strcat('net',num2str(1));
tr_arq=strcat('tr',num2str(1));
save(net_arq,'net')
save(tr_arq,'tr')
i=1
R
clear a cL i min_max net net_M net_arq perfor perfor_M tr_arq trn x k
save R R

% observação 2 ate a 491
load R
for i=2:491
    a=dados(:,1:i-1);
    b=dados(:,i+1:492);
    x=[a b];%dados para treinamento fora a observação i
    pp=dados(:,i);%teste com a observação de fora
    min_max=minmax(x);
    tt=[t1 t2 t3 t4 t5 t6 t7 t8 t9];%resposta desejada para tudo
    aa=tt(:,1:i-1);
    bb=tt(:,i+1:492);
    tt=[aa bb];%resposta desejada fora a observação de fora
    perfor_M=1;
    for k=1:3 %chutes
        net=newff(min_max,[n1 n2 9],{'tansig','tansig','logsig'},'trainoss');%cria a rede
        net.trainparam.epochs=1000;
        net.trainparam.show=500;
        net.trainParam.min_grad=1.0000e-010;
        [net,tr]=train(net,x,tt);%treina a rede
        perfor=min(tr.perf);

        if perfor < perfor_M
            net_M=net;
            perfor_M=perfor;
        end
    end
end
net=net_M;
tr=perfor_M;
trn=sim(net,pp);%simula a rede com uma observação que ficou de fora
[va cL]=max(trn);
Ra=[cL];
R=[R Ra];

```

```

net_arq=strcat('net',num2str(i));
tr_arq=strcat('tr',num2str(i));
i
R
save(net_arq,'net')
save(tr_arq,'tr')
save R R
end

clear a cL i min_max net net_M net_arq perfor perfor_M tr_arq trn x k

% interv=[1:108 109:117 118:133 134:167 168:221 222:284 285:289 290:305 306:492]

% observação 492 de fora
x=dados(:,1:491);%dados fora observação 492
pp=dados(:,492:492);%observação 492 para teste
min_max=minmax(x);
ttf=[t1 t2 t3 t4 t5 t6 t7 t8 t9(:,1:186)];%resposta desejada fora a observação 492
perfor_M=1;
for k=1:3 %chutes
    net=newff(min_max,[n1 n2 9],{'tansig','tansig','logsig'},'trainoss');%cria a rede
    net.trainparam.epochs=1000;
    net.trainparam.show=500;
    net.trainParam.min_grad=1.0000e-010;
    [net,tr]=train(net,x,ttf);%treina a rede
    perfor=min(tr.perf);

    if perfor < perfor_M
        net_M=net;
        perfor_M=perfor;
    end

end
net=net_M;
tr=perfor_M;
trn=sim(net,pp);%simula a rede com uma observação que ficou de fora
[va cL]=max(trn);
Ra=[cL];
R=[R Ra];
net_arq=strcat('net',num2str(492));
tr_arq=strcat('tr',num2str(492));
i
R
save(net_arq,'net')
save(tr_arq,'tr')

save R R

clear a cL i min_max net net_M net_arq perfor perfor_M tr_arq trn x k

%interv=[1 81, 82 88, 89 100, 101 126, 127 167, 168 215, 216 219, 220 231, 232 372]

% repet=[sbg_16910 sesm sfen sfins smed sns_78 spod src sml]
% '108 9 16 34 54 63 5 16 187'

```



```
% interv=[1:108 109:117 118:133 134:167 168:221 222:284 285:289 290:305 306:492]
```

```
confus=
```

```
[sum(R(1:108)==1) sum(R(1:108)==2) sum(R(1:108)==3) sum(R(1:108)==4) sum(R(1:108)==5)
sum(R(1:108)==6) sum(R(1:108)==7) sum(R(1:108)==8) sum(R(1:108)==9)
sum(R(109:117)==1) sum(R(109:117)==2) sum(R(109:117)==3) sum(R(109:117)==4) sum(R(109:117)==5)
sum(R(109:117)==6) sum(R(109:117)==7) sum(R(109:117)==8) sum(R(109:117)==9)
sum(R(118:133)==1) sum(R(118:133)==2) sum(R(118:133)==3) sum(R(118:133)==4) sum(R(118:133)==5)
sum(R(118:133)==6) sum(R(118:133)==7) sum(R(118:133)==8) sum(R(118:133)==9)
sum(R(134:167)==1) sum(R(134:167)==2) sum(R(134:167)==3) sum(R(134:167)==4) sum(R(134:167)==5)
sum(R(134:167)==6) sum(R(134:167)==7) sum(R(134:167)==8) sum(R(134:167)==9)
sum(R(168:221)==1) sum(R(168:221)==2) sum(R(168:221)==3) sum(R(168:221)==4) sum(R(168:221)==5)
sum(R(168:221)==6) sum(R(168:221)==7) sum(R(168:221)==8) sum(R(168:221)==9)
sum(R(222:284)==1) sum(R(222:284)==2) sum(R(222:284)==3) sum(R(222:284)==4) sum(R(222:284)==5)
sum(R(222:284)==6) sum(R(222:284)==7) sum(R(222:284)==8) sum(R(222:284)==9)
sum(R(285:289)==1) sum(R(285:289)==2) sum(R(285:289)==3) sum(R(285:289)==4) sum(R(285:289)==5)
sum(R(285:289)==6) sum(R(285:289)==7) sum(R(285:289)==8) sum(R(285:289)==9)
sum(R(290:305)==1) sum(R(290:305)==2) sum(R(290:305)==3) sum(R(290:305)==4) sum(R(290:305)==5)
sum(R(290:305)==6) sum(R(290:305)==7) sum(R(290:305)==8) sum(R(290:305)==9)
sum(R(306:492)==1) sum(R(306:492)==2) sum(R(306:492)==3) sum(R(306:492)==4) sum(R(306:492)==5)
sum(R(306:492)==6) sum(R(306:492)==7) sum(R(306:492)==8) sum(R(306:492)==9)]
save confus_32_14x14_1000ep confus
```

Apêndice C

Código-fonte dos algoritmos implementados no MATLAB utilizados no capítulo 3

Algoritmo de classificação da face da tábua

```
Autor Joseph Kalil Khoury Junior
numero=input('numero da tabua=', 's');
lado=input('lado da tabua=', 's');
num=input('numero de imagens=>');
tp=strcat('t', numero, '_', lado, '_');
%local onde sera lida as imagens originais
local=strcat('C:\Meus documentos\Kalil\cap_3_image\RGB\');
%local onde sera salva as imagens classificadas
local1=strcat('C:\Meus documentos\Kalil\cap_3_image\RGB\classificada32\');
%extensao do arquivo de imagem
ext='.tif'; tp2=' - RGB';
imcor_tot=[]; %imagem imendada
for i=1:num
    path=strcat(local, 't', numero, '_', lado, '_', num2str(i), ' - RGB.tif');
    imcor=imread(path); %ler a imagem
    imcor=imcor(6:1034, :, :);
    imcor_tot=[imcor; imcor_tot];
end
im=imcor_tot(:, :, 1); %retira a banda 1 para fazer executar o corta_mad_t
%subrotina(A) – corta_mad_t(im);
[L1, L2, C1, C2]=corta_mad_t(im); %imagem cortada da banda 1
clear im
imcor_tot_s=imcor_tot(L1:L2, C1:C2, :); %image color sem fundo color
imwrite(imcor_tot, strcat(local1, tp, 'tot', ext));
imwrite(imcor_tot_s, strcat(local1, tp, 'cort', ext));
clear imcor_tot
bloco=32;
TL=(L2-L1+1); TC=(C2-C1+1); %tamanho da image cortada
%subrotina (B) - ajustabloco2(TL, TC, bloco)
[L, C, lin_ini, lin_fim, col_ini, col_fim]=ajustabloco2(TL, TC, bloco); %ajusta a image para o tamanho de bloco
imcor_tot_q=imcor_tot_s(lin_ini:lin_fim, col_ini:col_fim, :);
imcor_tot_aj=imcor_tot_s(lin_ini:lin_fim, col_ini:col_fim, :);
clear imcor_tot_s
%criar quadros na image classificada – apresentação de resultados
for i=1:L/bloco-1
    imcor_tot_q(i*bloco:(i*bloco)+3, :, :)=0;
    for j=1:C/bloco-1
        imcor_tot_q(:, j*bloco:(j*bloco)+3, :)=0;
    end
end
imwrite(imcor_tot_q, strcat(local1, tp, 'cort_q', ext));
clear imcor_tot_q imcor
% % mostrar eixo na image
% iptsetpref('ImshowAxesVisible', 'on')
% ha=gca;
% set(ha, 'xtick', [16:32:C-16]);
% set(ha, 'xticklabel', [1:C/32]);
% set(ha, 'ytick', [16:32:L-16]);
% set(ha, 'yticklabel', [1:L/32]);
```

```

classificada=zeros(L,C);
b=fix(L/bloco);
a=fix(C/bloco);
coef_can =[
0.82892155000000 -0.40446898200000 -13.88385684000000 -11.04990303000000 1.63379235600000
0.05010704600000 -1.70940636800000 5.59106273400000 15.84350454000000 -18.44530141000000
2.75680120700000 8.25972908800000 -7.98003887900000 -16.95667660000000 14.73088801000000
-2.04758111900000 -1.59729265600000 6.60179398400000 6.71364355300000 -2.04868984600000
-0.83105231300000 -4.50076798000000 9.67362658700000 5.08018572400000 4.14198694300000
1.37964163900000 -7.27761966900000 -8.82253939100000 -6.92221213700000 -3.90082124500000
-7.73943332900000 -1.06000619100000 9.66937374600000 -7.49930694600000 -0.04968416500000
1.22311232300000 7.71455291200000 -2.13796515800000 6.62191962600000 -6.55748125900000
-1.07063603600000 1.28200910900000 7.38781868400000 6.56950236500000 7.22537190800000
6.65221833600000 -4.54981341900000 1.36460788100000 -3.35963841800000 -1.22380956100000
-0.50087124000000 3.18613185400000 -6.14007731800000 3.48352247900000 3.95528030300000
-1.90864861100000 -6.94986346900000 -0.57229442400000 2.90624282100000 -1.76735750900000
-0.85988583100000 12.00693796000000 3.12353028100000 -3.69933335100000 10.53140743000000
3.42078920500000 4.36561148800000 12.29005932000000 10.86085902000000 -7.46205639100000
4.10321973400000 -4.98649204600000 -8.44759735700000 2.39055877400000 7.90365612200000
0.14846239800000 -9.17578481900000 -2.26283468400000 -8.94744754600000 -1.92743439700000
2.59270848800000 6.55453945000000 -20.86602190000000 4.41065733400000 0.55964038000000
-7.57084841400000 -0.54612336700000 14.08620539000000 -4.16470956500000 -6.47808961600000 ];

```

```

% escores canônicos centralizados na media
m_esc=[1.646511789 -0.502915304 7.294427764 7.38892328 5.983294321 5.71668506 6.243528938];

```

```

matriz de média das classes de defeitos [9x5];
mx=[-2.030454027931400000000 -1.064103942348320000000 0.134501432089803000000 -
0.209922281847656000000 -0.069817505275474600000 1.966210440402150000000 -
-1.502376824594500000000 -1.750875968425490000000 1.966210440402150000000 -
0.540398152820370000000 -0.835811879578715000000 0.698975984490209000000 -
-1.284005450386540000000 3.498974956611330000000 0.698975984490209000000 -
1.702247051762390000000 1.167474868883070000000 -1.519948541138670000000 -
0.069105621610480900000 1.002538337427360000000 -0.246063250620191000000 -
0.664184986939203000000 -0.892626277275254000000 -0.246063250620191000000 -
-0.959476064885202000000 0.421185378914755000000 -0.246063250620191000000 -
0.723206624399230000000 0.681048202295362000000 -0.579216129459858000000 -
-0.634739451536261000000 0.449761564275151000000 -0.579216129459858000000 -
0.525605852453857000000 -0.003451366425864970000 0.255890733950848000000 -
0.747214342684029000000 0.517708338805278000000 0.255890733950848000000 -
0.114659114416365000000 -0.594410491415466000000 1.758018889117750000000 -
-0.576016071965581000000 3.211552739146750000000 1.758018889117750000000 -
0.988378234109760000000 -1.271437457472060000000 0.153170384018226000000 -
1.862488774855490000000 -0.344605175507945000000 0.153170384018226000000 -
0.053762836346230500000 0.072129293843313000000 ];

```

```

%probabilidade a priori(9x1)
pr=[0.21951219512195100000
0.01829268292682930000
0.03252032520325200000
0.06910569105691060000
0.10975609756097600000
0.12804878048780500000
0.01016260162601630000
0.03252032520325200000
0.38008130081300800000];

```

```

% bg(108) esmoado(9) fendilhado(16) f.inseto(34) medula(54) no sadio(63) podridao(5) racha(16) ml
%probabilidade a priori(9x1) modificada
pr=[(138*.1/305) (9*.1/305) (16*.1/305) (24*.1/305) (34*.1/305) (63*.1/305) (5*.1/305) (16*.1/305) (.9)];

```

```

cov=matriz identidade de 5x5, escores são descorrelacionados, inversa cov = cov
icov=cov;
%rotina para classificação dos blocos
for u=1:b
    for y=1:a
        subim=imcor_tot_aj((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco,:);
        A=subim;
% sub-rotina (C) - perc_im_cor_select_32(A)
        f=perc_im_cor_select_32(A);
        esc_f=[f*coef_can(:,1)-m_esc(1) f*coef_can(:,2)-m_esc(2) f*coef_can(:,3)-m_esc(3) f*coef_can(:,4)-
m_esc(4) f*coef_can(:,5)-m_esc(5)];
        x=esc_f;%caracteristica retirada de um bloco(18x1), multiplicada pelas cinco primeiras canonicas
resultando num vetor de caracteristicas de (5x1)
%funções discriminantes
        d1= (x-mx(1,:))*icov*(x-mx(1,:))'- 2*log(pr(1));
        d2= (x-mx(2,:))*icov*(x-mx(2,:))'- 2*log(pr(2));
        d3= (x-mx(3,:))*icov*(x-mx(3,:))'- 2*log(pr(3));
        d4= (x-mx(4,:))*icov*(x-mx(4,:))'- 2*log(pr(4));
        d5= (x-mx(5,:))*icov*(x-mx(5,:))'- 2*log(pr(5));
        d6= (x-mx(6,:))*icov*(x-mx(6,:))'- 2*log(pr(6));
        d7= (x-mx(7,:))*icov*(x-mx(7,:))'- 2*log(pr(7));
        d8= (x-mx(8,:))*icov*(x-mx(8,:))'- 2*log(pr(8));
        d9= (x-mx(9,:))*icov*(x-mx(9,:))'- 2*log(pr(9));
%classificação pela maior probabilidade a posteriori
[dmin,classe]=min([d1;d2;d3;d4;d5;d6;d7;d8;d9]);
%bolsa goma
        if classe==1
            classificada((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco)=1;
        end
%esmnoado
        if classe==2
            classificada((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco)=2;
        end
%fendilhado
        if classe==3
            classificada((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco)=3;
        end
%furo de inseto
        if classe==4
            classificada((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco)=4;
        end
%medula
        if classe==5
            classificada((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco)=5;
        end
%no sadio
        if classe==6
            classificada((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco)=6;
        end
%podridao
        if classe==7
            classificada((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco)=7;
        end
%racha
        if classe==8
            classificada((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco)=8;
        end
%madeira limpa
        if classe==9
            classificada((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco)=9;
        end
    end
end

```

```

end
end
end
% subplot(1,2,1),imshow(imcor_tot_q);
% subplot(1,2,2),imshow(classificada_show);
save(strcat(local1,tp,'classificada'),'classificada');
classificada_show=zeros(b*32,a*32,3);
class_bin=zeros(b*32,a*32);
for u=1:b
    for y=1:a
        subim=classificada((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco);
        A=subim;
        classe=A(1,1);
%bolsa goma
        if classe==1
            classificada_show((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco,1)=0;
            classificada_show((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco,2)=0;
            classificada_show((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco,3)=0;
            class_bin((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco)=1;
        end
%esmnoado
        if classe==2
            classificada_show((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco,1)=255;
            classificada_show((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco,2)=0;
            classificada_show((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco,3)=0;
            class_bin((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco)=1;
        end
%fendilhado
        if classe==3
            classificada_show((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco,1)=0;
            classificada_show((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco,2)=255;
            classificada_show((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco,3)=0;
            class_bin((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco)=1;
        end
%furo de inseto
        if classe==4
            classificada_show((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco,1)=0;
            classificada_show((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco,2)=0;
            classificada_show((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco,3)=255;
            class_bin((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco)=1;
        end
% medula
        if classe==5
            classificada_show((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco,1)=200;
            classificada_show((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco,2)=100;
            classificada_show((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco,3)=0;
            class_bin((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco)=1;
        end
%no sadio
        if classe==6
            classificada_show((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco,1)=0;
            classificada_show((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco,2)=250;
            classificada_show((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco,3)=230;
            class_bin((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco)=1;
        end
%podridao
        if classe==7
            classificada_show((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco,1)=125;
            classificada_show((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco,2)=125;
            classificada_show((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco,3)=125;
        end
    end
end
end

```

```

        class_bin((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco)=1;
    end
    %racha
    if classe==8
        classificada_show((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco,1)=252;
        classificada_show((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco,2)=254;
        classificada_show((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco,3)=40;
        class_bin((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco)=1;
    end
    %madira limpa
    if classe==9
        classificada_show((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco,1)=255;
        classificada_show((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco,2)=255;
        classificada_show((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco,3)=255;
        class_b((u-1)*bloco+1:(u-1)*bloco+bloco,(y-1)*bloco+1:(y-1)*bloco+bloco)=0;
    end
end
end
classificada_show=uint8(classificada_show);
imwrite(classificada_show,strcat(local1,tp,'classificada_show',ext));
imwrite(class_bin,strcat(local1,tp,'class_bin',ext));
class_bin=logical(class_bin);
L=bwlabel(class_bin,4);
STATSA =[IMFEATURE(L,'area',4)];
STATSP =[IMFEATURE(L,'PixelList',4)];
[Ctb Ltb]=size(L);
Ctb=Ctb*1.04;
Ltb=Ltb*1.13;
Nobj=size(STATSA,1);
resp=[ ];
for i=1:Nobj
    local=STATSP(i).PixelList;
    CL=[ ];
    for k=1:size(local,1)
        CL=[CL;classificada(local(k,2),local(k,1))];
    end
    %classificação pela predominância de defeitos no bloco
    d1=sum(CL==1);d2=sum(CL==2);d3=sum(CL==3);d4=sum(CL==4);d5=sum(CL==5);d6=sum(CL==6);
    d7=sum(CL==7);d8=sum(CL==8);d9=sum(CL==9);
    [Maior CLf]=max([d1+1 d2 d3 d4-1 d5-1 d6 d7 d8 d9]);
    maxobj=max(STATSP(i).PixelList);
    minobj=min(STATSP(i).PixelList);
    % area pixel n_linhas_obj(comprim)*cm n_colunas ob(larg)*cm area objeto comprimento*mm
    resp=[resp;i STATSA(i).Area 100*(maxobj(2)-minobj(2)+1)/Ctb 100*(maxobj(1)-minobj(1)+1)/Ltb
    100*STATSA(i).Area/(Ltb*Ctb) 1000*(maxobj(2)-minobj(2)+1)/Ctb CLf];
end
Loc=[ ];
for i=1:size(resp,1)
    if resp(i,2) <=3072
        Loc=[Loc;i];
    end
end
respf=resp;
%elimina objetos pequenos
respf(Loc,:)=[];
save(strcat(local1,tp,'resp'),'resp');
save(strcat(local1,tp,'respf'),'respf');
%armazena imagem depois de retirado ruidos
classificada_show_f=classificada_show;
for i=1:size(Loc,1)

```

```

ii=Loc(i);
local2=STATSP(ii).PixelList;
for k=1:size(local2,1)
    classificada_show_f(local2(k,2),local2(k,1),:)=255;
end
end
imwrite(classificada_show_f, strcat(local1,tp,'classificada_show_f,ext));
%Algoritmo para somar as medidas das dimensoes dos defeitos de uma mesma tabua
LC=[ ];
% Local onde tem mesmos defeitos na matriz respf
for i=1:size(respf,1)
    if respf(i,7)==1
        LC=[LC i];
    end
end
% soma as linhas de mesmo defeito
sd1=zeros(1,size(respf,2));
for i=1:size(LC,2)
    sd1=[sd1+respf(LC(i,:),:)];
end
LC=[ ];
% Local onde tem mesmos defeitos na matriz respf
for i=1:size(respf,1)
    if respf(i,7)==2
        LC=[LC i];
    end
end
% soma as linhas de mesmo defeito
sd2=zeros(1,size(respf,2));
for i=1:size(LC,2)
    sd2=[sd2+respf(LC(i,:),:)];
end
LC=[ ];
% Local onde tem mesmos defeitos na matriz respf
for i=1:size(respf,1)
    if respf(i,7)==3
        LC=[LC i];
    end
end
% soma as linhas de mesmo defeito
sd3=zeros(1,size(respf,2));
for i=1:size(LC,2)
    sd3=[sd3+respf(LC(i,:),:)];
end
LC=[ ];
% Local onde tem mesmos defeitos na matriz respf
for i=1:size(respf,1)
    if respf(i,7)==4
        LC=[LC i];
    end
end
% soma as linhas de mesmo defeito
sd4=zeros(1,size(respf,2));
for i=1:size(LC,2)
    sd4=[sd4+respf(LC(i,:),:)];
end
LC=[ ];
% Local onde tem mesmos defeitos na matriz respf
for i=1:size(respf,1)
    if respf(i,7)==5

```

```

    LC=[LC i];
end
end
% soma as linhas de mesmo defeito
sd5=zeros(1,size(respf,2));
for i=1:size(LC,2)
sd5=[sd5+respf(LC(i,:))];
end
LC=[ ];
% Local onde tem mesmos defeitos na matriz respf
for i=1:size(respf,1)
if respf(i,7)==6
    LC=[LC i];
end
end
% soma as linhas de mesmo defeito
sd6=zeros(1,size(respf,2));
for i=1:size(LC,2)
sd6=[sd6+respf(LC(i,:))];
end
LC=[ ];
% Local onde tem mesmos defeitos na matriz respf
for i=1:size(respf,1)
if respf(i,7)==7
    LC=[LC i];
end
end
end
% soma as linhas de mesmo defeito
sd7=zeros(1,size(respf,2));
for i=1:size(LC,2)
sd7=[sd7+respf(LC(i,:))];
end
LC=[ ];
% Local onde tem mesmos defeitos na matriz respf
for i=1:size(respf,1)
if respf(i,7)==8
    LC=[LC i];
end
end
end
% soma as linhas de mesmo defeito
sd8=zeros(1,size(respf,2));
for i=1:size(LC,2)
sd8=[sd8+respf(LC(i,:))];
end
respff=[sd1;sd2;sd3;sd4;sd5;sd6;sd7;sd8];
save(strcat(local1,tp,'respff'),'respff');
Cbg=respff(1,6);
Lesm=respff(2,4);
Cesm=respff(2,3);
Cfen=respff(3,3);
fins=respff(4,5);
med=respff(5,5);
ns=respff(6,6);
Apod=respff(7,5);
Crc=respff(8,6);
resp2=[Cbg;Lesm;Cesm;Cfen;fins;med;ns;Apod;Crc]
%Classifica a tabua de acordo com a quantidade de defeitos presentes
%1 comp.mm/m de bolsa goma
bg=resp2(1);
if bg==0

```



```

    cL1=1;
elseif bg<=25
    cL1=4;
else
    cL1=5;
end
%2 Largura esmoado cm/m ou %
Lesm=resp2(2);
if Lesm<=10
    cL2=1;
elseif Lesm<=15
    cL2=3;
elseif Lesm<=20
    cL2=4;
elseif Lesm<=30
    cL2=5;
else
    cL2=6;
end
%3 comprim. esmoado cm/m ou %
Cesm=resp2(3);
if Cesm<=20
    cL3=1;
elseif Cesm<=50
    cL3=3;
elseif Cesm<=60
    cL3=4;
elseif Cesm<=70
    cL3=5;
else
    cL3=6;
end
% resp2=[Cbg;Lesm;Cesm;Cfen;fins;med;ns;Apod;Crc]
%4 comp. fendilhado cm/m
fen=resp2(4);
if fen==0
    cL4=1;
elseif fen<=10
    cL4=2;
elseif fen<=30
    cL4=3;
elseif fen<=50
    cL4=4;
else
    cL4=5;
end
%5 furo de inseto em area(%)
fins=resp2(5,1);
if fins==0
    cL5=1;
else
    cL5=5;
end
% 6 medula area(%)
med=resp2(6);
if med==0
    cL6=1;
else
    cL6=5;
end

```

```

%7 no sadio em comprmrnto em mm/m
ns=resp2(7);
if ns==0
    cL7=1;
elseif ns<=25
    cL7=2;
else
    cL7=3;
end
%8 podridao area(%)
pod=resp2(8);
if pod==0
    cL8=1;
elseif pod<=15
    cL8=4;
elseif pod<=20
    cL8=5;
else
    cL8=6;
end
%9 racha em comprmrnto (mm/m)
rc=resp2(9);
if rc<=25
    cL9=1;
elseif rc<=50
    cL9=2;
elseif rc<=100
    cL9=3;
elseif rc<=150
    cL9=4;
else
    cL9=5;
end
% cLi corresponde a classe da tabua que seria inserida de acordo com a medida do defeito i
cL=max([cL1 cL2 cL3 cL4 cL5 cL6 cL7 cL8 cL9])
save(strcat(local1,tp,'DEF'),'resp2');
save(strcat(local1,tp,'CLASSE'),'cL');
sub-rotina (A) – function para cortar fundo da imagem com madeira
corta_mad_t
function [L1,L2,C1,C2]=f(im);
[L C P]=size(im);
%cortar tabua de fundo
i=4;
tt=im(i,:,1);
aa=sum(tt>50);
L1=25;
while aa <200 & i<(L-10)%quando encontrar mais de 1000 pixels
    i=i+1; %com valor superior ao brilho 50, eh o inicio da tabua
    tt=im(i,:,1);
    aa=sum(tt>50);
    L1=i+25;
end
i=i+3000;
while aa >=200 & i<(L-10)
    i=i+1;
    tt=im(i,:,1);
    aa=sum(tt>50);
    L2=i-25;
end
j=2;

```

```

t=im(:,j,1);
a=sum(t>50);
while a<4000 & j<C
    j=j+1;
    t=im(:,j,1);
    a=sum(t>50);
    C1=j+25;
end
j=C1+300;
t=im(:,j,1);
a=sum(t>50);
while a>=4000 & j<C
    j=j+1;
    t=im(:,j,1);
    a=sum(t>50);
    C2=j-25;
end
end

```

sub-rotina (B) - function para ajustar imagem para multiplo de blocos ajustabloco2

```

%Autor: Darly Senna Júnior
function [L,C,lin_ini,lin_fim,col_ini,col_fim] = ajustabloco(lin,col,n)
lin_blocos=fix(lin/n)*n;
col_blocos=fix(col/n)*n;
lin_ini=fix((lin-lin_blocos)/2)+1;
lin_fim=lin_blocos+lin_ini-1;
col_ini=fix((col-col_blocos)/2)+1;
col_fim=col_blocos+col_ini-1;
L=(lin_fim-lin_ini)+1;
C=(col_fim-col_ini)+1;

```

Sub-rotina (B) - function para retirada das características de blocos perc_im_cor_select_32

```

function vp = f(im);
r=im(:,:,1);
r=double(r(:));
Pr= PRCTILE(r,[2,5,10,20,30,40,50,60,70,80,90,95]);P95_5r=Pr(12)-Pr(1);
f5r=Pr(2)/P95_5r;%x2
f10r=Pr(3)/P95_5r;%x3
f20r=Pr(4)/P95_5r;%x4
f30r=Pr(5)/P95_5r;%x5
f95r=Pr(12)/P95_5r;%x12
g=im(:,:,2);
g=double(g(:));
Pg= PRCTILE(g,[2,5,10,20,30,40,50,60,70,80,90,95]);P95_5g=Pg(12)-Pg(1);
f5g=Pg(2)/P95_5g;%14
f20g=Pg(4)/P95_5g;%16
f30g=Pg(5)/P95_5g;%17
f60g=Pg(8)/P95_5g;%20
f70g=Pg(9)/P95_5g;%21
f80g=Pg(10)/P95_5g;%22
b=im(:,:,3);
b=double(b(:));
Pb= PRCTILE(b,[2,5,10,20,30,40,50,60,70,80,90,95]);P95_5b=Pb(12)-Pb(1);
f2b=Pb(1)/P95_5b;%25
f5b=Pb(2)/P95_5b;%26
f10b=Pb(3)/P95_5b;%27
f20b=Pb(4)/P95_5b;%28
f40b=Pb(6)/P95_5b;%30
f60b=Pb(8)/P95_5b;%32
f70b=Pb(9)/P95_5b;%33
vp=[f5r f10r f20r f30r f95r f5g f20g f30g f60g f70g f80g f2b f5b f10b f20b f40b f60b f70b];

```