



UNIVERSIDADE
ESTADUAL DE LONDRINA

JOSÉ LUIS SEIXAS JUNIOR

**APRENDIZADO DE MÁQUINA APLICADO
À SEGMENTAÇÃO DE IMAGENS DE LESÃO DE PELE**

Londrina
2015

JOSÉ LUIS SEIXAS JUNIOR

**APRENDIZADO DE MÁQUINA APLICADO
À SEGMENTAÇÃO DE IMAGENS DE LESÃO DE PELE**

Dissertação apresentada ao Programa de Mestrado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Alan Salvany Felinto

Londrina
2015

**Catálogo elaborado pela Divisão de Processos Técnicos da Biblioteca Central da
Universidade Estadual de Londrina.**

Dados Internacionais de Catalogação-na-Publicação (CIP)

S462a José Luis Seixas Junior
Aprendizado de Máquina Aplicado à Segmentação de Imagens de Lesão de Pele/
José Luis Seixas Junior. – Londrina, 2015
89 p. : il.

Orientador: Prof. Dr. Alan Salvany Felinto
-Dissertação (Mestrado em Ciência da Computação) -Universidade Estadual de
Londrina. Centro de Ciências Exatas. Programa de Pós-Graduação em Ciência da
Computação -Universidade Estadual de Londrina, 2015.
Inclui bibliografia.

1. Redes neurais(Computação) -Teses. 2. Processamento de sinais -Técnicas
digitais -Teses. 3. Algoritmos de computador -Teses. 4. Pele -Doenças -Teses. 5.
Aprendizado do computador -Teses. I. Felinton, Alan Salvany Felinto. II.
Universidade Estadual de Londrina. Centro de Ciências Exatas. Programa de Pós-
Graduação em Ciência da Computação. III. Título

CDU 519.683

JOSÉ LUIS SEIXAS JUNIOR

**APRENDIZADO DE MÁQUINA APLICADO À SEGMENTAÇÃO DE
IMAGENS DE LESÃO DE PELE**

Dissertação apresentada ao Programa de Mestrado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Mestre em Ciência da Computação.

BANCA EXAMINADORA

Orientador: Prof. Dr. Alan Salvany Felinto
Universidade Estadual de Londrina – UEL

Prof. Dr. Sylvio Barbon Jr.
Universidade Estadual de Londrina – UEL

Prof. Dr. José Alexandre de França
Universidade Estadual de Londrina – UEL

Prof. Dr. Vitor Valério de S. Campos
Universidade Estadual de Londrina – UEL

Londrina, 23 de março de 2015

Este trabalho é dedicado à meus pais, família e amigos próximos, por toda a compreensão, confiança e apoio.

AGRADECIMENTOS

Agradeço ao meu orientador Dr. Alan Salvany Felinto por mais uma orientação, pelos projetos propostos, pela amizade e confiança.

À professora Dra. Jandira Guenka Palma pelo aprendizado na disciplina.

Ao professor Dr. Rodolfo Miranda de Barros pela disciplina que levou à publicação de um artigo, em anexo neste trabalho (Anexo B).

Ao professor Dr. Sylvio Barbon Jr., pelas disciplinas, artigos escritos em conjunto e projetos paralelos.

Aos alunos companheiros do programa de mestrado, Márcio de Abreu Moreira, Wilson Hissamu Shirado, Rodrigo Augusto Igawa por estarem sempre dispostos à ajudar.

Ao aluno de doutorado da USP, Me. Rafael Gomes Mantovani com ajudas em artigos e bibliografias utilizadas neste e em outros trabalhos.

Aos demais colegas que enfrentaram os desafios e as disciplinas comigo.

À CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pelo suporte financeiro.

*“Computação não é mais sobre computadores.
É sobre viver.”
-Nicholas Negroponte*

SEIXAS JR., J. L.. **Aprendizado de Máquina Aplicado à Segmentação de Imagens de Lesão de Pele**. 89 p. Dissertação de Mestrado (Mestrado em Ciência da Computação) – Universidade Estadual de Londrina, Londrina-PR, 2015.

RESUMO

O diagnóstico incorreto de doenças de pele pode resultar em complicações durante o processo de cura. As imagens da pele fornecem informações importantes para a equipe médica no armazenamento e troca de informações na tentativa de prevenir que esse diagnóstico incorreto ocorra. Para tal, é necessário um bom processo de segmentação. A segmentação dessas imagens já está sendo usada e tem sido uma ferramenta eficaz para o reconhecimento de doenças de pele. Este trabalho apresenta diferentes métodos para segmentação, como descoberta de sementes para crescimento de regiões, já que vários algoritmos de crescimento de regiões apresentam bons resultados de clusterização, porém são sensíveis a sementes, usando a energia do canal da cor em regiões, ao invés de usar o sistema de RGB separados para cada pixel. Outras abordagens foram construídas com diferentes paradigmas de aprendizagem de máquina para a segmentação de imagens médicas de úlceras de pele em membros inferiores. Para o aprendizado de máquina, foram usados os algoritmos de árvore de decisão, que trazem uma abordagem mais intuitiva e também, com redes neurais, que possuem desempenho adequado para problemas não lineares. Os resultados foram comparados com modelo ouro obtido com a ajuda de especialistas, os resultados se mostraram adequados para a análise das características das lesões e abriram caminhos que poderão ser seguidos em trabalhos futuros, visto que, mesmo que adequados, os resultados ainda podem ser melhorados.

Palavras-chave: Imagens Médicas. Processamento de Imagens. Energia da Cor. Árvore de Decisão. Rede Neural Artificial.

SEIXAS JR., J. L.. **Machine Learning Applied to Skin Lesion Images Segmentation**. 89 p. Master's Thesis (Master in Science in Computer Science) – State University of Londrina, Londrina-PR, 2015.

ABSTRACT

Misleading diagnosis of skin diseases can result in complications during the healing process. Skin images provide important information for the medical staff for information storage and exchange, to trying to prevent this misdiagnosis from happening. For such, a good segmentation process is needed. The segmentation of these images is already being used and has been an effective tool for skin diseases recognition. This paper presents various methods for targeting seeds for region growing algorithms, as several of region growing algorithms have good clustering results, but are sensitive to seed, the use of color channel energy from regions instead of to use of separate RGB system for each pixel. Other approaches are built with different paradigms of machine learning for segmentation of medical images of skin ulcers in the lower limbs. For machine learning, decision tree algorithms were used, which bring a more intuitive approach, and also with neural networks, which have great performance for nonlinear problems. The results were compared with gold standard obtained with the help of experts, the results were good and opened paths that can be followed for further work since, even though good results, they can still be improved.

Keywords: Medical Images. Image Processing. Color Energy. Decision Tree. Artificial Neural Network.

LISTA DE ILUSTRAÇÕES

Figura 1 – Árvore de decisão abstrata.	31
Figura 2 – Modelo de arquitetura de uma RBFN.	35
Figura 3 – Diferença de forma de clusterização da MLP e RBFN.	36
Figura 4 – Exemplo de amostragem.	36
Figura 5 – Representação da clusterização feita com a RBFN.	37
Figura 6 – Relação de distância do centroide.	37
Figura 7 – Variações da curva Gaussiana.	39
Figura 8 – Imagem obtida nas condições supracitadas.	42
Figura 9 – Imagem com região delimitada por médicos.	43
Figura 10 – Região com a maior diferença de energia encontrada pelo algoritmo.	46
Figura 11 – Porcentagem de reconhecimento pela árvore.	56
Figura 12 – Segmentação feita pelo algoritmo J48.	57
Figura 13 – Primeiro resultado obtido com a RBFN.	58
Figura 14 – Resultado da RBFN após a modificação do conjunto de dados.	59
Figura 15 – Porcentagem de reconhecimento pela RBFN.	59
Figura 16 – Segmentação feita com a rede neural RBFN.	60
Figura 17 – Imagens originais usadas nos exemplos de J48 e RBFN.	61
Figura 18 – Imagens segmentadas utilizando as marcações dos especialistas.	62

LISTA DE TABELAS

Tabela 1 – Regiões de lesão encontradas.	53
Tabela 2 – Tempos de execução por tamanho de região.	55
Tabela 3 – Porcentagem de reconhecimento de pixel pela árvore.	75
Tabela 4 – Porcentagem de reconhecimento pela rede neural.	76

LISTA DE ABREVIATURAS E SIGLAS

HSI	Matiz, Saturação e Intensidade (Hue, Saturation e Intensity)
RGB	Vermelho, Verde, Azul (Red, Green e Blue)
UEL	Universidade Estadual de Londrina
CISMEPAR	Consórcio Intermunicipal de Saúde do Médio Paranapanema
HU	Hospital Universitário
IDE	Ambiente de Desenvolvimento Integrado (Integrated Development Environment)
ML	Aprendizado de Máquina (Machine Learning)
ABCD	Assimetria, Borda, Cor e Diâmetro (Asymetry, Border, Color e Diameter)
CART	Árvore de Regressão e Classificação (Classification And Regression Tree)
RBFN	Rede de Função de Base Radial (Radial Basis Function Network)
MLP	Perceptron Multi Camadas (Multilayer Perceptron)
RNA	Rede Neural Artificial

SUMÁRIO

1	INTRODUÇÃO	23
2	FUNDAMENTAÇÃO TEÓRICA	27
2.1	Trabalhos Correlatos	27
2.2	Energia da Cor	28
2.3	Aprendizado de Máquina	30
2.3.1	Árvore De Decisão	30
2.3.1.1	Algoritmo J48	32
2.3.2	Rede Neural Artificial	33
2.3.2.1	Radial Basis Function Network	35
3	MÉTODO DE PESQUISA	41
3.1	Aquisição de informação	41
3.2	Critérios de avaliação	43
4	EXPERIMENTOS	45
4.1	Experimentos com Energia da Cor	45
4.1.1	Experimento para encontrar uma região de lesão	45
4.1.2	Experimento com tamanhos diferentes de regiões	46
4.1.3	Experimento para contar quantas regiões são encontradas	47
4.2	Experimentos com Aprendizado de Máquina	48
4.2.1	Experimento com Árvore de Decisão	48
4.2.2	Experimento com Rede Neural Artificial	49
5	RESULTADOS	53
5.1	Energia da Cor	53
5.2	Árvore De Decisão	55
5.3	Rede Neural Artificial	58
6	CONCLUSÃO	63
	REFERÊNCIAS	67
	APÊNDICES	73
	APÊNDICE A – TABELAS DE RESULTADOS	75

ANEXOS	77
ANEXO A – ARTIGO PUBLICADO PELO AUTOR	79
Trabalhos Publicados pelo Autor	89

1 INTRODUÇÃO

O diagnóstico incorreto de doenças de pele pode conduzir à complicações durante o processo de cura. Weenig et al. [1] citam o caso de pioderma gangrenoso que tem várias causas e pode ter diferentes tipos de tratamento que não seriam recomendados para pacientes com outros tipos de doenças. O reconhecimento de lesões por meio de características como tamanho, severidade, presença de infecção e vascularidade tem um importante impacto na determinação do tratamento mais apropriado para cada paciente [2]. Muitos dos descritores de lesão de pele apresentados por Frykberg [2] podem ser obtidos com a análise das imagens segmentadas das lesões de pele, porém é necessária uma boa segmentação para o trabalho com dados corretos e confiáveis.

O grande volume de informações que uma imagem médica pode oferecer requer o uso de técnicas sofisticadas de análise para que elas possam ajudar a equipe médica na tomada de uma decisão [3]. Análise esta que pode determinar a situação e evolução de uma lesão, mas que depende de suas características citadas por Frykberg [2].

Estas características podem ser obtidas por meio da análise das imagens, mas antes é necessário o processo de segmentação da imagem para evidenciar a área de lesão. Em Kekre et al. [4], vemos que a segmentação de imagens é um processo que demanda uma grande intervenção humana ou o sistema será bastante específico, ou seja, é um problema orientado à aplicação.

A segmentação é o primeiro passo para identificar o estado da lesão, também possibilita a automatização na obtenção da informação em larga escala e a padronização das análises finais que deixam de ser subjetivas, aumentando a precisão dos métodos comparativos.

A cor é uma característica de baixo nível que pode transmitir informações relevantes, como as descritas por Frykberg [2], de uma determinada imagem de acordo com Fu et al. [5], assim muitos algoritmos se voltam para cores na segmentação das áreas de interesse. Cheng et al. [6] descrevem vários algoritmos e diferentes propostas sobre a segmentação por cores e cita abordagens baseadas em regiões como um boa opção não sensível à ruídos. Nas lesões de pele, os algoritmos baseados em crescimento de regiões funcionam de maneira satisfatória por que a região de lesão é bastante homogênea.

Muitos problemas nas análises médicas dependem ou podem ser solucionados com o uso de imagens digitais. Estas imagens dependem de especialistas para descrever cada uma delas. Em alguns casos, o médico não pode estar com o paciente, nestes casos, imagens são opções para enviar ou guardar o histórico dos pacientes. Porém, a análise feita totalmente pelo médico ainda será necessária.

Um dos objetivos deste trabalho é criar um método para encontrar automaticamente uma região de certeza de lesão através do cálculo das energias de cada canal, região essa que seria a entrada para o algoritmo de crescimento de regiões, na tentativa de auxiliar a precisão do diagnóstico feito por médicos, já que o volume de imagens e informações estão em constante crescimento.

Com este crescimento do volume de imagens e informação, muitos são os esforços e abordagens na tentativa de melhorar os resultados para que fiquem mais eficazes e eficientes. De acordo com Cheng et al. [6], existem muitas técnicas diferentes para a segmentação, como por histograma, lógica fuzzy, abordagens de regiões, bordas e inteligência artificial. Bhattacharyya [7] ainda menciona técnicas baseadas em wavelets e algoritmos genéticos. Na Seção 2.1 são apresentados exemplos de trabalhos que fazem uso de técnicas computacionais na busca de soluções para problemas da medicina.

Este trabalho avalia e propõe o reconhecimento automático de lesão com a segmentação de imagens por três abordagens, uma delas voltada para processamento de sinais, com a análise da energia da cor e outras duas com aprendizado de máquina, utilizando-se das técnicas de árvore de decisão e rede neural.

A energia da cor pode suprir uma desvantagem dos algoritmos de crescimento de regiões, que está no fato de que as sementes tem um impacto profundo na segmentação. As técnicas de fusão de regiões implementam funções que reduzem este impacto, mas consequentemente aumentando a complexidade dos algoritmos em tempo e memória.

O Aprendizado de Máquina, que é uma subárea da Inteligência Artificial (IA), está sendo aplicado à muitos métodos diferentes, pela sua abordagem e similaridade com a representação de aprendizado do ser humano, neste trabalho serão usados com duas subdivisões, árvore de decisão e rede neural.

Esta pesquisa foi aprovada pelo Comitê de Ética em Pesquisa da Universidade Estadual de Londrina (UEL) de acordo com a lei 315/06 de 02/02/07 (emenda 1/2010). O indivíduo foi informado sobre a natureza da pesquisa e assinou documentação de consentimento. Nos experimentos estavam presentes pacientes com úlceras crônicas de membros inferiores causadas por insuficiência venosa. A pesquisa foi realizada no CISMENPAR (Consórcio Intermunicipal de Saúde do Médio Paranapanema) em parceria com o Hospital Universitário (HU) de Londrina/UEL.

Com esta pesquisa espera-se criar métodos, ou iniciar a criação de tais métodos, que possam auxiliar profissionais das equipes médicas na extração de características das lesões, facilitando diagnósticos e tratamentos.

Os objetivos específicos e gerais deste trabalho foram alcançados com ajuda de diferentes técnicas. Técnicas que terão suas características e metodologias descritas neste trabalho da seguinte forma: O Capítulo 2 traz a fundamentação teórica dos algoritmos

para as abordagens propostas. O Capítulo 3 traz os métodos usados na pesquisa, os experimentos realizados para cada algoritmo estarão descritos no Capítulo 4. Os resultados das metodologias propostas são apresentadas no Capítulo 5, e no Capítulo 6 temos as conclusões e considerações finais deste trabalho.

No Apêndice A encontram-se as Tabelas de Resultados dos algoritmos da árvore de decisão e da rede neural, como será visto adiante, para a visualização das informação de maneira mais completa.

O Anexo A traz um artigo traduzido do autor deste trabalho. Artigo este que também foi desenvolvido durante o período de mestrado, mas que não possui o mesmo tema deste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, serão abordadas as áreas e subáreas onde o trabalho está inserido, tal como, explicitando seu caráter multidisciplinar, trazendo trabalhos de diferentes áreas do conhecimento. Ainda serão apresentados os principais conceitos e descrições das técnicas utilizadas.

2.1 Trabalhos Correlatos

Não é difícil encontrar artigos sobre processos computacionais que ajudam equipes médicas. O trabalho de Bi et al. [8] não trabalha com imagens, mas sim, desenvolve um sistema de apoio à diagnósticos e tratamentos. Ainda, é possível encontrar trabalhos que extraíam características de imagens. Por exemplo, Yang et al. [9] usa imagens digitais para a detecção de catarata, mas extrai várias características da imagem sem segmentação, pois a aquisição retorna a imagem como deve ser analisada. Com segmentação de imagens, temos trabalhos como o trabalho de Arigi et al. [10], que segmenta imagens de rins. Neste caso, as imagens, mesmo sendo em tons de cinza, possuem alto contraste da região de interesse e um número conhecido de regiões a serem segmentadas.

Em Bochko et al. [11] são usadas imagens de úlceras de pele de membros inferiores, como neste trabalho, porém com imagens adquiridas por infravermelho, que necessitam de uma aquisição diferenciada. Trabelsi et al. [12] buscam segmentar doenças de pele utilizando a adaptação das bordas encontradas, através do cálculo da variância, mas que se torna dependente de um valor de limiar, Azevedo et al. [13] focam em processos de clusterização de componentes de cor pelo sistema de cor HSI e representa matematicamente a morfologia da úlcera. Para tal, necessita de cor de plano de fundo controlada, além de uma paleta de cores de amostragem.

Existem também estudos que buscam descritores para áreas de interesse, como o estudo de Jacquelin et al. [14], que não traz resultados tão precisos por buscar texturas ruidosas, além de buscar características em imagens em tons de cinza que reduzem o número de características que possam servir como análise. Esta abordagem pode também causar um grande custo de tempo. Houhou et al. [15] propõem um algoritmo rápido mas que não lida tão bem com ruídos em texturas, necessitando assim de uma superfície bastante homogênea em relação à textura, que infelizmente não se encaixa nas texturas de lesão de pele. Qing et al. [16] usam operadores de padrões binários locais como um descritor de texturas, este algoritmo propõe entradas aleatórias que demoram a convergir, como segmentação de regiões. Assim, a técnica para detecção da região inicial da área de interesse proposta por nós também pode ser usada por ele. Outros estudos mostram

técnicas que usam texturas na identificação de objetos de interesse como em Ilea et al. [17] ou Belongie et al. [18], ou ainda com texturas na detecção de bordas como em Malik et al. [19] e Xiaohan et al. [20], mas todos estes sem aplicações específicas que dificultam a utilização em problemas reais, como pode ser inferido pelo conceito trazido por Kekre et al. [4], descrito na introdução deste trabalho.

Várias outras técnicas utilizam diferentes propostas, mas que esbarram no mesmo contexto de não conseguirem bons resultados em aplicações reais, pois a busca de uma única região de interesse nestas propostas, traria altas taxas de erro, como em Us et al. [21] onde é apresentada uma função de energia que combina cor e bordas para segmentação e em An et al. [22] que também usa duas técnicas para a extração da área de interesse, mas que usa transformadas wavelets para texturas e contourlets para regiões de borda. Ainda, em Chen et al. [23] é apresentada uma segmentação que é baseada em cores e texturas juntas.

Entre todos, o trabalho mais similar com a proposta deste é Shan et al. [24], que seleciona uma semente para algoritmos de crescimento de regiões em imagens de ultrassom em mamas. A similaridade da proposta está no fato da busca por uma semente em um problema de imagens médicas. Já a diferença, está na aquisição que no caso de Shan et al. vem de imagens de ultrassom, que tem menos definição de cor, mas mais homogeneidade de texturas.

Para o objetivo deste trabalho, busca-se uma descrição de coloração da área que descreva se ela pertence à uma lesão ou não.

2.2 Energia da Cor

A energia da cor pode descrever a ideia de quanto vermelha, azul ou qualquer outra cor é a região observada. A ideia de energia da cor vem de teoria de processamento de sinais, onde energia do sinal é a soma do sinal ao quadrado. Assim, a energia da cor é o valor ao quadrado do canal de cor sendo analisado [25], como pode ser melhor vista na forma de equação, denotada pela Equação (2.1). Quando esta técnica é utilizada, os valores passam a ter uma maior disparidade, aumentando o extensão das possibilidades de valores, facilitando a expressão de valores diferentes. A diferença entre dois números é mais facilmente vista quando os valores são elevados ao quadrado, principalmente quando são usadas técnicas computacionais que trabalham com valores limitados e, ainda mais importante, valores próximos.

$$\varepsilon(x, y) = \sum_a^b |p(x, y)|^2 \quad (2.1)$$

A Equação (2.1) é a expressão formal do cálculo da energia de uma região de

pixels, onde o pixel p é o ponto esquerdo superior da janela sendo analisada. Dados (x, y) , a energia é a soma dos valores de um canal específico (vermelho, verde ou azul) de a até b , onde a é a coordenada (x, y) do ponto superior esquerdo da região e b coordenadas (x, y) mais a variação do tamanho da região nas direções direita para baixo, respectivamente.

Por exemplo, a energia mostrará a importância de um canal na região calculada e a diferença entre elas mostram o quanto a região atual se assemelha à região que deve ser segmentada. O algoritmo apresentado usa a energia dos canais de cor para calcular o quão visível cada um deles vai ser na região e, conseqüentemente, na imagem. Por exemplo, quando a tentativa for de segmentar uma região avermelhada o canal do vermelho será maior que os demais, e ao quadrado esse valor deve parecer ainda mais distante.

Caso a região seja predominantemente amarelada, o canal azul terá sua energia muito menor que os outros canais. Em outras palavras, será possível separar a intensidade da cor dentro da região baseado apenas na distância das energias em relação aos outros canais na região sendo analisada.

Um exemplo de como a energia pode ser um indicador melhor que o valor simples é o triângulo de Pitágoras, o triângulo com lados 3, 4 e 5. Analogamente, os lados do triângulo representariam os canais no sistema RGB, assim, vermelho, verde e azul respectivamente. Somados os valores de vermelho e verde o resultado seria um número maior que o que representa o canal do azul ($3 + 4 > 5$). Porém, se a comparação for feita com cada canal de cor elevado ao quadrado os valores se igualariam ($9 + 16 = 25$). Suponhamos, então, que os valores fossem 3, 4 e 6 para vermelho, verde e azul respectivamente. A soma dos valores de vermelho e verde continuaria ultrapassando o valor de azul ($3 + 4 > 6$), mas ao utilizar os valores ao quadrado haveria uma noção melhor de que o azul está tendo mais impacto na região que vermelho e verde no ponto em questão ($9 + 16 < 36$), que é o que está sendo procurado pelo descritor, ou seja, um valor que possa descrever melhor a região.

Assim como na percepção humana, olhar para uma região e ver que esta região tem uma coloração próxima de uma área qualquer da imagem, será assumido que esta região pertence à área previamente reconhecida.

Portanto, o método consiste em calcular a diferença entre as energias de dois canais para todas as partes possíveis da imagem. Quando a energia de uma cor conseguir se manifestar de forma mais forte em relação às outras, esta região terá a característica procurada. No caso apresentado, a cor de maior interesse é o vermelho, ou seja, procura-se por regiões onde a influência do vermelho é maior que as dos outros dois canais. Quando os valores das energias denotarem a maior diferença dentro da imagem, essa região será dada como objeto de interesse. Visto que a região de interesse se trata de uma lesão de pele e a mesma possui uma coloração avermelhada.

O algoritmo propõe tamanhos diferentes de região, essas diferenças de tamanho podem ser usadas de duas formas diferentes para a segmentação:

- **Regiões pequenas como semente:** Algoritmos conhecidos, como visto em Likas et al. [26] e Friedland et al. [27], usam sementes para realizar a segmentação por crescimento de regiões. Esses métodos não exigem um grande conjunto de dados para iniciarem. Assim, mesmo com regiões pequenas, a energia já terá informação suficiente para que estes algoritmos iniciem seu processo de crescimento na área de lesão.
- **Regiões grandes como textura:** A segmentação de texturas necessita de um conjunto maior de informação em relação às de semente, pois texturas precisam de regiões e descritores que possam gerar padrões que serão os focos de busca para o algoritmo. Portanto, quando são geradas regiões grandes, elas seriam melhores utilizadas como texturas do que reprocessadas como sementes, já que elas já teriam um conjunto grande de informações para descrever padrões [28] [29].

2.3 Aprendizado de Máquina

O Aprendizado de Máquina (ML – da sigla em inglês para *Machine Learning*) é uma disciplina, parte da IA, que constrói e estuda algoritmos que possam aprender com base em dados de amostragem, construindo modelos que operam baseados em um conjunto de entrada [30].

Em Singh et al. [31], ML é uma alternativa para a automatização da segmentação de imagens. ML é uma área da IA onde os computadores são programados para aprender com experiências passadas. Para tal, eles usam princípios de inferências chamadas de indução, nos quais uma generalização de conclusões podem ser obtidas a partir de um conjunto de exemplos [32]. Mais especificamente, através do aprendizado supervisionado, que é uma forma de ML, na qual uma função ou hipótese é inferida proveniente de um conjunto de dados rotulado, capaz de resolver ou generalizar resultados desconhecidos até o momento [33].

Neste trabalho, foram utilizadas duas técnicas vindas de ML, uma árvore de decisão e uma rede neural.

2.3.1 Árvore De Decisão

A árvore de decisão é uma ideia intuitiva de como adquirir uma informação. Pois não é raro usar uma série de perguntas e respostas como filtros para a conclusão de uma decisão, ou seja, usar uma série de descrições como filtros para chegar na informação desejada. Na árvore de decisão um nó é determinado como folha e declara um valor de

retorno (ou decisão tomada) ou esse nó é a raiz de uma subárvore contendo uma nova especificidade daquilo que já foi testado, assim criando um novo nível da árvore e filtrando o conjunto de dados possíveis em uma resolução mais simples, por meio da divisão das características obtidas.

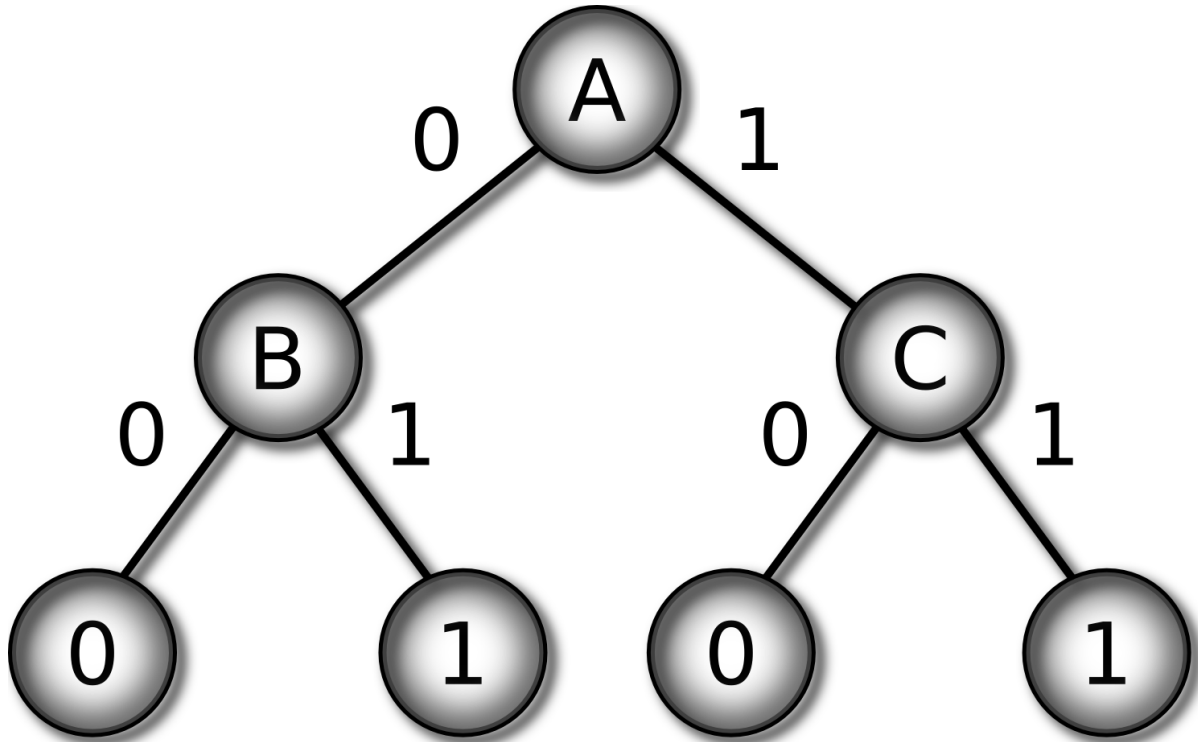


Figura 1 – Árvore de decisão abstrata.

[34]

Na Figura 1 está disponível, de forma visual, uma simulação simples de como uma série de questões podem determinar uma decisão, onde cada nó representa uma especificidade da característica testada previamente e cada folha um retorno da decisão que seria tomada. C, então, representaria uma especificidades do conjunto de dados que tem A como verdadeiro e B especificidade do conjunto onde A é falso. Ainda assim, B e C não precisam fazer testes das mesmas características de A e nem das mesmas características entre si.

Exemplo: A é uma pergunta do tipo “É um imóvel ou um automóvel?”, se imóvel, C seria a pergunta “Residencial ou comercial?”. B portanto, no nosso exemplo, representaria o conjunto dos automóveis e seria a pergunta “Passeio ou transporte?”. Ou seja, B e C filtrariam os resultados obtidos pela decisão tomada em A, com perguntas do seu próprio conjunto, que podem ou não ter interferência das outras atividades previstas na árvore.

Este modelo genérico de árvore se chama CART (sílabas em inglês de árvore de classificação e regressão - *Classification And Regression Tree*) [35].

Em forma de algoritmo teria a seguinte forma:

Algorithm 1 Algoritmo da árvore de decisão abstrata.

```

1: se A então
2:   se C então
3:     retorna verdadeiro;
4:   senão
5:     retorna falso;
6:   fim se
7: senão
8:   se B então
9:     retorna verdadeiro;
10:  senão
11:    retorna falso;
12:  fim se
13: fim se

```

Como é possível perceber pelo Algoritmo 1, uma árvore simples, como a mostrada na Figura 1, pode gerar uma série longa de códigos aninhados.

A principal vantagem da árvore de decisão é tempo reduzido ao obter a decisão, como o resultado precisa apenas descer uma árvore, o resultado pode ser obtido em tempo $O(\log n)$ [35], onde n é o número de regras encontradas pelo algoritmo para se tomar a decisão.

A desvantagem é encontrada na construção na árvore, que é feita com a abordagem de dividir-para-consquistar, onde cada conjunto é dividido em subconjuntos mais simples. Assim o nó folha define uma classe e a subida da árvore até o nó raiz são descritores que generalizam a classe encontrada na folha [36]. Esta divisão em subconjuntos e generalizações podem ser realizadas várias vezes e assim, essas atividades podem levar bastante tempo.

2.3.1.1 Algoritmo J48

O algoritmo J48 é a implementação do algoritmo C4.5 de árvores de decisão, que é o refinamento de outro algoritmo, o ID3. O ID3 foi desenvolvido para trabalhar com valores nominais apenas, ou seja, não havendo interpretação numérica. Como no exemplo mostrado anteriormente, mas as respostas de filtros posteriores não poderiam assumir valores numéricos ou booleanos [35]. Soluções de níveis numéricos do ID3 tomam um pré-processamento para modificar o conjunto de dados de forma que este pudesse então ser executado.

O C4.5 trabalha as divisões dos conjuntos dados não binários como o ID3, ou seja, para casos onde um nó tem mais de dois filhos. Também trata no sistema CART os dados do conjunto de valores reais, incluindo os booleanos que podem ser facilmente visto como

valores reais, mas com menos sensibilidade à valores desconhecidos, pois o C4.5 retorna uma subárvore de especialização com as probabilidades de cada nó com seus pesos no casos de informações desconhecidas [35].

O algoritmo inclui também um sistema de podas, onde a informação macro, elimina uma pesquisa menor se essa especificidade não contiver duas classes, tal como se visto que um número pertence ao conjunto dos naturais elimina a necessidade de verificar se este número pertence aos inteiros, mas também como visto previamente, a descida pelo nó que nega a afirmação correta, não eliminaria qualquer teste.

Outro exemplo, que pode ser encontrado em Duda et al. [37] seria a dos seguintes testes:

SE [(0.04 x_1 + 0.16 x_2 < 0.11)
 E (0.27 x_1 - 0.44 x_2 < -0.02)
 E (0.96 x_1 - 1.77 x_2 < -0.45)
 E (5.43 x_1 - 13.33 x_2 < -6.03)]
 ENTÃO $x \in \omega_1$

Por exemplo, estes testes poderiam caracterizar valores de canais de cores do modelo RGB, sendo x_1 e x_2 , valores de vermelho e verde, comparados com os valores de azul, que então, aplicaria x em uma determinada classe.

Se cada um dos testes acima precisassem ser feitos, cada uma das etapas “E” que seriam testadas separadamente, poderiam ser simplificadas com os seguintes testes:

SE [(0.04 x_1 + 0.16 x_2 < -0.11)
 E (5.43 x_1 - 13.33 x_2 < -6.03)]
 ENTÃO $x \in \omega_1$

Ou seja, os mesmos valores de x aceitos nos primeiros testes, também seriam aceitos na segunda formação de testes.

2.3.2 Rede Neural Artificial

Rede Neural Artificial (RNA) é um sistema de processamento que tem como principal fonte de inspiração as redes neurais biológicas [38].

Haykin [39] define como rede neural artificial, um processo massivamente distribuído paralelamente feito por unidades simples de processamento, que tem uma propensão natural à armazenar conhecimento experimental e liberando-o para ser usado quando necessário. Que se assemelha ao cérebro de duas formas:

- Conhecimento é adquirido pelo ambiente através de um processo de aprendizado;
- Conexões entre neurônios, conhecidas como sinapses, são usadas para manter o conhecimento armazenado.

Além desta definição, Haykin [39] também propõe várias propriedades e capacidades da utilização de redes neurais que se tornam interessante para o uso neste trabalho. Essas propriedades são:

- **Não Linearidade:** Redes neurais podem resolver problemas tanto lineares como não lineares;
- **Mapeamento de entrada e saída:** Possibilidade de aprendizado pela realização de indicadores (aprendizado supervisionado);
- **Adaptatividade:** Uma rede pode se adaptar e se restringir ao aprendizado de uma determinada característica se mais importante e pode se modificar à variações do ambiente;
- **Evidência de resposta:** A rede pode informar o grau de certeza dado na resposta;
- **Informação contextual:** Como todos os neurônios trabalham juntos para os resultados, as modificações em um podem gerar alterações de resultados em outros, assim, não apenas uma característica seria determinante, mas o contexto na qual é aplicada também;
- **Tolerância à falhas:** Redes neurais quando implementadas em forma de hardware tem boa tolerância à falhas, pois sua performance tende a ser melhorada com o tempo.
- **Analogia Neurobiológica:** O funcionamento que simula o funcionamento das células do cérebro é um grande indicativo de que essas estruturas tendem a propiciar um grande aprendizado.

Entre outras vantagens, que podem ser extraídas das características previamente citadas, temos também o fato de um novo aprendizado utilizar a situação atual da rede, não precisando de dados que já tenham sido usados. Ou seja, caso uma nova imagem seja usada como dado de aprendizado, ela pode ser colocada como elemento de entrada e a rede neural vai se adaptar com os valores que ela já tem para receber valores novos. Enquanto a árvore de decisão precisaria ser inteiramente remontada.

Segundo Braga et al. [40], uma grande propriedade das RNAs reside no fato de possuir a capacidade de produzir inferências produzidas através de conteúdo aprendido

por meio de exemplos, assim obtendo desempenho progressivamente melhorado. Para isto, utilizam-se de aprendizado vindo de ajustes de pesos das conexões de suas unidades de processamento.

2.3.2.1 Radial Basis Function Network

A RBFN (sigla em inglês para Rede de Função de Base Radial - *Radial Basis Function Network*) tem como modelo de arquitetura um vetor de entrada, seguido de uma camada de neurônios e um camada de nós de saída [41]. Como demonstrado na Figura 2.

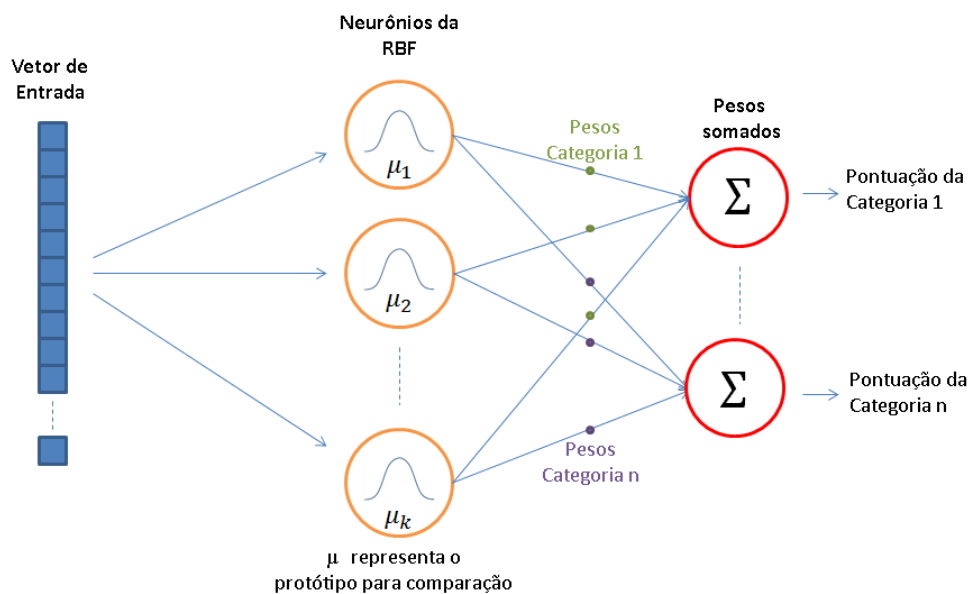


Figura 2 – Modelo de arquitetura de uma RBFN.

[34]

O vetor de entrada é o responsável por inserir o dado a ser treinado pela RBFN e classe ao qual os valores pertencem. A camada de neurônios é responsável por armazenar "exemplos" de classe, a distância do dado de entrada com estes exemplos, chamados centróides, que resultarão no grau de certeza ou confiança da classe escolhida. Na última camada, temos as somas dos pesos da RBFN, que indicarão uma proximidade com algum dos pontos de exemplo armazenado, assim, a distância do centróide mais perto do dado é a saída e o grau de confiança da rede [42].

A classificação da RBFN é feita pela similaridade da entrada com os exemplos usados para treinamento [43]. Ou seja, seus neurônios tem um comportamento semelhante ao algoritmo K-Média [26], onde é encontrado um centróide e é definido como saída o centróide mais próximo da entrada. Como a RBFN foi usada de forma supervisionada, os exemplos de entrada servem para encontrar os centróides e qual o raio de ação de cada um deles.

Os centróides podem fazer diferentes classificações dependendo da arquitetura, ou seja, os elementos agrupados pelas proximidades dos centróides podem fazer parte de um único grupo, ou cada centróide ter uma classificação diferente.

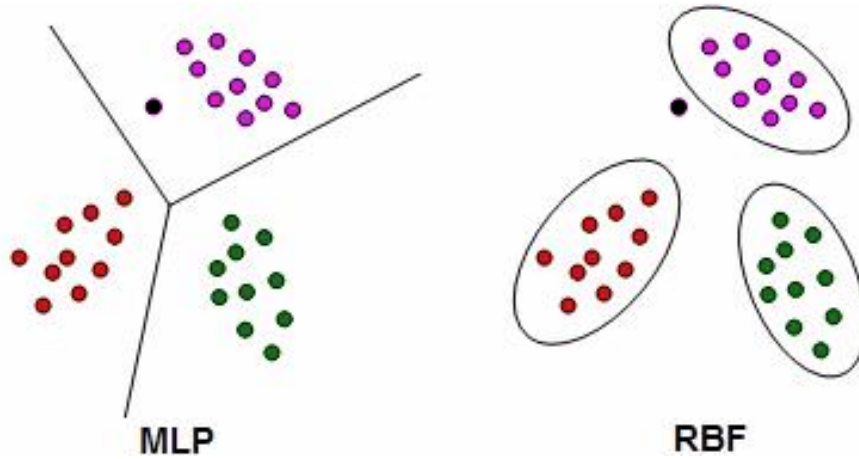


Figura 3 – Diferença de forma de clusterização da MLP e RBFN.

[34]

A Figura 3 mostra um exemplo do conceito em comparação com a MLP (sigla em inglês para Perceptron Multicamadas - *Multilayer Perceptron*). A MLP separaria os clusters pelas retas (ou hiperplanos, dependendo das dimensões do problema) e uma nova característica, representado pelo ponto preto, faria parte do cluster indicado. Já a RBFN conseguiria separar os clusters, que poderiam descrever uma mesma classe, e o ponto externo seria considerado de outra classificação.

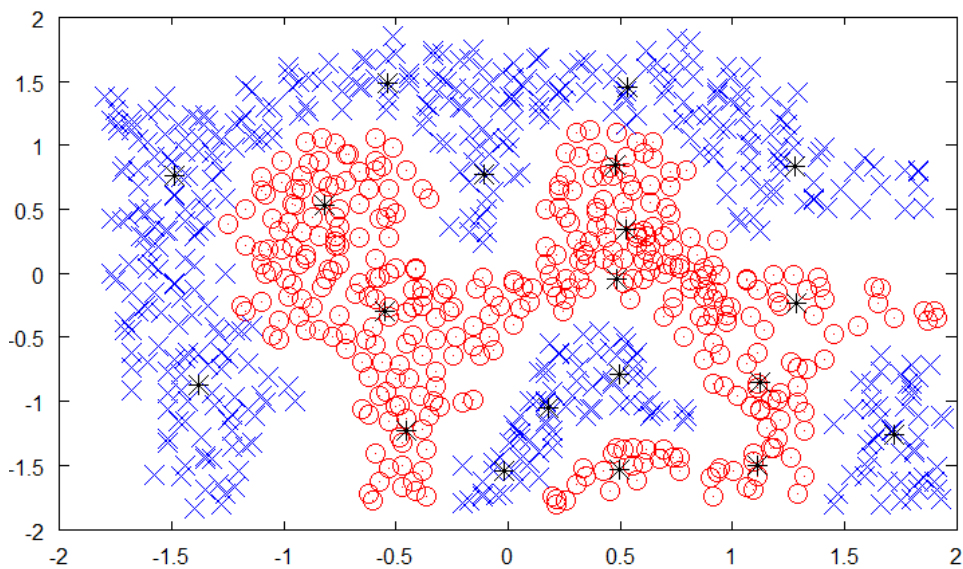


Figura 4 – Exemplo de amostragem.

[34]

Classificações como vistas na Figura 4, demonstram características que dificultam muito a utilização da rede como a MLP. Onde seria necessário um bom conhecimento da distribuição do problema para a montagem de uma arquitetura que pudesse representar de maneira aceitável o aprendizado.

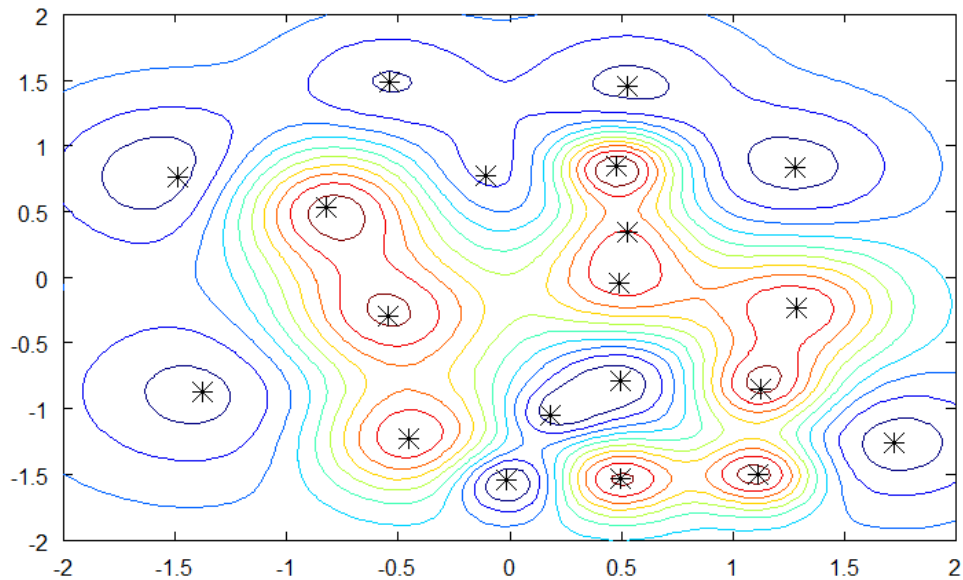


Figura 5 – Representação da clusterização feita com a RBFN.

[34]

A Figura 5 demonstra como ficaria a divisão dos clusters pela RBFN com entradas demonstradas pela Figura 4. A coloração das divisões indicam a proximidade das curvas com os pontos dos centroides, classificando classes diferentes.

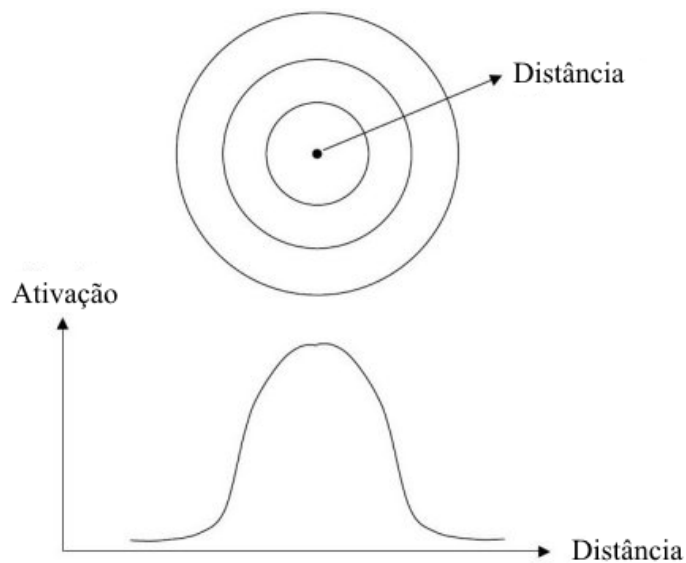


Figura 6 – Relação de distância do centroide.

[34]

Essa relação de distância do núcleo do centróide que é delimitante para a determinação de cada classe para o conjunto de entrada e é através dela que é feita a função de ativação do resultado. Existem formas diferentes de distribuir a relação de distância demonstrada na Figura 6:

$$f(x) = \frac{\sigma^2}{\sigma + |x - c|^2} \quad (2.2)$$

$$f(x) = (x^2 + \sigma^2)^{-\alpha} \quad \alpha > 0 \quad (2.3)$$

$$f(x) = x^2 \ln(x) \quad (2.4)$$

Porém a mais comum é baseada na curva Gaussiana [44], que tem a Equação 2.5 para uma dimensão:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (2.5)$$

Onde x é a entrada, μ é a média e σ é o desvio padrão.

Na aplicação da RNA, a parte da Equação (2.5) que multiplica a exponencial não é utilizada, pois essa é determinada pelos pesos das arestas da construção da RBFN [45]. Assim, temos a Equação (2.6):

$$f(x) = e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (2.6)$$

Onde x é a entrada, μ é a média e σ é o desvio padrão. A média será o ponto onde se encontra o centróide do referido conjunto. Outra forma de representação que pode ser encontrada é onde a razão da variância (quadrado do desvio padrão) pode ser multiplicada por uma constante de estreitamento da curva Gaussiana, que pode ser representada pela Equação (2.7):

$$f(x) = e^{-\beta(x-\mu)^2} \quad (2.7)$$

Mantendo x como a entrada, μ e como o centro do conjunto, portanto, β seria a razão da variância de acordo com o estreitamento da curva a qual se deseja na Gaussiana.

Como pode ser visto na Figura 7, variações na constante obtida pela variância, modificariam o comportamento da curva Gaussiana, dando valores diferentes de retorno

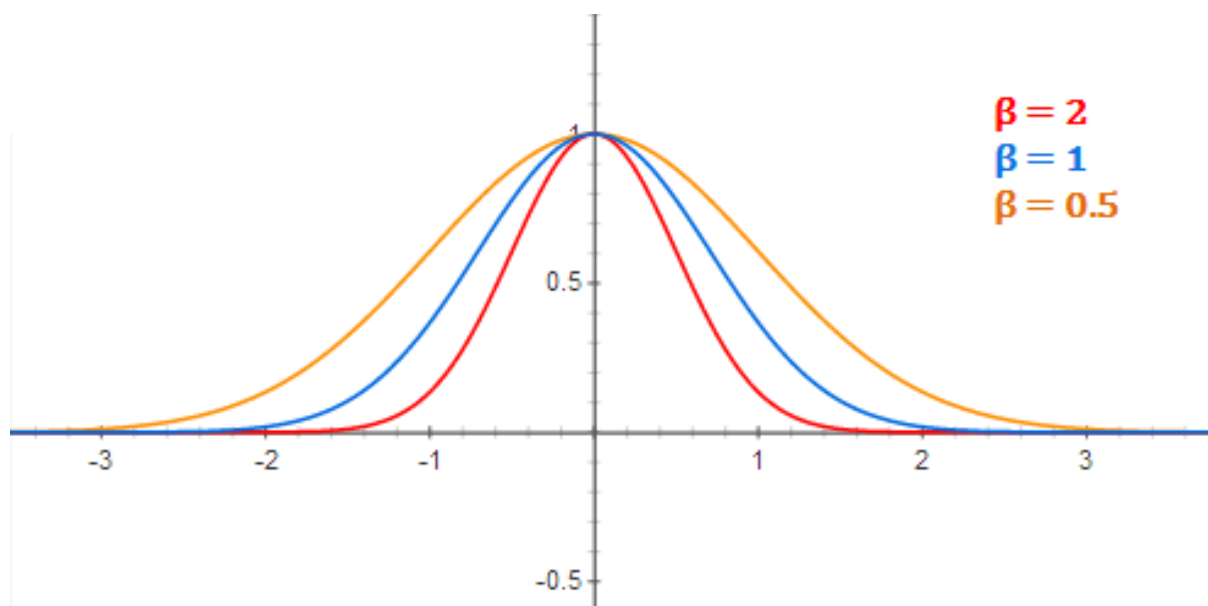


Figura 7 – Variações da curva Gaussiana.

[34]

de acordo com a distância dos centróides. Esses valores podem trazer mais ou menos rigorosidade na saída, dependendo apenas do objetivo em questão.

Os métodos de pesquisa, que serão apresentados no próximo Capítulo, vem não só buscar uma área que ainda se produz poucos métodos computacionais, que seria a área de lesões de pele, mas também, trazer métodos e descritores que podem ser usados pelos trabalhos citados na tentativa de melhorar os resultados obtidos até o presente momento.

Assim, tentaremos motivar outros pesquisadores a usar os métodos propostos na busca pela melhor segmentação possível. Utilizando-nos de métodos ainda não utilizados como o de energia da cor, ou de eficácia já testadas como no caso do aprendizado de máquina, direcionados a nosso objetivo de encontrar certezas de regiões de lesão.

3 MÉTODO DE PESQUISA

O objetivo dos métodos é basicamente o mesmo, encontrar regiões de certeza de lesão. Porém, as diferentes formas de chegar a esse objetivo traçam caminhos diferentes.

Na utilização de energia da cor, voltaremos a busca por uma região menor que pertença totalmente a lesão. Já para algoritmos de aprendizado de máquina, buscaremos a maior região possível, contanto que o algoritmo possua um nível elevado de certeza da região sendo de lesão.

Para fins de futuras comparações e para que este trabalho possa ser usado por outros pesquisadores, o processo de aquisição das imagens e avaliação de resultados foi mantido para todas as estratégias.

3.1 Aquisição de informação

As imagens foram adquiridas de oito pacientes diferentes em fases diferentes de lesão, com uma câmera Sony, modelo DSC-S1900, que permite uma boa portabilidade, sendo ela pequena, e também aceita configurações manuais. As configurações foram usadas em modo manual para que houvesse controle da similaridade entre as imagens, deixando apenas o foco automático. Assim, a câmera poderia ser configurada apenas uma vez e a própria equipe médica poderia fazer aquisições posteriores.

Conjuntos de imagens foram adquiridos para ajustes das configurações de câmera para serem usadas em todas as aquisições. As configurações tem como propósito principal, manter a qualidade de cor e evitar superexposição de luz, evitando assim, reflexos que pudessem danificar a planilha de cores. O conjunto com as melhores características que se encaixam no propósito do trabalho tiveram as seguintes características:

- Valor de exposição: -0.3;
- F-number 3.1;
- ISO: 400;
- Flash: desabilitado.

As imagens, adquiridas com as configurações previamente citadas, tiveram os limites da área da lesão demarcada por médicos no hospital, assim seria possível comparar o que pode ser considerado como lesão para futuros resultados.

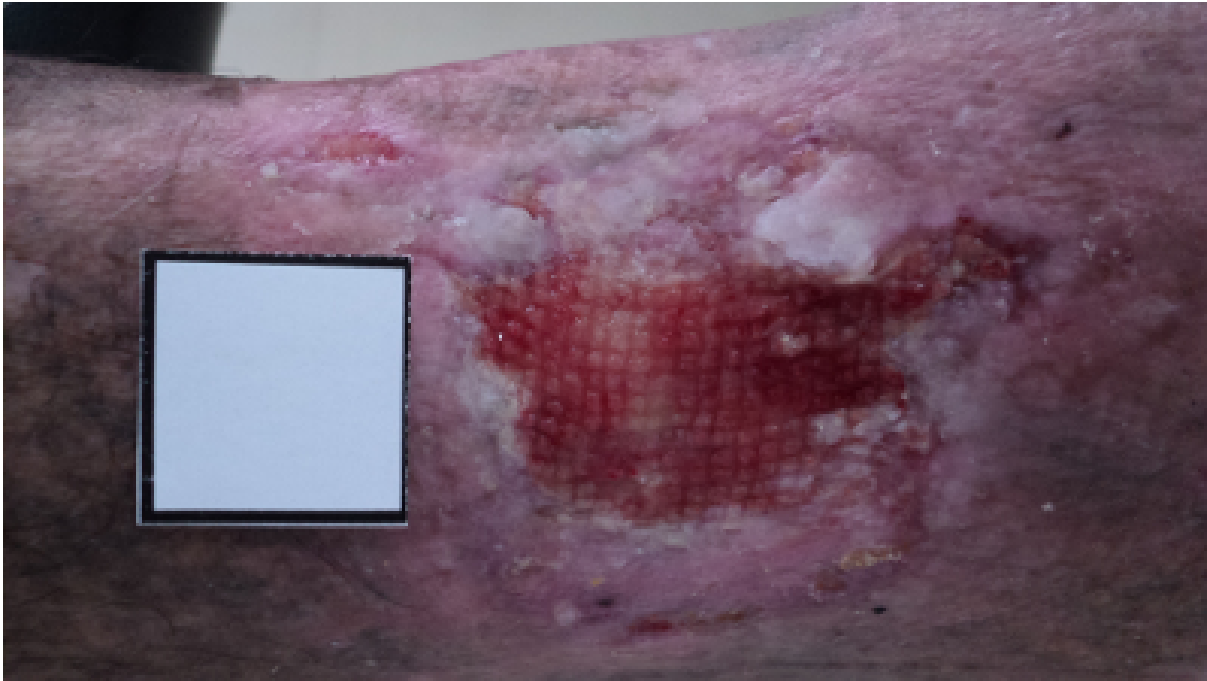


Figura 8 – Imagem obtida nas condições supracitadas.

A Figura 8 mostra um exemplo de imagem adquirida com as configurações previamente descritas. As configurações foram mantidas em todas as imagens, para que, sendo imagens do mesmo tipo de lesão, houvesse similaridade entre as imagens.

Foram adquiridas trinta e três imagens em formato bitmap. Como a utilização das imagens não necessitavam de imagens com alta resolução, elas foram reduzidas e as imagens usadas nos experimentos foram mantidas em resolução 400 x 225 pixels, além disso, um dos experimentos necessitou do armazenamento das partes da imagem em lista, assim, imagens com resolução muito alta exigiriam processamento desnecessário, mas não teriam uma diferença significativa de eficácia.

Para que fossem feitos experimentos com tamanhos diferentes da janela a ser analisada, os experimentos foram feitos com porcentagens das medidas da imagens ao invés do número real de pixels. Assim, o experimento não fica atrelado à imagens com apenas a resolução usadas para este trabalho, como é melhor detalhado na Seção 4.1.2.

O quadrado branco presente na Figura 8 é um marcador para cálculo de área, pois o processo de segmentação é um dos passos para a extração de características da lesão. Assim, um objeto de fácil detecção com tamanho conhecido foi posicionado na imagem como referência. Além disso, o quadrado também pode ser utilizado como um bom objeto para o balanço do branco da câmera, que é o ajuste de cores feito para compensar o efeito causado pela temperatura da forma de iluminação, aproximando as cores da coloração real dos objetos [46].

3.2 Critérios de avaliação

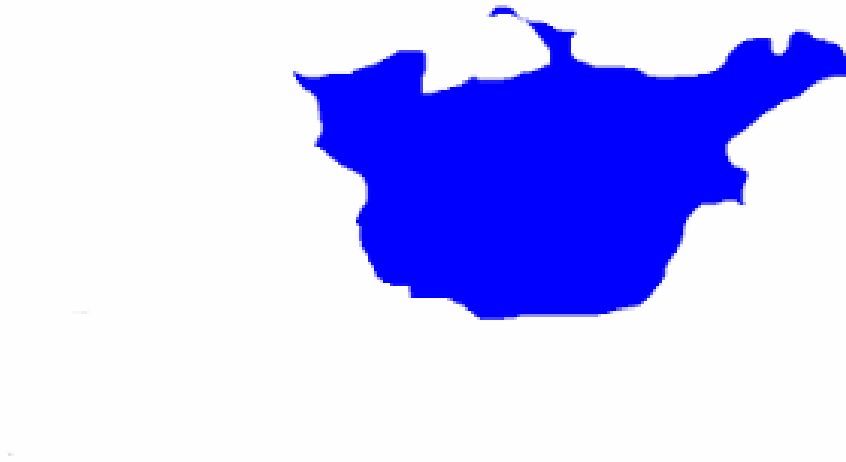


Figura 9 – Imagem com região delimitada por médicos.

Na Figura 9 é mostrada uma região marcada pelos médicos, sendo a marcação da área de lesão da Figura 8. Estas imagens foram obtidas com a ajuda de médicos especialistas que identificaram nas imagens as regiões que seriam consideradas lesão. Seguindo esta marcação, a área demarcada foi preenchida para facilitar a visualização e comparações posteriores. O preenchimento também auxiliou na identificação automática da área de lesão.

Para as abordagens de ML, o conjunto de dados não pode conter todas as trinta e três imagens adquiridas, pois, parte delas foi usada como estratégia para o aprendizado supervisionado, ou seja, utilizando as marcações previamente descritas pelo processo da energia da cor, foram classificados os pixels pertencentes ou não à lesão, assim seria possível informar as características de cada pixel e seus status de pertencer à lesão.

Das trinta e três imagens adquiridas, dez foram usadas na construção do conjunto de dados, ou seja, aproximadamente trinta por cento das imagens. Assim, o conjunto de dados que pode ser usado nas abordagens de aprendizado contém aproximadamente um milhão e meio de conjuntos de entradas. As características informadas, as quais formam os conjuntos de entradas, variam de acordo com a abordagem de aprendizado utilizada.

Em [47], os autores estabelecem que a mais famosa forma de diagnóstico é o da regra ABCD (importado da sigla em inglês para Assimetria, Borda, Cor e Diâmetro –

Asymetry, Border, Color, Diameter), e que as novas tendências para segmentação de imagens de pele são através de métodos de cores e texturas.

Alguns estudos buscam por sistemas de cores diferentes para analisar homogeneidade de cores e texturas pela dependência da intensidade de luz encontrado no sistema RGB [48]. Entretanto, o sistema RGB é um espaço de cor que tem uma representação muito próxima da percepção da visão humana [49]. Assim, o sistema RGB foi o escolhido como alvo de nossos experimentos.

A estratégia de avaliação usará as imagens provenientes das marcações dos especialistas como modelo ouro. Assim, como buscamos a decisão final da equipe médica no auxílio do diagnóstico de lesão, teremos esta decisão como o nosso grau de certeza de 100% de região de lesão.

De tal forma, como veremos nos experimentos, as comparações de acertos serão feitos utilizando essas imagens de marcação para atestar o que é ou não parte da lesão.

4 EXPERIMENTOS

Os experimentos foram realizados usando um computador HP de processador Intel Core 2 Quad Q9505 de 2,83GHz, 8GB de memória RAM, usando Windows 7 profissional de 64 bits como sistema operacional, os algoritmos foram desenvolvidos na linguagem Java em sua versão 1.7 com IDE Eclipse Juno. Além disso, a árvore de decisão utilizada foi a que pode ser encontrada no *software Weka* versão 3.6.7.

4.1 Experimentos com Energia da Cor

O objetivo principal da energia da cor foi o de encontrar uma região com certeza de lesão. Objetivos secundários foram traçados em conjunto com este, como por exemplo, encontrar intervalos onde a região de lesão ainda possa ser encontrada. Ainda, como especificidade dos experimentos, a tentativa de melhorar a performance para futuros estudos.

Assim, os experimentos sobre a energia da cor foram divididos em três.

4.1.1 Experimento para encontrar uma região de lesão

O objetivo do primeiro experimento foi encontrar uma região de certeza de lesão, para que esta região pudesse ser usada como semente pelos algoritmos de crescimento de regiões.

Algorithm 2 Encontra a região com a maior diferença do vermelho para os outros canais.
divide a imagem em regiões;

para todo regiões divididas **faça**

 calcule a energia dos canais vermelho, verde e azul;

 subtraia as energias de verde e azul da energia do vermelho;

 adicione o valor em uma lista ordenada;

fim para

retorna a lista;

A entrada do Algoritmo 2 é uma imagem de lesão, como exemplificada pela Figura 8 da Seção 3.1, na página 42. Já a saída será uma lista ordenada de forma crescente das diferenças do canais verde e azul em relação ao canal vermelho.

A primeira região da lista é obtida com a diferença de energia mas parâmetros auxiliares foram armazenados. Entre estes parâmetros temos as coordenadas x e y do ponto esquerdo superior. A ordenação foi estipulada em maiores primeiro.

Como pode ser visto no Algoritmo 2, não seria necessário manter a lista pelo simples fato de obter a região com a maior diferença das energias, seria possível e mais

fácil retornar apenas as informações da região com a maior diferença, durante o processo de cálculo. Porém, o algoritmo descrito também foi usado em outros experimentos como serão descritos posteriormente, assim, foi necessário o armazenamento da lista.

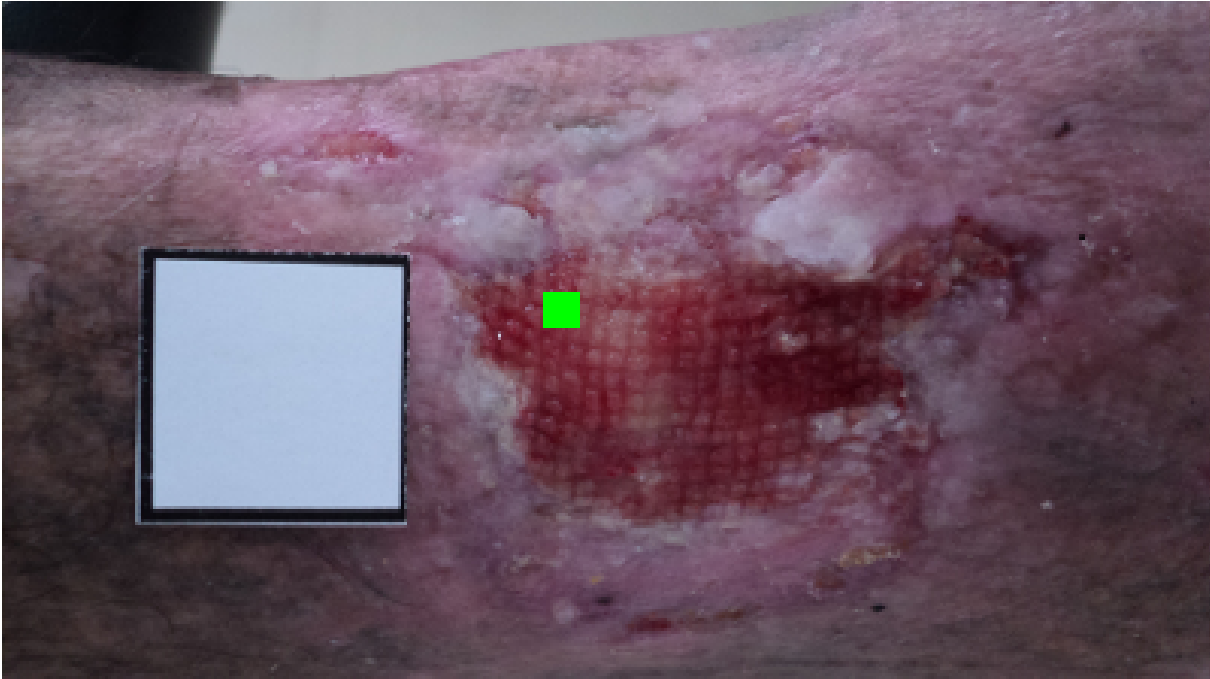


Figura 10 – Região com a maior diferença de energia encontrada pelo algoritmo.

Na Figura 10, temos uma região de lesão encontrada pelo Algoritmo 2, a região em verde denota a região com a maior diferença entre a energia do canal vermelho e as energias dos canais verde e azul somadas.

4.1.2 Experimento com tamanhos diferentes de regiões

No segundo experimento, a mesma tentativa foi feita, porém com tamanhos diferentes de região para o cálculo da energia. Para algoritmos onde uma região pequena é suficiente como amostragem, seria interessante manter a menor região possível armazenada para não precisar de muita informação em memória. Assim como, as regiões menores tendem a conter menos ruído, pois caso a região pequena contiver ruídos, como visto anteriormente, ela não estaria nas primeiras posições das diferenças de energia.

Porém, dividir a imagem em muitas regiões toma mais tempo ao comparar os resultados com um número maior de regiões possíveis no momento da escolha da melhor diferença. Assim, variações dos tamanhos de região implicam em possibilidades de realizar análises mais precisas ou eficientes, dependendo então, da aplicação ou do objetivo da extração que está sendo realizada.

As alternativas de tamanhos de regiões foram feitas com porcentagens da largura e altura da imagem, para que pudessem ser usadas com qualquer imagem de entrada, mesmo que não estivessem na mesma resolução das anteriores. No caso deste trabalho, não haveria esta necessidade pois todas as imagens tinham o mesmo tamanho.

Inicialmente, foram usadas regiões de 2% da largura e altura da imagem, então foram adicionados 2% em cada execução até atingir 10% da largura e altura, ou seja, para as imagens previamente descritas de resolução 400 x 225, foram usadas regiões de 8x4 (2%), 16x9 (4%), 24x13 (6%), 32x18 (8%) e 40x22 (10%) pixels, calculados diretamente pelo método com base na imagem de entrada.

Além disso, alguns algoritmos são adequados para regiões maiores que poderiam conter um montante de informação superior para decisões para junção de novas áreas à semente. Nestes casos, não seria interessante manter uma região muito pequena como semente, pois a economia feita de memória, não compensaria o gasto elevado com outros processos que poderiam ainda, gerar outras informações para a memória, não havendo economia. Outros processos pode ser necessários para tratar ruídos, que teriam um impacto maior, pois representariam uma porção maior da região analisada, caso essa fosse pequena.

4.1.3 Experimento para contar quantas regiões são encontradas

O terceiro experimento visou identificar quantas regiões da lista ordenada obtida no primeiro experimento seriam partes da região de lesão, ou seja, quantas das maiores diferenças de energias da lista encontrariam a região de lesão. Este experimento foi realizado para identificar outras áreas de lesão para usar como sementes, em casos onde a lesão é menos avermelhada ou quando a lesão estiver em fase diferente do processo de cura, que pode variar a coloração da mesma.

Para a execução do Algoritmo 3, o conjunto de entrada necessário seria a lista ordenada resultante do Algoritmo 2 e sua saída é o número de regiões com as maiores energias que fazem parte integralmente da lesão.

Algorithm 3 Encontra quantas regiões com as maiores diferenças que encontram a região de lesão.

```

1: para todo elementos da lista ordenada faça
2:   para  $(w, y) \leftarrow (x, y)$  até  $(x + dx, y + dy)$  faça
3:     se  $(w, h) \notin WoundRegion$  então
4:       retorna índice da lista;
5:     fim se
6:   fim para
7: fim para

```

O Algoritmo 3 busca quantas regiões pertencem à lesão. Portanto, procura em

todas as regiões calculadas se existe algum pixel que não pertença à lesão. A comparação de pertencer ou não foi feita com a região preenchida da marcação feita pelos médicos mostrada na Figura 9 da Seção 3.2, encontrada na página 43.

Qualquer pixel que não faça parte da região demarcada pelos especialistas, faz com que a região inteira passe a ser considerada fora da região de interesse. Esta suposição é bastante extrema, já que um único pixel representaria a não presença do resto todo da região consigo, mas esta afirmação é essencial para o trabalho. Visto que assim, seria garantido que a região escolhida faria parte inteiramente da região e não conteria nenhum ruído para a semente do algoritmo de crescimento de regiões. Assim, mesmo para o algoritmo de crescimento que trabalhar com regiões pequenas ou que não tenham boa tolerância a ruído, não seria necessário o tratamento posterior destes ruídos.

4.2 Experimentos com Aprendizado de Máquina

As características do conjunto de entrada são os valores usados pelos algoritmos para a generalização do resultado, quando junto com as características é informado o grupo a qual estas características pertencem, este aprendizado é chamado de supervisionado, ou seja, são informadas as características que descrevem os exemplos com a classe a qual essas características pertencem. Aprendizado não supervisionado informa os valores das características, mas não a quem pertencem, nestes casos os algoritmos buscam a generalização do problema buscando por grupos bem definidos ou com grau de erro aceitável.

A dificuldade da generalização do problema faz com que a abordagem de aprendizado não supervisionado não seja interessante, pois nesse caso, os grupos deveriam possuir características distintas o suficiente para a generalização das duas classes, caso contrário, o custo computacional seria muito alto e/ou com resultados não satisfatórios. Aliada a esse fator, existe a vantagem dos experimentos possuírem uma determinação dos médicos do que deve ser considerado lesão. Assim, o treinamento, tanto para a árvore de decisão quanto para a rede neural, foi feito com abordagem supervisionada.

4.2.1 Experimento com Árvore de Decisão

Como características de construção das instâncias do *software Weka*, foram usados os atributos de cor RGB da imagem, junto com o rótulo de qual a região do pixel em questão pertence. Como exemplos de linhas de entrada para o algoritmo temos:

54, 41, 35, false;

27, 23, 24, true;

26, 22, 23, true;

225, 206, 192, false;

Onde o primeiro valor corresponde ao valor de vermelho (R - *Red*), o segundo de verde (G - *Green*), o terceiro de Azul (B - *Blue*) e o último valor é o correspondente a relação fundo, ou seja, será utilizado verdadeiro (*true*) quando o pixel citado fizer parte do fundo, assim, não fazendo parte da lesão.

Os experimentos foram realizados com o conjunto de funções do *software Weka* com a importação dos seus métodos diretamente no código Java desenvolvido. Os valores dos atributos da árvore foram mantidos padrões do modo visual do software.

O *software Weka* foi usado como um recurso de tempo, já que possui as implementações da árvore que foi escolhida, além disso, o *software*, que é usado em pesquisas científicas, proporciona a sua utilização direta em código fonte. Ou seja, possibilita que suas funções sejam usadas diretamente na implementação, facilitando a automatização para execução em larga escala.

Facilidade esta também proveniente da linguagem Java, onde a simples importação do *Java Archive* (.jar) no projeto é suficiente para a importação e execução das funções requeridas.

4.2.2 Experimento com Rede Neural Artificial

Tal como visto na Seção 4.1, neste presente trabalho é mais interessante termos áreas de lesão não encontradas do que áreas de pele encontradas, por dois motivos principais. O primeiro, pelo fato de as áreas de pele constituírem maior parte da imagem, assim, caso a RNA esteja reconhecendo áreas de pele como lesão, provavelmente ira fazer isso em uma região muito grande, inviabilizando os resultados. Segundo motivo seria que devido à automatização dos resultados, as pequenas áreas de dúvidas que possam ser descartadas, seriam descartadas em todas as lesões, mas que provavelmente não atuam sobre grande parte da imagem.

Assim, é mais interessante reconhecer aquilo que se tem certeza de lesão, ou seja, incrementar a rigorosidade da saída da RBFN. Temos por rigorosidade, o nível de certeza da saída da RBFN, ou seja, aumentar a rigorosidade condiz com o fato de aceitar apenas valores de certeza maiores que 80%, ao invés de aceitar acima de 60%, por exemplo. Mesmo que 60% já demonstraria uma tendência a classe de não lesão pela RBFN,.

Este propósito levou ao pensamento de aumentar o valor de β , visto na Equação (2.7) que pode ser encontrada na página 38 para os casos de teste deste trabalho. Porém, não seria interessante encontrar um valor de β fixo, que tivesse que variar caso o conjunto de testes fosse modificado, ou seja, evitando que seja empírico. Com isto, a solução encontrada foi uma modificação, utilizando os próprios valores já presentes nas equações descritas, tal que, pudesse fazer este estreitamento da curva, gerando uma maior

rigorosidade.

Desta forma, utilizando como base a Equação (2.7), mas com o intuito de estreitar a curva Gaussiana, chegamos à equação utilizada para a curva Gaussiana deste trabalho representada pela Equação (4.1):

$$f(x) = e^{-\frac{(x-\mu)^2}{\gamma^3}} \quad (4.1)$$

Onde x continua representando a entrada, μ como o centro do cluster e γ sendo o valor da variância. Com isto, a saída da RBFN passa a ser mais rigorosa, tendo um valor encontrado dentro da própria rede como limiar.

Este estreitamento foi necessário para melhorar o desempenho da RBFN com dados de entrada muito próximos. Da forma como foram passados à árvore de decisão, ou seja, somente os valores de RGB não foram suficientes para a determinação de um resultado satisfatório.

Assim, a RBFN deste trabalho foi constituída com a seguinte arquitetura:

- Sete valores no vetor de entrada.
- Dez neurônios na camada de neurônios.
- Um nó na camada de saída.

Tendo somente a alteração da função Gaussiana como diferencial em comparação a RBFN comum. O conjunto de dados utilizados como entrada da RBFN foram constituídos de valores de região e a diferença do pixel local. Os sete os valores passados para a rede, foram divididos em três grupos.

Os três primeiros valores representavam as médias dos valores de RGB de uma região num raio de três pixels do pixel sendo visitado, e os três valores posteriores sendo a diferença do pixel visitado para esta média. O último valor foi utilizado como saída esperada no processo de aprendizado supervisionado.

Como exemplos dos valores de entrada temos:

117, 73, 91, -5, -10, -40, falso;

116, 72, 90, -1, -5, -36, falso;

116, 83, 98, 23, 65, 32, verdadeiro;

115, 83, 98, 22, 65, 31, verdadeiro;

115, 84, 99, 26, 71 36, verdadeiro;

A utilização dos valores por área se faz interessante pelo acompanhamento da visão humana, onde não é fácil descobrir se um único ponto faz parte de uma região ou não, assim, é analisado toda a região para esta determinação. Para que a segmentação pudesse ser feita a nível de pixel, foram adicionados os valores de diferenças.

Assim, os três primeiros valores indicam se o valor pertence a lesão e os três últimos indicam se aquela região consegue descrever o comportamento daquele pixel sendo analisado.

O valores de verdadeiro e falso, foram substituídos pelos valores 1 e -1 respectivamente, durante o processo de aprendizado.

5 RESULTADOS

Os resultados encontrados neste trabalho variam de acordo com a categoria da técnica utilizada, pois para a energia da cor houve o cálculo da semente para ser utilizada como entrada em algoritmos de crescimento de regiões.

5.1 Energia da Cor

De acordo com Stockburger [50], durante o processo de construção de uma tabela de frequências é necessário encontrar um intervalo que corresponda ao dado o qual se têm interesse, neste caso, intervalos que serão considerados parte da lesão ou não. Para estes intervalos foram usadas as imagens marcadas pelos médicos, já que elas tinham a indicação do que seria considerado lesão, da forma na qual o algoritmo buscaria indicar.

Tabela 1 – Regiões de lesão encontradas.

Número do Arquivo de Imagem	Tamanhos das Regiões				
	2%	4%	6%	8%	10%
1	154	35	15	4	2
2	214	41	15	7	2
3	275	59	20	11	1
4	11	5	2	1	0
5	38	3	2	1	0
6	38	4	1	0	0
7	65	6	3	1	0
8	59	12	7	2	0
9	12	3	1	0	0
10	20	4	2	1	0
11	6	2	1	0	0
12	5	2	1	0	0
13	5	2	1	0	0
14	80	29	12	7	2
15	10	4	2	1	0
16	100	24	12	3	1
17	71	15	6	3	1
18	64	7	4	1	0
19	205	41	17	9	4
20	35	7	5	2	1

21	124	25	7	6	1
22	42	12	4	3	1
23	276	52	25	5	1
24	170	39	14	3	2
25	46	8	5	4	0
26	111	37	25	9	3
27	12	5	3	1	0
28	109	15	11	8	2
29	132	20	10	5	3
30	192	30	17	6	3
31	473	117	39	15	6
32	28	5	1	0	0
33	16	4	2	1	0

Na Tabela 1 encontram-se o número de regiões que podem ser encontradas no topo das listas para diferentes tamanhos de janelas, denotando que a estratégia apresentada consegue encontrar ao menos uma região de lesão contanto que tenha tamanho suficiente para se encaixar dentro da lesão, ou seja, o primeiro elemento da lista ordenada de energias é uma região que poderia ser usada como semente pra crescimento de regiões. É importante lembrar que as regiões de cálculo são retangulares, assim algumas curvas da lesão ou lesões estreitas fazem com que uma região retangular não consiga se encaixar perfeitamente dentro da área de lesão.

Usando 2% da largura e altura foi possível gerar 2812 regiões, 625 foram geradas com 4%, 286 regiões de 6%, 155 de 8% e 102 foram geradas com 10%. Alguns pixels podem ter sido usados em mais de uma região graças à aproximação.

Com isto, algumas porcentagens não cabem completamente dentro da lesão, estas regiões ainda poderiam ser usadas dependendo do erro que seria considerado aceitável no crescimento de regiões. No presente trabalho, preferiu-se não trabalhar com falsos positivos, ou seja, melhor uma região menor que caiba completamente na lesão do que uma maior que ultrapasse a linha demarcada. A justificativa para esta escolha é de que a área de pele saudável toma uma parte muito grande da imagem e se usada como semente, provavelmente o algoritmo de crescimento de regiões cresceria demais e tomaria muitas partes da imagem externas à lesão.

A Tabela 2 indica o tempo gasto com a fase de divisões da imagem em regiões e calcular as energias, esses cálculos de tempo de execução foram usados para indicar as melhores abordagens para cada caso, assim, para casos onde as imagens contém pequenas

Tabela 2 – Tempos de execução por tamanho de região.

Tamanho da Região	Tempo de Execução
2%	161"52'
4%	39"08'
6%	16"46'
8%	9"6'
10%	6"9'

lesões seria necessária a divisão em pequenas regiões, porém o processo toma muito mais tempo que com regiões maiores. Os resultados mostrados pela Tabela 2 são referentes aos procedimentos para a execução em todas as imagens do experimento em cada uma das porcentagens de tamanho.

Nos casos apresentado neste trabalho, seriam possíveis usar valores médios, tal como 6% que podem assegurar ao menos uma região presente dentro da lesão em todos os testes e ainda possível reduzir o tempo de processamento em quase 90%. Outro fator que deve ser levado em consideração é quanta informação deve ser passada para cada algoritmo de crescimento. Ou seja, os algoritmos podem exigir regiões maiores na busca por padrões, mas também alguns deles apenas precisam de um único ponto, neste quesito as regiões podem ser divididas de duas formas:

- Pequenas regiões (2%): Estes tipos terão o mesmo problema de abordagens de um único pixel, pode não haver informação suficiente para descrever com precisão a lesão por completo para uma boa segmentação. Ainda assim, como pode ser visto na Tabela 2, teria um grande tempo de execução para a divisão da imagem em muitas pequenas regiões.
- Grandes regiões (10%): Nestes casos, algumas lesões podem não serem grandes o suficiente para preencher a região sem tocar a área externa da lesão, assim a região, mesmo com maior energia, terá partes que não pertencem à lesão. Assim, mesmo com tempo de execução menores, a abordagem não seria a melhor escolha, pois iria causar um erro muito grande graças à inserção do ruído externo como semente.

5.2 Árvore De Decisão

Para melhor visualização, os resultados obtidos com a árvore de decisão estão representados em forma de figuras, o Anexo A deste trabalho possui as informações em forma de tabelas com a numeração precisa. Lembrando que os índices não representados nas figuras e tabelas, foram escolhidos para a montagem da árvore.

A escolha das imagens que foram utilizadas na montagem da árvore foi feita com fases diferentes da lesão e pacientes diferentes. Foram usadas pelo menos uma imagem de cada um dos oito pacientes e em etapas diferentes do estado da lesão.

Os valores foram ordenados por verdadeiros positivos para o melhor acompanhamento da relação de acertos do algoritmo para cada imagem, representadas pelos índices, que também podem ser vistos na Tabela 1.

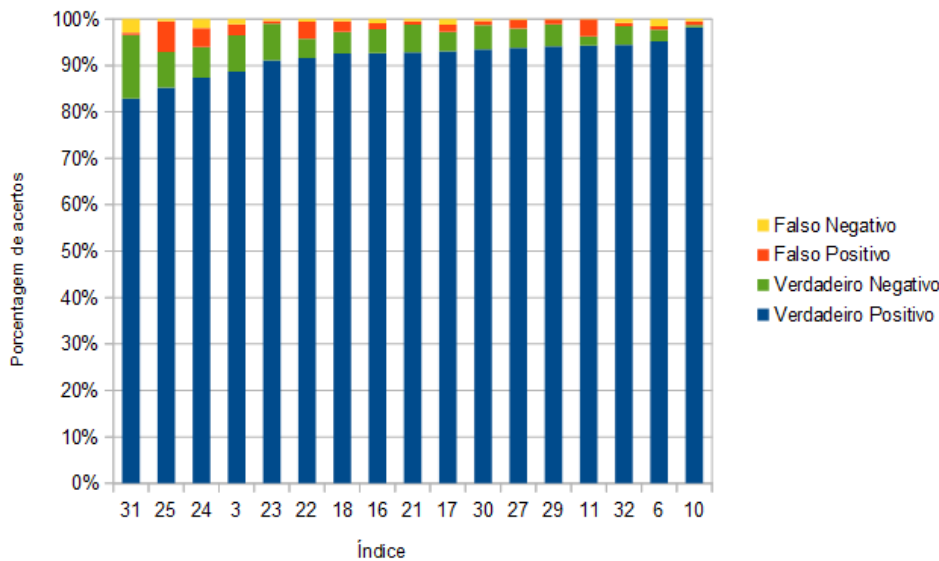


Figura 11 – Porcentagem de reconhecimento pela árvore.

Os indicadores da Figura 11 são distribuídos da seguinte forma:

- **Verdadeiro Positivo:** Aquilo que o algoritmo reconheceu como fundo e é fundo;
- **Falso Positivo:** O que foi tido pelo algoritmo como fundo mas era parte da lesão;
- **Falso Negativo:** O que foi tido pelo algoritmo como lesão mas era fundo;
- **Verdadeiro Negativo:** Aquilo que foi dado pelo algoritmo como lesão e era mesmo lesão;
- **Verdadeiro:** Soma dos verdadeiros positivos e verdadeiros negativos;
- **Falso:** Soma dos falsos positivos e falsos negativos;

Desta forma, em média, 97,20% dos pixels de cada imagem foi classificado da forma correta de fundo e objeto de interesse. Representando uma acurácia média de 97,21% e precisão média de 97,91%.

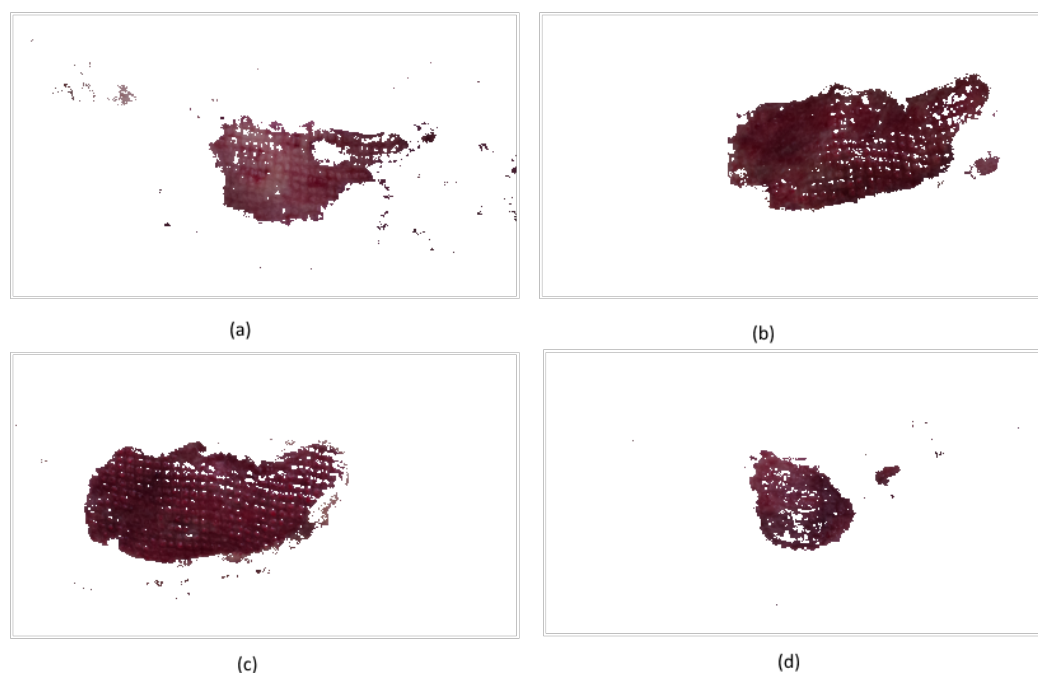


Figura 12 – Segmentação feita pelo algoritmo J48.

Na Figura 12, temos visualmente quatro exemplos de segmentações feitas com a árvore de decisão. Onde as Figuras 12 (b) e (c) são do mesmo paciente em tempos diferentes de aquisição. As Figuras 12 (a) e (d) são dois outros pacientes.

As principais desvantagens encontradas na árvore de decisão podem ser descritas em duas formas:

- **Tempo de Construção:** Com o número elevado de entradas para a construção da árvore, esta levou um tempo considerável para ser montada. Já que além da construção existe o processo de poda que elimina testes desnecessários, como visto na Seção 2.3.1.1. O processo de leitura do arquivo de entrada até o fim do processo de podas levou 37'13' (trinta e sete minutos e treze segundos).
- **Adição de Informação:** O sistema de construção da árvore não possibilita a inserção de informação puramente dito. Caso queiramos inserir informações de uma nova imagem para o aprendizado na árvore existente, ela deve ser desfeita e reconstruída, inserindo novamente a base usada anteriormente, para então adicionar a nova informação.

Já entre as vantagens da árvore, podemos explicitar:

- **Conjunto de Informação Simples:** A árvore de decisão conseguiu gerar resultados aceitáveis e interessantes mesmo com um conjunto de características bastante simples, necessitando apenas dos recursos de um único pixel e ainda de características simples de serem adquiridas do mesmo.
- **Tempo de Execução:** Assintoticamente a complexidade da descida da árvore para execução de um conjunto de dados é $O(n \log n)$, que ainda não é tão eficiente quanto estratégias $O(n)$, mas, como dito anteriormente, o conjunto de características usadas por ela pode ser bastante simples, o que tornou o tempo de execução bastante rápido.

5.3 Rede Neural Artificial

Os primeiros resultados da RBFN foram obtidos com a função de Gaussiana tradicional e treinamento com valores de RGB como da árvore de decisão. Resultados estes que não foram considerados satisfatórios com apenas 38% de acerto.

Neste momento, o limiar de aceitação estava fixado em 50% de certeza da saída da RBFN. E alterações do grau de certeza não geraram melhoras significativas na segmentação. A Figura 13 traz um exemplo do resultado da segmentação da RBFN nestas condições.



Figura 13 – Primeiro resultado obtido com a RBFN.

A primeira alteração foi a do conjunto de dados, passando então, para as características descritas na Seção 4.2.2, que conseguiram uma melhora considerável, obtendo 84% de acertos que também não variaram muito com alteração no grau de certeza.

A Figura 14 traz um exemplo de segmentação utilizando a RBFN após esta alteração.

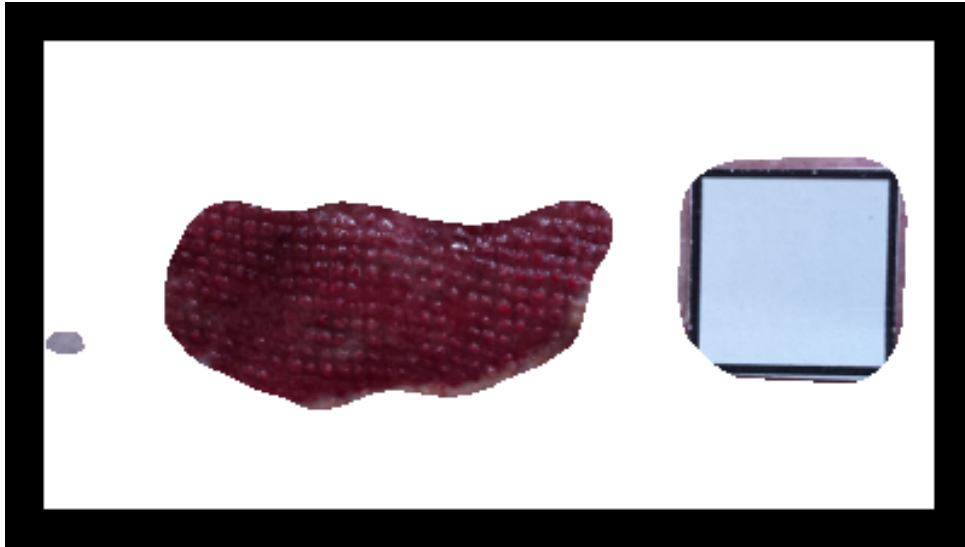


Figura 14 – Resultado da RBFN após a modificação do conjunto de dados.

Por fim, foi efetuada a modificação na função Gaussiana, como descrita na Seção 4.2.2, expressa pela Equação 4.1, página 50. Que foram os resultados que se mostraram gradativamente melhores com variações no grau de confiança e que então foram considerados satisfatórios.

Assim como na árvore de decisão, os resultados serão expressos em forma de figura. As mesmas imagens usadas na construção da árvore foram usadas no treinamento da RBFN para efeitos de comparação. Além do fato de que as imagens tinham sido escolhidas para representar todo o conjunto de dados necessários.

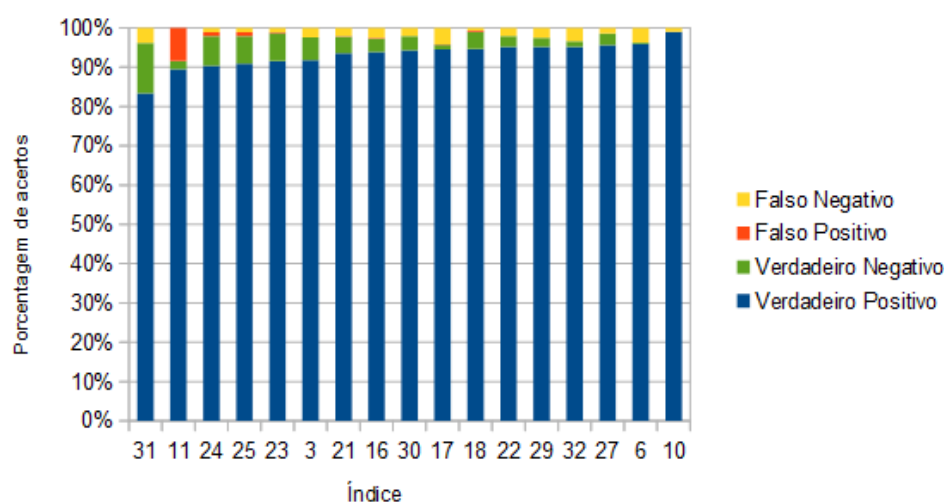


Figura 15 – Porcentagem de reconhecimento pela RBFN.

Foram mantidos os indicadores e a ordenação da Figura 11.

Assim como descritos nos experimentos da seção anterior, a RBFN foi usada de

maneira mais restritiva, o que aumentaria o resultado de falsos negativos. Porém os falsos positivos tiveram uma diminuição pelo fato de não haver buracos na segmentação da lesão, já que a RBFN manteve descritores voltados à região também, não só por pixel. A acurácia média da RBFN foi de 95,51% e sua precisão média foi de 97,49%.

Foi usada a base com regiões também na árvore de decisão, mas os resultados não foram tão bons quanto voltados à pixels. Tal como, a RBFN também foi feita com apenas pixels, mas que não geram resultados satisfatórios. Assim, este trabalho focou nos melhores resultados e as justificativas desses resultados.

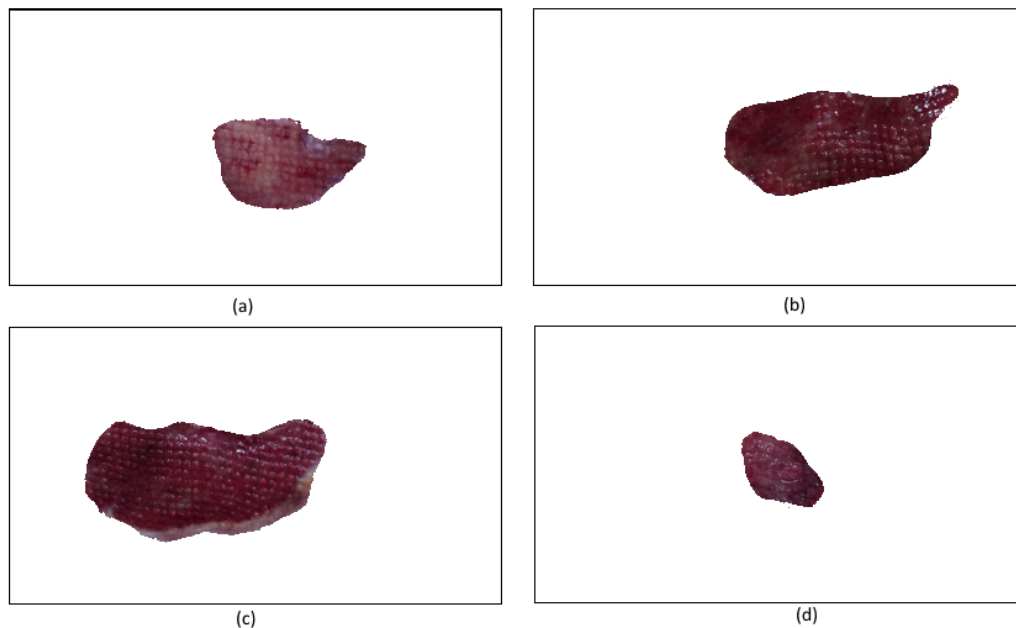


Figura 16 – Segmentação feita com a rede neural RBFN.

Visualmente as segmentações representadas nas Figura 16 (a), (b), (c) e (d). A visualização demonstra o fator de região empregado na RBFN, assim, a segmentação tende a não criar ruídos externos e nem picos, onde o fator de confiança nos resultados da RNA geram as curvas. Os resultados mostrados são para resultados da RNA, onde a confiança na saída da rede é maior que 80%, ou seja, o valor de saída da rede tem que ser maior ou igual à 80% da possibilidade de saída. Para tal, uma primeira busca na imagem foi feita para determinar as margens de saída da rede e foi usado como medida dos 0% e 100%.

Das desvantagens da RBFN, podemos citar:

- **Arquitetura Ideal Não Intuitiva:** É difícil definir uma arquitetura ideal para a RBFN em um conjunto de dados tão grande e complexo. A arquitetura utilizada foi tida como satisfatória e testes com algumas outras arquiteturas geram resultados que obtiveram níveis inferiores de acurácia e gastos de tempo superiores. Alguns casos

ainda causaram erros de execução e suas características nem ao menos puderam ser mensuradas. Podemos comparar esta situação com a árvore de decisão, onde a estrutura da árvore é diretamente gerada pelo algoritmo.

- **Limiar Inconstante:** O limiar para a determinação do grau de certeza pode variar com o conjunto de imagens de teste, por este motivo, este trabalho não contém a forma numérica deste valor. O valor é obtido com o conjunto de testes após o treinamento e provavelmente irão modificar o valor que seria usado como limiar de aceitação de um determinado conjunto de dados.

Entre as vantagens temos:

- **Adição de informação:** Em comparação direta com a árvore de decisão, uma inserção de informação na RBFN pode ser feita sem grandes problemas, seria como uma continuação do treinamento. Todos os treinamentos que já tiverem sido realizados já deixaram suas marcas na rede e não precisariam serem refeitos.
- **Tempo de Construção:** Tendo complexidade $O(n)$ em sua construção e tendo passos de cálculos matemáticos, a RBFN consegue uma montagem bastante eficiente, levando 53' (cinquenta e três segundos) desde a leitura dos dados até a completude de seu treinamento.

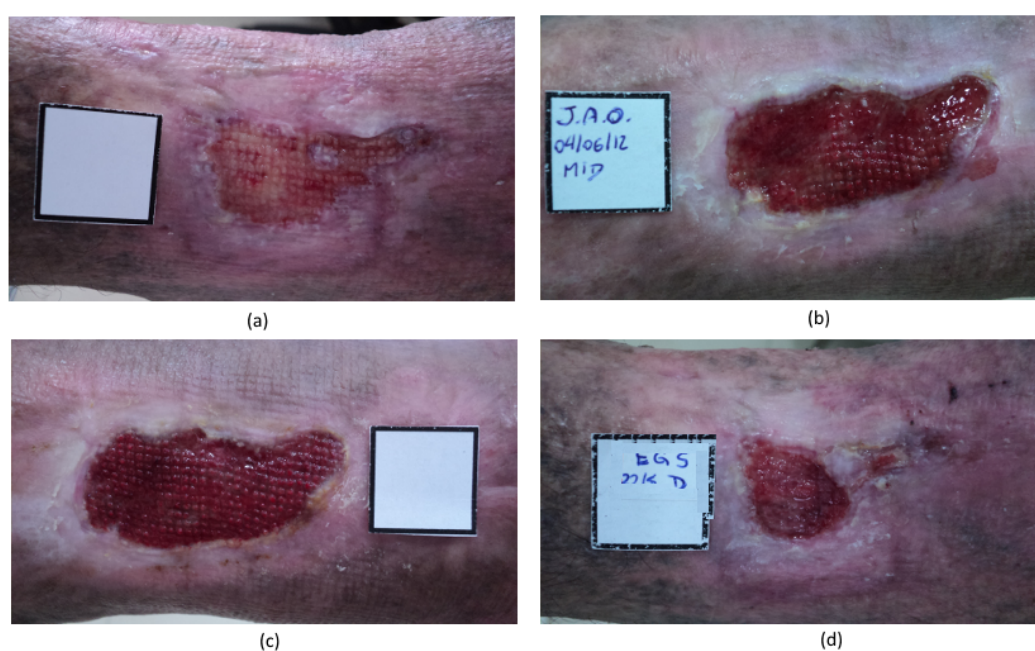


Figura 17 – Imagens originais usadas nos exemplos de J48 e RBFN.

As Figuras 17 (a), (b), (c) e (d) representam as imagens originais das mostradas segmentadas para uma comparação visual do resultado. As imagens foram as mesmas usadas na geração das Figuras 12 e 16.

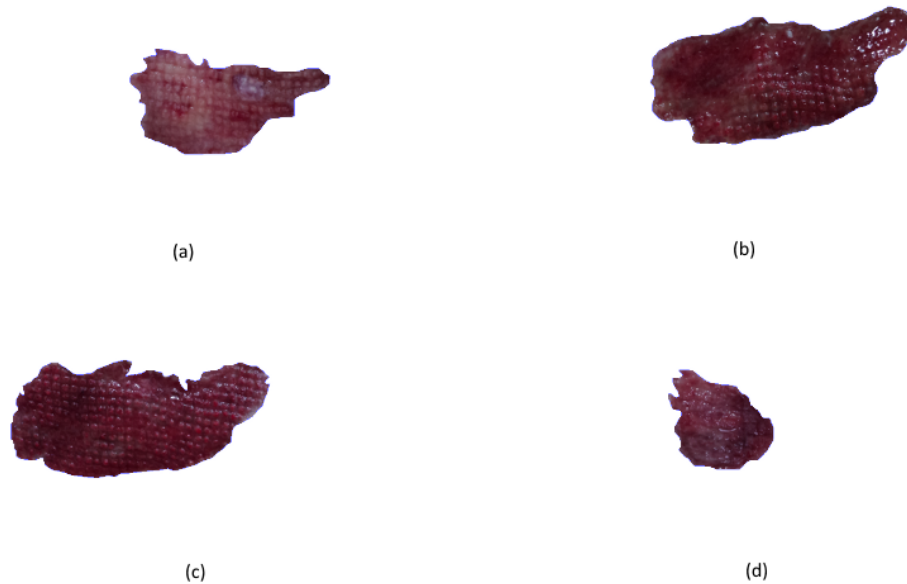


Figura 18 – Imagens segmentadas utilizando as marcações dos especialistas.

A Figura 18 traz as segmentações das imagens mostradas anteriormente nos casos em que receberiam 100% de acertos, ou seja, elas foram obtidas através da marcação dos médicos, para fins de comparação daquilo que se esperava da segmentação dos métodos utilizados.

6 CONCLUSÃO

Neste trabalho abordamos diferentes técnicas e formas de encontrar regiões de lesão de pele, causadas por insuficiência venosa, em imagens digitais.

Entre estas técnicas, tivemos uma vinda de processamento de sinais, para encontrar descritores eficazes em tal busca. Técnicas de processamento de sinais tratam imagens de maneira puramente numérica, não necessariamente se preocupando com o que o sinal representa, assim, conseguimos resultados não somente eficazes, mas também, bastante eficientes.

Como alternativa, buscamos experimentos com algoritmos de aprendizado de máquina, que já se demonstraram capazes de trabalhar em cenários de reconhecimentos de padrões e clusterização, que impactam diretamente sem segmentação de imagens. Nossa busca foi então a de experimentar arquiteturas, descritores, algoritmos, de forma que possamos motivar pesquisas cada vez mais aprofundadas de aprendizado de máquina com função de segmentação de imagens.

Além disso, ainda pudemos trazer uma alteração na já reconhecida rede neural artificial RBFN, de forma a viabilizar resultados satisfatórios no processo de segmentação. Ainda trouxemos a comparação desta alternativa com outro paradigma de aprendizado, da árvore de decisão.

Com processamento de sinais percebemos que é possível encontrar uma região de lesão baseado na diferença entre as energias dos canais de cor, mesmo com tamanhos diferentes, evitando assim, problemas com performance. Os tamanhos alternativos de região podem ajudar na aplicação da técnica para trabalhos diferentes, inclusive em variações de análises médicas, reduzindo o gasto de tempo e demonstrando que a técnica não necessariamente precisa se manter atrelada a um tipo de lesão ou problema.

Portanto, obtendo resultados satisfatórios, ao contrário de alguns casos, vistos na literatura, que não demonstraram bons resultados graças à sementes inapropriadas para a abordagem de crescimento de regiões. Como o propósito das imagens é a identificação da área de lesão, é bastante provável que a lesão tenha tamanho relativamente grande dentro da imagem, sendo assim, não há a necessidade de dividir essa imagem em regiões extremamente pequenas para o cálculo das energias.

Importante notar que o trabalho proposto não explicita qual o algoritmo de crescimento de regiões que deve ser usado. A pesquisa apresenta como encontrar uma boa semente para a segmentação que é um problema encontrado em alguns dos algoritmos e é o dado de entrada para outros. Ohta et al. [51], Ohlander et al. [52], Kurugollu et al. [53], Celenk [54], Tremeau e Borel [55], Cheng e Sun [56] são exemplos de trabalhos que

contém algoritmos de segmentação que usam a técnica de crescimento de regiões e tem dependência de uma escolha adequada da semente de segmentação.

Portanto, o desenvolvedor pode usar algoritmos conhecidos e já implementados ou propor uma nova solução usando os dados que podem ser extraídos da região encontrada. Estas características fazem com que mesmo pessoas que não fazem parte do contexto de ciência da computação possam usar esta estratégia. Assim a segmentação pode ser feita por aplicações que já tenham métodos de crescimento de regiões implementados.

Como trabalho futuro, é possível analisar as energias, porém de formas diferentes, na busca de outras áreas que possam identificar fases diferentes das doenças. Assim, a análise pode incluir outras características da lesão, tornando a análise mais eficaz. Outro trabalho que pode ser realizado, é baseado em características específicas de um determinado algoritmo de crescimento de regiões, na tentativa de obter alguma informação diferenciada que possa ser relevante para análises da lesão, ou seja, a busca de uma informação que faria o algoritmo ter melhores resultados ou mesmo uma execução mais rápida.

Analogamente, o aprendizado de máquina se mostrou um alternativa bastante eficaz para o problema de segmentação. No caso específico da árvore de decisão, algumas etapas de pós processamento podem ser incorporadas para refinar a saída. Como exemplo, poderia ser usado um algoritmo de preenchimento de regiões pequenas para evitar buracos dentro da área de lesão. Pensando ainda nos tamanhos de regiões, também pode ser usado um algoritmo que elimina áreas desconexas muito pequenas que poderiam ser consideradas ruídos.

Tanto para as árvores de decisão como para as RNAs, um bom conjunto de entrada é crucial, as características informadas aos métodos de aprendizado de máquina deste trabalho, constituem-se de informações simples que podem ser intuitivas, estudos mais aprofundados talvez possam mostrar melhores resultados.

A mudança na variância da Gaussiana para o estreitamento da curva se mostrou importante para o a melhoria no desempenho da segmentação, chegando a atingir 99,05% de reconhecimento correto que era de aproximadamente 84% da RBFN sem a alteração, enquanto a árvore de decisão conseguiu atingir até 98,99%.

Alguns estudos deverão ser aprofundados para RNA, para evitar a forma arredondada de bordas, esta resolução pode causar problemas em lesões de morfologias diferentes das estudadas. A saída da RNA com um grau de certeza mais elevado pode ainda ser usado como uma entrada de crescimento de regiões como foi usada a medida das energias.

Também podem ser usados modelos híbridos onde, por exemplo, ao invés de usar algoritmo de preenchimento de regiões, poderia ser usada uma rede neural para suprir um necessidade vista pelas árvores de decisão.

Além disso, outros métodos de aprendizado de máquina podem ser usados com a mesma constituição da base de dados, assim, havendo meios de compará-los e até identificar o melhor para cada etapa do processo.

Por último, ainda podem ser feitos estudos com quaisquer umas das técnicas apresentadas na tentativas de separação das áreas diferentes da lesão, como presença de infecção, área epitelizada, necrose, ou qualquer outra área que possa constituir a lesão.

Esperamos que este trabalho possa auxiliar outros pesquisadores de diferentes áreas de como usar estas técnicas nas resoluções de seus respectivos problemas de segmentação. Esperamos também, que se possa dar continuidade com estes estudos e que estes resultados possam sempre ser melhorados. E principalmente que este trabalho possa, brevemente, levar a aplicações que consigam auxiliar as equipes médicas e, consequentemente, beneficiar a comunidade e alavancar mais pesquisas que possam trazer ainda mais benefícios a comunidade e a medicina.

REFERÊNCIAS

- [1] WEENIG, R. H. et al. Skin ulcers misdiagnosed as pyoderma gangrenosum. *The New England journal of medicine*, v. 347, n. 18, p. 1412–1418, 2002.
- [2] FRYKBERG, R. G. Diabetic foot ulcers: Current concepts. *The Journal of Foot and Ankle Surgery*, v. 37, n. 5, p. 440 – 446, 1998. ISSN 1067-2516.
- [3] WERNICK, M. et al. Machine learning in medical imaging. *Signal Processing Magazine, IEEE*, v. 27, n. 4, p. 25–38, 2010.
- [4] KEKRE, H. B.; SARODE, T. K.; RAUL, B. C. Color image segmentation using kekre’s fast codebook generation algorithm based on energy ordering concept. In: *Proceedings of the International Conference on Advances in Computing, Communication and Control*. New York, NY, USA: ACM, 2009. (ICAC3 '09), p. 357–362. ISBN 978-1-60558-351-8.
- [5] FU, K. et al. Superpixel based color contrast and color distribution driven salient object detection. *Sig. Proc.: Image Comm.*, v. 28, n. 10, p. 1448–1463, 2013.
- [6] CHENG, H.-D. et al. Color image segmentation: advances and prospects. *Pattern Recognition*, v. 34, n. 12, p. 2259–2281, 2001.
- [7] BHATTACHARYYA, S. A brief survey of color image preprocessing and segmentation techniques. *Journal of Pattern Recognition Research*, v. 1, p. 120–129, 2011.
- [8] BI, L. et al. Multidimensional analysis for traditional chinese medicine diagnosis and treatment on hepatitis diseases. In: *International Conference on e-Health Networking, Applications and Services*. [S.l.: s.n.], 2012. p. 52–56. ISBN 978-1-4577-2039-0.
- [9] YANG, M. et al. Classification of retinal image for automatic cataract detection. In: *International Conference on e-Health Networking, Applications and Services*. [S.l.: s.n.], 2013. p. 674–679. ISBN 978-1-4673-5800-2.
- [10] ARIBI, Y.; WALI, A.; ALIM, A. M. An intelligent system for renal segmentation. In: *International Conference on e-Health Networking, Applications and Services*. [S.l.: s.n.], 2013. p. 11–15. ISBN 978-1-4673-5800-2.
- [11] BOCHKO, V. et al. Lower extremity ulcer image segmentation of visual and near-infrared imagery. *Skin Research and Technology*, Blackwell Publishing Ltd, v. 16, n. 2, p. 190–197, 2010. ISSN 1600-0846.
- [12] TRABELSI, O. et al. Skin disease analysis and tracking based on image segmentation. In: *Electrical Engineering and Software Applications (ICEESA), 2013 International Conference on*. [S.l.: s.n.], 2013. p. 1–7.
- [13] AZEVEDO-MARQUES, P. M. et al. Segmentation of dermatological ulcers using clustering of color components. In: *CCECE*. [S.l.]: IEEE, 2013. p. 1–4. ISBN 978-1-4799-0031-2.

- [14] JACQUELIN, C.; AURENGO, A.; HEJBLUM, G. Evolving descriptors for texture segmentation. *Pattern Recognition*, v. 30, n. 7, p. 1069–1079, 1997.
- [15] HOUHOU, N.; THIRAN, J.; BRESSON, X. Fast texture segmentation based on semi-local region descriptor and active contour. *Numerical Mathematics: Theory, Methods and Applications*, v. 2(4), p. 445–468, 2009.
- [16] QING, X.; JIE, Y.; SIYI, D. Texture segmentation using lbp embedded region competition. *Electronic Letters on Computer Vision and Image Analysis*, v. 5(1), p. 41–47, 2005.
- [17] ILEA, D. E.; WHELAN, P. F. Image segmentation based on the integration of colour-texture descriptors - a review. *Pattern Recognition*, v. 44, n. 10-11, p. 2479–2501, 2011.
- [18] BELONGIE, S. et al. Color- and texture-based image segmentation using em and its application to content-based image retrieval. In: *Proceedings of the Sixth International Conference on Computer Vision*. Washington, DC, USA: IEEE Computer Society, 1998. (ICCV '98), p. 675–. ISBN 81-7319-221-9.
- [19] MALIK, J. et al. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, v. 43, n. 1, p. 7–27, 2001.
- [20] XIAOHAN, Y.; YLä-JääSKI, J.; YUAN, B. A new algorithm for texture segmentation based on edge detection. *Pattern Recognition*, v. 24, n. 11, p. 1105–1112, 1991.
- [21] USó, A. M.; PLA, F.; GARCÍA-SEVILLA, P. A novel energy minimization criterion for color image segmentation. In: *ICPR (3)*. [S.l.: s.n.], 2004. p. 206–209.
- [22] AN, N.-Y.; PUN, C.-M. Color image segmentation using adaptive color quantization and multiresolution texture characterization. *Signal, Image and Video Processing*, Springer London, p. 1–12, maio 2012. ISSN 1863-1703.
- [23] CHEN, J. et al. Adaptive image segmentation based on color and texture. In: *ICIP (3)*. [S.l.: s.n.], 2002. p. 777–780.
- [24] SHAN, J.; CHENG, H.-D.; WANG, Y. A novel automatic seed point selection algorithm for breast ultrasound images. In: *ICPR*. [S.l.]: IEEE, 2008. p. 1–4. ISBN 978-1-4244-2175-6.
- [25] SMITH, S. M.; BRADY, J. M. Susan — a new approach to low level image processing. *International Journal of Computer Vision*, Kluwer Academic Publishers, v. 23, n. 1, p. 45–78, 1997. ISSN 0920-5691.
- [26] LIKAS, A.; VLASSIS, N.; VERBEEK, J. J. The global k-means clustering algorithm. *Pattern Recognition*, v. 36, n. 2, p. 451 – 461, 2003. ISSN 0031-3203. Biometrics.
- [27] FRIEDLAND, G.; JANTZ, K.; ROJAS, R. Siox: Simple interactive object extraction in still images. In: *ISM*. [S.l.]: IEEE Computer Society, 2005. p. 253–260. ISBN 0-7695-2489-3.

- [28] MA, J.; DUAN, F.; GUO, P. Improvement of texture image segmentation based on visual model. In: *Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), 2012 6th International Conference on*. [S.l.: s.n.], 2012. p. 151–154.
- [29] ZULIANG, A.; LIMIN, Z.; JIANTING, Z. A practical method of texture segmentation and transformation for radar image simulation. In: *Computer Science and Automation Engineering (CSAE), 2012 IEEE International Conference on*. [S.l.: s.n.], 2012. v. 3, p. 311–317.
- [30] BISHOP, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006. ISBN 0387310738.
- [31] SINGH, M.; SINGH, S. A machine learning approach for optimising image segmentation algorithms. In: *Intelligent Computer Communication and Processing (ICCP)*. [S.l.: s.n.], 2011. p. 261–265.
- [32] MITCHELL, T. M. *Machine Learning*. New York: McGraw Hill, 1997.
- [33] MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. *Foundations of Machine Learning*. [S.l.]: The MIT Press, 2012. ISBN 026201825X, 9780262018258.
- [34] MCCORMICK, C. *Radial Basis Function Network (RBFN) Tutorial, Computer Vision and Machine Learning Projects and Tutorials*. 2013. Disponível em: <<https://chrisjmcormick.wordpress.com/2013/08/15/radial-basis-function-network-rbfn-tutorial/>>.
- [35] QUINLAN, J. R. *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN 1558602402.
- [36] WITTEN, I. H.; FRANK, E. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005. ISBN 0120884070.
- [37] DUDA, R. O.; HART, P. E.; STORK, D. G. *Pattern Classification*. [S.l.]: Wiley-Interscience, 2001. ISBN 0-471-05669-3.
- [38] SILVA, L. N. d. C. *Análise e síntese de estratégias de aprendizado para redes neurais artificiais*. 232 f. Monografia (Mestrado) — Universidade Estadual de Campinas, Campinas, SP, 1998.
- [39] HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. [S.l.]: Prentice Hall, 1999.
- [40] BRAGA, A.; CARVALHO, A. C.; LUDERMIR, T. B. *Redes Neurais Artificiais: Teoria e aplicações*. LTC Editora, 2007. v. 2. ISBN 9788521615644. Disponível em: <<http://www.worldcat.org/isbn/9788521615644>>.
- [41] BROOMHEAD, D. S.; LOWE, D. Radial Basis Functions, Multi-Variable Functional Interpolation and Adaptive Networks. *Complex Systems*, v. 2, p. 321–355, mar. 1988.

- [42] PARK, J.; SANDBERG, I. W. Universal approximation using radial-basis-function networks. *Neural Comput.*, MIT Press, Cambridge, MA, USA, v. 3, n. 2, p. 246–257, jun. 1991. ISSN 0899-7667. Disponível em: <http://dx.doi.org/10.1162/neco.1991.3.2.246>.
- [43] RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. 3rd. ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2009. ISBN 0136042597, 9780136042594.
- [44] BOW, S.-T. *Pattern Recognition and Image Preprocessing*. [S.l.]: Dekker, 1992. (Electrical Engineering and Electronics, 77).
- [45] BISHOP, C. M. *Neural Networks for Pattern Recognition*. New York, NY, USA: Oxford University Press, Inc., 1995. ISBN 0198538642.
- [46] LIU, Y.-C.; CHAN, W.-H.; CHEN, Y.-Q. Automatic white balance for digital still camera. *IEEE Trans. on Consum. Electron.*, IEEE Press, Piscataway, NJ, USA, v. 41, n. 3, p. 460–466, ago. 1995. ISSN 0098-3063. Disponível em: <http://dx.doi.org/10.1109/30.468045>.
- [47] SERRANO, C.; ACHA, B. Pattern analysis of dermoscopic images based on markov random fields. *Pattern Recognition*, Elsevier, v. 42, n. 6, p. 1052–1057, 2009.
- [48] LITTMANN, E.; RITTER, H. Adaptive color segmentation—a comparison of neural and statistical methods. *Neural Networks, IEEE Transactions on*, IEEE, v. 8, n. 1, p. 175–185, 1997.
- [49] LOPES, F. M.; CONSULARO, L. A. A rbf perceptive model for image thresholding. In: IEEE. *Computer Graphics and Image Processing, 2005. SIBGRAPI 2005. 18th Brazilian Symposium on*. [S.l.], 2005. p. 225–232.
- [50] STOCKBURGER, D. *Introductory Statistics: Concepts, Models, and Applications, Second Edition*. 2nd. ed. [S.l.]: Atomic Dog Publishing, 2001. Paperback.
- [51] OHTA, Y.; KANADE, T.; SAKAI, T. Color information for region segmentation. *Computer Graphics and Image Processing*, v. 13, n. 3, p. 222 – 241, July 1980.
- [52] OHLANDER, R.; PRICE, K.; REDDY, D. R. Picture segmentation using a recursive region splitting method. *Computer Graphics and Image Processing*, v. 8, p. 313–333, 1978.
- [53] KURUGOLLU, F.; SANKUR, B.; HARMANCI, A. E. Color image segmentation using histogram multithresholding and fusion. *Image and Vision Computing*, v. 19, n. 13, p. 915–928, nov. 2001. ISSN 02628856. Disponível em: [http://dx.doi.org/10.1016/s0262-8856\(01\)00052-x](http://dx.doi.org/10.1016/s0262-8856(01)00052-x).
- [54] CELENK, M. A color clustering technique for image segmentation. *Computer Vision, Graphics, and Image Processing*, v. 52, n. 2, p. 145 – 170, 1990. ISSN 0734-189X. Disponível em: <http://www.sciencedirect.com/science/article/pii/0734189X9090052W>.

- [55] TREMEAU, A.; BOREL, N. A region growing and merging algorithm to color segmentation. *Pattern Recogn.*, Elsevier Science Inc., New York, NY, USA, v. 30, n. 7, p. 1191–1203, jul. 1997. ISSN 0031-3203. Disponível em: [http://dx.doi.org/10.1016/S0031-3203\(96\)00147-1](http://dx.doi.org/10.1016/S0031-3203(96)00147-1).
- [56] CHENG, H. D.; SUN, Y. A hierarchical approach to color image segmentation using homogeneity. *IEEE Transactions on Image Processing*, v. 9, p. 2071–2082, 2000.

Apêndices

APÊNDICE A – TABELAS DE RESULTADOS

No Apêndice A temos as tabelas com os valores expressos pelos gráficos do Capítulo 5. A Tabela 3 traz os valores detalhados do gráfico mostrado na Figura 11 que pode ser encontrada na páginas 56. Com a porcentagem em cada um dos indicadores de acerto do algoritmo J48.

Tabela 3 – Porcentagem de reconhecimento de pixel pela árvore.

Índice	Verdadeiro Positivo	Falso Positivo	Falso Negativo	Verdadeiro Negativo	Verdadeiro	Falso
3	88,65%	2,38%	1,12%	7,85%	96,50%	3,50%
6	95,23%	0,73%	1,58%	2,47%	97,70%	2,30%
10	98,20%	0,82%	0,51%	0,48%	98,67%	1,33%
11	94,33%	3,61%	0,12%	1,95%	96,27%	3,73%
16	92,69%	1,29%	0,93%	5,08%	97,78%	2,22%
17	93,06%	1,51%	1,25%	4,17%	97,24%	2,76%
18	92,60%	2,30%	0,47%	4,64%	97,24%	2,76%
21	92,86%	0,75%	0,50%	5,89%	98,75%	1,25%
22	91,59%	3,61%	0,59%	4,22%	95,81%	4,19%
23	91,15%	0,60%	0,41%	7,83%	98,99%	1,01%
24	87,38%	4,03%	1,98%	6,61%	93,99%	6,01%
25	85,29%	6,66%	0,45%	7,60%	92,89%	7,11%
27	93,80%	1,89%	0,14%	4,18%	97,98%	2,02%
29	94,18%	1,03%	0,08%	4,71%	98,89%	1,11%
30	93,45%	0,83%	0,44%	5,28%	98,73%	1,27%
31	82,88%	0,46%	2,92%	13,74%	96,62%	3,38%
32	94,44%	0,77%	0,78%	4,01%	98,45%	1,55%
Média	91,87%	1,96%	0,84%	5,34%	97,20%	2,80%

Assim como nas imagens, os indicadores das tabelas presentes neste Apêndice são os seguintes:

- **Verdadeiro Positivo:** Aquilo que o algoritmo reconheceu como fundo e é fundo;
- **Falso Positivo:** O que foi tido pelo algoritmo como fundo mas era parte da lesão;
- **Falso Negativo:** O que foi tido pelo algoritmo como lesão mas era fundo;
- **Verdadeiro Negativo:** Aquilo que foi dado pelo algoritmo como lesão e era mesmo lesão;
- **Verdadeiro:** Soma dos verdadeiros positivos e verdadeiros negativos;
- **Falso:** Soma dos falsos positivos e falsos negativos;

Na Tabela 4 os mesmos indicadores são usados, mas com o detalhamento dos valores usados na construção do gráfico presente na Figuras 15, página 59. Trazendo, então, o detalhamento da porcentagem de acertos obtidos com o algoritmo RBFN.

Tabela 4 – Porcentagem de reconhecimento pela rede neural.

Índice	Verdadeiro Positivo	Falso Positivo	Falso Negativo	Verdadeiro Negativo	Verdadeiro	Falso
3	91,80%	0,10%	2,37%	5,74%	97,53%	2,47%
6	96,38%	0%	2,41%	1,21%	97,59%	2,41%
10	95,08%	0%	1,87%	3,05%	98,13%	1,87%
11	95,96%	0%	3,73%	0,32%	96,27%	3,73%
16	94,65%	0,09%	3,03%	2,24%	96,88%	3,12%
17	97,37%	0%	2,42%	0,21%	97,58%	2,42%
18	89,52%	8,42%	0%	2,07%	91,58%	8,42%
21	96,65%	0%	1,98%	1,36%	98,02%	1,98%
22	94,53%	0,04%	4,21%	1,22%	95,75%	4,25%
23	94,69%	0,33%	0,62%	4,36%	99,05%	0,95%
24	93,58%	0%	5,48%	0,94%	94,52%	5,48%
25	95,19%	0%	2,04%	2,76%	97,96%	2,04%
27	95,66%	0,03%	1,38%	2,94%	98,60%	1,40%
29	94,48%	0,02%	2,26%	3,24%	97,72%	2,28%
30	95,19%	0,01%	2,50%	2,29%	97,48%	2,52%
31	94,28%	0%	2,11%	3,61%	97,89%	2,11%
32	94,33%	0,04%	2,82%	2,80%	97,13%	2,87%
Média	94,67%	0,53%	2,42%	2,37%	97,04%	2,96%

Os experimentos realizados para a construção das Tabelas 3 e 4 podem ser encontrados nas Seções 4.2.1 e 4.2.2 do Capítulo 4, que se inicia na página 45.

Anexos

ANEXO A – ARTIGO PUBLICADO PELO AUTOR

O artigo presente neste anexo foi um trabalho desenvolvido pelo autor, durante o programa de mestrado, em parceria com os professores Dr. Rodolfo Miranda de Barros, especialista em engenharia de software, Dr. Sylvio Barbon Jr. e Dr. Alan Salvany Felinto, gerentes do grupo de jogos, para a criação um processo de desenvolvimento direcionado a grupos acadêmicos de jogos.

O artigo foi publicado nos anais da 11^o Conferência de Computação Aplicada em Porto, Portugal.

O grupo alvo foi o grupo de jogos existente na Universidade Estadual de Londrina que com o seu crescimento rápido, precisou de uma carga de engenharia de software a qual não havia sido esperada.

Devido o fato de não fazer parte direta sobre o tema do trabalho proposto, o artigo segue em anexo.

PROCESSO DE DESENVOLVIMENTO DE SOFTWARE COM MELHORAMENTO CONTÍNUO DIRECIONADO À GRUPOS DE JOGOS ACADÊMICOS

1. Introdução:

O desenvolvimento de jogos pode ajudar os estudantes trazendo eles para o desenvolvimento de softwares complexos, assim, eles começam a lidar com tipos de problemas alguns softwares acadêmicos não possibilitam. Começando pelo fato que são usados conceitos de várias disciplinas, no caso dos jogos, várias disciplinas acabam sendo usadas porque jogos tem conceitos, implementações e conhecimentos ainda mais profundos de diferentes áreas da computação.

Usando a experiência recebida da fábrica de engenharia de software da universidade, chamada GAIA, foi criado um processo iterativo incremental que ajuda os estudantes. Assim, eles podem obter melhores resultados, aprendendo como manter-se usando e melhorando o processo de engenharia de software através de um passo de revisão de iteração.

É comum encontrar processos de desenvolvimento de jogos baseados em estruturas de metodologias ágeis [Larman 2003] [Rabin 2009] [Peres et al. 2011], a ideia de manter o produto atualizado e gerando conteúdo periodicamente, facilitando a análise de progresso. Além disso, esta metodologia reconhece que modificações podem ser necessárias e prepara-se para tais [Beck e Andres 2004] [Larman e Basili 2003] [Schwaber 2004].

Evitando planejamentos muito longos e evitando pensar muito à frente em algo que possa não acontecer. A aplicação de técnicas ágeis com práticas como controle de versão, padrões de código e montagens diárias de projeto tem se provado eficazes e eficientes nas construções do processos de desenvolvimento de jogos [Cockburn e Highsmith 2001]. Porém, ainda existe uma falta de especificidades técnicas para a visão acadêmica de desenvolvimento de jogos e como lidar com características que não fazem parte dos grupos comerciais.

Portanto nasceu a proposta de organizar um grupo de jogos dentro da universidade. O objetivo do artigo é criar um processo que possa organizar o desenvolvimento de jogos, assim, ajudando a produção de jogos enquanto mantêm a qualidade que é esperada de qualquer software e tendo um aprendizado colaborativo muito mais intenso durante todo o processo.

Na Seção 2, será visto que o processo construído com as ideias apresentadas. Os resultados do processo de aprendizado serão mostrados na Seção 3, seguido pelas Conclusões do artigo.

2. Processo de Desenvolvimento de Software:

O processo iterativo incremental proposto traz a ideia de mensurar a evolução, porque os pequenos passos ou incrementos mantêm sempre algo sendo feito, mas também podem ter controle para garantir que nada está sendo desenvolvido desnecessariamente. Este conceito, usado pelas metodologias ágeis, tem sugerem se preocupar com pequenos problemas primeiro, para prevenir que mais e maiores problemas ocorram, assim, pequenas porções de desenvolvimento são implementadas. O processo proposto precisa controlar toda a organização do desenvolvimento, mas também precisa garantir que o próprio processo esteja sendo atualizado, assim, todo desenvolvimento terá pontos de controle.

Para criara um processo que possa atingir as necessidades do grupo, mas também que possa ser flexível à futuras mudanças, foi usado como modelo, o processo de desenvolvimento de software da fábrica de engenharia de software GAIA. Baseado no processo descrito pela fábrica, foi possível gerar um processo que garante a qualidade e o controle procurados através de um processo que já foi utilizado e teve bons resultados. Esta técnica vista em [Góes e de Barros 2012], [Rautenberg et al. 2011], [Cakici e Boman 2011], [Simplerl 2009], [Brazhnik 2007], onde o processo é apresentado para dois grupos, especialistas e não especialistas, e tais, precisam responder uma série de questão, usando escalas, de acordo com a precisão do processo para vários pontos.

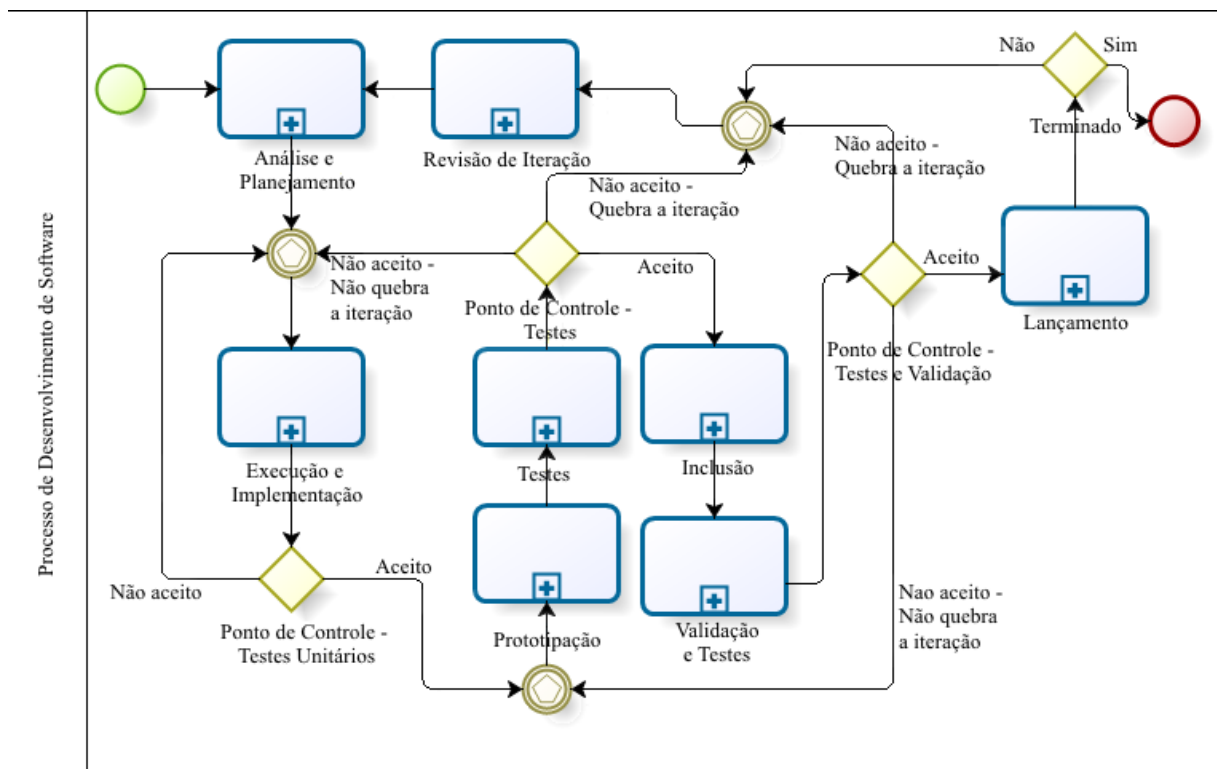


Figura 2.1. Diagrama do Processo

2.1 Análise e Planejamento:

Esta fase tem o poder de motivar o grupo, expondo as ideias iniciais e se abrindo para o surgimento de novas, então a equipe de desenvolvimento não ficará apenas responsável pela codificação mas também parte do projeto. Nesta fase, o escopo de projeto será finalmente decidido e será construído durante as iterações do processo. Graças à característica iterativa do processo, esta fase pode, e provavelmente será, executada várias vezes, em cada uma delas, será determinado o que será feito em um futuro imediato e quais devem ser as métricas para determinar que a iteração será dada como completa. Esta fase deve também conter informações dos testes à serem realizados e como devem ser suas aprovações.

2.2 Execução e Implementação:

A fase de implementação carrega o escopo, a estratégia e conclusões feitas na fase anterior e então desenvolve o que foi proposto. Todas as configurações, metodologias, ferramentas e padrões de desenvolvimento, não são especificados pelo processo, porque pode haver grande disparidade entre projetos. É importante que o processo funcione para todos os tipos de jogos, portanto deixando o gerenciamento decidir quais serão as melhores abordagens para cada projeto.

2.3 Prototipação:

A fase de prototipação é responsável por unir as unidades desenvolvidas do sistema em um ambiente maior e testável. É dependente do escopo do trecho à ser desenvolvido liberado pela fase de planejamento, então o grupo pode saber todo o conteúdo que a presente versão deve conter. O ambiente pode ser de pequenas iterações para desenvolvimento de cenários ou personagens, ou mesmo complexas, como um fase inteira.

2.4 Testes de Protótipo:

Testes de protótipo testam uniformemente todas as operações que devem ter sido implementadas na iteração vigente. Estes testes variam de acordo com a iteração, dependendo do foco da iteração.

2.5 Inclusão:

Inclui a iteração corrente no conteúdo produzido até agora. É importante sequenciar as iterações com um incremento que possa ser ajustável ao conteúdo preexistente. Caso não seja possível sempre criar conteúdo que seja ajustado ao já produzido, é recomendado que o conteúdo prévio possa ao menos disparar o que está sendo produzido, assim os testes podem simplesmente serem chamados e não “ter que fazer” o teste ser rodado.

2.6 Verificação e Validação:

Valida se a iteração foi incluída corretamente e que erros não foram gerados. Erros

em etapas prévias devem ser testados nesta fase também. É nesta fase, que as operações são autorizadas a serem executadas em porções que serão definidas como completas, portanto, importante prestar atenção nas documentações geradas previamente e modificações que possam ser feitas. Estas modificações também devem manter as documentações atualizadas.

2.7 Lançamento:

A fase de lançamento é caracterizada pela formação de um conteúdo testado e jogável, que devem conter as funcionalidades especificadas na fase de planejamento de todas as iterações realizadas até o momento. Esta fase também é responsável por determinar o que será feito na próxima iteração, como na fase inicial. A análise de próximo conteúdo também serve para controlar o que foi realizado.

2.8 Pontos de Controle:

A prototipação e validação irão determinar qual a próxima fase baseados na aceitação dos testes, não aceitação sem quebrar o fluxo normal da iteração ou a não aceitação e a quebra do fluxo da iteração. Esta decisão dependerá do tipo de problema encontrado durante a execução das fases. Estes pontos de controle serão responsáveis por apresentarem a atual situação do desenvolvimento dentro da iteração e da iteração dentro do projeto.

Gravidade	Descrição
Pequena	Todo erro que não comprometa, não impeça o uso do produto e que nenhum requerimento essencial esteja faltando, o procedimento não requer ação urgente e não requer que nenhuma parte do processo pare para consertar o erro. Dadas essas características, problemas relacionados à prioridades devem ser decididas à nível de gestão. Pode ser deixado para ser resolvido em futuras versões onde os recursos serão usados novamente por outra fase do desenvolvimento.
Grande	Erro que compromete, mas não impede o funcionamento do produto ou demonstra que requerimentos essenciais não estão totalmente desenvolvidos. Requer uma ação de conserto assim que possível.
Crítica	Erro que compromete, impede o uso e indica que um requerimento essencial não tenha sido produzido. Não pode ser resolvido com ações simples e requer tomada de decisão imediata.

Tabela 2.1. Classificação de erros de acordo com sua gravidade

A Tabela 2.1 mostra diferentes tipos de problemas quanto sua gravidade. Os pontos de controle durante o processo precisarão determinar o que será ou não considerado aceitável. Nos casos de não serem aceitos, se isso irá quebrar a iteração ou não. Esta informação deve ser considerada e deve fazer parte da fase de planejamento, pois irá variar baseado no projeto ou no tamanho da iteração corrente. Na grande maioria da vezes um

único erro crítico não será aceito e irá quebrar a iteração, porque o mesmo pode ser grande suficiente para demandar sua própria iteração.

Natureza	Descrição
Omissão	Falta ou ausência de seção, requerimento, característica ou algum outro item da documentação.
Violação	Inconsistência ou inexatidão de dois ou mais itens da documentação. Esta inconsistência pode ser referente à modificações futuras também podem refletir modificações em documentações prévias.
Imprecisão	Requerimento ou funcionalidade especificada de forma ambígua ou incompleta.
Estilo	Linguagem (natural ou de programação), formatação, entre outros.

Tabela 2.2. Classificação de erros de acordo com sua natureza

Outra referência de como os problemas podem ser classificados é de acordo com a sua natureza, mostrados na Tabela 2.2. A natureza não especificamente implica na profundidade do problema, mas pode ser usado e pode-se aprender com esta classificação. Pode também ser usado como um complemento, onde a natureza pode implicar e uma gravidade, ou a junção das duas implicar em uma outra característica ou forma de classificação para a tomada de alguma decisão, em outras palavras, natureza nem gravidade sozinhos seriam responsáveis por uma tomada de decisão, mas sim a junção dos dois.

2.9 Revisão de Iteração:

A revisão de iteração pode ser alcançada por três caminhos diferente, por pontos de controle de prototipação ou validação, ou pela fase de lançamento. Assim, o primeiro passo nesta fase é ter esta informação, ou seja, como ela foi alcançada. Na revisão, todas as lições que podem ser aprendidas da iteração corrente devem ser armazenadas.

Se todas as fases ocorrerem de forma correta e todo o conteúdo for produzido, é interessante armazenar a informação de como foi produzido, os recursos utilizados, tempo gasto, mensuração de testes, critérios e tudo que puder ajudar futuras iterações ou novos projetos. O mesmo pensamento pode ser usado para os casos onde a fase é alcançada por um ponto de controle, estes casos implicam que alguma coisa no meio do caminho deu errado e a iteração foi interrompida, assim a próxima iteração ainda terá algo a ser produzido que pertença a iteração anterior. Algumas decisões precisarão ser tomadas aqui, por exemplo, se será melhor começar a nova iteração resolvendo o problema ou continuar a implementação e voltar ao problema mais tarde? Que tipo de problemas podem acontecer e quando eles poderão ser resolvidos? Várias outras questões podem aparecer mas todas elas depende do grupo, do projeto e dos histórico que o processo já tiver gerado.

Baseado nesta base histórica, nas mudanças feitas por ela e na utilização de cada fase dos processo, o processo como um todo pode se adaptar, adicionando características

que podem ser importantes para o desenvolvimento, removendo ou alterando fases que não estejam sendo utilizadas completamente ou da maneira correta.

3. Resultados

A Tabela 3.1 mostra processos ou funções de cada parte das melhorias do processo, depois do uso em um projeto proposto.

Índice	Fase	Problema	Melhoria
1	Análise e Planejamento	Foi gasto muito tempo e reuniões para organizar o tempo disponível para cada tarefa.	Pode ser realizado com gasto menor de tempo.
2	Análise e Planejamento	Pontos de inspeção em pequena ou nenhuma produção.	Trazido um aluno para a gestão para controle do que foi desenvolvido.
3	Execução e Implementação	A junção de porções de código desenvolvidas por grupos diferentes tomaram muito tempo para serem unidos.	Redução significativa de erros de versionamento e regressão.
4	Execução e Implementação	A fase de planejamento não foi usada da maneira correta, decisões foram tomadas durante a implementação.	Melhor uso da fase de planejamento.
5	Testes Unitários	Gasto tempo em realizar testes de pouca relevância e que demandam muitos testes.	Dada a plataforma de desenvolvimento, não foi gasto tempo desnecessário.
6	Testes Unitários (Pontos de Controle)	O controle dos testes não estavam direcionados a código desenvolvido e sim a operações da <i>engine</i> de desenvolvimento.	Foi usado controle de compilação e conteúdo.
7	Prototipação	Muitos testes foram previstos para serem realizados nesta fase.	Foco em testes específicos para o protótipo.
8	Inclusão	Projetos contiveram muitas variações que dificultaram a inclusão e versionamento.	O uso de uma ferramenta de versionamento devem ser incluída no projeto. Junto com estratégias de padronização.
9	Revisão de Iteração	Problemas adiados podem ser adiados novamente se não tratados corretamente.	Relatórios de dificuldades, problemas e causas.

Tabala 3.1. Tabela de melhoramentos

Cada linha da Tabela 3 mostra um problema que foi resolvido e sua resolução armazenada como uma lição aprendida. Nem todas elas causam impacto direto no processo,

mas reforçaram ou confirmaram usos de algumas fases, tal como exibiram a necessidade de algo de alguma etapa.

A base histórica é uma forma de garantir que as lições aprendidas chegarão a todos os membros do grupo, como grupos de jogos tem uma grande variedade de pessoas e conhecimento, se torna interessante que qualquer informação dentro do grupo chegue qualquer novo integrante que possa ser incluído no meio do percurso, graças a facilidade de rotatividade por alunos no grupo pela característica acadêmica, assim reduzindo riscos de grupos muito novos ou inexperientes [de Barros et al. 2009].

Um grupo com as características previamente descritas desenvolveu um jogo usando o processo, com isso, foi possível criar as primeiras lições aprendidas que foram armazenadas pelo grupo na base de conhecimento, assim, possibilitando mudanças e remoções reduzindo os problemas encontrados. Com este estudo de caso, a natureza adaptativa do processo se provou importante e ajudou na evolução do processo, como visto na Seção 2.

4. Conclusão:

O processo tem características que permitem o desenvolvimento organizado de jogos de qualidade, aliado ao alto nível de aprendizado para os alunos, os preparando para trabalharem em um ambiente controlado. Como trabalhos futuros, o próprio processo deve ser adaptar e gerar sua própria versão mais refinadas voltadas a especificações de um grupo acadêmico de desenvolvimento de jogos. Outra proposta, seria a de adicionar mais funcionalidades de gerenciamento como gerenciamento de configurações ou voltadas a testes através de um analista de testes.

Mais do que eliminar a fase de testes unitários, a importância do estudo de caso foi de garantir que as outras fases estão sendo usadas da forma correta, especialmente, que elas são importantes para o processo. Projetos futuros terão a chance de buscar melhorias para o processo mas com um histórico daquilo que já tenha sido utilizado, o que já foi removido e a razões para isto, assim o processo pode evoluir tendo em mão os caminhos pelos quais não seguir.

Todas as melhorias mostradas na tabela vieram depois do uso do processo pelo estudo de caso, algumas modificações, outros foram reforços a alguma ideia, outros não foram fortes suficiente para alterações, mas podem gerar atenções ou modificações futuras, mas mais importante que todos vieram de experiências do grupo.

REFERÊNCIAS

- [1] BECK, K.; ANDRES, C. (2004). *Extreme Programming Explained: Embrace Change* (2nd Edition). Addison-Wesley, Boston.
- [2] BRAZHNIK, O. (2007). Databases and the geometry of knowledge. *Data & Knowledge Engineering*, 61(2):207–227.
- [3] CAKICI, B.; BOMAN, M. (2011). A workflow for software development within computational epidemiology. *J. Comput. Science*, 2(3):216–222.
- [4] COCKBURN, A.; HIGHSMITH, J. (2001). Agile software development: The people factor. *IEEE Computer*, 34(11):131–133.
- [5] de BARROS, R. L. B., et al (2009). Investigating the communication process in multidisciplinary game development teams. *Sistemas Colaborativos, Simpósio Brasileiro de*, 0:61–69.
- [6] GÓES, A. d. S.; de BARROS, R. M. (2012). Knowledge management in a software factory. In *XXXVIII Conferencia Latinoamericana En Informática (CLEI)*, Medellin, Colombia. CLEI, Universidad de Medellín, IEEE.
- [7] LARMAN, C. (2003). *Agile and Iterative Development: A Manager’s Guide*. Addison-Wesley Professional, 1st edition.
- [8] LARMAN, C.; BASILI, V. R. (2003). Iterative and incremental development: A brief history. *IEEE Computer*, 36(6):47–56.
- [9] PERES, A. L., et al (2011). Methods and processes definitions for multiplatform social network games development with distributed teams. In *SBGames*, pages 189–195. IEEE.
- [10] RABIN, S. (2009). *Introduction to Game Development*. Charles River Media, 2nd edition.
- [11] RAUTENBERG, S., et al (2011). Modelo de conhecimento para mapeamento de instrumentos da gestão do conhecimento e de agentes computacionais da engenharia do conhecimento baseado em ontologias. 16th.
- [12] SCHWABER, K. (2004). *Agile Project Management with Scrum*. Prentice Hall.
- [13] SIMPERL, E. P. B. (2009). Reusing ontologies on the semantic web: A feasibility study. *Data & Knowledge Engineering*, 68(10):905–925.

TRABALHOS PUBLICADOS PELO AUTOR

Trabalhos publicados pelo autor durante o programa.

1. SEIXAS Jr., J. L.; de BARROS, R. M.; BARBON Jr., S.; FELINTO, J. De C.; FELINTO, A. S., **Software development process with continuous improvement directed to academic gaming groups**, 11th Conference on Applied Computing, Oct. 2014, Porto, Portugal (Qualis CC 2012, B4)
2. IGAWA, R. A.; KIDO, G. S.; SEIXAS Jr., J. L.; BARBON Jr., S., **Adaptive Distribution of Vocabulary Frequencies: A Novel Estimation Suitable for Social Media Corpus**, Intelligent Systems (BRACIS), 2014 Brazilian Conference on , vol., no., pp.282,287, 18-22 Oct. 2014 (Qualis CC 2012, B2)
3. SEIXAS Jr., J. L.; BARBON Jr., S.; SIQUEIRA, C. M.; DIAS, I. F. L.; CASTALDIN, A. G.; FELINTO, A. S., **Color energy as a seed descriptor for image segmentation with region growing algorithms on skin wound images**, e-Health Networking, Applications and Services (Healthcom), 2014 IEEE 16th International Conference on, pp.387,392, 15-18 Oct. 2014 (Qualis CC 2012, B4)