

UNIVERSIDADE FEEVALE

NÍCOLAS POHREN

**O USO DE ALGORITMOS GENÉTICOS PARA SEGMENTAÇÃO AUTOMÁTICA
DE IMAGENS**

Novo Hamburgo

2018

NÍCOLAS POHREN

**O USO DE ALGORITMOS GENÉTICOS PARA SEGMENTAÇÃO AUTOMÁTICA
DE IMAGENS**

Trabalho de Conclusão de Curso apresentado
como requisito parcial à obtenção do grau
de Bacharel em Ciência da Computação pela
Universidade Feevale

Orientador: Marta Rosecler Bez

Novo Hamburgo

2018

AGRADECIMENTOS

Gostaria de agradecer a todos que, direta ou indiretamente, contribuíram para a realização deste trabalho de conclusão, em especial: Aos meus pais, Luis e Katharina, pelos ensinamentos e suporte incondicional. À Marina, que me apoiou durante todo o desenvolvimento deste trabalho. À professora Marta, que, além de me orientar durante o ano inteiro, sempre se mostrou uma grande amiga. E aos juízes deste trabalho, Anderson, Gabriel e Rodrigo, pela disponibilidade e interesse com o trabalho. Muito obrigado!

RESUMO

A segmentação automática de imagens vem se tornando cada vez mais utilizada em diversas áreas como Medicina e Agronomia. Cada vez mais o método tradicional de segmentação automática, que se baseia no uso de técnicas de processamento digital de imagens, vem perdendo lugar para o uso de algoritmos baseados em redes neurais. Isso devido a sua capacidade de trabalhar com imagens pouco homogêneas. No entanto, as redes neurais acabam se tornando uma espécie de caixa-preta, onde, muitas vezes, não é possível que um ser humano interprete como o sistema chegou ao seu resultado. Neste trabalho é proposto um sistema que utiliza algoritmos genéticos para treinar o sequenciamento e a parametrização de técnicas de processamento digital de imagens. Dessa forma, é demonstrada a possibilidade do sistema funcionar com imagens não homogêneas, permitindo ainda a análise do sistema evoluído por um ser humano. O método proposto obteve uma assertividade de 91,32% na sua base de treinamento, o que está de acordo com outros métodos de segmentação. Também foi realizado um questionário com três juízes, comprovando assim a capacidade de análise e entendimento dos modelos gerados.

Palavras-chaves: Segmentação de imagens; Processamento digital de imagens; Algoritmo Genético.

ABSTRACT

Automatic image segmentation is being used increasingly in several fields such as medicine and agronomy. The traditional method for automatic segmentation, which is based on digital image processing techniques, is losing space for algorithms based on neural networks. This is because of their ability to work with non-homogeneous images. However, neural networks are a kind of black box, where, frequently, it is not possible for a human being to analyze how the system achieved its result. This study proposes a system that uses genetic algorithms to train the sequencing and parametrization of digital image processing techniques. In that manner, the system's capability of working with non-homogeneous images is demonstrated, still allowing the analysis of the evolved model by a human being. The proposed method obtained an assertivity rate of 91,32% using the training dataset, which is in accordance with other segmentation methods. A questionnaire was also used with three judges, in order to prove the capability of analysis and understanding of the generated models.

Key-words: Image segmentation; Digital image processing; Genetic Algorithm.

LISTA DE ILUSTRAÇÕES

Figura 1 – Fluxo básico de um algoritmo genético	13
Figura 2 – Exemplo de uma representação cromossomial binária RGB	14
Figura 3 – Exemplo de um <i>crossover</i> de um ponto	18
Figura 4 – Exemplo de <i>crossover</i> uniforme	19
Figura 5 – Exemplo de um operador de mutação em um cromossomo de cinco bits	19
Figura 6 – Etapas do processamento digital de imagens	22
Figura 7 – Exemplo de uma imagem e seu histograma	25
Figura 8 – Demonstração da relação entre a imagem original e o histograma	26
Figura 9 – Demonstração de <i>thresholding</i>	26
Figura 10 – Demonstração de diferentes limiares	27
Figura 11 – Exemplo de brilho e contraste	28
Figura 12 – Demonstração de inversão	28
Figura 13 – Conversão para tons de cinza	29
Figura 14 – Demonstração do conjunto que representa a imagem binária	30
Figura 15 – Demonstração do processo de dilatação	31
Figura 16 – Elementos estruturantes na dilatação	31
Figura 17 – Demonstração do processo de erosão	32
Figura 18 – Demonstração de abertura e fechamento	33
Figura 19 – Exemplo do processo de convolução	34
Figura 20 – Exemplo de máscaras de convolução de média	34
Figura 21 – Demonstração dos filtros de média e mediana	35
Figura 22 – Demonstração do Filtro de Gauss	36
Figura 23 – Demonstração do Filtro de Sobel	37
Figura 24 – Demonstração do Filtro de Canny	38
Figura 25 – Exemplo de transformações geométricas	40
Figura 26 – Máscaras utilizadas pelo algoritmo de Stentiford	42
Figura 27 – Máscaras utilizadas pelo Zhang Suen	44
Figura 28 – Janela utilizada para a análise dos pixels vizinhos no algoritmo de Holt	44
Figura 29 – Resultado de limiarização na imagem Lena	47
Figura 30 – Resultado de limiarização na imagem Rio Solimões	47
Figura 31 – Cromossomo binário onde L corresponde a cada nível da quadtree	48
Figura 32 – Comparação da segmentação via <i>quadtree</i>	48
Figura 33 – Comparação da segmentação via crescimento de regiões	49
Figura 34 – Imagens utilizadas para teste	51

Figura 35	– Demonstração de um processo feito no VISNode	53
Figura 36	– Exemplo de imagens do FASSEG	54
Figura 37	– Imagens utilizadas para comparação	56
Figura 38	– Painel principal da aplicação	59
Figura 39	– Tabela de gerações	59
Figura 40	– Tela de indivíduos	60
Figura 41	– Gráfico de assertividade da simulação por geração	62
Figura 42	– Imagens e gabaritos utilizadas para treinamento	65
Figura 43	– Resultado da segmentação para o modelo 1	66
Figura 44	– Resultado da segmentação para o modelo 2	67
Figura 45	– Resultado da segmentação para o modelo 3	68
Figura 46	– Gráfico de respostas para as questões de identificação	70
Figura 47	– Gráfico de respostas para a questão: A área segmentada condiz com a área do rosto da imagem original	71
Figura 48	– Gráfico de respostas para a questão: O contorno da área segmentada está bem delineado e sem ruídos	72
Figura 49	– Gráfico de respostas para a questão: A qualidade geral da segmentação é consistente entre todas as imagens de entrada	72
Figura 50	– Gráfico de respostas para a questão: Existe clareza sobre quais algorit- mos de PDI estão sendo utilizados	74
Figura 51	– Gráfico de respostas para a questão: Existe clareza sobre quais parâ- metros são utilizados para cada algoritmo	74
Figura 52	– Gráfico de respostas para a questão: Está clara a sequência que estes algoritmos serão executados	75
Figura 53	– Gráfico de respostas para a questão: Pode-se compreender como a es- colha dos algoritmos, seu sequenciamento e sua parametrização geram a segmentação do rosto	76
Figura 54	– Gráfico de respostas para a questão: Pode-se extrair algum conheci- mento ou aprendizado quanto a como realizar segmentações a partir da análise destes resultados	77

LISTA DE TABELAS

Tabela 1 – Exemplo de avaliações e normalização	16
Tabela 2 – Exemplo de roleta viciada	17
Tabela 3 – Máscaras do operador de Sobel	37
Tabela 4 – Resultados da segmentação	49
Tabela 5 – Comparação do método proposto com outros métodos semelhantes . .	51
Tabela 6 – Comparação entre os métodos de comparação de imagens binários . . .	58
Tabela 7 – Algoritmos de PDI	61
Tabela 8 – Resultado dos três modelos gerados	65
Tabela 9 – Perfil dos juízes avaliadores	69
Tabela 10 – Questionário	69
Tabela 11 – Tabela comparativa do resultado obtido com outros trabalhos de seg- mentação	79

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application programming interface</i>
AG	Algoritmo Genético
CMYK	<i>Cyan, Magenta, Yellow, Black</i>
FA	Função de avaliação
GA	<i>Genetic Algorithm</i>
HSI	<i>Hue, Saturation, Intensity</i>
PDI	Processamento digital de imagens
RGB	<i>Red, Green, Blue</i>
RN	Rede Neural

SUMÁRIO

1	Introdução	9
2	Algoritmos Genéticos	12
2.1	Algoritmos Evolucionários	12
2.2	Algoritmos Genéticos	13
2.2.1	Representação Cromossomial	13
2.2.2	Inicialização	15
2.2.3	Função de Avaliação	15
2.2.4	Seleção	16
2.2.5	Operadores Genéticos	17
2.3	Espécies	20
2.4	Considerações	20
3	Processamento digital de imagens	21
3.1	Histograma	23
3.2	Limiarização	24
3.3	Brilho e contraste	27
3.4	Conversão em Tonalidades de Cinza	29
3.5	Morfologia Matemática	29
3.5.1	Dilatação e Erosão	30
3.5.2	Abertura e Fechamento	32
3.6	Filtragem da Imagem	32
3.6.1	Filtros passa-baixas	33
3.6.1.1	Filtro da Média e Mediana	33
3.6.1.2	Filtro Gaussiano	35
3.6.2	Filtros passa-altas	36
3.6.2.1	Operador de Sobel	36
3.6.2.2	Operador de Canny	37
3.7	Transformações Geométricas	38
3.7.1	Translação	38
3.7.2	Escala	39
3.7.3	Rotação	39
3.7.4	Espelhamento	40
3.7.5	Contornos ativos	40
3.7.6	Afinamento de Bordas	42
3.7.6.1	Stentiford	42

3.7.6.2	Zhang Suen	43
3.7.6.3	Holt	43
3.8	Considerações	45
4	Trabalhos Correlatos	46
4.1	O Uso do Algoritmo Genético em Segmentação de Imagens Digitais	46
4.2	A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation	49
5	Desenvolvimento da aplicação	52
5.1	Algoritmo Genético Proposto	52
5.1.1	Representação Cromossomial	54
5.1.2	Parametrização do Algoritmo Genético	54
5.1.3	Comparação de Imagens	56
5.2	Interface de usuário	58
5.3	Treinamento e Otimização do GA	60
5.3.1	Experimento A - Experimento de controle do GA	60
5.3.2	Experimento B - Variação no <i>crossover</i> e mutação	61
5.3.3	Experimento C - Simulação por muitas gerações	62
5.3.4	Experimento D - Maior quantidade de processos possíveis	63
5.3.5	Experimento E - Indivíduos aleatórios	63
5.3.6	Experimento F - <i>Crossover</i> de 1 ponto e 2 pontos	63
5.3.7	Experimento G	64
5.4	Resultados obtidos	64
5.5	Avaliação	67
5.5.1	Avaliação por Juízes	68
5.5.1.1	Identificação	70
5.5.1.2	Qualidade da segmentação	70
5.5.1.3	Clareza da solução	73
5.5.1.4	Análise e aprendizado	75
5.5.2	Comparativo com Outras Soluções	78
6	Conclusão	80
	Conclusão	80
	Referências	81

1 INTRODUÇÃO

Os avanços em métodos de aquisição de imagens, no poder computacional de *hardware* e na melhoria nas tecnologias, vêm tornando a análise automática de imagens uma técnica utilizada na resolução de diversos problemas. Destaca-se o uso da análise de imagens nas áreas da Medicina, Biologia, Sensoriamento Remoto, Meteorologia, Automação Industrial, Engenharia, Geologia e Agronomia.

Um problema comum na análise de imagens é a segmentação destas. O processamento para segmentação de imagens visa identificar e separar a imagem em diferentes regiões relevantes para o processamento em questão. Como exemplo é possível citar a segmentação de órgãos, como pulmões, em um exame de tomografia computadorizada (RÖNNAU, 2015).

Uma forma de realizar a análise das imagens se dá pela execução de uma sequência de algoritmos de processamento digital de imagens (PDI) (GONZALEZ; WOODS, 2012). Esses algoritmos podem ser orquestrados e parametrizados pelo criador do sistema, ou de forma empírica pelo usuário. Mais recentemente, outra forma utilizada para a resolução de problemas de segmentação e classificação de imagens é a aplicação de redes neurais (RN) simples ou convolucionais (NOH; HONG; HAN, 2015), onde um sistema de aprendizado de máquina é treinado para executar a tarefa especificada. Esse treinamento é efetuado de forma supervisionada, necessitando de uma base previamente anotada, o que é um recurso de construção bastante custoso.

A técnica de RN vem ganhando visibilidade no mundo acadêmico. Isso se deve a diversos fatores. O uso de RN permite que um único sistema seja treinado para solucionar diversos problemas. Também é possível que o treinamento supervisionado da RN seja realizado por um especialista no assunto em questão e que não necessite de conhecimento do sistema e da tecnologia utilizada. Além disso, o seu resultado normalmente é superior quando as imagens não são homogêneas, ou seja, não se enquadram perfeitamente na situação prevista pelo sistema (PAL; PAL, 1993).

No entanto, sistemas baseados em redes neurais oneram em tempo de processamento em relação aos sistemas baseados somente em PDI (HUANG; LIM; MING, 1992), o que pode ser um problema para sistemas de monitoramento em tempo real ou avaliação de uma grande quantidade de imagens.

Embora existam tentativas anteriores de analisar e compreender a saída de uma RN (ZEILER; FERGUS, 2014), isso não é possível em todas as soluções que as utilizam. Dessa forma, a rede neural acaba se tornando uma caixa-preta, de forma que geralmente não é possível a um ser humano interpretar como ela está chegando ao resultado de forma

que ele possa aprender com o sistema e reproduzir os resultados.

Nesta pesquisa explora-se, então, o problema de como realizar a segmentação de imagens digitais utilizando técnicas de *machine learning* com uma qualidade de segmentação que está de acordo com aquelas encontradas em outras pesquisas, mas que ainda permite a análise e interpretação por um ser humano. Este trabalho possui metodologia aplicada, exploratória e experimental, e utilizará abordagens metodológicas qualitativas, onde será avaliada a capacidade de interpretação dos modelos de segmentação gerados. O trabalho também possui aspectos quantitativos, onde será validada e comparada a assertividade da segmentação com outras pesquisas semelhantes.

Em outras soluções desenvolvidas para problemas específicos são utilizadas técnicas de PDI pré-selecionadas e algoritmos genéticos (HOLLAND, 1992) para o treinamento dos seus parâmetros. Pode-se citar o software que segmenta elementos em fotos de satélite (COSTA; CAZES, 2010) ou a comparação da evolução dos parâmetros em métodos de segmentação baseados em *quadrees*, *thresholding* e crescimento de regiões (MATIAS, 2007). Entretanto, todos os trabalhos utilizam o AG apenas para ajustar os parâmetros dos algoritmos de PDI, sem utilizar o AG para mudar o algoritmo a ser utilizado ou a sequência de passos para obter o resultado.

Propõe-se, neste trabalho, um sistema que seja capaz de utilizar algoritmos genéticos tanto para parametrizar quanto para sequenciar técnicas de PDI. Isso permite a obtenção de resultados de qualidade semelhante a de redes neurais, porém, também permitindo o estudo, por parte de um ser humano, para interpretar a sequência de passos utilizada, de forma a aprender e refinar o processamento.

Prévio a escrita deste projeto, foi desenvolvido um ambiente de manipulação de técnicas de PDI denominado VISNode (VISNODE, 2018). Este permite a utilização e parametrização de processos de PDI em formato de grafo, onde os processos não são executados necessariamente em sequência, e parâmetros podem ser transferidos entre processos, ou nodos. A ferramenta permite a execução dos grafos de processos, denominados projetos, via uma *application programming interface* (API) utilizando como entrada um conjunto de imagens, assim como a visualização destes projetos em uma interface gráfica.

Com base neste ambiente, o sistema proposto terá como saída um projeto da ferramenta VISNode. Assim, será possível visualizar facilmente os processos que estão sendo utilizados, em que sequência e com quais parâmetros, pela interface do sistema.

Este trabalho está dividido em 6 capítulos. O Capítulo 2 trata sobre algoritmos genéticos, onde é apresentado o conceito geral de algoritmos evolucionários, e as operações que são realizadas neste tipo de algoritmo. No Capítulo 3 são apresentados diversos algoritmos e técnicas que podem ser utilizados para trabalhar com imagens digitais de forma a segmentar a mesma. No Capítulo 4 estão descritos trabalhos que se utilizam de

algoritmos genéticos em conjunto com técnicas de PDI para a segmentação de imagens digitais. O Capítulo 5 especifica a proposta deste trabalho, quais técnicas de PDI foram utilizadas, a implementação do algoritmo genético e a forma de validação dos resultados. Por fim, o Capítulo 6 apresenta a conclusão do trabalho.

2 ALGORITMOS GENÉTICOS

Este capítulo trata da técnica de algoritmos genéticos (do inglês *genetic algorithms*, ou GA). São apresentados os conceitos de população, geração, cromossomos, genes, *crossover* e mutação.

2.1 ALGORITMOS EVOLUCIONÁRIOS

Na área de inteligência artificial, algoritmos evolucionários usam modelos computacionais dos processos naturais de evolução como uma ferramenta para resolver problemas de otimização. Apesar de haver uma grande variedade de modelos computacionais propostos, todos eles têm em comum o conceito de simulação da evolução das espécies através de seleção, mutação e reprodução, processos estes que dependem do desempenho dos indivíduos desta espécie dentro do seu ambiente (LINDEN, 2008).

Os algoritmos evolucionários funcionam mantendo uma população de estruturas, denominadas de indivíduos, cada um representando uma solução para o problema. Cada indivíduo recebe uma avaliação, que é uma qualificação de sua qualidade como solução do problema em questão. Esta qualificação é calculada utilizando uma função de avaliação (FA), ou *fitness function*. Com base nesta avaliação são aplicados operadores que simulam conceitos da biologia, como reprodução e mutação, de forma a simular a sobrevivência do mais apto (LINDEN, 2008).

O primeiro passo para um algoritmo evolucionário é a criação da população inicial, chamada de primeira geração. O tamanho de uma população varia de acordo com o problema, mas é comum se utilizar populações de centenas ou milhares de indivíduos. Normalmente esta geração é completamente aleatória para garantir a máxima diversidade nos indivíduos (LINDEN, 2008).

Após a criação da primeira geração, cada indivíduo é avaliado de acordo com a função de avaliação. A FA é utilizada para avaliar e sumarizar o quão próximo cada indivíduo está de alcançar o seu objetivo. A definição da função de avaliação é o passo mais importante para a criação de um algoritmo evolucionário, pois é ela quem dita o objetivo do algoritmo. Muitas vezes não é possível calcular diretamente a função de avaliação de um indivíduo, sendo necessária a realização de uma simulação para obter o seu valor (LINDEN, 2008).

Seleciona-se então os indivíduos com a maior avaliação para reprodução. Uma nova população (próxima geração) é criada com base nos indivíduos mais aptos da geração anterior. Podem ser utilizados diversos operadores para a criação dos novos indivíduos, como *crossover* (reprodução) e mutação. Ao fim do processo, existirá uma nova população,

que deverá ser avaliada novamente e o processo se repete para a criação da próxima geração.

Este processo é executado até que uma condição de terminação seja atingida. Essa condição de terminação pode ser o alcance de uma avaliação mínima, número de gerações ou quantidade de recursos de processamento utilizados.

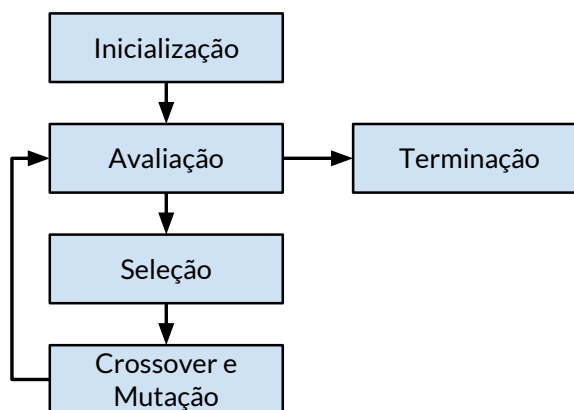
2.2 ALGORITMOS GENÉTICOS

Algoritmos genéticos são um ramo dos algoritmos evolucionários que traz conceitos da biologia genética como genes e cromossomos para os algoritmos evolucionários. Isso permite a melhor simulação de um sistema evolucionário próximo ao que existe na natureza, obtendo resultados de alta qualidade para problemas de otimização.

Como os algoritmos genéticos fazem parte do grupo de algoritmos evolucionários, eles possuem a mesma sequência de passos: inicialização, avaliação, simulação e criação de uma nova geração. O diferencial do algoritmo genético é o uso de cromossomos para representar um indivíduo. O cromossomo é uma sequência de valores, ou genes, que representa a solução daquele indivíduo.

Para a criação das novas gerações, são utilizadas funções com base nos cromossomos. Alguns exemplos de operadores são o *crossover*, que visa juntar dois cromossomos pais para a criação de cromossomos filhos que herdam características de seus pais, e o operador de mutação, que simula a possibilidade natural de uma mutação genética. O fluxo de um algoritmo genético pode ser visto na Figura 1.

Figura 1 – Fluxo básico de um algoritmo genético



Fonte: Adaptado de Linden (2008)

2.2.1 Representação Cromossomial

Em algoritmos genéticos todos os indivíduos são representados por uma cadeia de valores chamada de cromossomo. Cada pedaço indivisível desta representação é chamado

de gene, por analogia com as partes fundamentais que compõem um cromossomo biológico.

A codificação da informação em cromossomos é um ponto crucial dentro do GA, e é, junto com a função de avaliação, o que liga o GA ao problema a ser resolvido. Se a codificação for feita de forma inteligente, esta já incluirá as idiossincrasias do problema e permitirá que se evitem testes de viabilidade de cada uma das soluções geradas (LINDEN, 2008).

A representação mais simples e mais usada pelos praticantes da área de algoritmos genéticos é a binária, isto é, um cromossomo nada mais é do que uma sequência de *bits*. Essa representação foi adotada inicialmente por Holland (1992) e, hoje em dia, ela é amplamente adotada por pesquisadores da área de GA. Ademais, os operadores genéticos são compreensíveis e implementáveis rapidamente para cromossomos binários (LINDEN, 2008).

Um cromossomo para o algoritmo genético é apenas uma sequência de valores individuais. O GA não sabe a natureza do problema que o indivíduo está tentando solucionar. Quem fará a ligação do cromossomo com o problema será a função de avaliação, que será descrita mais detalhadamente na seção 2.2.3. Um exemplo de representação cromossomial binária pode ser um cromossomo de três genes, onde o primeiro gene representa a cor vermelha, o segundo a cor verde e o terceiro a cor azul. Assim, na Figura 2, é possível ver todo o espaço de soluções representáveis por este cromossomo.

Figura 2 – Exemplo de uma representação cromossomial binária RGB



Fonte: elaborado pelo autor

No entanto, a representação binária tem dificuldades ao lidar com múltiplas dimensões de variáveis contínuas, principalmente com alta precisão. Esta também não é uma representação ideal quando existe um número finito de estados distintos que não é múltiplo de dois, pois muitos estados se tornarão inválidos. Por esta razão existem outras formas de representação cromossomial, como a numérica e a categórica.

A representação cromossomial numérica assume que cada gene terá um valor

inteiro ou real ao invés de valores binários. Outra forma de representação cromossomial é a categórica, onde cada gene representa um valor de uma lista de valores possíveis.

Ainda existe a possibilidade da criação de um cromossomo híbrido. Em um cromossomo de representação híbrida, cada gene pode ter a sua própria representação (binária, numérica ou categórica), e os operadores genéticos devem funcionar de acordo com o tipo do gene (LINDEN, 2008).

2.2.2 Inicialização

O primeiro passo de um algoritmo genético é a criação da geração inicial. Neste passo é criada uma população de indivíduos, sendo que cada indivíduo possui o seu cromossomo como uma cadeia completamente aleatória de genes. A lei das probabilidades sugere que existirá uma distribuição de soluções que cobre praticamente todo o espaço de soluções, mas isto não pode ser garantido, pois a população possui um tamanho finito (LINDEN, 2008).

O desempenho do algoritmo genético é extremamente sensível ao tamanho da população, logo, este parâmetro deve ser definido com muito cuidado. Caso este número seja pequeno demais, não haverá espaço para uma variedade genética suficientemente grande dentro da população, o que fará com que o algoritmo seja incapaz de achar boas soluções. Caso este número seja grande demais, o algoritmo demorará demais e utilizará muitos recursos computacionais (LINDEN, 2008). Normalmente populações variam entre centenas e milhares de indivíduos.

2.2.3 Função de Avaliação

Segundo Linden (2008), a função de avaliação é a maneira utilizada pelos GAs para determinar a qualidade de um indivíduo como solução do problema em questão. A FA pode ser considerada como uma nota numérica dada ao indivíduo na resolução do problema. Essa avaliação será utilizada para a escolha dos indivíduos pelo módulo de seleção dos pais, sendo a forma de diferenciar entre as boas e as más soluções para um problema.

GAs são técnicas de maximização, logo, a função de avaliação deve ser tal que se o cromossomo C1 representa uma solução melhor que o cromossomo C2, então a avaliação de C1 deve ser maior do que a avaliação de C2. A função de avaliação deve refletir os objetivos a serem alcançados na resolução de um problema e é derivada diretamente das condições impostas por este problema (HOLLAND, 1992).

Um cuidado a ser tomado com a função de avaliação é o problema do super indivíduo. Este problema ocorre quando há um ou mais indivíduos cuja avaliação é muito superior àquela dos outros membros da população. Neste caso, este indivíduo será quase

sempre escolhido pelo módulo de seleção, causando uma perda imediata da diversidade genética nas gerações subsequentes.

Para solucionar este problema, é possível utilizar outras técnicas como a de normalização linear. A normalização de uma função de avaliação funciona atribuindo um valor constante k ao melhor cromossomo, $k - c$ para o segundo indivíduo, $k - 2c$ para o terceiro indivíduo, e assim sucessivamente, sendo k e c duas constantes definidas pelo criador do GA. Com a normalização, os indivíduos permanecerão na mesma ordem de avaliação, mas com um intervalo constante entre eles (LINDEN, 2008). Um exemplo de normalização pode ser visto na Tabela 1, utilizando as constantes $k = 10$ e $c = 1$.

Tabela 1 – Exemplo de avaliações e normalização

<i>Indivíduo</i>	<i>Avaliação</i>	<i>Avaliação normalizada</i>
0	123,48	10
1	68,42	9
2	18,45	8
3	18,38	7

2.2.4 Seleção

O próximo passo é a criação de uma nova população para a próxima geração. Esta população será criada com base nos indivíduos da população anterior e os operadores de *crossover* e mutação. O mais comum é que a nova geração tenha um tamanho idêntico a geração anterior (LINDEN, 2008).

Para cada indivíduo a ser criado na nova população são selecionados dois ou mais indivíduos da geração anterior com base em sua função de avaliação. Esses dois indivíduos servirão como pais do novo cromossomo. Com base nos operadores de *crossover* e mutação, o novo indivíduo tipicamente possuirá várias características de seus pais. Novos pares de pais são selecionados para cada novo indivíduo, de forma que quanto maior a avaliação de um indivíduo, maior é sua chance de ser selecionado como pai (LINDEN, 2008). Também é possível utilizar métodos que usam mais de dois indivíduos como pais, o que não é tão análogo com a reprodução natural, mas algumas pesquisas indicam que o resultado é superior (TING, 2005) (EIBEN; RAUE; RUTTKAY, 1994).

O método mais comum para seleção de pais é o da roleta viciada. Para a realização deste método, é atribuída uma probabilidade para cada indivíduo $p_i = p_i - 1 + \frac{a}{t}$, sendo a a avaliação daquele indivíduo e t a soma de todas as avaliações. Dessa forma, a soma de todas as probabilidades deverá ser 1 (ou 100%). É sorteado então um número aleatório r entre 0 e 1 e selecionado o indivíduo onde $p_i - 1 < r < p_i$. Dessa forma, quanto maior a avaliação de um indivíduo, maior sua chance de ser selecionado (LINDEN, 2008). Um exemplo de uma distribuição de roleta viciada pode ser visto na Tabela 2.

Tabela 2 – Exemplo de roleta viciada

<i>Indivíduo</i>	<i>Avaliação</i>	<i>Pedaço da roleta (%)</i>
0001	1	1.61
0011	9	14.51
0100	16	25.81
0110	36	58.07
Total	62	100

Ao final deste processo existirá uma nova população que é diferente da geração inicial. É mais provável que a avaliação média desta nova população seja maior que a da população anterior. Isso se deve ao fato de apenas os indivíduos mais aptos terem sido selecionados para a reprodução (LINDEN, 2008).

Os indivíduos com uma avaliação menor ainda devem ter chance de serem selecionados para reprodução, embora menor, o que garante uma diversidade genética entre a população. Uma maior diversidade genética é importante para garantir que um espectro maior de soluções sejam exploradas pelo algoritmo.

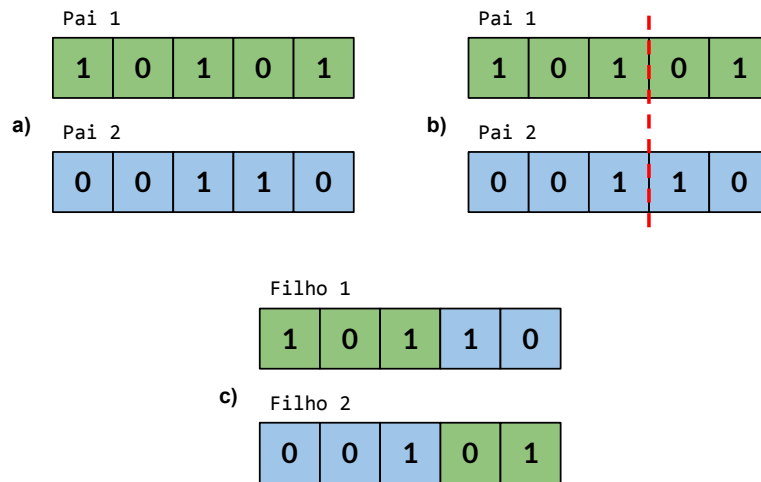
2.2.5 Operadores Genéticos

Os operadores genéticos são funções aplicadas sobre um ou mais cromossomos para a geração de novos cromossomos. Os exemplos mais simples de operadores genéticos são o *crossover* e a mutação. O operador de *crossover* é o responsável por gerar um novo cromossomo baseado em dois cromossomos pais. O operador de mutação é aplicado em um cromossomo para que ele tenha uma pequena chance de sofrer uma mutação genética em algum gene. Existe mais de uma técnica de *crossover* e mutação, ambas podem ser parametrizadas e podem ou não ser executadas.

A técnica de *crossover* mais simples é o *crossover* de um ponto. Neste são usados dois cromossomos pais para gerar dois cromossomos filhos. Para realizar um *crossover* de um ponto, é selecionada uma posição aleatória do cromossomo, e ambos os cromossomos pais são cortados nesta posição. O primeiro filho é composto através da concatenação da parte esquerda do primeiro pai com a parte direita do segundo pai. O segundo filho é composto através da concatenação das partes que sobraram, ou seja, a metade esquerda do segundo pai com a metade à direita do primeiro pai (LINDEN, 2008). É possível ver, na Figura 3, um exemplo de *crossover* de um ponto sendo executado com base em (a) dois pais, (b) um ponto de corte e (c) os dois filhos gerados, em um cromossomo de tamanho cinco com representação binária.

Um problema gerado por um *crossover* de um ponto é que se a combinação de melhor resultado depende que, por exemplo, o primeiro e o último gene sejam 1 (denominado de esquema 1 * * * 1 para um cromossomo de 5 genes), essa combinação pode não ser preservada no *crossover*. Isso se deve ao fato de que pelo menos uma das duas

Figura 3 – Exemplo de um *crossover* de um ponto



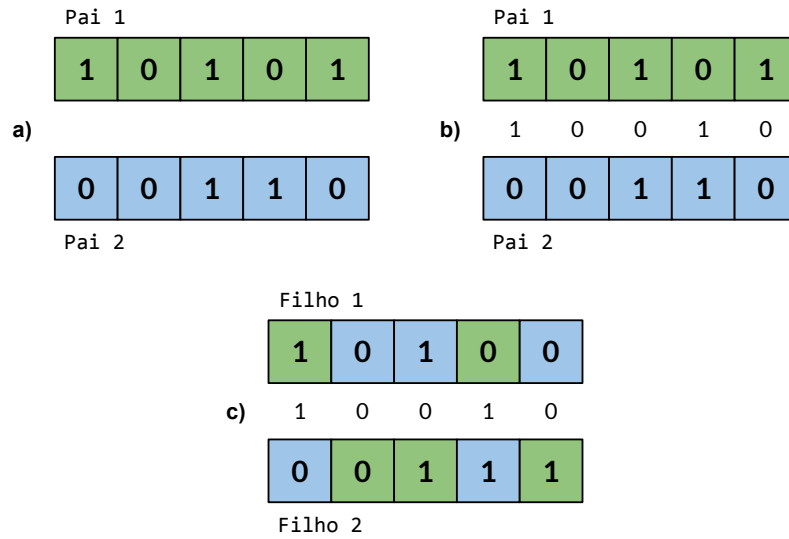
Fonte: Adaptado de Linden (2008)

metades do cromossomo será trocada com outro indivíduo, e se o indivíduo não tiver essa combinação, então este esquema será perdido (LINDEN, 2008).

Existe outra forma de operação de *crossover* que possibilita a preservação de tais esquemas, chamada de *crossover* uniforme. Neste, para cada gene é sorteado um número zero ou um. Se o valor for igual a um, o filho número um recebe o gene da posição corrente do primeiro pai e o segundo filho o gene corrente do segundo pai. Por outro lado, se o valor sorteado for zero, as atribuições serão invertidas: o primeiro filho recebe o gene da posição corrente do segundo pai e o segundo filho recebe o gene corrente do primeiro pai (LINDEN, 2008). Na Figura 4 é possível ver o funcionamento do *crossover* uniforme, com um cromossomo de cinco bits. Em (a) está representado o par de pais selecionados para o *crossover*, em (b) é feita a seleção aleatória binária para cada gene para a construção do par de filhos (c).

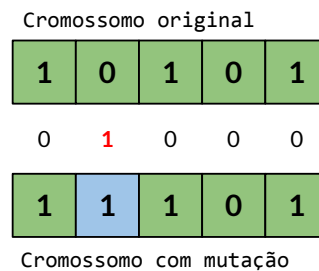
Outro operador utilizado na criação de uma nova geração é o operador de mutação. O objetivo do operador de mutação é incrementar a diversidade genética da população durante a execução do algoritmo, para explorar possibilidades que não foram concebidas na geração inicial e que não foram obtidas com o operador de *crossover*. O operador de mutação opera em cada gene do cromossomo, e possui uma chance normalmente muito baixa (da ordem de 0,5%) de alterar o valor do gene em questão. Para um gene binário, o operador de mutação inverterá o *bit* em questão, como demonstrado na Figura 5, para genes numéricos ou categóricos, um novo valor totalmente aleatório será escolhido (LINDEN, 2008).

Anteriormente foi demonstrado o operador de *crossover* para genes binários. Para

Figura 4 – Exemplo de *crossover* uniforme

Fonte: Adaptado de Linden (2008)

Figura 5 – Exemplo de um operador de mutação em um cromossomo de cinco bits



Fonte: Adaptado de Linden (2008)

genes numéricos, o operador de mutação ocorre trocando um gene por um novo valor completamente aleatório. Já para valores categóricos, deve ser garantido que todas as categorias tenham o mesmo peso ou chance de serem obtidas (LINDEN, 2008).

Normalmente os operadores de *crossover* e mutação não são executados em todos os indivíduos de todas as gerações. No início do GA, é desejável executar muita reprodução e pouca mutação, visto que há muita diversidade genética e é importante explorar o máximo possível o espaço de soluções. Depois de um grande número de gerações, ocorre a convergência genética, o que implica na pouca diversidade na população, tornando extremamente interessante que o operador de mutação fosse escolhido mais frequentemente do que o de *crossover*, para permitir que seja reinserida a diversidade genética na população (LINDEN, 2008).

Seria necessário então que a probabilidade do operador de *crossover* fosse caindo com o decorrer do algoritmo e que a probabilidade do operador de mutação fosse concomitantemente aumentando. Para tal, são utilizadas diversas técnicas de interpolação, como a linear, quadrática e descontínua (LINDEN, 2008).

2.3 ESPÉCIES

Um algoritmo genético pode ainda utilizar o conceito de espécies. Conforme Li et al. (2002), uma espécie é uma subdivisão da população do AG, definida por N indivíduos semente e um parâmetro de distância de espécie. Quando existem diversas espécies em uma simulação, o melhor indivíduo de cada espécie é mantido para a próxima geração. Os outros indivíduos são replicados utilizando os operadores de *crossover* e mutação da mesma forma que um algoritmo genético padrão.

O principal motivo para utilizar segmentação por espécies em um algoritmo genético é garantir que os indivíduos que não alcançaram o melhor resultado em uma determinada geração, mas têm potencial para alcançar a máxima global, não sejam imediatamente descartados. Desta forma, eles sobreviverão por um número maior de gerações, tendo uma chance maior de explorar diferentes variações e obter um resultado ainda melhor.

Para segmentar uma população em espécies, são agrupados indivíduos onde a distância euclidianda de seus parâmetros for menor do que o parâmetro de distância de espécie. Em cada espécie, é selecionado um indivíduo dominante, cuja função de avaliação é a maior da espécie. Estes indivíduos dominantes serão sempre mantidos para a próxima geração.

2.4 CONSIDERAÇÕES

Este capítulo apresentou a técnica de algoritmos genéticos como uma poderosa ferramenta para a otimização de problemas. Foram demonstrados os seus principais conceitos, como população, geração, *fitness*, indivíduo, gene e cromossomo, tal como as técnicas utilizadas, entre elas a mutação, *crossover* de um ponto e *crossover* uniforme.

No próximo capítulo serão apresentadas técnicas de processamento digital de imagens. O uso destas técnicas para segmentação de imagens será considerado como um problema de otimização global, o qual será resolvido utilizando algoritmos genéticos.

3 PROCESSAMENTO DIGITAL DE IMAGENS

Este capítulo aborda o tema sobre processamento digital de imagens. Serão explicados os conceitos de imagem digital, pré-processamento e segmentação, tal como diversas técnicas para manipulação de imagens.

Segundo Pedrini e Schwartz (2008), o processamento digital de imagens consiste em um conjunto de técnicas para capturar, representar e transformar imagens com o auxílio do computador. O emprego dessas técnicas permite extrair e identificar informações das imagens e melhorar a qualidade visual de certos aspectos estruturais, facilitando a percepção humana e a interpretação automática por meio de máquinas.

Azevedo e Conci (2003) definem uma imagem digital como um conjunto finito de pontos que são representados por um número finito e discreto de tons ou cores. A representação adequada de uma imagem em tons de cinza é como uma matriz bidimensional cujas linhas e colunas identificam um ponto na imagem. Cada ponto, também chamado de pixel, é representado no computador como um número inteiro que corresponde a intensidade de luz no ponto. Frequentemente, a cor do pixel é representada como um inteiro de 8 bits variando entre 0 e 255, sendo o valor 0 correspondente à cor preta, 255 à cor branca, e as outras tonalidades de cinza distribuídas entre esses valores limites.

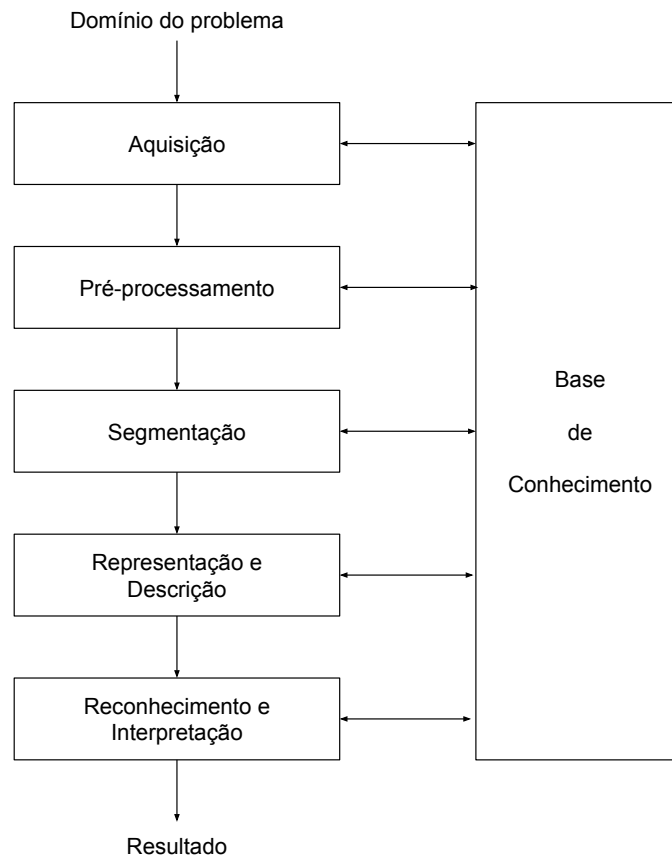
Nessa representação de uma imagem como uma matriz de pixels bidimensional, os índices da matriz são valores inteiros que especificam a linha e a coluna na matriz. O pixel (0,0) está localizado no canto superior esquerdo da imagem. As posições dos pontos no plano da imagem têm coordenadas x e y . A coordenada y corresponde à direção vertical, e a coordenada x corresponde à direção horizontal. O eixo y é positivo para baixo e o eixo x positivo para a direita. Assim, o valor de cada pixel é representado pela função $f(x, y)$.

Ao trabalhar com imagens coloridas, é utilizado um conjunto de planos bidimensionais para representar diferentes cores ou propriedades da imagem, onde cada plano é denominado canal. O modelo de representação mais comum para imagens é o RGB, onde cada plano bidimensional da imagem representa, respectivamente, Vermelho (*Red*), Verde (*Green*) e Azul (*Blue*). Outros modos de representação bastante utilizados são CMYK (*Cyan, Magenta, Yellow e Black*) e HSI (*Hue, Saturation e Intensity*). Em todas as representações ainda é possível adicionar um novo canal, denominado Alpha, que serve para representar a transparência do pixel.

Segundo Pedrini e Schwartz (2008), um sistema de processamento digital de imagens é constituído por um conjunto de cinco etapas, representados na Figura 6, capazes de produzir um resultado a partir do domínio do problema. As etapas são: aquisição; pré-processamento; segmentação; representação e reconhecimento. Todo o conhecimento

sobre o domínio do problema está codificado na forma de uma base de conhecimento. Esta é dependente da aplicação, cujo tamanho e complexidade podem variar significativamente. A base de conhecimento é utilizada para guiar a comunicação entre os módulos de processamento a fim de executar uma determinada tarefa.

Figura 6 – Etapas do processamento digital de imagens



Fonte: Adaptado de Pedrini e Schwartz (2008)

A etapa de aquisição captura a imagem por meio de um dispositivo ou sensor e converte-a em uma representação adequada para o processamento digital subsequente. Os principais dispositivos para aquisição de imagens são câmeras de vídeo, tomógrafos médicos, satélites e scanners. Dentre os aspectos envolvidos nesta etapa estão a escolha do tipo de sensor, as condições de iluminação da cena, a resolução e o número de níveis de cinza ou cores da imagem digitalizada.

A imagem digital resultante do processo de aquisição pode apresentar imperfeições ou degradações decorrentes, por exemplo, das condições de iluminação ou características dos dispositivos. Segundo Gonzalez e Woods (2012), o objetivo principal das técnicas de realce é processar uma imagem, de modo que o resultado seja mais apropriado para uma aplicação específica do que a imagem original. Isso se dá por meio da aplicação de técnicas para atenuação de ruído, correção de contraste ou brilho e a suavização de determinadas propriedades da imagem. Como a interpretação dos dados contidos em

imagens digitais é uma atividade complexa, um processo intermediário de segmentação é necessário para particionar o conjunto de dados de entrada em estruturas com conteúdo semântico relevante para a aplicação em questão. Essas estruturas correspondem a objetos ou partes de objetos que auxiliarão no processo de interpretação de imagens (PEDRINI; SCHWARTZ, 2008).

Segundo Gonzalez e Woods (2012), a segmentação de imagens digitais divide uma imagem em suas partes ou objetos constituintes. O nível até o qual essa subdivisão deve ser realizada depende do problema a ser resolvido. Ou seja, a segmentação pode parar quando os objetos de interesse na aplicação tiverem sido isolados.

Em geral, a segmentação autônoma é uma das tarefas mais difíceis em processamento de imagens. Esse passo determina o eventual sucesso ou fracasso na análise. De fato, a segmentação efetiva quase sempre garante sucesso no reconhecimento. Por essa razão, um cuidado considerável deve ser tomado para melhorar as chances de uma segmentação robusta (GONZALEZ; WOODS, 2012). Ruídos na imagem podem levar os métodos de segmentação a distorcer as formas dos objetos, comprometendo seu reconhecimento, tal que regiões distintas poderiam ser incorretamente identificadas como uma única região ou, por outro lado, uma região homogênea poderia ser dividida em regiões menores (PEDRINI; SCHWARTZ, 2008).

Após a segmentação, estruturas adequadas de representação devem ser utilizadas para armazenar e manipular os objetos de interesse extraídos da imagem. O processo de descrição visa a extração de características ou propriedades que possam ser utilizadas na discriminação entre classes de objetos. Essas características são, em geral, descritas por atributos numéricos que formam um vetor de características.

A última etapa envolve o reconhecimento e a interpretação dos componentes de uma imagem. Reconhecimento ou classificação é o processo que atribui um identificador ou rótulo aos objetos da imagem, baseado nas características providas pelos seus descritores. O processo de interpretação consiste em atribuir um significado ao conjunto de objetos reconhecidos.

Nas próximas seções deste trabalho serão apresentados alguns conceitos de PDI, como histograma e convolução, que são utilizados por diversas técnicas de PDI. Ademais, serão demonstrados algoritmos de pré-processamento e segmentação de imagens que serão utilizados na proposta deste trabalho.

3.1 HISTOGRAMA

Segundo Pedrini e Schwartz (2008), o histograma de uma imagem corresponde à distribuição dos níveis de cinza da imagem, o qual pode ser representado por um gráfico indicando o número de pixels na imagem para cada nível de cinza. Seja $f(x, y)$ uma

imagem representada por uma matriz bidimensional, com dimensões M por N pixels e contendo L níveis de cinza no intervalo $[0, L_{max}]$. O cálculo do histograma é apresentado no Listing 3.1. O histograma é representado por um vetor H com L elementos.

Listing 3.1 – Cálculo do histograma de uma imagem em tons de cinza

```

1 // Atribuir valor zero a todos os elementos do vetor
2 para i = 0 ate lmax faca
3     H[i] <- 0
4
5 // Calcula distribucao dos tons de cinza para cada pixel
6 para x = 0 ate M - 1 faca
7     para y = 0 ate N - 1 faca
8         H[f(x,y)] <- H[f(x,y)] + 1

```

Na Figura 7 está representado (a) um exemplo de uma imagem de 64 por 64 pixels, cada um com 4 *bits* de dados, permitindo 16 tonalidades de cinza. Ao lado há (b) o array H retornado pelo cálculo de histograma e (c) o gráfico representando o histograma da imagem digital.

O histograma de uma imagem fornece uma indicação de sua qualidade quanto ao nível de contraste e quanto a sua luminosidade média, ou seja, se a imagem é predominantemente clara ou escura (AZEVEDO; CONCI, 2003). Esta relação está demonstrada na Figura 8.

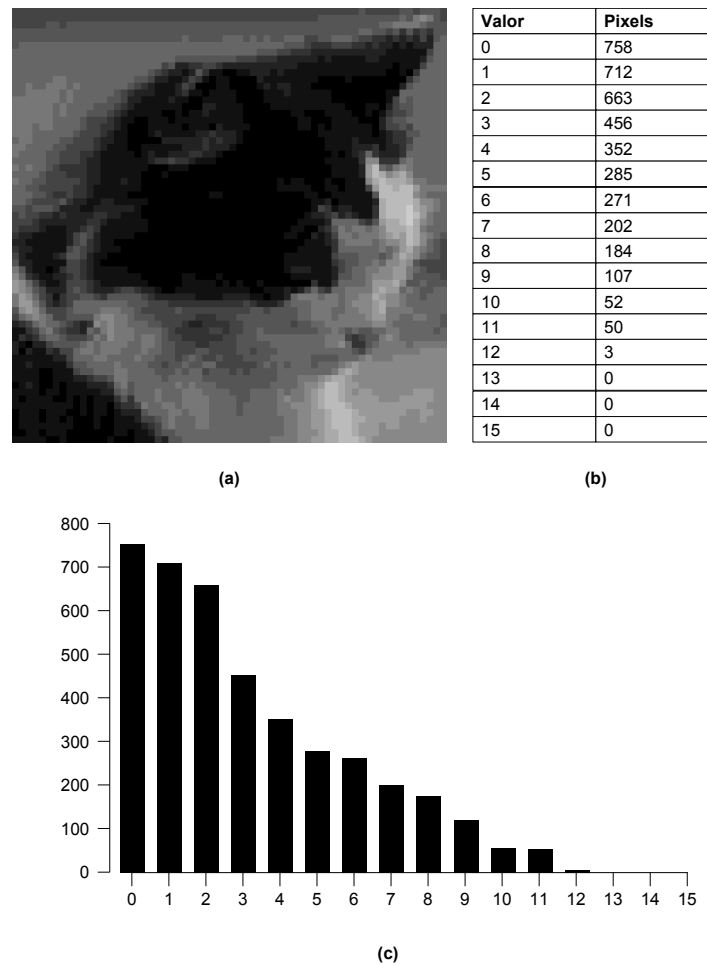
3.2 LIMIA RIZAÇÃO

Segundo Pedrini e Schwartz (2008), a limiarização ou *thresholding* é uma das técnicas mais simples de segmentação e consiste na classificação dos pixels de uma imagem de acordo com a especificação de um ou mais limiares. Cada pixel da imagem será classificado com base nos limiares definidos, de forma que a limiarização pode ser definida como

$$f(x, y) = \begin{cases} l_1, se f(x, y) \leq T_1 \\ l_2, se T_1 < f(x, y) \leq T_2 \\ l_3, se T_2 < f(x, y) \leq T_3 \\ \dots \\ l_k - 1, se f(x, y) < T_n \\ l_k, se f(x, y) \geq T_n \end{cases}$$

tal que, para cada intervalo T_n , é especificado um nível de cinza correspondente l_k . O algoritmo para a limiarização pode ser visto no Listing 3.2.

Figura 7 – Exemplo de uma imagem e seu histograma



Fonte: elaborado pelo autor

Listing 3.2 – Algoritmo para limiarização

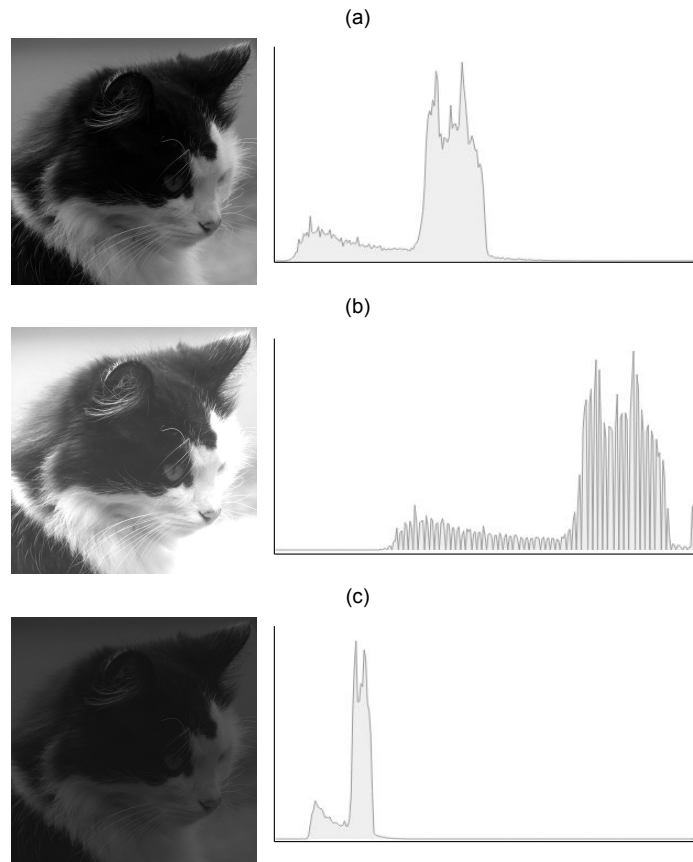
```

1 para x = 0 ate M - 1 faca
2     para y = 0 ate N - 1 faca
3         se H[f(x,y)] < T1
4             H[f(x,y)] = l1
5         senao se H[f(x,y)] < T2
6             H[f(x,y)] = l2
7         [...]
8         senao
9             H[f(x,y)] = ln

```

Quando a limiarização possui apenas um limiar ela é denominada binarização, pois a imagem resultante possui apenas dois valores de intensidade, 0 (preto) ou 1 (branco) (PEDRINI; SCHWARTZ, 2008). A forma mais comum para a identificação de um limiar adequado é a análise do histograma da imagem, visando identificar o ponto de corte entre as grandes concentrações de valores (GONZALEZ; WOODS, 2012). Um exemplo de

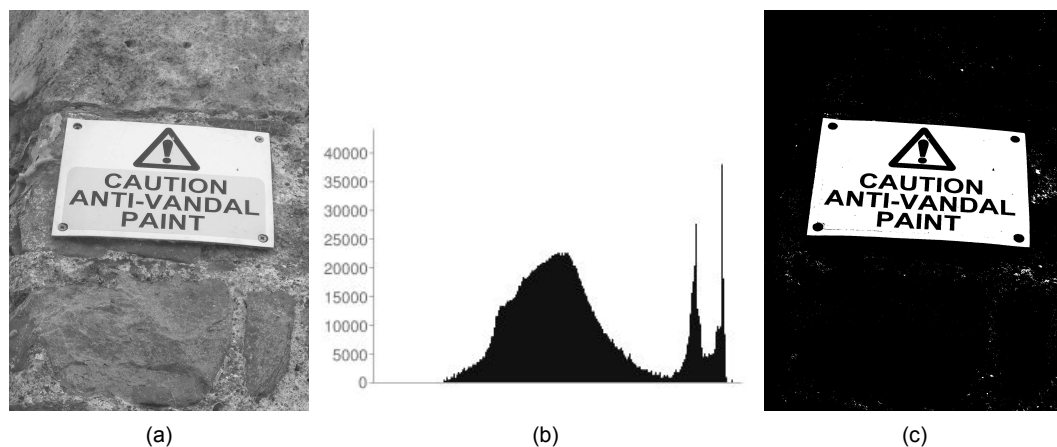
Figura 8 – Demonstração da relação entre a imagem original e o histograma



Fonte: elaborado pelo autor

binarização pode ser visto na Figura 9, demonstrando a imagem original (a), o histograma da imagem (b) e a imagem resultante da aplicação do limiar de 203 (c).

Figura 9 – Demonstração de *thresholding*

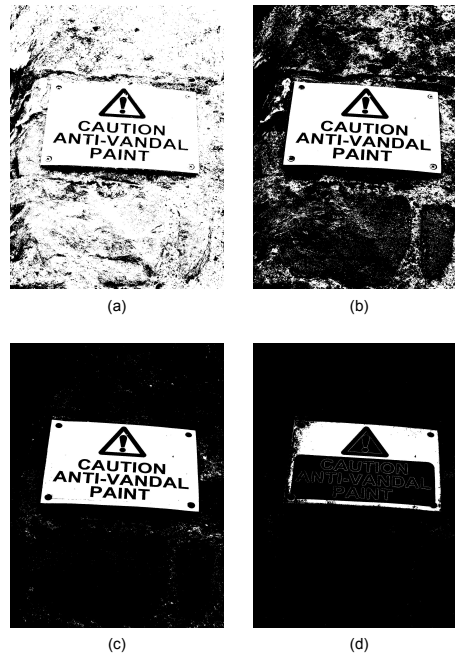


Fonte: elaborado pelo autor

A seleção correta do valor de limiar é crucial para que o processo de segmentação baseado na limiarização produza bons resultados (PEDRINI; SCHWARTZ, 2008). O resultado do uso de diferentes valores para os limiares pode ser visto na Figura 10, utili-

zando os valores (a) 100, (b) 150, (c) 200, e (d) 230. Na prática, espera-se que o tipo de limiarização global descrita obtenha sucesso apenas em ambientes altamente controlados (GONZALEZ; WOODS, 2012).

Figura 10 – Demonstração de diferentes limiares



Fonte: elaborado pelo autor

3.3 BRILHO E CONTRASTE

Conforme Pedrini e Schwartz (2008), o brilho está associado à sensação visual de intensidade luminosa ou luminância de uma fonte. Em imagens digitais o brilho é diretamente relacionado ao valor do pixel de uma imagem. Já o contraste pode ser definido como uma medida de variação relativa da luminância, ou seja, da intensidade luminosa por unidade de área.

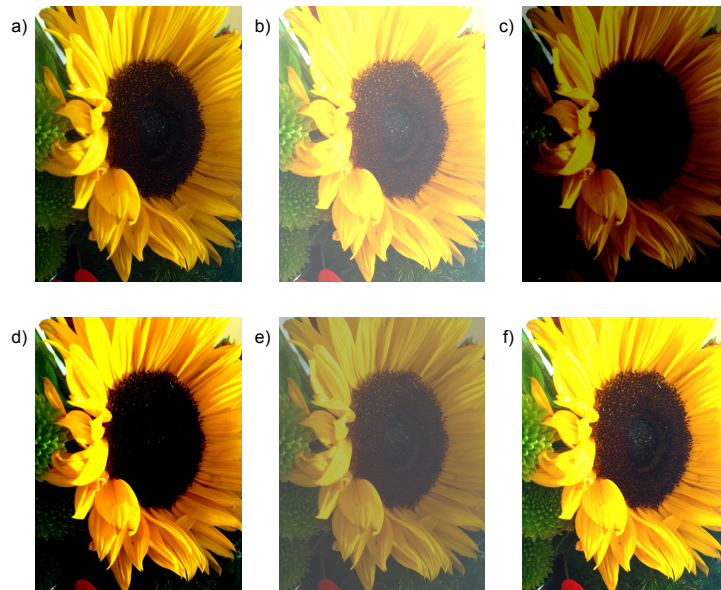
Para realizar alterações no brilho ou contraste de um pixel da imagem, pode ser utilizada a transformação descrita pela função $g(x, y) = a \cdot f(x, y) + b$, tal que o parâmetro a controla a escala de níveis de cinza da imagem (contraste) e b ajusta seu brilho (PEDRINI; SCHWARTZ, 2008). O valor de $f(x, y)$ é o valor atual do pixel na posição calculada e o valor de $g(x, y)$ é o valor do pixel na nova imagem. Na Figura 11 é possível ver a diferença causada pela aplicação da função com diferentes parâmetros para a e b em uma imagem original (a). Os parâmetros são: (b) Imagem transformada com $a = 1$ e $b = 100$; (c) Imagem com $a = 1$ e $b = -100$; (d) Imagem transformada com $a = 1.5$ e $b = 0$; (e) Imagem transformada com $a = 0.5$ e $b = 0$; (f) Imagem transformada com $a = 1.4$ e $b = 50$. O algoritmo para a aplicação do brilho e contraste pode ser visto no Listing 3.3

Listing 3.3 – Algoritmo para aplicação de brilho e contraste

```

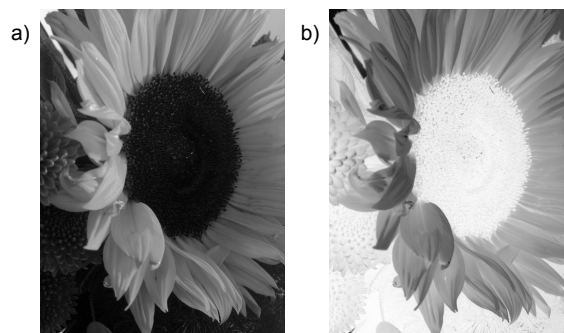
1 para x = 0 ate M - 1 faça
2     para y = 0 ate N - 1 faça
3          $H[f(x,y)] = a * H[f(x,y)] + b$ 

```

Figura 11 – Exemplo de brilho e contraste

Fonte: elaborado pelo autor

Outra função linear comum é a transformação inversa, que produz o negativo de uma imagem. Nessa transformação, a intensidade da imagem de saída diminui à medida que a intensidade da imagem de entrada aumenta (PEDRINI; SCHWARTZ, 2008). Ela pode ser representada com a função de transformação linear demonstrada acima utilizando os parâmetros $a = -1$ e $b = f_{max}$, sendo f_{max} o valor máximo do pixel. Um exemplo de inversão de imagens pode ser vista na Figura 12.

Figura 12 – Demonstração de inversão

Fonte: elaborado pelo autor

3.4 CONVERSÃO EM TONALIDADES DE CINZA

A conversão de uma imagem para tonalidades de cinza consiste em descartar a informação de cor da imagem, mantendo apenas a informação de sua luminância. Para que uma imagem digital utilizando o modelo RGB possua uma tonalidade de cinza, todos os seus canais (*red*, *green* e *blue*) devem possuir exatamente o mesmo valor. Desta forma, a técnica mais simples para converter uma imagem colorida em uma imagem em tonalidades de cinza é realizar a média aritmética dos seus canais, atribuindo o resultado novamente a cada um dos canais, de forma que $C_R = C_G = C_B = (C_R + C_G + C_B)/3$.

No entanto, é possível realizar a conversão para tonalidades de cinza utilizando uma média ponderada. Esta é uma técnica comum, pois o olho humano não consegue perceber as cores vermelho, verde e azul com a mesma precisão. A definição de cada pixel pela média ponderada é dado pela fórmula $C_R = C_G = C_B = C_R * w_R + C_G * w_G + C_B * w_B$. Diversos valores para os pesos da média ponderada foram propostos. Entre eles, é possível citar o modelo CIE 1931 (BROADBENT, 2004) e o modelo ITU-R BT.709 (ANDERSON et al., 1996). Na Figura 13 é possível visualizar o resultado de uma conversão em tons de cinza utilizando a média simples.

Figura 13 – Conversão para tons de cinza



Fonte: elaborado pelo autor

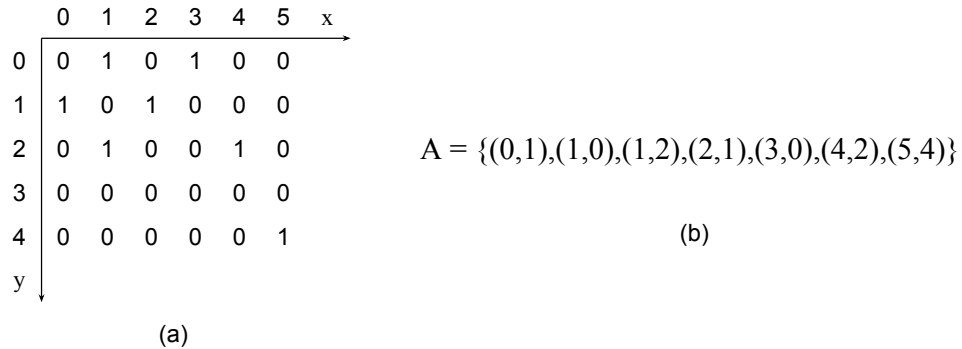
3.5 MORFOLOGIA MATEMÁTICA

Segundo Pedrini e Schwartz (2008) a morfologia matemática consiste em uma metodologia para manipulação de imagens que permite a construção de operadores úteis para a descrição de objetos em imagens. Ela foi originalmente desenvolvida para manipular imagens binárias, sendo posteriormente estendida para tratar imagens em níveis de cinza.

A morfologia matemática utiliza a teoria de conjuntos para representar a forma dos objetos em uma imagem. Por convenção, objetos em uma imagem binária serão representados por pixels com valor 1 enquanto o fundo será formado por pixels com valor

0. Dessa forma, é possível representar uma imagem binária como uma coleção de coordenadas discretas que correspondem aos pontos que fazem parte dos objetos da imagem, o que pode ser visualizado na Figura 14.

Figura 14 – Demonstração do conjunto que representa a imagem binária



Fonte: Adaptado de Pedrini e Schwartz (2008, p.328)

Um operador morfológico é um mapeamento entre o conjunto A que define a imagem e um conjunto B , chamado de elemento estruturante. O elemento estruturante é expresso com respeito a uma origem local (PEDRINI; SCHWARTZ, 2008).

3.5.1 Dilatação e Erosão

A operação de dilatação entre o conjunto A e um elemento estruturante B é definida como a adição de Minkowski (1911), ou seja:

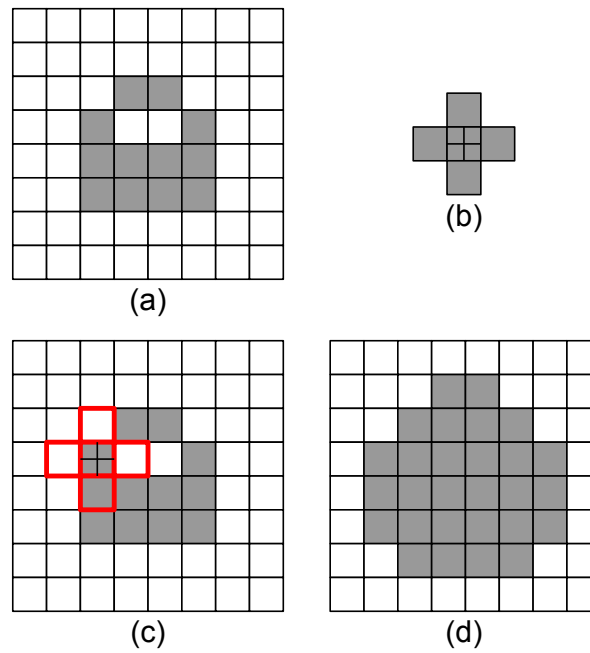
$$D(A, B) = A \oplus B = \bigcup_{b \in B} (A + b)$$

De acordo com esta equação, o processo de dilatação entre A e B corresponde ao conjunto de todas as translações de B com os pontos da imagem em que há pelo menos um elemento não nulo (com valor 1) em comum com o conjunto A (PEDRINI; SCHWARTZ, 2008).

Na Figura 15 está demonstrada a aplicação da dilatação em uma imagem de entrada (a) utilizando o elemento estruturante (b). A origem do elemento estruturante está indicada por uma cruz. Em (c) é possível ver o elemento estruturante sendo aplicado sobre o pixel da posição (3, 4). Na dilatação, caso a origem do elemento estruturante seja não nula, todos os pixels vizinhos receberão o valor do elemento estruturante. A imagem final está demonstrada em (d).

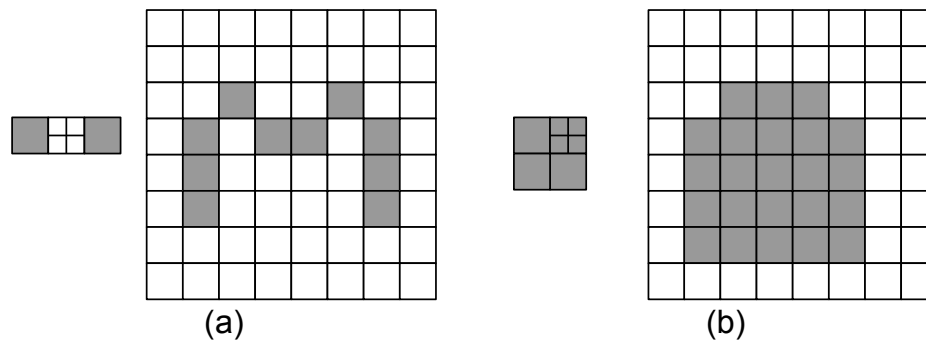
O resultado do processo de dilatação é dependente do elemento estruturante utilizado. Na Figura 16 é possível ver o efeito de dois elementos estruturantes diferentes sendo aplicados sobre a mesma imagem de origem da Figura 15 (a).

Figura 15 – Demonstração do processo de dilatação



Fonte: elaborado pelo autor

Figura 16 – Elementos estruturantes na dilatação



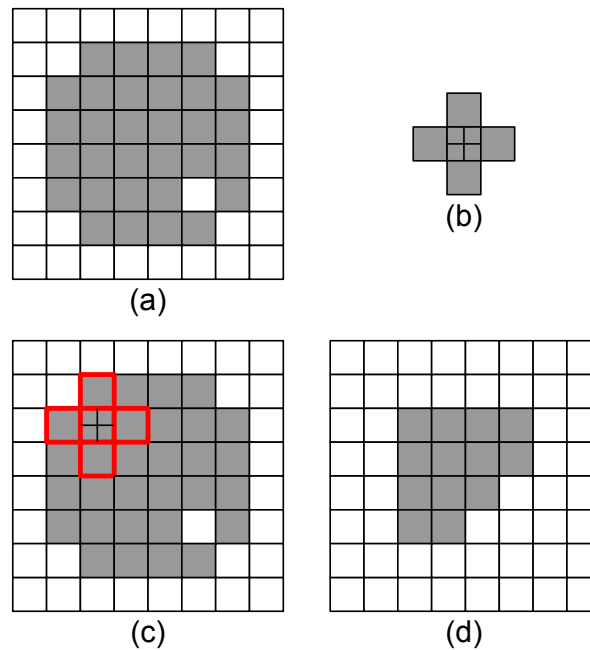
Fonte: elaborado pelo autor

A operação de erosão entre o conjunto A e o elemento estruturante B é definida como a subtração de Minkowski (1911), ou seja

$$\varepsilon(A, B) = A \ominus B = \bigcap_{b \in B} (A - b)$$

De acordo com esta equação, o resultado da erosão pode ser calculado como o conjunto de pixels, tal que o elemento estruturante B , transladado com respeito a cada um dos pixels dos objetos na imagem A , corresponde ao conjunto de pixels dos objetos em A . Assim, os pixels que não correspondem ao padrão definido pelo elemento estruturante não pertencerão ao resultado. O processo de erosão está descrito na Figura 17.

Figura 17 – Demonstração do processo de erosão



Fonte: elaborado pelo autor

3.5.2 Abertura e Fechamento

Duas outras operações morfológicas importantes em manipulação de imagens são a abertura e o fechamento. A abertura de A por B é definida como $A \circ B = (A \ominus B) \oplus A$ (PEDRINI; SCHWARTZ, 2008). Isso significa que a abertura é a erosão de A por B , seguida da dilatação da imagem resultante por A .

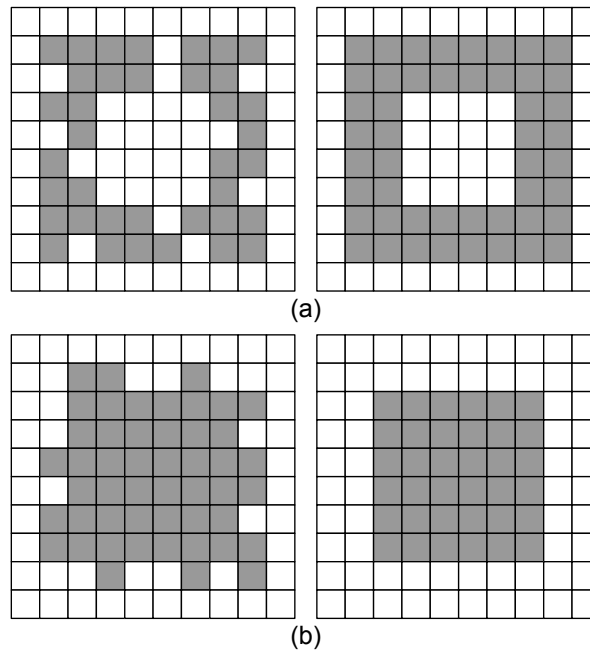
O fechamento de A por B é definido como $A \bullet B = (A \oplus B) \ominus A$ (PEDRINI; SCHWARTZ, 2008). Ou seja, é realizada a dilatação de A por B , seguida da erosão da imagem resultante por B . Na Figura 18 é possível visualizar o efeito dos processos de fechamento (a) e abertura (a) em uma imagem binária utilizando um elemento estruturante de vizinhança-8.

3.6 FILTRAGEM DA IMAGEM

Conforme Azevedo e Conci (2003), o uso de filtros em imagens objetiva melhorar a qualidade das imagens através da amplificação do seu contraste, eliminação de padrões periódicos ou aleatórios (como ruídos ou imperfeições das imagens), melhoria no seu foco e acentuação de características. Os filtros podem ser classificados como passa-baixa e passa-alta.

Para a aplicação de filtros em imagens pode ser utilizado um processo denominado convolução. Conforme (PEDRINI; SCHWARTZ, 2008), a convolução é um processo que envolve a aplicação de uma máscara quadrada w de tamanho n por n . Cada posição da máscara possui um valor numérico denominado peso ou coeficiente. Cada pixel $f(x, y)$

Figura 18 – Demonstração de abertura e fechamento



Fonte: elaborado pelo autor

da imagem é substituído por um novo valor, que depende do valor dos pixels vizinhos e dos pesos da máscara. Os coeficientes da máscara são multiplicados pela intensidade dos vizinhos correspondentes e então somados, resultando no valor do pixel central.

Denotando os níveis de cinza da imagem sob a máscara por $z_i = f(x, y)$, $1 \leq i \leq 9$, a resposta da máscara é $R = w_1.z_1 + w_2.z_2 + \dots + w_9.z_9 = \sum_{i=1}^9 w_i.z_i$ em que w_i representa os coeficientes das máscaras. Um exemplo de aplicação de convolução pode ser visualizado na Figura 19, onde (a) representa a máscara w e (b) representa a imagem de origem. Em (c), (d) e (e) representam os passos da convolução para o pixel (2, 2).

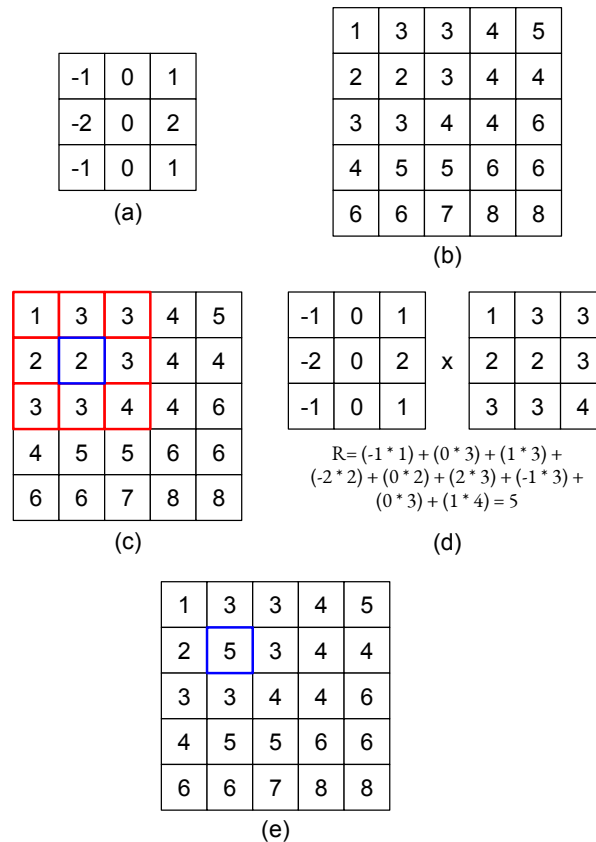
3.6.1 Filtros passa-baixas

São denominados filtros passa-baixas aqueles que atenuam ou eliminam os componentes de alta-frequência enquanto deixam as frequências baixas inalteradas. Os componentes de alta-frequência caracterizam bordas e outros detalhes finos de uma imagem, de forma que o efeito resultante da filtragem passa-baixas é o borramento ou desfoque da imagem (GONZALEZ; WOODS, 2012). Existem diversos filtros de classe passa-baixas, porém, neste trabalho, serão apresentados apenas os mais utilizados.

3.6.1.1 Filtro da Média e Mediana

Um dos filtros passa-baixa mais simples é o filtro da média, em que cada pixel é substituído pelo valor médio de seus vizinhos. O filtro da média é um filtro passa-baixa utilizado para fins de suavização de imagens (PEDRINI; SCHWARTZ, 2008). Na Figura 20

Figura 19 – Exemplo do processo de convolução



Fonte: elaborado pelo autor

estão representadas máscaras de filtro da média de tamanho 3x3 e 5x5, respectivamente.

Figura 20 – Exemplo de máscaras de convolução de média

$$h_1 = \begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix} \quad h_2 = \begin{bmatrix} \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \end{bmatrix}$$

Fonte: adaptado de Pedrini e Schwartz (2008), pg. 328

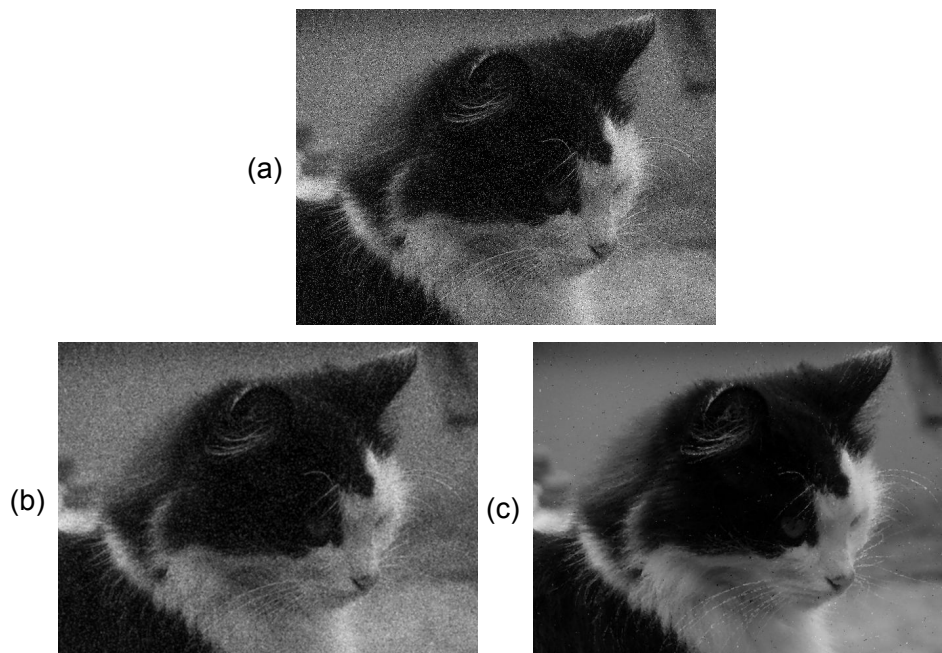
O filtro de média é considerado um filtro linear, pois ele efetua a suavização homogênea de detalhes finos e bordas da imagem, causando uma perda de detalhes. Existe outra classe de filtros denominada filtros não-lineares que procura evitar esta perda (PEDRINI; SCHWARTZ, 2008).

Um filtro não-linear bastante empregado no processamento digital de imagem é o filtro de mediana, que consiste em substituir a intensidade de cada pixel pela mediana das intensidades na vizinhança do pixel. O filtro de mediana não utiliza a convolução, mas sim ordena a intensidade da vizinhança e utiliza a intensidade do pixel que estiver

na posição central. Para uma vizinhança de $n \times n$ pixels, sendo n ímpar, a mediana das intensidades ordenada encontra-se na posição $\frac{n^2+1}{2}$. Caso n seja par, é necessário realizar a média aritmética dos dois elementos mais próximos ao centro (GONZALEZ; WOODS, 2012).

Seus resultados são geralmente melhores que o filtro da média. Isso ocorre devido ao fato de que, se existe um ruído entre os elementos da máscara, este valor estará presente nas primeiras ou últimas posições da vizinhança ordenada. Assim, valores de intensidade discrepantes serão completamente desconsiderados, ao invés de impactar no valor da intensidade (AZEVEDO; CONCI, 2003). Na Figura 21 é possível ver a comparação entre os filtros de média (b) e mediana (c) para a remoção do ruído do tipo sal e pimenta da imagem original (a).

Figura 21 – Demonstração dos filtros de média e mediana



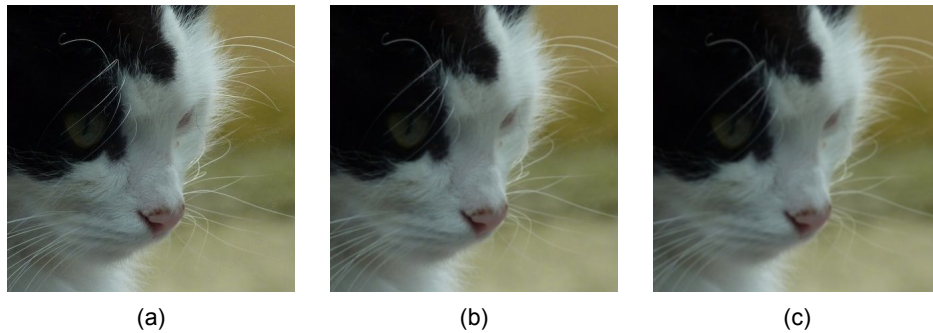
Fonte: elaborado pelo autor

3.6.1.2 Filtro Gaussiano

Segundo Azevedo e Conci (2003), o filtro de Gauss ou Gaussiano é o filtro linear passa-baixa mais importante. Assim como outros filtros dessa classe, é utilizado para reduzir a quantidade de variação de intensidade entre um pixel e seus vizinhos, efetuando a redução de ruídos e outras trocas bruscas de frequência da imagem. Na Figura 22 é possível ver a aplicação do filtro Gaussiano em uma imagem (a) com máscaras de tamanho 3×3 (b) e 5×5 (c).

Segundo Pedrini e Schwartz (2008), os coeficientes da máscara de processamento do filtro Gaussiano são obtidos através de uma função Gaussiana bidimensional. A função

Figura 22 – Demonstração do Filtro de Gauss



Fonte: elaborado pelo autor

Gaussiana com média zero e desvio padrão σ é descrita por

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

A aplicação do filtro Gaussiano pode ser feita com operações de convolução, então é necessário transformar a função gaussiana em estrutura de uma máscara. O tamanho da máscara que descreve o filtro de Gauss é muito grande, mas é possível descartar os valores que estão além de três posições do centro (AZEVEDO; CONCI, 2003). Duas máscaras desse filtro estão exemplificadas abaixo, ambas com desvio-padrão igual a 1.

$$Z = \frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 5 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} \quad Z = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

3.6.2 Filtros passa-altas

Os filtros passa-altas atenuam ou eliminam os componentes de baixa-frequência, mantendo frequências altas. Como as baixas frequências são responsáveis pelas características que variam lentamente em uma imagem, como o contraste total e a intensidade média, o efeito resultante da filtragem passa-altas é uma redução destas características, correspondente a uma aparente agudização de bordas, trocas repentinas de intensidade e também ruídos (GONZALEZ; WOODS, 2012). Existem inúmeros filtros do tipo passa-alta, alguns destes sendo apresentados neste trabalho.

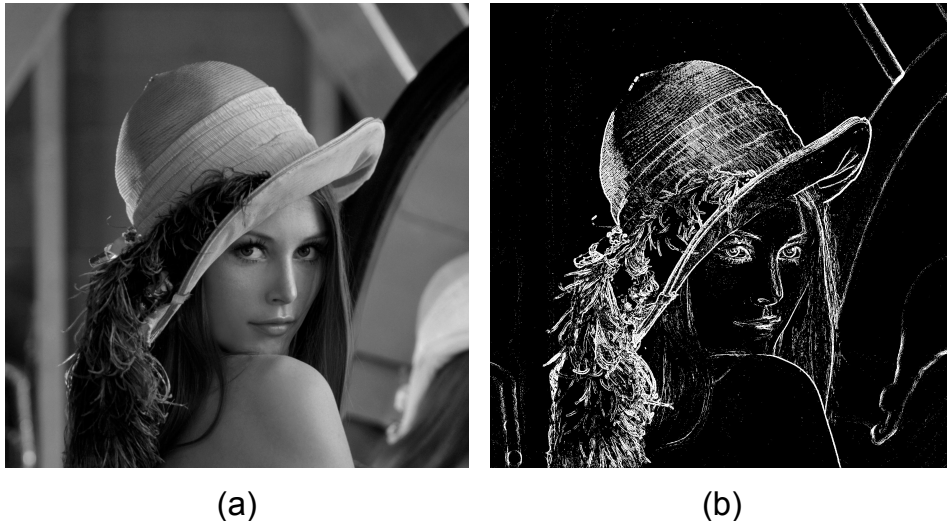
3.6.2.1 Operador de Sobel

O operador de Sobel foi proposto por Sobel e Feldman (1968). Ele é composto por duas máscaras que aproximam a magnitude do gradiente com as diferenças dos níveis de cinza da imagem. Os dois gradientes representam, respectivamente, as variações de

Tabela 3 – Máscaras do operador de Sobel

-1	0	1	-1	-2	-1
-2	0	2	0	0	0
-1	0	1	1	2	1
G_x			G_y		

intensidade no eixo horizontal (G_x) e vertical (G_y). Ao efetuar operações de convolução dessas máscaras com a imagem, as variações de intensidade dos níveis de cinza em cada eixo são detectadas. O resultado final da convolução é dado por $\sqrt{G_x^2 + G_y^2}$. Este valor resultante deve então ser aplicado a um limiar, identificando se o ponto é ou não a borda (PEDRINI; SCHWARTZ, 2008). Um exemplo da aplicação do filtro de Sobel pode ser visto na Figura 23 com um limiar de 40.

Figura 23 – Demonstração do Filtro de Sobel

Fonte: elaborado pelo autor

3.6.2.2 Operador de Canny

O detector de bordas de Canny é mais complexo, mas geralmente superior aos demais métodos passa-alta (GONZALEZ; WOODS, 2012). Ele foi proposto por Canny (1987) e é composto por 4 etapas básicas: suavização da imagem de entrada com a aplicação do filtro gaussiano; cálculo da magnitude do gradiente e ângulos das imagens; identificação dos pontos de máxima ou mínima intensidade de acordo com o gradiente; e, por fim, é definido o limiar e feita a análise de conectividade dos pixels classificados como borda (GONZALEZ; WOODS, 2012). Um exemplo do operador de Canny com limiar de 25 pode ser visto na Figura 24.

Figura 24 – Demonstração do Filtro de Canny



Fonte: elaborado pelo autor

3.7 TRANSFORMAÇÕES GEOMÉTRICAS

Conforme Azevedo e Conci (2003), as transformações geométricas são operações realizadas em imagens digitais que movem os pixels da imagem de origem entre uma posição (x_o, y_o) para a posição (x_d, y_d) da imagem de destino, ou seja, modificam a posição dos pixels no espaço de imagem. As principais transformações geométricas são: translação, mudança de escala, rotação e espelhamento.

Para realizar a conversão de coordenadas é utilizado um sistema de coordenadas homogêneo, ou seja, de dimensão $n+1$. No caso de imagens bidimensionais, é acrescentada a terceira coordenada w com o valor 1 aos pontos (x, y) . Com o sistema de coordenadas homogêneas, todas as transformações geométricas principais são tratadas através de multiplicação de matrizes.

3.7.1 Translação

Transladar um objeto significa deslocar, mover ou somar a cada um de seus pontos (x_o, y_o) uma quantidade (T_x, T_y) . A translação 2D na forma matricial em coordenadas homogêneas é definida pela equação

$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ 1 \end{bmatrix}$$

3.7.2 Escala

A mudança de escala significa variar o seu tamanho, de forma semelhante a um *zoom in* ou *zoom out*. A operação básica consiste em multiplicar cada um de seus pontos (x_o, y_o) por um fator escalar (S_x, S_y) . A escala 2D na forma matricial em coordenadas homogêneas é definida pela equação

$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ 1 \end{bmatrix}$$

O fato de os valores de x e y serem inteiros (pois representam as coordenadas da imagem) pode gerar problemas. Considerando, por exemplo, um escalar de $(2, 2)$. O pixel da posição $(1, 1)$ irá virar $(2, 2)$, enquanto o pixel da posição $(2, 2)$ irá virar $(4, 4)$. Isso deixará o pixel $(3, 3)$ sem valor de origem. Isso pode ser corrigido realizando uma operação de interpolação linear ou bilinear entre os pontos existentes, dando assim o valor dos pontos intermediários.

3.7.3 Rotação

A rotação utilizando um ângulo relativo à origem consiste em encontrar, para cada pixel (x_o, y_o) um outro pixel (x_d, y_d) sobre uma circunferência centrada na origem que passa pelos dois pontos com ângulo $\theta = p\hat{o}q$.

A rotação da imagem sobre o eixo Z na forma matricial em coordenadas homogêneas é definida pela equação

$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ 1 \end{bmatrix}$$

A rotação da imagem sobre o eixo X na forma matricial em coordenadas homogêneas é definida pela equação

$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ 1 \end{bmatrix}$$

A rotação da imagem sobre o eixo Y na forma matricial em coordenadas homogêneas é definida pela equação

$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ 1 \end{bmatrix}$$

3.7.4 Espelhamento

Conforme Azevedo e Conci (2003), a transformação de espelhamento (também chamada de reflexão ou *flip*) em torno de um eixo é uma operação que combina a rotação por ângulos múltiplos de 90° com a inversão das coordenadas. Um espelhamento horizontal é uma rotação de 180° no sentido anti-horário com coordenadas y_o invertidas. Já o *flip* vertical é uma rotação de 180° no sentido horário com os valores das coordenadas x_o invertidos. Em coordenadas homogêneas o *flip* horizontal é obtido por

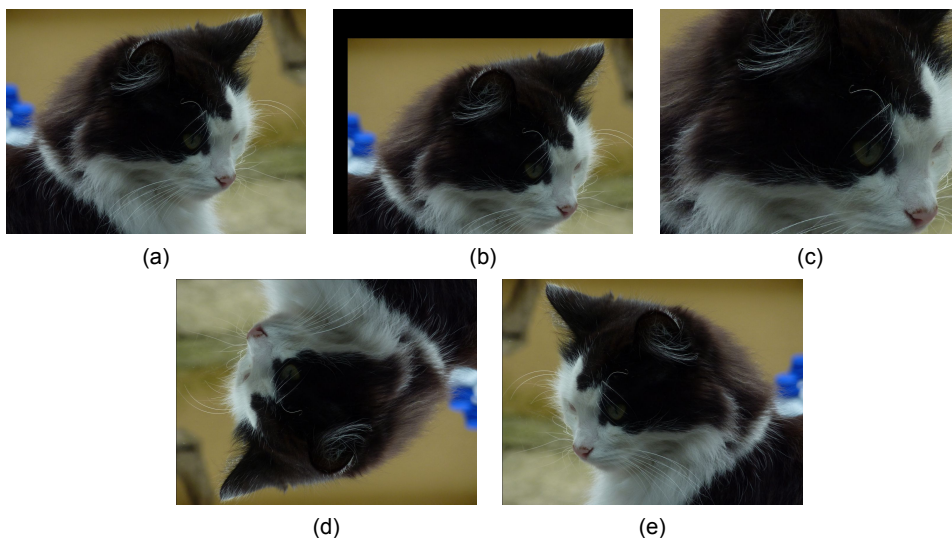
$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ 1 \end{bmatrix}$$

enquanto o vertical é dado por

$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ 1 \end{bmatrix}$$

Na Figura 25 é possível visualizar o efeito das transformações geométricas. Em (a) está representada a imagem original. Em (b) a imagem está transladada utilizando $T_x = 50$ e $T_y = 100$. Em (c) foi aplicada uma escala com $S_x = 1,6$ e $S_y = 1,6$. Em (d) a imagem foi rotacionada utilizando $\theta = 180$. E em (e), foi aplicado um espelhamento horizontal na imagem.

Figura 25 – Exemplo de transformações geométricas



Fonte: elaborado pelo autor

3.7.5 Contornos ativos

O modelo de contornos ativos, também chamado de *active contour model* ou ainda Snakes, é um framework em visão computacional proposto por Kass, Witkin e Ter-

zopoulos (1988) para delinear o contorno de um objeto em uma imagem possivelmente ruidosa. O algoritmo consiste em uma linha deformável (*snake*) onde a curva sofre deformações com base em energias que empurram a linha contra o centro da imagem, enquanto as forças internas do objeto mantêm a linha se deformando ao redor dos contornos do objeto.

Uma linha deformável simples pode ser definida como um conjunto de n pontos v_i onde $i = 0 \dots n - 1$, a energia elástica interna $E_{interna}$, e a energia externa do contorno $E_{externa}$. O propósito da energia interna é controlar as doformações feitas a linha, e o propósito da energia externa é controlar o encaixe do contorno sobre o objeto.

A função de energia da curva é a soma da sua energia externa e energia interna, ou

$$E_{snake}^* = \int_0^1 E_{snake}(\mathbf{v}(s)) ds = \int_0^1 (E_{interna}(\mathbf{v}(s)) + E_{image}(\mathbf{v}(s)) + E_{con}(\mathbf{v}(s))) ds$$

A energia interna da curva é composta pela continuidade do contorno E_{cont} e a suavidade do contorno E_{curv} .

$$E_{interna} = E_{cont} + E_{curv}$$

Isto pode ser expandido para

$$E_{interna} = \frac{1}{2}(\alpha(s) |\mathbf{v}_s(s)|^2) + \frac{1}{2}(\beta(s) |\mathbf{v}_{ss}(s)|^2) = \frac{1}{2} \left(\alpha(s) \left\| \frac{d\bar{v}}{ds}(s) \right\|^2 + \beta(s) \left\| \frac{d^2\bar{v}}{ds^2}(s) \right\|^2 \right)$$

onde $\alpha(s)$ e $\beta(s)$ são pesos definidos de forma empírica. Estes parâmetros controlam a sensibilidade da função de energia interna a quantidade de elasticidade e a quantidade de curvatura do *snake* respectivamente, e portanto controlam o número de limites do formato da curva. Na prática, um peso elevado para $\alpha(s)$ penaliza mudanças nas distâncias entre pontos no contorno. Já um peso elevado para $\beta(s)$ penaliza oscilações no contorno.

A energia na imagem é uma função sobre suas características. Esse é o ponto de modificação mais comum para métodos derivados. As características da imagem podem ser processadas de várias formas. Para uma imagem $I(x, y)$, linhas, bordas e terminações presentes na imagem, a formulação generalizada para a energia em relação a imagem é

$$E_{image} = w_{line} E_{line} + w_{edge} E_{edge} + w_{term} E_{term}$$

onde w_{line} , w_{edge} e w_{term} são os pesos destas características. Pesos maiores indicam que esta característica terá uma contribuição maior para a energia da imagem.

A função para a linha é a intensidade da imagem, que pode ser representada como $E_{line} = I(x, y)$. O sinal desta função determinará se a linha será atraída para linhas escuras ou claras. A função para as bordas é baseada no gradiente da imagem, e pode ser representada como $E_{edge} = -|\nabla I(x, y)|^2$. Por fim, a função da terminação pode ser definida como

$$E_{term} = \frac{C_{yy}C_x^2 - 2C_{xy}C_xC_y + C_{xx}C_y^2}{(C_x^2 + C_y^2)^{3/2}}$$

onde $C(x, y)$ é a imagem de entrada suavizada por um coeficiente.

3.7.6 Afnamento de Bordas

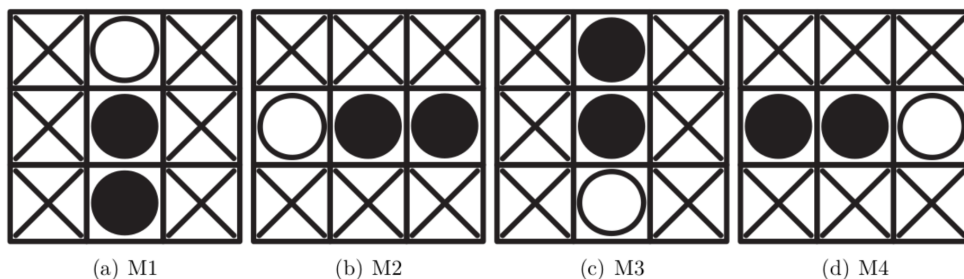
Conforme Guilherme (2007) o processo de afinamento de bordas ou esqueletização consiste na remoção de pixels redundantes da imagem para a criação do seu esqueleto. O esqueleto de uma imagem é a representação básica das formas contidas na imagem, de forma que todos os pixels são necessários. Os algoritmos de esqueletização sempre recebem como entrada uma imagem binarizada.

Uma das características mais importantes que define um bom algoritmo de esqueletização é a preservação das características das formas da imagem original. Estas características englobam posição, orientação, tamanho e conectividade (GUILHERME, 2007).

3.7.6.1 Stentiford

O algoritmo de Stentiford foi proposto por Stentiford e Mortimer (1983), e adota uma abordagem baseada na utilização de máscaras para o afinamento de objetos. São utilizadas quatro máscaras que são aplicadas de forma sucessiva e ordenada (GUILHERME, 2007). As máscaras são utilizadas para encontrar pixels na imagem que coincidam com um determinado padrão, sendo então marcados para remoção.

Figura 26 – Máscaras utilizadas pelo algoritmo de Stentiford



Fonte: Guilherme (2007, p.13)

Na Figura 26 são demonstradas as máscaras utilizadas pelo algoritmo, onde os círculos brancos representam pixels de valor zero, os círculos pretos representam os pixels

de valor um e os X representam os pixels com valores irrelevantes. Cada máscara percorre a imagem em uma determinada ordem: M1 da esquerda para a direita, de cima para baixo; M2 da esquerda para a direita, de baixo para cima; M3 da direita para a esquerda, de baixo para cima; e M4 da direita para a esquerda, de cima para baixo.

Para a aplicação da máscara, são percorridos os pixels da imagem e é carregado a sua vizinhança. Caso a vizinhança seja compatível com a máscara em processamento, seu valor será alterado para zero (fundo). Caso contrário, o pixel continuará com seu valor original um.

3.7.6.2 Zhang Suen

O algoritmo de Zhang Suen (ZHANG; SUEN, 1984) tem como base a comparação do pixel em processamento com seus 8 vizinhos. A exclusão de pixel por parte do algoritmo somente é realizada mediante a algumas regras. Estas regras têm como objetivo obter a exclusão segura dos pixels, garantindo, desta forma, que áreas interligadas não percam a conectividade e que a eliminação ocorrerá nas bordas do objeto (GUILHERME, 2007).

O algoritmo de Zhang Suen percorre cada pixel da imagem, tentando identificar se ele deverá ou não ser marcado para exclusão (trocar de branco para preto, considerando que preto é a cor do fundo). Para que um pixel possa ser excluído, a sua vizinhança-8 deve se encaixar nestas quatro regras:

- Possuir sua conectividade maior que 1;
- O objeto deve ser composto de pelo menos dois e no máximo seis pixels pretos;
- Ao menos um dos pixels da máscara 1 (Máscara (a) da Figura 27) deve ser branco;
- Ao menos um dos pixels da máscara 2 (Máscara (b) da Figura 27) deve ser branco;

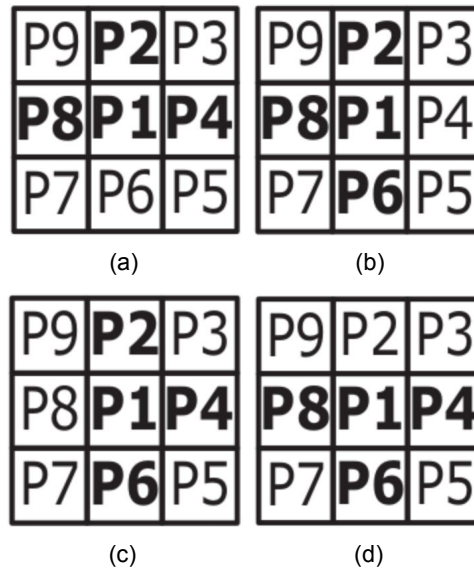
Após o processamento ser concluído para todos os pixels, este é executado uma segunda vez. No entanto, nos últimos dois passos, são utilizadas as máscaras 3 e 4, respectivamente, demonstradas pelas máscaras (c) e (d) da Figura 27.

3.7.6.3 Holt

O algoritmo de Holt (HOLT et al., 1987) utiliza uma vizinhança 3 x 3 para a análise dos pixels a serem removidos. O formato da matriz permite que seja feita uma análise do pixel central e seus vizinhos. Segundo Guilherme (2007), o algoritmo de Holt é composto por duas expressões lógicas, onde uma é aplicada na primeira iteração do algoritmo e a outra na segunda iteração.

$$\text{Primeira iteração: } v(C) \wedge (\sim \text{edge}(C) \vee (v(L) \wedge v(S) \wedge (v(N) \vee v(O))))$$

Figura 27 – Máscaras utilizadas pelo Zhang Suen



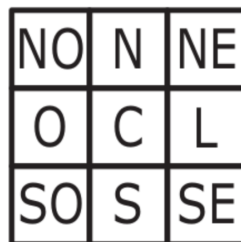
Fonte: Guilherme (2007, p.16)

Segunda iteração: $v(C) \wedge (\sim edge(C) \vee (v(O) \wedge v(N) \wedge (v(S) \vee v(L))))$

Para um ponto ser removido, o resultado das expressões lógicas devem ser verdadeiros. Nas expressões, os termos C , O , N , S e L representam os vizinhos da imagem conforme definido na Figura 28. Já $edge()$ e $v()$ representam funções definidas por:

1. $v()$: Retorna verdadeiro se o valor do ponto for o mesmo valor do objeto (valor preto) e falso se o valor do ponto for igual ao valor do plano de fundo (valor branco)
2. $edge()$: Retorna verdadeiro se o valor processado estiver na borda do objeto e falso se não estiver
 - 2.i. Um pixel está na borda quando sua conectividade é igual a 1 e quando possuir de 2 a 6 vizinhos conectados

Figura 28 – Janela utilizada para a análise dos pixels vizinhos no algoritmo de Holt



Fonte: Guilherme (2007, p.21)

3.8 CONSIDERAÇÕES

Este capítulo apresentou o tema de processamento digital de imagens. Foi apresentado o conceito de imagem digital. Além disso, foram apresentadas diversas técnicas de pré-processamento e segmentação, como *thresholding*, detecção de bordas e morfologia matemática.

No próximo capítulo serão apresentadas trabalhos que tentaram unir o uso de algoritmos genéticos com PDI para realizar a segmentação de imagens digitais. Nos trabalhos indicados, o GA foi utilizado apenas para parametrizar as técnicas de PDI para alcançar o objetivo especificado.

4 TRABALHOS CORRELATOS

Neste capítulo serão apresentados trabalhos relacionados a área de pesquisa deste projeto.

4.1 O USO DO ALGORITMO GENÉTICO EM SEGMENTAÇÃO DE IMAGENS DIGITAIS

A tese desenvolvida por Matias (2007) aplica algoritmos genéticos com diferentes funções de avaliação para evoluir a parametrização de três métodos de segmentação de imagens digitais, comparando seu resultado. Os algoritmos de segmentação analisados são *quadtrees*, limiarização e crescimento de regiões.

Todos os métodos visam minimizar uma função objetivo (entropia, segmentação excedente, entre outras) ou maximizar tal função (pixels corretamente agrupados). Para a avaliação dos resultados, foram geradas várias imagens que utilizam o GA no algoritmo de segmentação e outras imagens geradas com o mesmo algoritmo, mas sem utilizar o GA.

O método mais simples trabalhado por Matias (2007) é a segmentação utilizando o limiar (*threshold*). O objetivo dos AGs, neste tipo de algoritmo, é gerar de forma automática uma imagem segmentada por atribuição de um *threshold*. A função de avaliação utilizada para este algoritmo foi a entropia, que representa a aleatoriedade das tonalidades da imagem. Para a segmentação baseada em limiar a representação cromossômica foi utilizado um cromossomo de 8 genes binários, representando o valor do *threshold* entre 0 e 255.

Uma imagem utilizada para testes foi uma foto de 256x256 pixels denominada Lena. Para treinamento do GA foi utilizada uma população de 25 indivíduos, com uma probabilidade de mutação de 0.05 e uma probabilidade de *crossover* de 0.6. O número máximo de gerações delimitado para o GA foi de 1000 gerações. A segmentação com a melhor função objetivo obtida por este método está exemplificada na Figura 29.

Também foram realizados os mesmos testes utilizando uma foto de satélite do Rio Solimões em época de seca em alguns de seus afluentes. Utilizando os mesmos parâmetros do GA descritos anteriormente, a segmentação com melhor avaliação está demonstrada na Figura 30, onde também é apresentado o uso do limiar ótimo descrito (GONZALEZ; WOODS, 2012) (c).

Outro método implementado pela teste foi o método de segmentação por quadrantes, ou *quadtree*. Neste método, o algoritmo genético foi utilizado para a geração dos

Figura 29 – Resultado de limiarização na imagem Lena

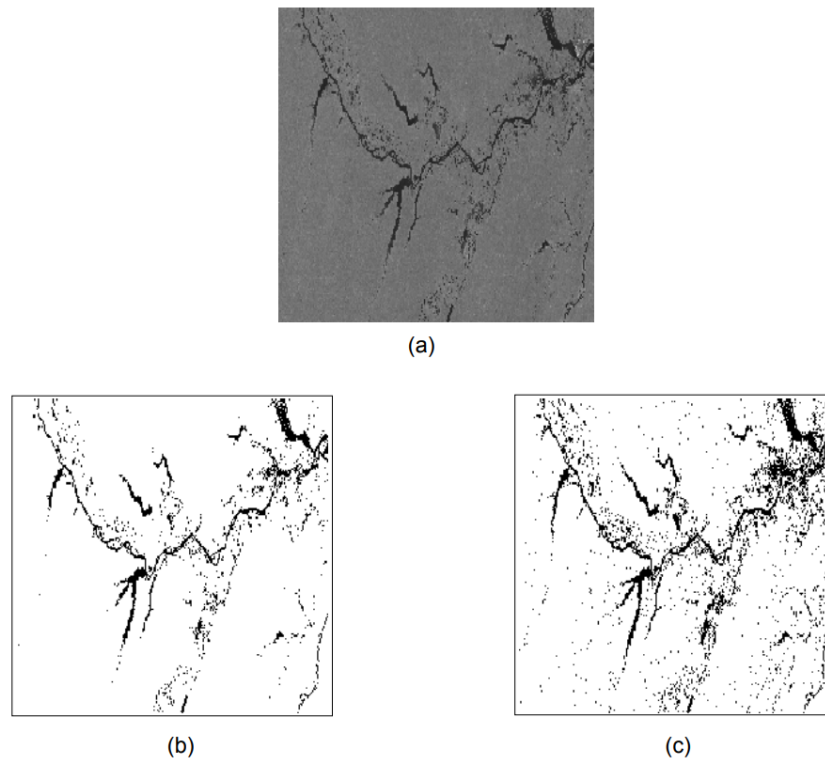


(a)

(b)

Fonte: Matias (2007, p.71)

Figura 30 – Resultado de limiarização na imagem Rio Solimões



(a)

(b)

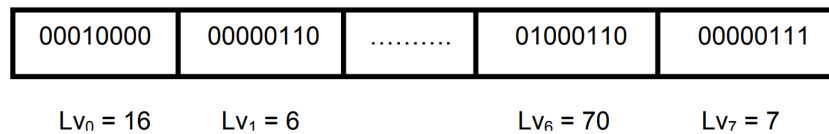
(c)

Fonte: Matias (2007, p.72)

valores de entropia de cada nível da *quadtree*. Em uma imagem de resolução $N \times N$ pixels, o número de níveis representados por cada cromossomo é $L_v = \log_2 N$. Cada valor do nível L_{v_n} representado no cromossomo indica o limiar de entropia daquele quadrante para que o mesmo seja subdividido novamente. O exemplo do cromossomo pode ser visto na Figura 31.

Para validação da técnica de segmentação por *quadtrees* foi utilizada a foto de satélite do Rio Solimões. A resolução da imagem é 256×256 , o que gera uma *Quadtree* de 8 níveis. Para esta imagem foi executado o GA com uma população inicial aleatória de 300 indivíduos. A probabilidade de mutação utilizada foi de 0,1 e a probabilidade de *crossover*

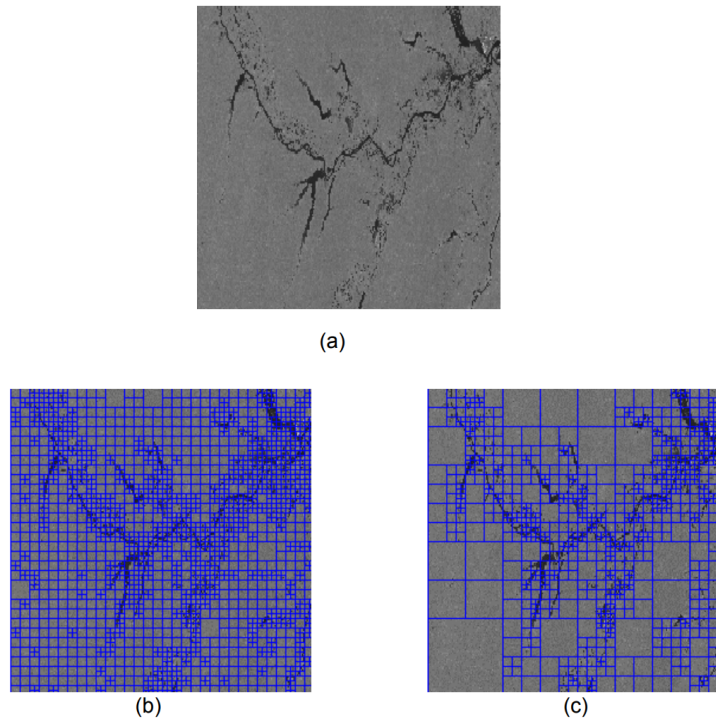
Figura 31 – Cromossomo binário onde L corresponde a cada nível da quadtree



Fonte: Matias (2007, p.54)

foi de 0,6. A GA foi executada por um total de 9 gerações. O resultado da segmentação pode ser visto na Figura 32. É possível identificar que a segmentação utilizando *quadtrees* (c) e GAs foi mais eficiente que o método usual (b), gerando maiores áreas homogêneas, o que pode ser útil para a compressão de imagens.

Figura 32 – Comparação da segmentação via *quadtree*



Fonte: Matias (2007, p.68)

Outro método de segmentação de imagens testado por Matias (2007) foi o método de crescimento de regiões proposto por Bins et al. (1996), um método que é direcionado a imagens de sensoriamento remoto. Este algoritmo utiliza dois parâmetros de entrada: o limiar de similaridade, utilizado para agrupar os pixels que estejam dentro deste valor, e a área máxima das regiões, para evitar problemas de *under-segmentation*.

A representação cromossômica deste método são os dois parâmetros de entrada concatenados em uma *string* de K bits, onde $K = \log_2 T + A$, sendo T o número máximo de tons de cinza da imagem e A um número suficiente de bits para armazenar a área máxima, neste caso $A = 16$. Para a validação do método, ao invés da entropia da imagem, foram utilizadas três métricas de avaliação de algoritmos de segmentação: CG (*Correctly Grouped*), US (*Undersegmentation*) e OS (*Oversegmentation*) (ROLI, 2005).

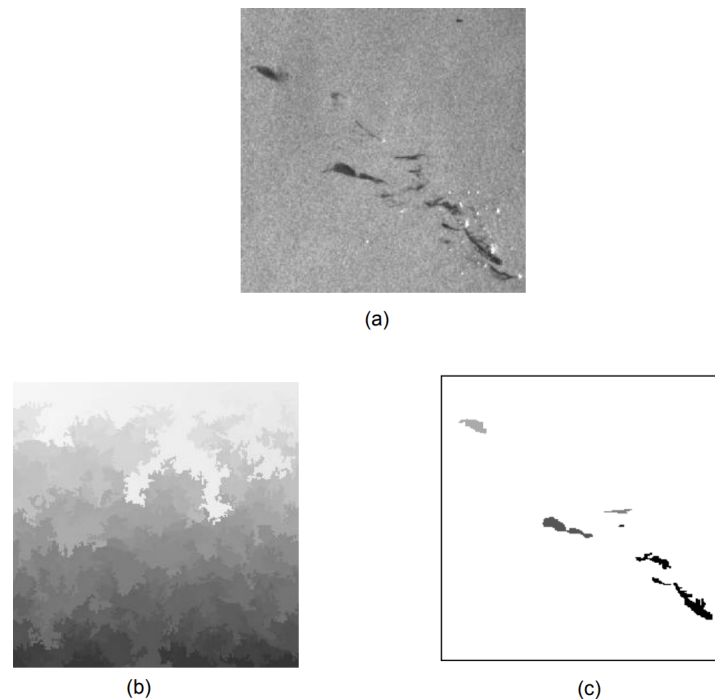
Tabela 4 – Resultados da segmentação

<i>Caso</i>	<i>CG</i>	<i>US</i>	<i>OS</i>
Sem GA	10,17%	18,33%	82,21%
Com GA	90,44%	3,74%	4,71%

Fonte: Matias (2007)

Para a validação do processo de segmentação foi utilizada uma foto de satélite de uma mancha de petróleo no mar, de resolução 256x256. Para a simulação do GA foi utilizada uma população de tamanho de 25 indivíduos aleatórios. A probabilidade de mutação utilizada foi de 0.05 e a probabilidade de *crossover* foi de 0,6. O número máximo de gerações do GA foi de 1000. Ao final do processamento, foi obtida a saída representada na Figura 33. Os valores do método de avaliação para a imagem segmentada sem GA (b) e com GA (c) pode ser visto na Tabela 4.

Figura 33 – Comparação da segmentação via crescimento de regiões



Fonte: Matias (2007, p.68)

Dessa maneira, Matias (2007) verificou que o uso do GA com uma função objetivo adequada pode trabalhar muito bem em conjunto com algoritmos de segmentação, na busca de bons parâmetros de entrada iniciais. Isso foi comprovado nos três algoritmos testados: limiarização, *quadtrees* e crescimento de regiões.

4.2 A MULTILEVEL AUTOMATIC THRESHOLDING METHOD BASED ON A GENETIC ALGORITHM FOR A FAST IMAGE SEGMENTATION

Neste trabalho, Hammouche, Diaf e Siarry (2008) propõem uma técnica de *thresholding* multinível para a segmentação de imagens baseada no uso de algoritmos ge-

néticos para determinar os limiares apropriados. Segundo os autores, o uso de GAs tem várias vantagens sobre métodos de otimização tradicionais, particularmente a capacidade de encontrar a solução ótima global, prevenindo que o algoritmo fique preso em soluções localmente ótimas. Ademais, algoritmos genéticos podem ter uma grande redução no tempo de processamento com implementações paralelas. A técnica de *thresholding* genética proposta é baseada em um GA padrão. Ele permite a determinação do número de limiares, assim como o valor de cada um.

Antes de realizar a busca dos limiares utilizando o GA, o tamanho do histograma da imagem é reduzido de forma a acelerar a convergência do GA. A redução do histograma é feita utilizando a técnica de *wavelet transform* (KIM; SHIM; PARK, 2003). O histograma é reduzido para dois sinais. O primeiro é o sinal de tendência, ou sinal de aproximação, enquanto o segundo é o sinal de detalhe.

No GA proposto, o cromossomo é codificado como uma String binária A com o mesmo comprimento do histograma reduzido L^r , de forma que $A = a_0, a_1, \dots, a_{L^r-1}$, onde o gene a_i possui o valor 0 ou 1, representando se o índice é um pico ou um vale no histograma, respectivamente. A posição i onde $a_i = 0$ indica o valor onde existirá um *threshold*. Dessa forma, o número de *bits* 0 no cromossomo indica o número de limiares.

Para permitir a solução ótima do número de *thresholds* e seus valores, a função objetiva é calculada usando a função de custo proposta por Yen, Chang e Chang (1995). Esta função avalia a entropia de cada classe criada, levando em consideração o número de limiares utilizados. Com este método, é garantida a segmentação boa da imagem utilizando um número tão pequeno quanto possível de limiares.

A população do algoritmo genético é inicializada com uma população de indivíduos gerada aleatoriamente. Para a avaliação do GA, o indivíduo com a melhor função objetivo em cada geração é copiada para uma posição isolada da população de forma a armazenar a melhor solução atual. Então, antes de começar a próxima geração, a melhor solução pode melhorar a sua avaliação realizando diversas pequenas modificações em seus genes. Esta simples técnica pode economizar diversas gerações de aprimoramento e não afeta o comportamento geral do GA.

A população corrente evolui para a próxima geração utilizando as três técnicas padrões de GAs: seleção, *crossover* e mutação. Para a seleção, é utilizada a técnica da roleta viciada. Para o *crossover*, é utilizado o método de ponto único. Também é utilizada a mutação tradicional. Após o *crossover* e mutação, o novo gene pode conter uma sequência de vários *bits* 0 seguidos, o que não é desejável. Para evitar essa situação, é realizado um ajuste no cromossomo de forma que apenas o primeiro zero sucessivo é mantido, trocando os subsequentes pelo valor 1.

Para a validação dos resultados, foram utilizadas algumas imagens comuns para

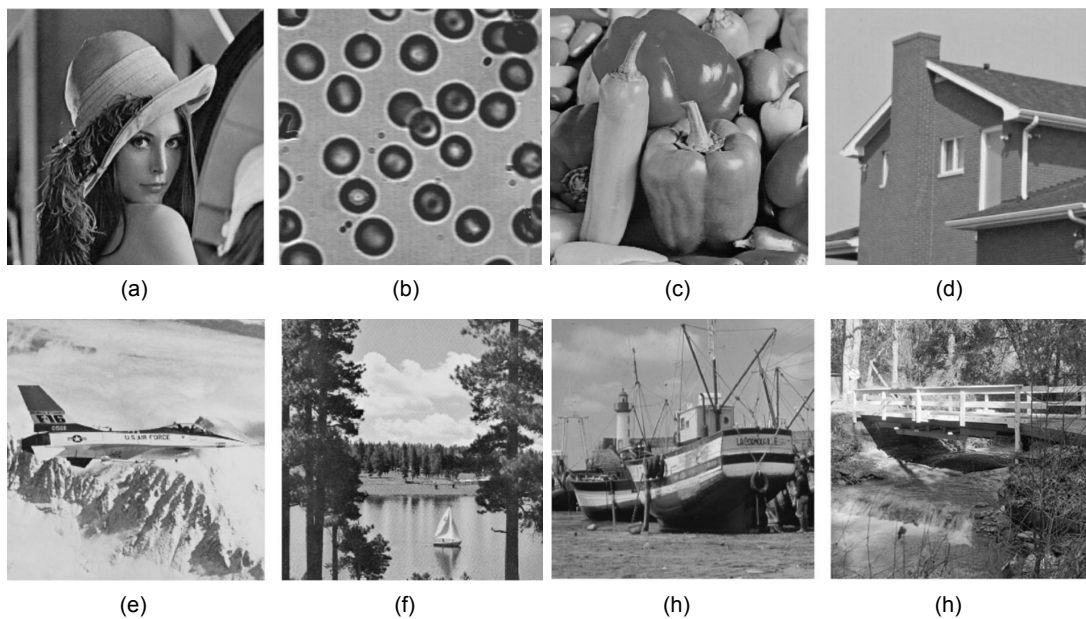
Tabela 5 – Comparação do método proposto com outros métodos semelhantes

<i>Método</i>	<i>Classes</i>	<i>Limiares</i>	<i>FA</i>	<i>Uniformidade</i>	<i>Tempo (ms)</i>
ES-Otsu	5	47-84-119-164	10,80	0,97877	601.766
ES-Kapur	4	60-109-160	11,00	0,97159	50.281
GA-Otsu	5	46-80-113-162	10,82	0,97805	578
GA-Kapur	4	59-107-159	10,97	0,97177	1.118
Iterative-Otsu	5	47-83-118-163	10,80	0,97874	360
Iterative-Kapur	4	60-108-159	10,99	0,97160	1516
Método proposto	5	46-83-119-164	10,80	0,97875	32

Fonte: Hammouche, Diaf e Siarry (2008)

validação de técnicas de processamento de imagens: *Lena*, *Blood*, *Peppers*, *House*, *Airplane*, *Lac*, *Boats* e *Bridge*. Na Figura 34 estão demonstradas as figuras utilizadas tal como o resultado da segmentação das imagens. Os resultados obtidos demonstram a robustez do método proposto, no aspecto de acurácia na segmentação. Uma comparação dos resultados da imagem *Lena* pode ser visto na Tabela 5.

Figura 34 – Imagens utilizadas para teste



Fonte: Adaptado de Hammouche, Diaf e Siarry (2008)

Neste capítulo foram apresentados alguns trabalhos relacionados ao assunto abordado. É possível perceber que o principal foco acadêmico ao tentar unir algoritmos genéticos com processamento digital de imagem é na otimização dos parâmetros dos algoritmos de PDI. Portanto, este trabalho tem como diferencial o uso dos GAs para tentar otimizar não só a parametrização, como também o sequenciamento destes algoritmos. No próximo capítulo será descrita a solução proposta e como esta foi desenvolvida.

5 DESENVOLVIMENTO DA APLICAÇÃO

Neste capítulo será apresentado o desenvolvimento do trabalho proposto. Este capítulo está dividido em quatro seções, cada uma apresentando um aspecto do trabalho. Primeiramente é detalhado o algoritmo genético, como ele foi implementado e parametrizado. Após, é apresentada a interface gráfica que foi desenvolvida para que o usuário possa visualizar o andamento e o resultado da simulação, e interagir com a mesma. Na sequência, são demonstrados os diversos experimentos realizados com diferentes parâmetros do algoritmo genético para obter o melhor resultado possível, e será apresentado este resultado. Por fim, é detalhada a metodologia utilizada para validação do trabalho.

5.1 ALGORITMO GENÉTICO PROPOSTO

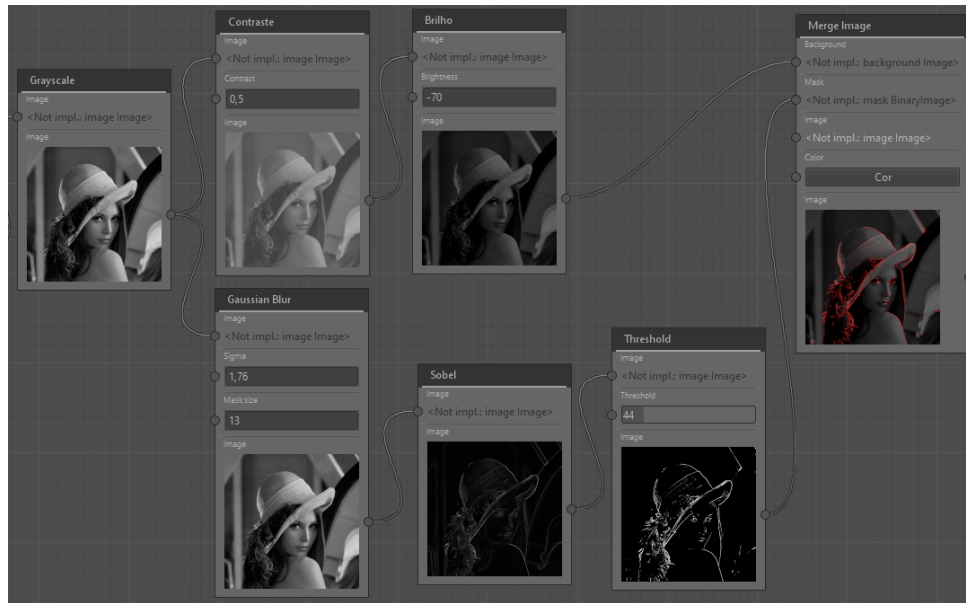
Nesta seção serão apresentados detalhes sobre a implementação do algoritmo genético, técnicas utilizadas e como o mesmo foi parametrizado para a simulação. O algoritmo genético foi desenvolvido utilizando a linguagem de programação Java. A estrutura e lógica do algoritmo genético foi completamente desenvolvido neste trabalho, sem o uso de *frameworks* ou bibliotecas prontas.

Neste trabalho é proposto um sistema de aprendizado de máquina utilizando algoritmos genéticos com técnicas de processamento digital de imagens para a segmentação de imagens, focando na segmentação de fotos de rostos. De forma semelhante aos trabalhos apresentados como correlatos, os GAs serão utilizados para evoluir os parâmetros das técnicas de processamento de imagem. No entanto, o principal objetivo deste trabalho é conseguir que o GA selecione também quais técnicas de PDI e em qual sequência os mesmos serão executados para obter o melhor resultado do processo de segmentação.

Este trabalho se baseia no *software* VISNode (VISNODE, 2018), uma ferramenta para execução de técnicas de PDI em formato de grafo. Ele possui dezenas de algoritmos de PDI implementados que serão utilizados neste trabalho. Cada nodo possui entradas e saídas, sendo que estas podem ser conectadas entre si, garantindo uma maior flexibilidade no sequenciamento e execução dos algoritmos. Um exemplo de um projeto do VISNode pode ser visto na Figura 35. Neste projeto existem dois caminhos de execução, um para gerar uma imagem mais escura e com menos contraste e outro para realizar uma segmentação utilizando o algoritmo de Sobel. Ao final, estes dois caminhos de execução são unificados em uma imagem no nodo mais a direita (*Output*). No GA proposto, cada indivíduo representará um grafo de processos de PDI, que serão executados de forma linear, permitindo a parametrização estática (valor fixo para o parâmetro).

Para a avaliação dos indivíduos do GA e validação do trabalho, está sendo utili-

Figura 35 – Demonstração de um processo feito no VISNode



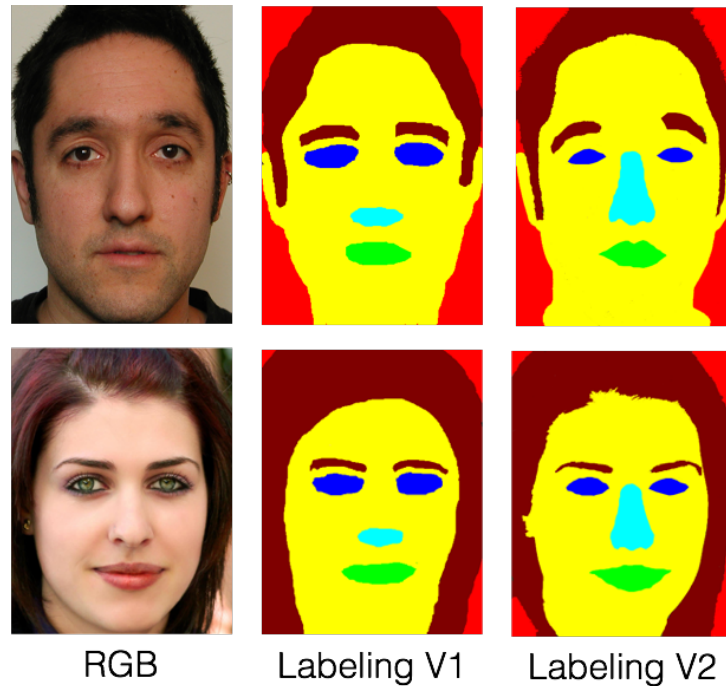
Fonte: elaborado pelo autor

zada a base de rostos segmentados FASSEG (FASSEG, 2018). Esta base possui 70 fotos de rostos normalizados (com posição e iluminação semelhantes) e as imagens já segmentadas delineando o rosto, olhos, cabelo, boca e nariz em cores diferentes. Exemplos de imagens de entrada e segmentadas podem ser vistos na Figura 36, onde estão representadas imagens de entrada (esquerda), imagens segmentadas da versão 1 (meio) e imagens segmentadas da versão 2 (direita). Para este trabalho, serão utilizados os gabaritos da versão 2 por sua qualidade superior. Neste trabalho o objetivo principal é que seja possível segmentar de forma satisfatória o contorno do rosto, ou seja, sem o fundo e sem os cabelos, mas contendo os olhos e a boca.

Pela natureza do problema, foram selecionadas algumas técnicas da literatura para a implementação do algoritmo genético. O modelo escolhido para estruturação do cromossomo foi o híbrido, contendo cromossomos categóricos (processo de PDI a ser executado) e numérico (parametrização do processo). Após alguns experimentos entre as técnicas de *crossover* de um ponto, dois pontos e uniforme, foi selecionado o *crossover* de um ponto pela existência de máscaras de agrupamento dentro do cromossomo que seriam facilmente perdidas com o uniforme.

Além disso, foram implementadas outras heurísticas para tentar acelerar a convergência do GA para o resultado esperado. Uma destas heurísticas foi a sobrevivência garantida do melhor indivíduo, onde ele sempre será copiado para a próxima geração antes de realizar os outros operadores genéticos. Isso garante que o resultado final do algoritmo genético nunca retroceda para um resultado inferior a um resultado já obtido, e pode acelerar na convergência. Além disso, foram realizados testes com uma regra para não permitir um número elevado de indivíduos da mesma espécie, para aumentar a diversi-

Figura 36 – Exemplo de imagens do FASSEG



Fonte: FASSEG (2018)

dade genética e o espectro de soluções que serão avaliadas.

5.1.1 Representação Cromossomial

Para a representação cromossomial dos indivíduos foi optada pela abordagem de representação híbrida. Genes categóricos são utilizados para representar os algoritmos a serem utilizados, e genes numéricos são utilizados para representar os seus parâmetros.

O cromossomo é composto por N grupos de 6 genes. Cada grupo representa um processo de PDI. O primeiro gene do grupo é categórico, representando o algoritmo de PDI. Os próximos 5 genes são numéricos e representam os parâmetros do algoritmo. O valor dos genes é normalizado para o intervalo de 0 a 1, e na hora de construir o processo este valor é transformado para a faixa necessária. Vários processos não utilizam todos os 5 parâmetros numéricos, ou até mesmo nenhum deles.

Esta estrutura pode se repetir quantas vezes for necessário. Foram testadas diferentes variações do número de processos para encontrar aquele que é o mais eficaz.

5.1.2 Parametrização do Algoritmo Genético

O sistema desenvolvido permite que diversos parâmetros do algoritmo genético sejam alterados. Isso foi feito com o intuito de testar e comparar diversas combinações de parâmetros para encontrar a melhor solução possível.

A estrutura de parâmetros foi criada permitindo que cada parâmetro mude ao

longo da simulação. Isto é útil para diversas situações, como, por exemplo, a variação da taxa de mutação e *crossover*. Conforme indicado no Capítulo 2, Linden (2008) indica que o algoritmo genético deve iniciar com uma taxa de *crossover* maior do que a taxa de mutação, e ao longo da simulação esta proporção deve ser invertida.

Para a configuração de parâmetros que variam ao longo da simulação, são definidas gerações-chave onde o modelo de parâmetros mudará. Podemos citar como exemplo a taxa de mutação: na geração 0 é definida uma taxa de mutação de 1%; a partir da geração 5, esta taxa aumenta para 5%; na geração 10, aumenta para 10%, e assim por diante.

No total são seis parâmetros que podem ser configurados para a simulação, cada um podendo variar por geração: tamanho da população, percentual de indivíduos gerados com *crossover*, percentual de mutação, percentual de indivíduos gerados aleatoriamente, probabilidade de mutação e o percentual máximo de indivíduos por espécie.

O tamanho da população é um dos parâmetros básicos da simulação. Ele é um valor inteiro maior que zero, e indica quantos indivíduos existirão em cada geração.

O percentual de indivíduos criados com *crossover* indica quantos cromossomos da próxima geração são criados puramente utilizando o operador genético de *crossover*. O método de *crossover* utilizado é o de um ponto com dois pais. Os dois pais são selecionados utilizando o método da roleta viciada.

O percentual de mutação indica a chance de um cada gene dos indivíduos filhos ser alterado aleatoriamente. Como cada gene de cada indivíduo tem uma chance de sofrer uma mutação, este percentual normalmente é muito baixo, sendo muitas vezes menor do que 1%.

O percentual de indivíduos criados aleatoriamente indica quantos cromossomos são criados de forma totalmente aleatória, como se fosse a primeira geração. Isso pode ser feito para garantir que o algoritmo genético continue explorando soluções completamente novas mesmo após uma redução na sua diversidade genética. Foram realizados experimentos com este parâmetro, mas na versão final do algoritmo genético ele não foi utilizado (0%).

Por fim, o percentual máximo de indivíduos por espécie limita quantos indivíduos de uma única espécie podem existir em uma mesma geração. Isso garante uma maior diversidade genética do GA, de forma que nunca existirá uma super-espécie.

Para todas as operações que requerem a escolha de um ou mais cromossomos pais, como *crossover* ou mutação, foi utilizada a técnica da roleta viciada normalizada. Foi utilizada a variação normalizada pois em protótipos iniciais do trabalho foi identificado que o problema do super-indivíduo era muito acentuado no contexto do problema.

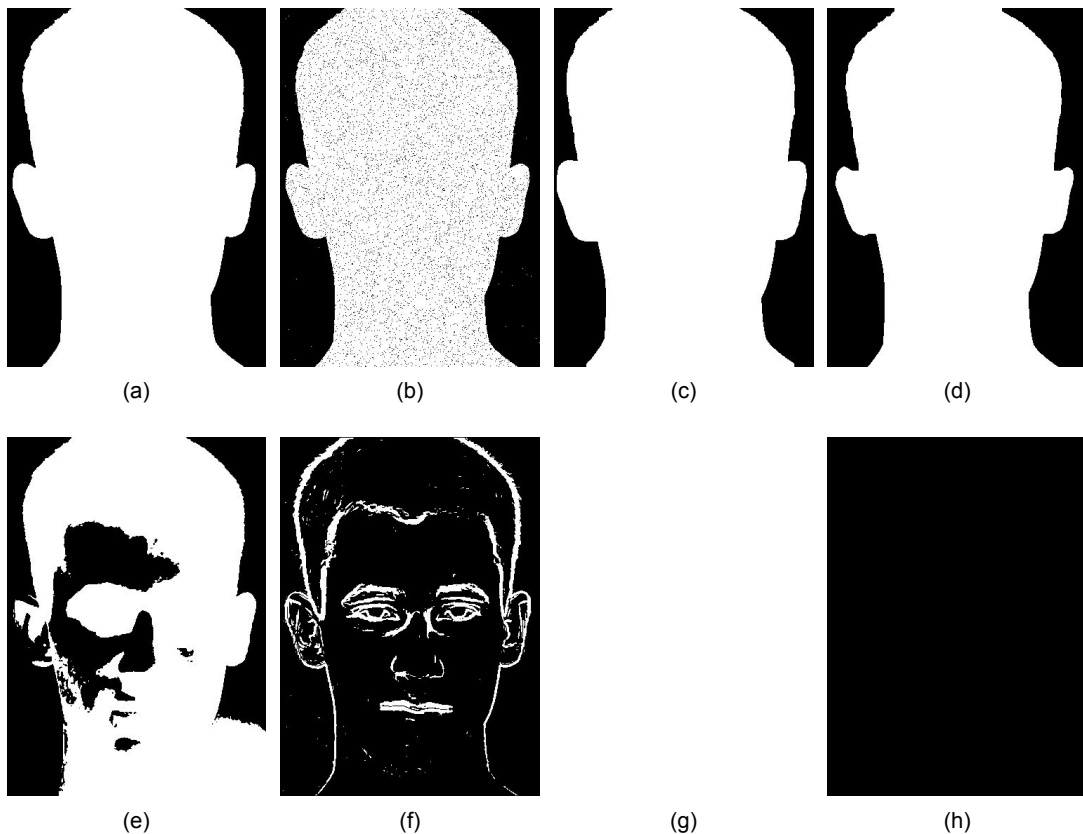
5.1.3 Comparação de Imagens

Para o cálculo da *fitness function* de um indivíduo é realizada a comparação da sua imagem de saída com um gabarito. A saída desta comparação deve ser um valor numérico entre 0 e 1, sendo o valor 1 a imagem perfeitamente correta. Quando existem diversas imagens que são testadas, a média de suas comparações é considerada sua avaliação.

Ao longo do desenvolvimento deste trabalho foram testadas várias funções de comparação. As imagens a ser comparadas são imagens binárias, onde o valor 1 (branco) faz parte do objeto em questão (neste caso, rosto), e 0 não fazendo parte do objeto. Para a escolha da melhor função, foi criado um conjunto de imagens que seriam comparadas com um gabarito, e estas foram ordenadas de acordo com o quão boa sua nota deveria ser em relação as outras imagens do conjunto, sem levar em consideração sua avaliação numérica.

Na Figura 37 é possível ver o conjunto de imagens utilizados composto por, em ordem decrescente de avaliação desejada: (a) Gabarito perfeito; (b) gabarito perfeito com ruído sal e pimenta; (c) gabarito perfeito reduzido (erosão); (d) gabarito perfeito amplificado (dilatação); (e) imagem segmentada com *threshold*; (f) imagem segmentada com Sobel; (g) imagem puramente branca; (h) imagem puramente preta.

Figura 37 – Imagens utilizadas para comparação



Fonte: elaborado pelo autor

O primeiro método testado era a comparação pixel a pixel da imagem, retornando o percentual de pixels corretos. Este método funcionou bem em testes preliminares deste trabalho, mas foi identificado um problema onde imagens puramente brancas recebiam um percentual acima da média e eram consideradas soluções boas pelo GA. Isso se devia ao fato de que, na base selecionada, os rostos ocupam grande parte da imagem, então uma imagem totalmente branca possui mais pixels corretos do que incorretos. Foi por causa deste problema que a pesquisa por outros métodos foi iniciada.

O segundo método testado foi o índice de Sørensen–Dice (SØRENSEN, 1948) (DICE, 1945). Este índice representa a similaridade entre dois conjuntos, podendo ser facilmente aplicado a imagens binárias. Para tal, é necessário considerar que a imagem binária é um conjunto dos pixels que fazem parte do objeto (valor 1). O índice de Sørensen–Dice pode ser dado pela equação

$$DSC = \frac{2Tp}{2Tp + Fp + Fn}$$

onde Tp representa o número de verdadeiro-positivos (pixels corretos), Fp representa o número de falso-positivos (pixels que não deveriam fazer parte do objeto) e Fn a quantidade de falso-negativos (pixels que deveriam estar no objeto). O mesmo problema das imagens brancas foi identificado com este método.

Após os testes comparativos com estes métodos, foi identificado que nenhum dos dois apresentava o resultado desejado para comparação das imagens binárias. Por esta razão, foi proposto um método próprio para comparação de imagens, derivado do método de Sørensen–Dice. O valor para o método de comparação proposto é obtido calculando o índice de Sørensen–Dice de verdadeiros-positivos, ou seja, $I_T = \frac{2Tp}{2Tp + Fp + Fn}$, além de calcular o índice considerando os verdadeiros-negativos, ou seja, $I_F = \frac{2Tn}{2Tn + Fp + Fn}$. Então, é selecionado o índice com menor valor. Desta forma, ele pode ser generalizado como

$$I = \frac{2 \times \min(Tp, Tn)}{2 \times \min(Tp, Tn) + Fp + Fn}$$

A comparação entre os métodos pode ser vista na Tabela 6. Na tabela, as ocorrências marcadas com um asterisco indicam percentuais que são menores do que o de ocorrências posteriores. É possível ver que utilizando os métodos de comparação de pixels ou o método de Sørensen–Dice, imagens totalmente brancas possuem um percentual muito maior (43% e 67% respectivamente) do que a imagem com Sobel. Ao utilizar Sørensen–Dice a imagem branca também teve um percentual melhor do que a imagem limiarizada, embora por um valor menor (5%). No método proposto, o percentual de assertividade das imagens totalmente branca e totalmente preta é 0%, sem gerar uma alteração significativa para as outras imagens.

Com base nesta comparação, foi utilizado o método de comparação proposto, pois ele mantém a mesma ordem de assertividade dos itens, porém não dando uma vantagem

Tabela 6 – Comparação entre os métodos de comparação de imagens binários

Imagem	Comparação de pixels	Sørensen–Dice	Proposta
Gabarito perfeito	100%	100%	100%
Gabarito com ruído	97,52%	98,31%	97,33%
Gabarito reduzido	96,82%	97,90%	96,63%
Gabarito aumentado	96,79%	97,79%	96,66%
Entrada limiarizada	80,86%	80,86% *	63,73%
Sobel	31,86% *	18,13% *	13,18%
Imagem branca	74,23%	85,21%	0%
Imagem preta	25,76%	0%	0%

a imagens totalmente brancas dentro do algoritmo genético, pois seu percentual não é tão alto. Esta decisão foi tomada levando em consideração o tipo de imagem foco deste trabalho: rostos. Não foi realizada comparação destes métodos para outros tipos de imagens, e os resultados poderão variar.

5.2 INTERFACE DE USUÁRIO

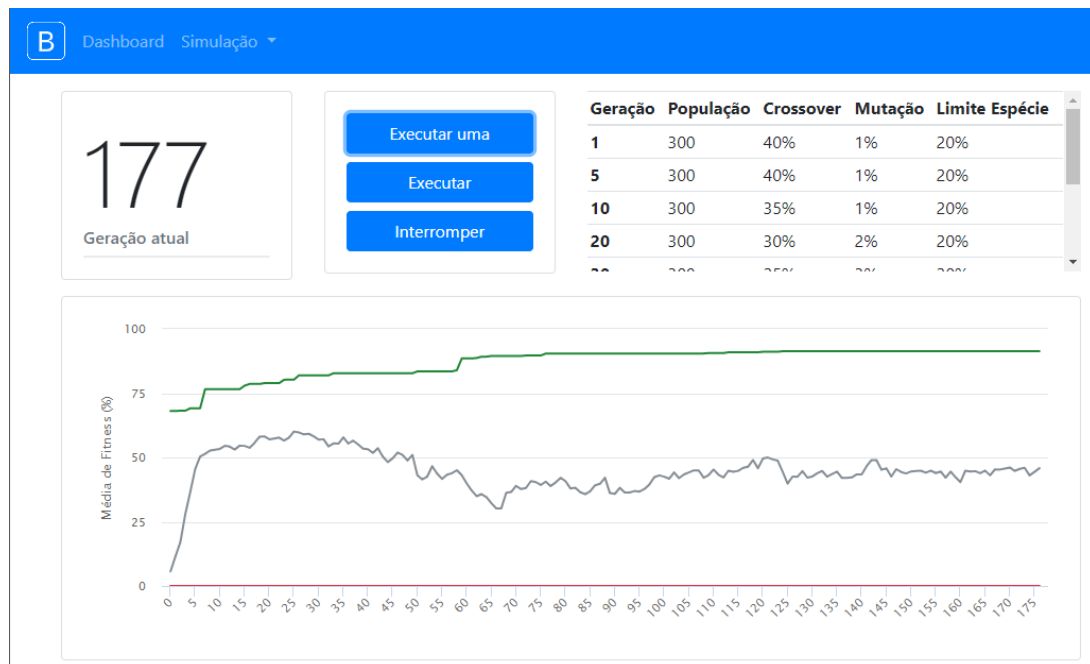
Nesta seção serão detalhadas as funcionalidades da interface de usuário desenvolvida. A interface foi desenvolvida com o objetivo de controlar a simulação e visualizar os resultados obtidos.

Ao iniciar o sistema de simulação, será exibida a tela principal, com os controles da simulação e visão geral dos resultados. Esta tela proporciona três controles principais ao usuário: “Executar uma”, “Executar” e “Interromper”. A opção “Executar uma” realiza a simulação da próxima geração do algoritmo genético, e interrompe a execução assim que ela for concluída. A opção de “Executar” executará gerações por tempo indeterminado. Por fim, a opção “Interromper” aborta a execução da geração atual. O painel principal pode ser visto na Figura 38.

Também é possível iniciar uma nova simulação, salvar ou carregar uma simulação existente à partir do menu “Simulação”. Para armazenar os dados da simulação estes são serializados utilizando a notação JSON (BRAY, 2017) e após compactados utilizando a implementação padrão do ZLIB (DEUTSCH; GAILLY, 1996) para Java. Além disso, a cada geração é realizada uma cópia de segurança da simulação para disco, para que, caso ocorra algum problema com a aplicação ou com o ambiente, seja possível retomar da última geração.

Como ferramental de análise, a interface proporciona na tela inicial um gráfico e uma tabela. O gráfico demonstra a média da função de avaliação, tal como melhor e pior indivíduo, ao longo das gerações. Estes mesmos dados são exibidos na visão tabular. Pela tabela é possível acessar a visualização da geração, clicando sobre seu número. A tabela pode ser vista na Figura 39.

Figura 38 – Painel principal da aplicação



Fonte: elaborado pelo autor

Figura 39 – Tabela de gerações

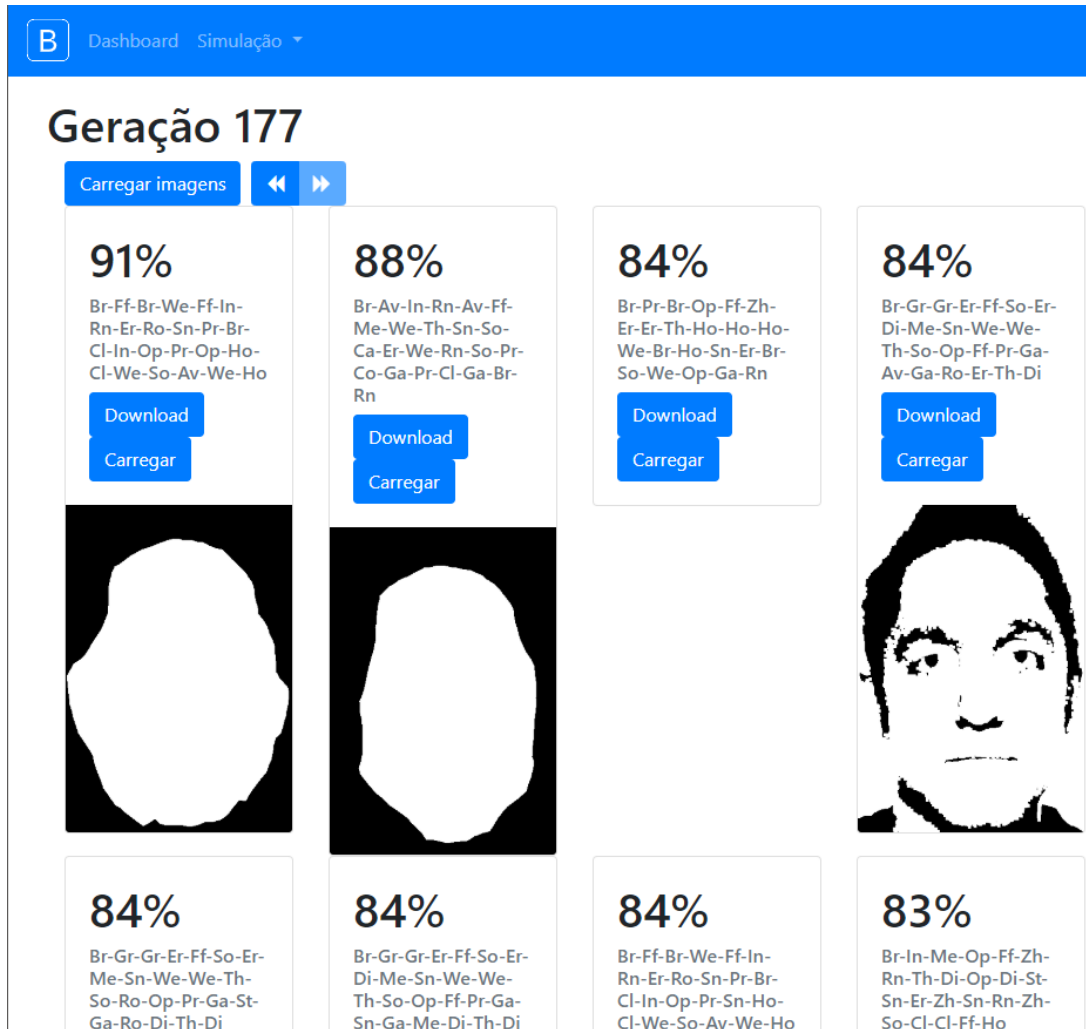
#	Média	Melhor	Pior
177	46%	91%	0%
176	44%	91%	0%
175	43%	91%	0%
174	46%	91%	0%
173	46%	91%	0%
172	45%	91%	0%
171	46%	91%	0%
170	46%	91%	0%
169	45%	91%	0%
168	45%	91%	0%

< Página 1 >

Fonte: elaborado pelo autor

Na tela da geração, são exibidos todos os indivíduos ordenados de forma decrescente de sua função de avaliação. Para cada indivíduo é exibido o valor de sua função de avaliação, tal como a sua espécie. É possível ainda exibir a imagem resultante de todos os indivíduos a partir do botão “Exibir imagens”, ou individualmente no botão “Carregar” de cada indivíduo. Em cada indivíduo existe um botão para baixar o arquivo projeto do VISNode, que pode, posteriormente, ser aberto com a ferramenta para visualização. A tela de indivíduos pode ser vista na Figura 40.

Figura 40 – Tela de indivíduos



Fonte: elaborado pelo autor

5.3 TREINAMENTO E OTIMIZAÇÃO DO GA

Ao longo do desenvolvimento do trabalho, foram realizados diversos experimentos com a parametrização do algoritmo genético e quais processos de PDI seriam utilizados, com o intuito de obter o melhor resultado no menor número possível de gerações. Neste capítulo serão demonstrados, em ordem cronológica, os experimentos realizados e os seus resultados.

5.3.1 Experimento A - Experimento de controle do GA

O primeiro experimento realizado no trabalho foi com o intuito de construir um controle para os próximos experimentos. Foram utilizados algoritmos comuns de PDI, detalhados na Tabela 7. O Algoritmo Genético também foi parametrizado da forma mais simples possível.

Para este experimento, foi utilizada uma população de 50 indivíduos, cada indi-

Tabela 7 – Algoritmos de PDI

<i>Algoritmo</i>	<i>Parâmetros</i>	<i>Algoritmo</i>	<i>Parâmetros</i>
Tons de Cinza		Roberts	
	[1] Peso <i>Red</i>		
Tons de Cinza Ponderados	[2] Peso <i>Green</i>	Robinson	
	[3] Peso <i>Blue</i>		
Threshold	[1] Limiar	Prewitt	
Inversão de valores		Canny	
Erosão		Zhang Suen	
Dilatação		Stentiford	
Abertura		Holt	
Fechamento		Desfoque Média	
Brilho	[1] Brilho	Desfoque Mediana	
Contraste	[1] Contraste	Desfoque de Gauss	[1] Sigma
			[2] Máscara
Sobel		<i>Flood Fill</i>	[1] Valor

víduo com até 10 processos. Para a validação, foram utilizadas as primeiras 5 imagens da base de treinamento. O percentual de *Crossover* utilizado foi de 40% e o de mutação foi de 0,5%.

A simulação foi executada três vezes até 100 gerações, e foi selecionada aquela de assertividade mediana para utilizar como controle. A assertividade final foi de 70,47%. Este resultado foi obtido na geração 88, mas o resultado de 70% já havia sido obtido na geração 4. Isso demonstra que houve uma convergência muito rápida do resultado. Além disso, a melhor simulação teve resultado de 84,56%, o que indica que esta convergência foi para uma máxima local. Uma forma de solucionar o problema da convergência para máxima local é mudar o parâmetro de *crossover* para reduzir ao longo da simulação, enquanto o parâmetro de mutação aumenta. Este experimento está descrito na seção seguinte.

5.3.2 Experimento B - Variação no *crossover* e mutação

Este experimento foi realizado para tentar resolver o problema da convergência para uma máxima local identificado no Experimento A. Para este experimento, o percentual de *crossover* inicia como 40% na geração 1, e decai linearmente até 5% na geração 60. Da mesma forma, a chance de mutação inicia em 0,5% na geração 0 até 7% na geração 60.

De forma semelhante ao anterior, este experimento foi executado três vezes até a geração 100. O percentual de assertividade mediano foi de 70,75% (+0,28% comparando ao Experimento A) e o melhor foi 84,96% (+0,40% comparando ao Experimento A). A convergência do resultado ocorreu na geração 98, mas, de forma idêntica ao teste anterior, o resultado de 70% foi atingido já na geração 4.

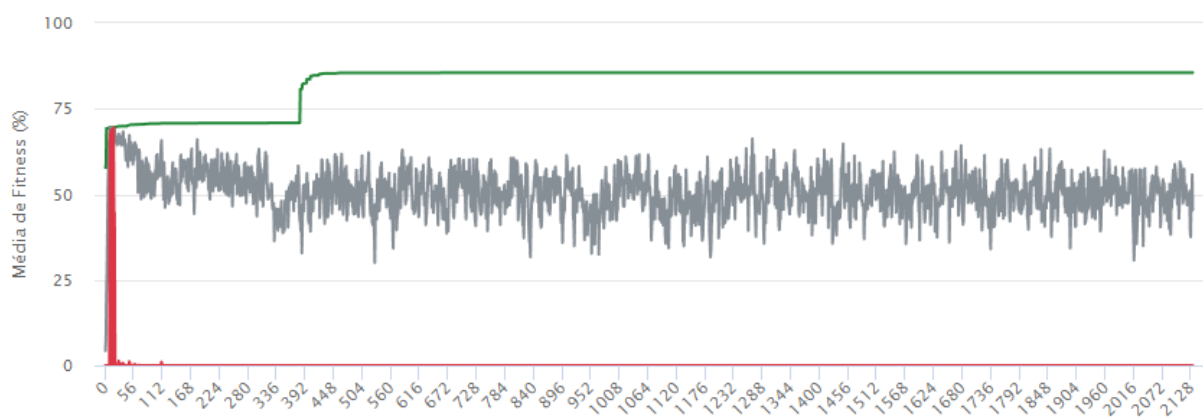
Isso indica que a convergência à uma máxima local continua ocorrendo, mesmo com um percentual de mutação incremental. No entanto, ocorreu uma melhora na penúltima geração da simulação, enquanto no Experimento A se passaram 12 gerações sem melhoras. Isso pode significar que os indivíduos iriam continuar melhorando por muito mais tempo, possivelmente até saindo desta máxima local. Por esta razão foi executado um novo experimento, deixando o GA executar por um número mais elevado de gerações. Este experimento está detalhado na próxima seção.

5.3.3 Experimento C - Simulação por muitas gerações

Neste experimento a simulação do algoritmo genético foi executada com a mesma parametrização do experimento anterior, mas por muito mais tempo. Ao final deste experimento, 2132 gerações foram simuladas, totalizando 106 mil indivíduos avaliados (contra 5.000 dos experimentos anteriores). O objetivo deste experimento era identificar se o GA conseguiria obter um resultado melhor do que o anterior, saindo de uma possível máxima local.

Ao final da simulação, o melhor resultado foi de 85,47%, 14,72% melhor que a mediana do experimento anterior e 0,51% acima que o melhor resultado. Isto mostra que o melhor resultado obtido após poucas centenas de gerações não melhorará muito após milhares, o que pode indicar que esta é a solução ótima global, ou muito próxima dela. Na Figura 41 é possível ver o gráfico de assertividade máxima, média e mínima da simulação ao longo das gerações.

Figura 41 – Gráfico de assertividade da simulação por geração



Fonte: elaborado pelo autor

Com base no resultado deste experimento, foi identificada a possibilidade de que esta seria uma possível solução ótima global para esta parametrização. Portanto, no experimento realizado na sequência, foi aumentado o número máximo de processos por indivíduo. Para compensar o aumento de possibilidades, também foi aumentado o tamanho da população.

5.3.4 Experimento D - Maior quantidade de processos possíveis

O próximo experimento realizado foi para verificar se com uma quantidade maior de processos disponíveis para execução o GA conseguiria resultados superiores. O número de processos possíveis foi incrementado de 10 para 25 processos por indivíduo, de forma que agora cada indivíduo possui 150 genes. Por causa do grande aumento na quantidade de possíveis soluções, o tamanho da população também foi aumentado, de 50 para 300 indivíduos. Além disso, foi adicionado o algoritmo Snakes.

Neste experimento, a assertividade média final obtida foi de 87,42%, proporcionando um aumento de 1,95% em relação ao experimento anterior. Ao final da simulação, 337 gerações foram simuladas, totalizando em 101 mil indivíduos avaliados, valor muito próximo ao experimento anterior. Nos próximos experimentos foram realizados testes para tentar aumentar a velocidade de conversão do Algoritmo Genético.

5.3.5 Experimento E - Indivíduos aleatórios

Neste experimento foi adicionado um novo parâmetro ao algoritmo genético: percentual de indivíduos aleatórios por geração. Se este parâmetro for maior que zero, um número de indivíduos de cada geração será criado de forma totalmente aleatória, como se fosse a primeira geração. Isto foi feito com o objetivo de continuar explorando possibilidades muito diferentes das existentes na população, de forma a facilitar que o GA saia de soluções localmente ótimas. Neste experimento foi utilizado 10% de indivíduos aleatórios por geração.

Ao final da simulação, 448 gerações foram avaliadas, obtendo um resultado final de 85,71% de assertividade, 1,71% menor do que o experimento anterior. Como o resultado foi inferior e a convergência do GA não ocorreu de forma significativamente mais rápida que o experimento anterior, o parâmetro de aleatoriedade foi removido para os próximos experimentos.

5.3.6 Experimento F - Crossover de 1 ponto e 2 pontos

O próximo experimento foi o comparativo entre os diferentes métodos de *crossover* para tentar identificar o mais adequado para o problema em questão. Os testes realizados até o momento foram com o *crossover* uniforme. A taxa de *crossover* foi mantida em 40% para todos os testes.

O melhor percentual de assertividade obtido até o momento utilizando o *crossover* uniforme foi de 87,42% em 337 gerações. A simulação utilizando o *crossover* de dois pontos obteve um percentual de 85,99%, 1,45% menor do que o Experimento D. Já o resultado com *crossover* de um ponto foi 89,80% em 156 gerações, garantindo uma melhora de

2,38% em quase 300 gerações a menos processadas. Por este motivo, o crossover de 1 ponto foi mantido para os próximos testes.

5.3.7 Experimento G

Neste experimento foi adicionado um novo algoritmo de PDI à biblioteca de algoritmos que o GA pode utilizar. Este algoritmo é o *threshold* modificado para permitir dois limiares diferentes. Ele recebe por parâmetro o valor dos dois limiares, e mais três parâmetros que indicam o valor que deve ser utilizado para os pixels que estiverem abaixo do primeiro *threshold*, entre os dois *thresholds* e acima do segundo, respectivamente. Além disso, alguns algoritmos baseados em convolução receberam um novo parâmetro para indicar o tamanho da máscara utilizada, que pode ser alterado pelos genes do algoritmo genético. Estes algoritmos foram: abertura, fechamento, dilatação, erosão, desfoque de média e desfoque de mediana.

Também foi adicionada uma nova heurística ao algoritmo genético: um limite de indivíduos por espécie. Para definir a espécie de um indivíduo, são utilizados os genes categóricos que representam os seus processos. Desta forma, todos os indivíduos que utilizam os mesmos processos na mesma ordem serão da mesma espécie, por mais que seus parâmetros sejam largamente diferentes. Foi adicionado um novo parâmetro a simulação para indicar um percentual da população que pode ser ocupada por uma única espécie. Este parâmetro foi iniciado com o valor de 20% da população. Isso foi feito para aumentar a diversidade genética as populações, evitando que o GA avalie apenas indivíduos de uma única espécie.

Ao final da simulação, foram avaliadas 177 gerações. O percentual médio de assertividade foi de 91,32%, garantindo uma melhora de 1,52% com um número semelhante de gerações. A heurística de limite de indivíduos por espécie garantiu uma maior diversidade genética durante a simulação, o que pode ter influenciado a melhoria do percentual de assertividade. Após mais algumas simulações, este modelo foi utilizado como resultado final da aplicação.

5.4 RESULTADOS OBTIDOS

Após os testes realizados para tentar otimizar o algoritmo genético, foram realizadas novas simulações para obter os resultados finais do trabalho. Cada simulação era executada até que não houvesse melhoria no seu resultado por uma quantidade elevada de tempo, onde então era iniciada a próxima simulação. Ao final, três simulações foram executadas, cada uma tomando entre 40 e 50 horas de tempo de processamento. Para o treinamento e cálculo da assertividade dos modelos foram utilizadas as 5 primeiras imagens da base FASSEG. As imagens e seus gabaritos podem ser vistos na Figura 42.

Figura 42 – Imagens e gabaritos utilizadas para treinamento



Fonte: FASSEG (2018)

Os três modelos gerados pelas simulações apresentaram resultados bastante distintos. O modelo com maior assertividade apresentou 91,32% de pixels corretos, enquanto o segundo e o terceiro apresentaram 85,51% e 83,24% respectivamente. O melhor modelo foi simulado por 177 gerações, embora a convergência tenha ocorrido na geração 125. Na Tabela 8 estão detalhadas as informações sobre os modelos gerados.

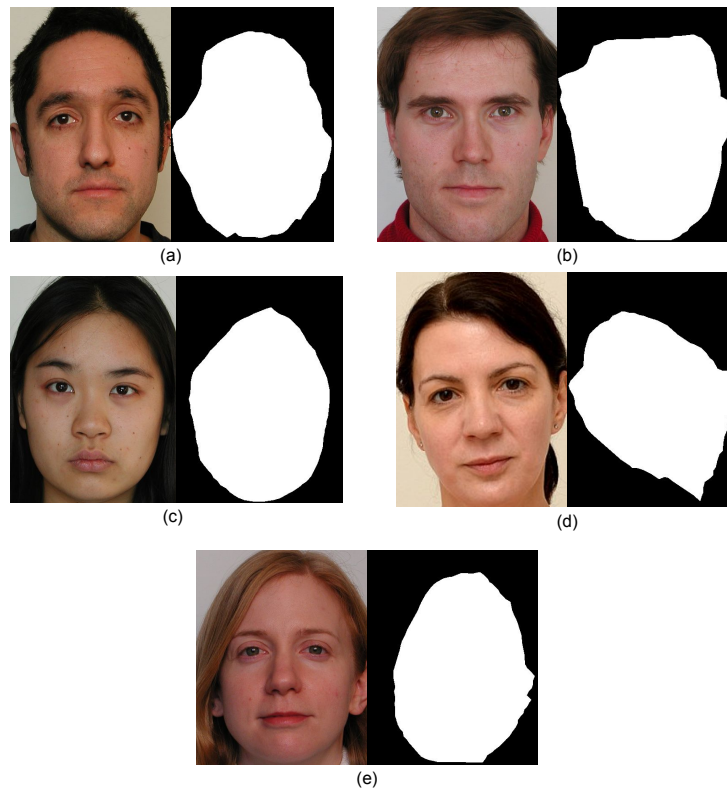
Tabela 8 – Resultado dos três modelos gerados

Modelo	Assertividade	Total de gerações	Geração de convergência
1	91,32%	177	125
2	85,81%	85	84
3	83,24%	118	93

O primeiro modelo, que obteve uma assertividade de 91,32%, utilizou uma combinação de algoritmos de pré-processamento, detecção de bordas e *snakes* para realizar a segmentação. O seu resultado final pode ser visto na Figura 43. Sua sequência completa de algoritmos foi: Brilho (202); Brilho (-237); Conversão para tons de cinza ponderada (*vermelho* = 0,79; *verde* = 0,69; *azul* = 0,63); *Flood Fill*; Inversão de cores; Detecção de bordas de Robinson; Erosão; Detecção de bordas de Roberts; *Snakes* ($\alpha = 0,895$;

$\beta = 0,209; \gamma = 0,145$).

Figura 43 – Resultado da segmentação para o modelo 1



Fonte: elaborado pelo autor

O segundo modelo obteve uma assertividade de 85,81% e utilizou apenas algoritmos de pré-processamento e limiarização para realizar a segmentação. O seu resultado final pode ser visto na Figura 44. Sua sequência completa de algoritmos foi: Conversão para tons de cinza ponderada (*vermelho* = 0,90; *verde* = 0,34; *azul* = 0,57); Desfoque de média (*mascara* = 3); Dilatação (*mascara* = 3); Fechamento (*mascara* = 3); Contraste (2,96); Contraste (2,78); Erosão (*mascara* = 3); Brilho (-182); Desfoque de mediana (*mascara* = 3); *Flood Fill*; Limiarização (125).

Por fim, o terceiro modelo obteve uma assertividade de 83,24% e também utilizou uma combinação de algoritmos de pré-processamento, detecção de bordas e snakes para realizar a segmentação, mas de forma menos eficaz. O seu resultado final pode ser visto na Figura 45. Sua sequência completa de algoritmos foi: Abertura (*mascara* = 3); Inversão de cores; Dilatação (*mascara* = 7); Dilatação (*mascara* = 3); Detecção de bordas de Sobel; Erosão (*mascara* = 3); Detecção de boras de Sobel; Abertura (*mascara* = 7); Desfoque de média (*mascara* = 3); Inversão de cores; Threshold múltiplo (*limiar*₁ = 143; *limiar*₂ = 81; *valor*₁ = 90); *valor*₂ = 81; *valor*₃ = 62); *Flood Fill*; *Snakes* ($\alpha = 0,284$; $\beta = 0,536$; $\gamma = 0,720$)

O próximo passo realizado foi a avaliação dos resultados obtidos, e se estes estavam de acordo com as hipóteses deste trabalho.

Figura 44 – Resultado da segmentação para o modelo 2



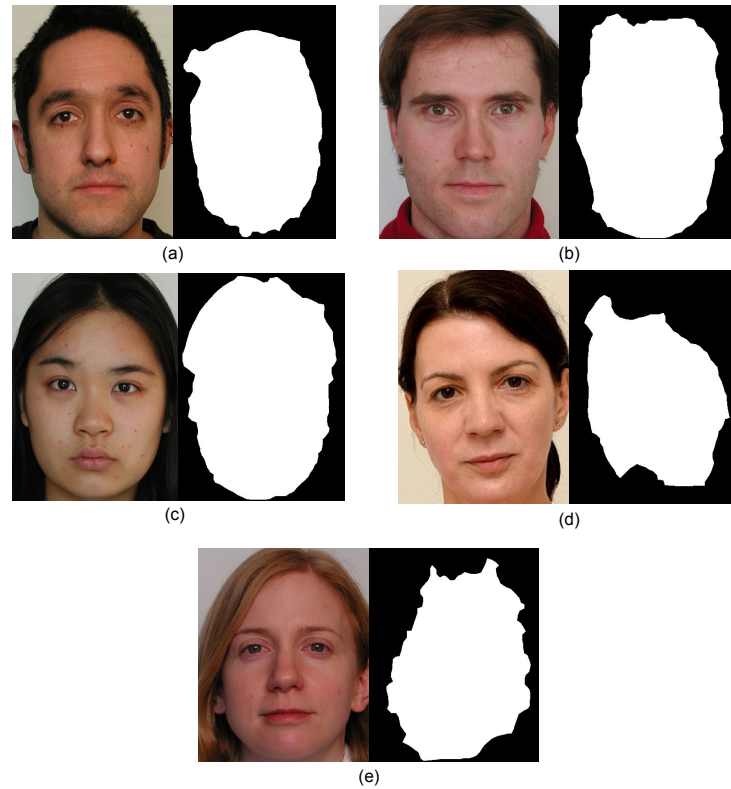
Fonte: elaborado pelo autor

5.5 AVALIAÇÃO

Para este trabalho, existem duas teorias que precisam ser validadas. Primeiro, é necessário validar de forma qualitativa se o resultado deste trabalho pode ser analisado e compreendido por seres humanos, de forma que eles possam extrair informações e possivelmente aprender novas técnicas. Também é necessário validar de forma quantitativa se o resultado da segmentação possui uma qualidade semelhante ou superior a outras técnicas de segmentação, como redes neurais.

Para realizar a validação qualitativa deste trabalho, foi realizada uma avaliação por juízes. Três especialistas foram convidados para validar a hipótese deste trabalho de que os modelos gerados permitem a análise dos resultados e o aprendizado quanto a segmentação de imagens. Para a avaliação, foram analisados os três modelos gerados pela aplicação. Para realizar a validação quantitativa foram selecionados outros trabalhos de segmentação de faces utilizando diferentes técnicas de segmentação e onde o percentual de assertividade estava explícito. Então, é apresentado um comparativo entre os trabalhos e o modelo com a melhor assertividade obtido neste trabalho. Os dois métodos de validação estão descritos nas seções a seguir.

Figura 45 – Resultado da segmentação para o modelo 3



Fonte: elaborado pelo autor

5.5.1 Avaliação por Juízes

Este trabalho possui uma etapa de validação qualitativa que visa validar a hipótese de que o resultado do sistema proposto pode ser analisado e compreendido por um ser humano, de forma que ele possa extrair alguma informação ou conhecimento do mesmo. Considerando a natureza subjetiva do problema, foi escolhida a metodologia de avaliação por juízes para realizar a validação qualitativa.

Para a avaliação, foram selecionados três juízes especialistas na área de Processamento Digital de Imagens. Foram apresentados os três modelos de segmentação para os juízes, para que fosse realizada a análise crítica dos processos. Durante a análise, os juízes avaliaram a qualidade do resultado, a clareza do modelo, e se era possível identificar oportunidades de aprendizado. Com base nesta análise, cada juiz respondeu a um questionário para avaliar a qualidade da segmentação e, principalmente, a capacidade de compreensão e aprendizado com o resultado gerado. O perfil dos juízes está representado na Tabela 9.

O questionário foi composto por 12 questões, sendo quatro delas para identificação dos juízes. As outras 8 questões são relacionadas ao projeto analisado e o seu resultado. Na Tabela 10 estão detalhadas todas as questões que constituem o questionário aplicado. As questões 5 e 6 foram respondidas para cada imagem de entrada. Para todas as questões

Tabela 9 – Perfil dos juízes avaliadores

Juíz	Bacharelado	Mestrado	Doutorado
A	Ciências da Computação Feevale, 2015	Computação Aplicada Unisinos, 2017	-
B	Ciências da Computação Feevale, 2013	Ciências da Computação PUCRS, 2017	-
C	Análise de Sistemas UCPEL, 2005	Ciências da Computação UFRGS, 2008	Ciências da Computação PUCRS, Em andamento

Tabela 10 – Questionário

#	Seção	Tipo	Questão
1	Identificação	Texto	Nome
2	Identificação	Texto	Formação
3	Identificação	Lickert	Conhecimento sobre Processamento Digital de Imagens
4	Identificação	Lickert	Conhecimento sobre Algoritmos Genéticos
5	Qualidade	Lickert	A área segmentada condiz com a área do rosto da imagem original
6	Qualidade	Lickert	O contorno da área segmentada está bem delineado e sem ruídos
7	Qualidade	Lickert	A qualidade geral da segmentação é consistente entre todas as imagens de entrada
8	Clareza	Lickert	Existe clareza sobre quais algoritmos de PDI estão sendo utilizados
9	Clareza	Lickert	Existe clareza sobre quais parâmetros são utilizados para cada algoritmo
10	Clareza	Lickert	Está clara a sequência que estes algoritmos serão executados
11	Aprendizado	Lickert	Pode-se compreender como a escolha dos algoritmos, seu sequenciamento e sua parametrização geram a segmentação do rosto
12	Aprendizado	Lickert	Pode-se extrair algum conhecimento ou aprendido quanto a como realizar segmentações a partir da análise destes resultados

que envolviam a avaliação de um aspecto do sistema, foi utilizada a Escala de Likert (LIKERT, 1932). Para Collis e Hussey (2005), a Escala de Likert permite traduzir uma opinião ou percepção em um valor numérico, sendo simples para o respondente e para o pesquisador, sendo fácil de codificar e analisar. Além disto, em cada seção do questionário foi adicionado um campo livre para que os juízes pudessem informar comentários sobre o questionário ou sobre o resultado analisado.

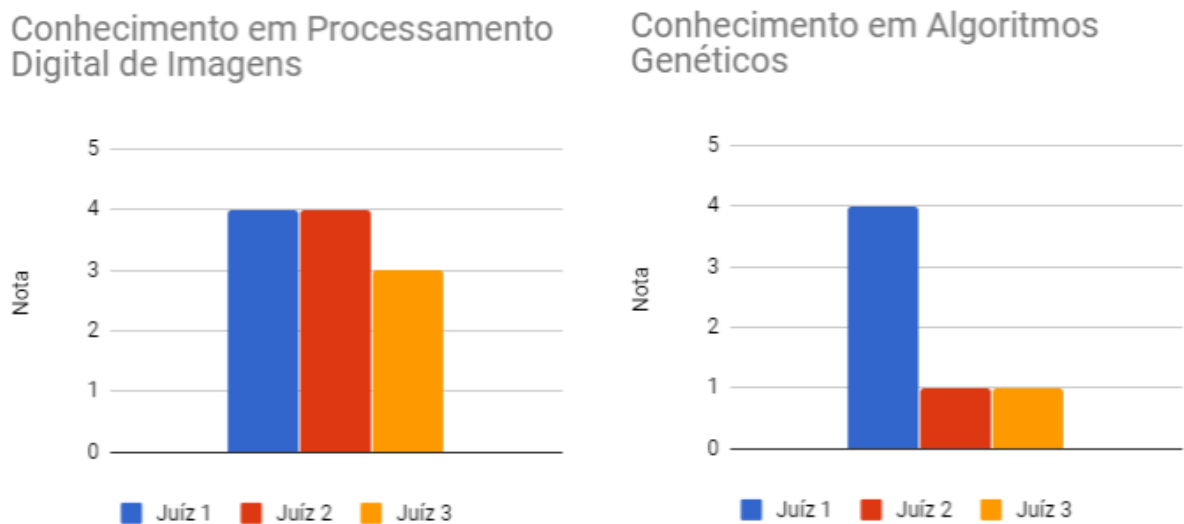
A avaliação foi realizada de forma presencial, com os três juízes simultaneamente. No dia da validação foi realizada uma breve apresentação sobre o trabalho e seu objetivo, e os avaliadores receberam este trabalho para leitura um mês antes. Após a apresentação, foram repassadas as perguntas do questionário para verificar se alguma havia ficado am-

bígua ou de difícil entendimento, e então foi iniciado o processo de avaliação. Do início ao fim, o processo da avaliação levou em torno de 1 hora e 20 minutos.

5.5.1.1 Identificação

Nesta primeira seção do questionário, o objetivo foi identificar o perfil auto-avaliativo dos juízes com relação aos temas abordados por este trabalho: Algoritmos genético e processamento digital de imagens. A seção era composta por quatro perguntas. A primeira pergunta identifica o juiz pelo nome, e foi utilizada apenas para organização dos resultados, sendo que este será omitido. A próxima pergunta solicitava que cada juiz identificasse sua formação acadêmica. Dos três juízes, dois são Mestres em Ciências da Computação, enquanto o terceiro é Mestre em Computação Aplicada. As duas próximas perguntas solicitavam que os juízes auto-avaliassem seu nível de conhecimento nas áreas de PDI e GA, respectivamente. As respostas podem ser vistas na Figura 46.

Figura 46 – Gráfico de respostas para as questões de identificação



Fonte: elaborado pelo autor

É importante ressaltar que no momento de selecionar os juízes, foi levado em consideração principalmente seu conhecimento na área de processamento de imagens, pois este seria o tema mais relevante durante a etapa de avaliação. Isso pode ser identificado nas respostas, de forma que todos os avaliadores identificam que possuem um conhecimento médio ou superior de PDI, mas apenas um juiz indicou que possui conhecimento também em algoritmos genéticos.

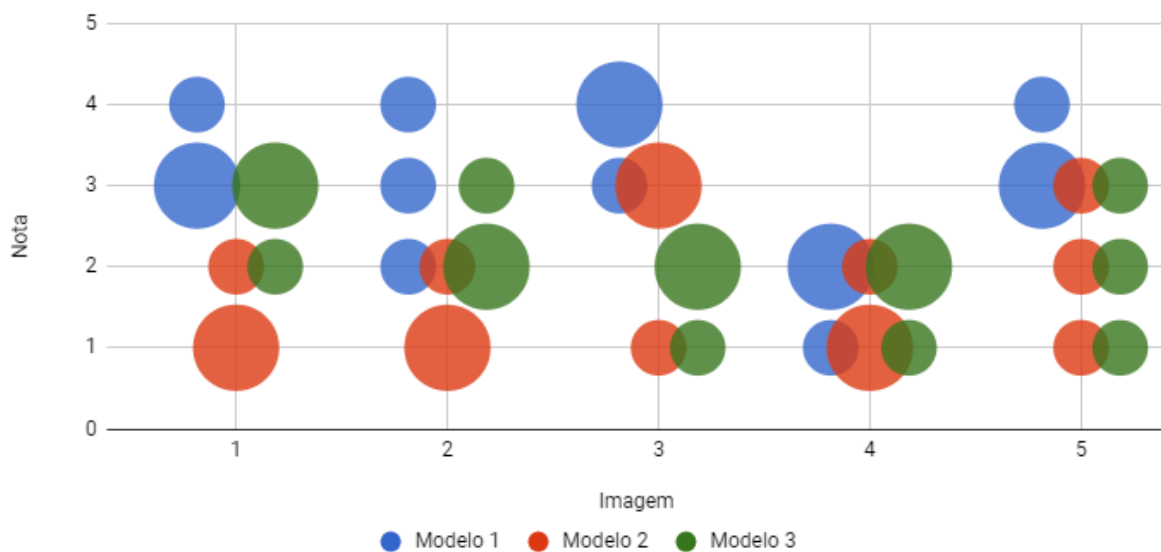
5.5.1.2 Qualidade da segmentação

A segunda seção do questionário tem como objetivo identificar a opinião dos avaliadores sobre a qualidade da segmentação dos rostos. Cada avaliador respondeu todas as perguntas desta seção três vezes, uma para cada modelo avaliado.

A primeira pergunta da seção pedia que os juízes avaliassem se a área segmentada pelo modelo correspondia a área esperada. Os resultados da questão podem ser vistos na Figura 47. Neste gráfico, o eixo vertical representa a nota dada na escala Likert, o eixo horizontal representa a imagem que estava sendo avaliada, a cor da série representa o modelo que estava sendo avaliado, e o tamanho das bolhas representa o número de votos (0, 1, 2 ou 3).

Analisando os resultados é possível identificar claramente a relação do percentual de assertividade obtido para cada modelo com a avaliação dos juízes, onde o modelo 1 tem uma avaliação superior aos modelos 2 e 3. Também é possível identificar que, em todos os modelos, a imagem número 4 ficou com uma avaliação inferior as outras, o que pode identificar uma situação com complicadores para segmentação do rosto. Considerando o modelo um, a qualidade da segmentação média entre os juízes e entre as imagens foi de 3, o que foi considerado um resultado positivo, visto que para o segundo modelo algumas imagens ficaram com média próxima de 1.

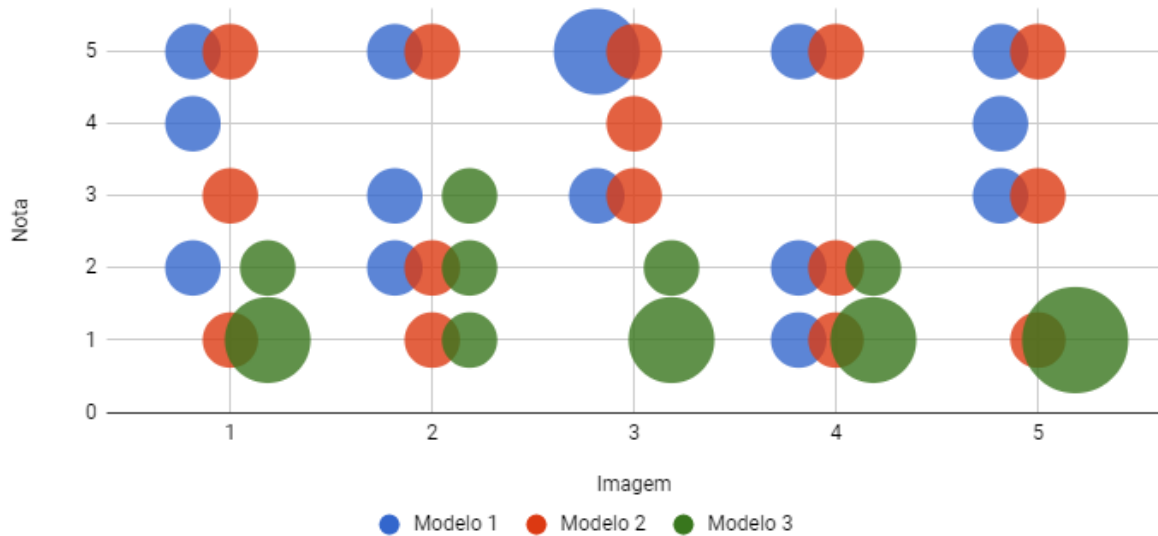
Figura 47 – Gráfico de respostas para a questão: A área segmentada condiz com a área do rosto da imagem original



Fonte: elaborado pelo autor

A segunda pergunta desta seção relativa a qualidade da segmentação questionava sobre a delimitação do contorno dos rostos, e se estava clara e sem ruídos. O gráfico das respostas pode ser visto na Figura 48, e segue o mesmo formato do gráfico anterior. Os resultados desta questão foram positivos, principalmente para o modelo 1. O modelo 3 teve resultados bem inferiores aos outros modelos, devido ao fato do contorno ser muito inconsistente com relação aos outros. É possível identificar, ainda, que as respostas variam muito mais entre os juízes, sendo que em 5 casos (Modelo 1, imagem 4 e Modelo 2, imagens 1, 2, 4 e 5) houve uma resposta com avaliação máxima e uma com avaliação mínima. Isso pode indicar uma deficiência na clareza da questão ou no que deveria ter sido avaliado.

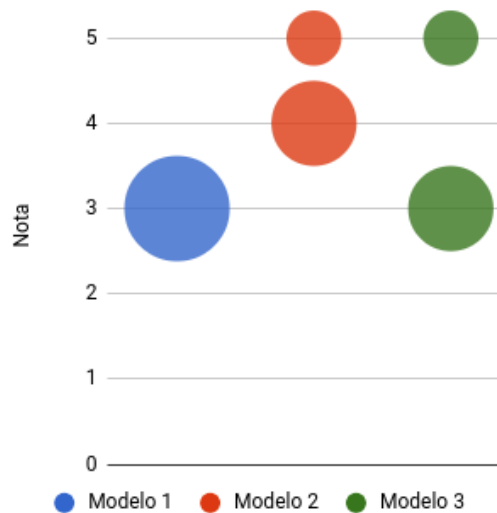
Figura 48 – Gráfico de respostas para a questão: O contorno da área segmentada está bem delineado e sem ruídos



Fonte: elaborado pelo autor

A última questão da seção relativa a qualidade da segmentação questionava se a qualidade geral da segmentação era consistente entre todas as imagens. O gráfico das respostas pode ser visto na Figura 49. Diferente das questões anteriores, esta questão não era por imagem, de forma que o eixo horizontal não possui variação neste gráfico. É possível identificar que houve pouca variação entre as opiniões dos juízes com relação a esta questão. Para o modelo 1, a opinião foi unânime, com uma avaliação de 3. Para os outros modelos, a opinião não foi unânime, mas teve pouca variação nos resultados.

Figura 49 – Gráfico de respostas para a questão: A qualidade geral da segmentação é consistente entre todas as imagens de entrada



Fonte: elaborado pelo autor

Os comentários dos juízes com relação as questões desta seção foram de acordo com as respostas obtidas e com a análise realizada. Os comentários relacionados a qualidade da segmentação foram:

- Modelo 1 - “As imagens 2 e 4 aparentemente apresentam resultados inferiores quando relacionados com as segmentações observadas nas outras imagens”
- Modelo 2 - “A segmentação não apresenta ruídos e mantém seu padrão de delineamento, no entanto, segmenta o fundo branco como rosto”
- Modelo 2 - “Apenas uma imagem possui um segmentação aceitável, com bom contorno, entretanto, possui ruídos no interior”
- Modelo 3 - “A qualidade geral é compatível, mas apresenta resultados bastante ruins. Os limites da segmentação são bastante serrilhados”

5.5.1.3 Clareza da solução

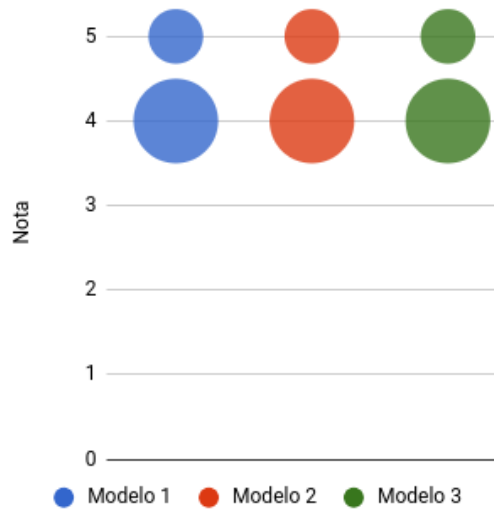
A segunda seção do questionário foi relacionada a clareza da solução. O principal objetivo destas questões era identificar se a ferramenta VISNode possuía os recursos suficientes para realizar a análise do resultado e compreendê-lo. Desta forma, isolando a possibilidade de baixas avaliações relacionadas a análise e aprendizado serem por causa da ferramenta, e não do modelo de segmentação em si.

A primeira questão desta seção foi relacionada a clareza sobre quais algoritmos de PDI estavam sendo utilizados. O gráfico de respostas pode ser visualizado na Figura 50, onde o eixo vertical representa a nota, a série representa o modelo avaliado, e o tamanho da bolha representa o número de votos. Conforme esperado, a avaliação dos três modelos quanto a esta questão foi idêntica, pois a ferramenta utilizada foi a mesma, independente do modelo. A avaliação média dos juízes foi positiva neste quesito.

A segunda pergunta desta seção questionava sobre a clareza dos parâmetros utilizados para cada algoritmo. O gráfico das respostas pode ser visualizado na Figura 51, e segue o mesmo formato do gráfico anterior. O resultado foi um pouco inferior ao da questão anterior, o que indica que os parâmetros de cada algoritmo não estão tão claros quanto quais algoritmos são utilizados. Da mesma forma da questão anterior, o resultado foi o mesmo entre todos os modelos, o que já era esperado.

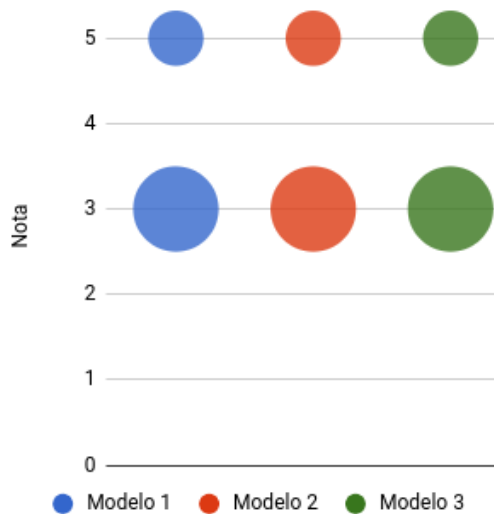
A última questão da seção de segmentação perguntava sobre a clareza da sequência que os algoritmos foram executados. O gráfico das respostas pode ser visualizado na Figura 52. As respostas desta questão foram extremamente positivas, sendo que a avaliação máxima foi escolhida de forma unânime entre os juízes e para todos os modelos.

Figura 50 – Gráfico de respostas para a questão: Existe clareza sobre quais algoritmos de PDI estão sendo utilizados



Fonte: elaborado pelo autor

Figura 51 – Gráfico de respostas para a questão: Existe clareza sobre quais parâmetros são utilizados para cada algoritmo

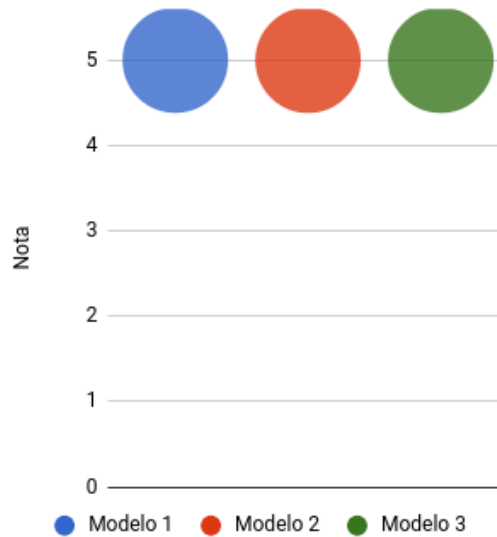


Fonte: elaborado pelo autor

Os comentários obtidos para a seção de clareza apresentaram uma opinião positiva sobre a clareza da ferramenta VISNode para entendimento e análise dos modelos, e estavam de acordo com as respostas obtidas. Os comentários foram:

- Modelo 1: “A ferramenta deixa clara todas as etapas, bem como seus valores de entrada”
- Modelo 1: “A ferramenta é uma excelente opção para a experimentação e aprendi-

Figura 52 – Gráfico de respostas para a questão: Está clara a sequência que estes algoritmos serão executados



Fonte: elaborado pelo autor

zado das técnicas e de algoritmos de processamento de imagens”

- Modelo 1: “Achei claro e facilmente interpretável”
- Modelo 2: “Novamente, a ferramenta torna a sequência muito clara”
- Modelo 2: “A ferramenta é uma excelente opção para a experimentação e aprendizado das técnicas e de algoritmos de processamento de imagens”
- Modelo 3: “Novamente, a ferramenta mantém tudo muito claro com relação a sequência de etapas”
- Modelo 3: “A ferramenta é uma excelente opção para a experimentação e aprendizado das técnicas e de algoritmos de processamento de imagens”
- Modelo 3: “Novamente, bem claro, como comentamos na avaliação, este bloco poderia ser único, e não por modelo”

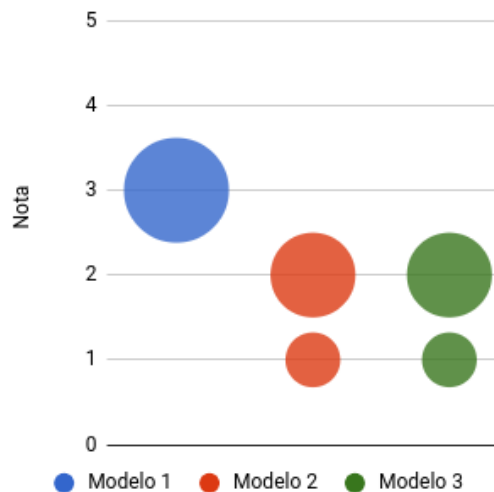
5.5.1.4 Análise e aprendizado

A última seção do questionário consiste de questões relacionadas a capacidade de análise dos modelos e aprendizado a partir desta análise. O objetivo desta seção era validar se os modelos de segmentação gerados podem ser considerados "caixa-branca", permitindo a fácil compreensão por um ser humano.

A primeira questão desta seção tinha como objetivo identificar se foi possível compreender como o modelo gerado obtém uma segmentação do rosto. O gráfico das respostas pode ser visto na Figura 53, e segue o mesmo formato dos últimos gráficos. As

respostas obtidas para o modelo 1 foram no valor médio da escala, enquanto para os outros modelos foi abaixo da média. Desta forma, a qualidade da segmentação e a capacidade de compreensão aparentam estar proporcionalmente relacionados.

Figura 53 – Gráfico de respostas para a questão: Pode-se compreender como a escolha dos algoritmos, seu sequenciamento e sua parametrização geram a segmentação do rosto



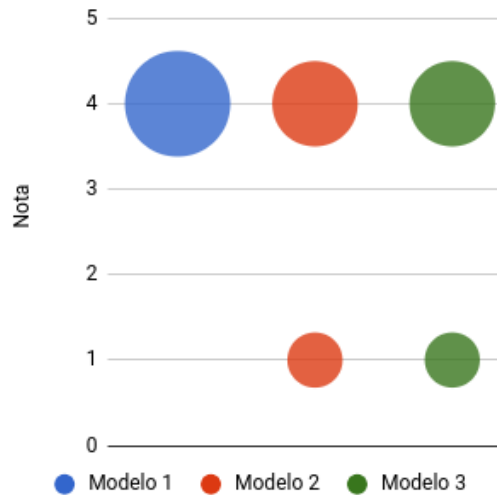
Fonte: elaborado pelo autor

A segunda e última pergunta da seção questionava sobre a possibilidade de extrair algum conhecimento ou aprendizado quanto a como realizar a segmentação de rostos a partir da análise realizada. O gráfico das respostas pode ser visto na Figura 54. É possível ver que o modelo 1, cuja assertividade foi a maior obtida, também apresenta a melhor possibilidade de aprendizado, obtendo a nota média 3. Os outros dois modelos não obtiveram um resultado tão bom, o que pode ser devido a sua segmentação não ter obtido um resultado inferior.

Além da questão quantitativa, foi solicitado que os juízes destacassem qual tipo de informação eles conseguiram extrair do modelo, caso sua avaliação tenha sido positiva na questão quantitativa. Os comentários podem ser vistos na sequência:

- Modelo 1 - “Sim, no entanto, algumas etapas parecem nulas entre si. Exemplo das etapas 1 e 2, onde o brilho é adicionado e logo removido”
- Modelo 1 - “É possível, através da ferramenta, identificar técnicas que podem ser empregadas no processo de segmentação, já que estão disponíveis para uso mais de 50 métodos diferentes. O pesquisador pode desconhecer alguns dos métodos sugeridos, e pela solução gerada pelo algoritmo pode adquirir conhecimento”
- Modelo 1 - “Algumas escolhas mostram uma grande alteração na imagem e parecem ser bem efetivas”

Figura 54 – Gráfico de respostas para a questão: Pode-se extrair algum conhecimento ou aprendizado quanto a como realizar segmentações a partir da análise destes resultados



Fonte: elaborado pelo autor

- Modelo 2 - “Igual ao que foi descrito para o modelo anterior”
- Modelo 2 - “Novamente, algumas escolhas ficam claras e mostram bom resultado, entretanto, outras parecem pouco efetivas e sem justificativa de uso”
- Modelo 3 - “Igual ao que foi descrito para o modelo anterior”
- Modelo 3 - “Este foi os que teve mais etapas sem compreensão”

Ao final da seção também foi disponibilizada uma área para comentários gerais sobre a seção de análise e aprendizado. Os comentários podem ser vistos na sequência:

- “Como não são apresentadas justificativas para as escolhas dos métodos (e seus respectivos parâmetros) é bastante complicado o entendimento do motivo de determinada técnica ter sido executada”
- “Acredito ser clara a saída e qual o ‘passo a passo’ realizado, entretanto, algumas escolhas não parecem ser efetivas, não colaborando para o processo”

Por fim, ao final do questionário, foi disponibilizada uma questão para comentários gerais que não se encaixavam nas outras seções. Os comentários gerais estão descritos na sequência:

- “A natureza aleatória das decisões fica clara ao avaliar a sequência, os métodos e os parâmetros utilizados no modelo 1”

- “Como comentário final, o trabalho apresenta um problema interessante, apresenta uma solução inovadora e com possibilidades de continuidade e melhoria”

5.5.2 Comparativo com Outras Soluções

Para validar a viabilidade da solução proposta para segmentação facial da pele, esta passou por uma etapa de validação quantitativa, comparando o seu percentual de assertividade com outras soluções para o problema da segmentação facial da pele. Para realizar a comparação, foi utilizado o estudo de Kakumanu, Makrogiannis e Bourbakis (2007). Neste estudo foi realizada a comparação de 27 diferentes estratégias para realizar a segmentação facial da pele, sendo que destas, 20 possuem no trabalho o seu percentual de assertividade pixel-a-pixel, sendo este contabilizado em pixels com o caso verdadeiro-positivo. Ou seja, o percentual representa quantos pixels da face realmente foram segmentados como face.

É importante ressaltar que ao longo do desenvolvimento do trabalho, os percentuais de assertividade estavam representados utilizando o método de comparação proposto, não sendo diretamente comparáveis com os percentuais apresentados no trabalho de Kakumanu, Makrogiannis e Bourbakis (2007). Por esta razão, o percentual foi recalculado para a melhor solução selecionada, sendo demonstrado aqui como percentual de pixels verdadeiro-positivos. A Tabela 11 representa a comparação entre os trabalhos. Os trabalhos que não demonstravam seu percentual de verdadeiros-positivos foram omitidos, pois não seriam relevantes para comparação.

Conforme comentado por Kakumanu, Makrogiannis e Bourbakis (2007), não é possível fazer uma comparação direta entre os trabalhos e métodos pelo fato de que as bases de imagens possuem tamanhos e características diferentes. A solução proposta foi treinada com um conjunto de imagens bastante reduzido, sendo utilizadas apenas 5 imagens da base selecionada. Mesmo assim, é possível notar que o percentual de assertividade obtido está de acordo com a qualidade esperada de métodos de segmentação de faces.

Devido a natureza exploratória deste trabalho e a falta de referências de soluções semelhantes, não foi possível obter resultados desta qualidade ao trabalhar com a base completa. Acredita-se que com maior refinamento do algoritmo genético e dos processos de PDI disponíveis, é possível manter a qualidade da segmentação mesmo com conjuntos maiores de imagens.

Tabela 11 – Tabela comparativa do resultado obtido com outros trabalhos de segmentação

Autor(es)	Método	Base (Imagens)	Percentual
Jones e Rehg (2002)	Bayes	Compaq (4675)	90.0%
	GMM (16)	Compaq	90.0%
Brown et al. (2001)	SOM	Compaq	78.0%
Jedynak et al. (2003)	MaxEnt. model	Compaq	82.9%
Lee e Yoo (2002)	Ellip. model	Compaq	90.0%
	SGM	Compaq	90.0%
	GMM(6)	Compaq	90.0%
Brand e Mason (2000)	Bayes	Compaq	93.4%
	I-axis Thresh.	Compaq	94.7%
	Thresh. ratios	Compaq	94.7%
Sebe et al. (2004)	BN	Compaq	99.4%
Phung et al. (2001)	Thresholding	ECU (4000)	82.0%
	Bayes	ECU	88.9%
	MLP	ECU	88.5%
	SGM	ECU	88.0%
	GMM (2)	ECU	85.2%
Anagnostopoulos et al. (2003)	Fuzzy rules+PNN	Própria (317)	82.4%
Caetano et al. (2002)	GMM (2)	Própria (800)	87.0%
Jayaram et al. (2004)	Bayes	Própria (815)	98.2%
	SGM	Própria (815)	94.4%
Trabalho Proposto	GA + PDI	FASSEG (5)	91,3%

Fonte: Adaptado de Kakumanu, Makrogiannis e Bourbakis (2007)

6 CONCLUSÃO

A segmentação de imagens digitais é uma das etapas mais importante do processo de análise e extração de características, e vem cada vez mais sendo automatizada para suas diversas aplicações. Existe um grande leque de possíveis soluções, algoritmos e técnicas para realizar a segmentação de imagens, cada uma com seus pontos fortes e fracos. Geralmente, estas técnicas são orquestradas pelo próprio criador do sistema automatizado, o que depende fortemente do seu conhecimento sobre estes algoritmos e quais foram considerados para a solução.

Uma forma de tirar esta responsabilidade do criador do sistema é utilizando redes neurais, técnicas de *machine learning* que aprendem como realizar a segmentação utilizando uma base de treinamento de imagens pré-segmentadas. Uma desvantagem das redes neurais é seu tempo de processamento, geralmente maior que das soluções que utilizam apenas técnicas de PDI. Além disso, após uma RN ter aprendido a realizar a segmentação, não é possível que um ser humano facilmente analise a rede para aprender como segmentar melhor imagens no futuro.

Dessa forma, o objetivo deste trabalho foi propor um sistema capaz de utilizar algoritmos genéticos em conjunto com técnicas de PDI, visando a criação de sistemas de segmentação automáticos baseados em treinamento, mas ainda permitindo a análise do seu resultado. Dessa forma, a pesquisa iniciou no Capítulo 2 abordando os conceitos e técnicas utilizados para desenvolver um algoritmo genético. No Capítulo 3, as técnicas de PDI utilizadas foram apresentadas.

O detalhamento do sistema proposto e como ele foi desenvolvido está descrito no Capítulo 5. Neste capítulo ainda são demonstrados os resultados do o trabalho, onde foi obtida uma taxa de assertividade de até 91,32% na base de treinamento. Além disso, foi demonstrado um comparativo com outros trabalhos de segmentação facial. Também foi realizado um questionário com três juízes, demonstrando a possibilidade de análise e aprendizado com base nos modelos de segmentação gerados.

Como trabalhos futuros, pode-se tentar realizar testes com o método proposto com bases de imagens maiores e mais comuns no meio acadêmico, de forma a poder compará-lo de forma mais precisa com outros métodos de segmentação. Além disso, é necessário ainda evolução com relação a sua taxa de assertividade para que possa se tornar uma solução viável. Alguns experimentos que podem ser citados são: a implementação de novos algoritmos de processamento digital de imagens; a execução de modelos como grafos, ou seja, conexões paralelas ao invés de apenas sequenciais; modelo de segmentação que utiliza informações da imagem para parametrizar os algoritmos.

REFERÊNCIAS

- ANAGNOSTOPOULOS, I et al. A probabilistic neural network for human face identification based on fuzzy logic chromatic rules. **IEEE MED03**, 2003.
- ANDERSON, Matthew et al. Proposal for a standard default color space for the internet—srgb. In: SOCIETY FOR IMAGING SCIENCE AND TECHNOLOGY. **Color and imaging conference**. [S.l.], 1996. v. 1996, n. 1, p. 238–245.
- AZEVEDO, Eduardo; CONCI, Aura. **Computação gráfica: teoria e prática**. [S.l.]: Elsevier, 2003.
- BINS, L Sant’anna et al. Satellite imagery segmentation: a region growing approach. **Simpósio Brasileiro de Sensoriamento Remoto**, Imagem Multimídia, São Paulo. Proceedings, CD Salvador, Bahia, Brazil, v. 8, n. 1996, p. 677–680, 1996.
- BRAND, Jason; MASON, John S. A comparative assessment of three approaches to pixel-level human skin-detection. In: IEEE. **wecwis**. [S.l.], 2000. p. 5056.
- BRAY, Tim. **The javascript object notation (json) data interchange format**. [S.l.], 2017.
- BROADBENT, Arthur D. A critical review of the development of the cie1931 rgb color-matching functions. **Color Research & Application**, Wiley Online Library, v. 29, n. 4, p. 267–272, 2004.
- BROWN, David A; CRAW, Ian; LEWTHWAITE, Julian. A som based approach to skin detection with application in real time systems. In: CITESEER. **BMVC**. [S.l.], 2001. v. 1, p. 491–500.
- CAETANO, Tiberio S; OLABARRIAGA, Silvia D; BARONE, Dante Augusto Couto. Performance evaluation of single and multiple-gaussian models for skin color modeling. In: IEEE. **Computer Graphics and Image Processing, 2002. Proceedings. XV Brazilian Symposium on**. [S.l.], 2002. p. 275–282.
- CANNY, John. A computational approach to edge detection. In: **Readings in Computer Vision**. [S.l.]: Elsevier, 1987. p. 184–203.
- COLLIS, Jill; HUSSEY, Roger. **Pesquisa em administração: um guia para alunos de graduação e pós-graduação**. Trad. Lucia Simonini. [S.l.]: Porto Alegre: Bookman, 2005.
- COSTA, GA; CAZES, TB.
Aplicação de Algoritmos Genéticos para a Evolução de Parâmetros para a Segmentação de Imagens — PUCRio, 2010.
- DEUTSCH, Peter; GAILLY, JeanLoup. **Zlib compressed data format specification version 3.3**. [S.l.], 1996.
- DICE, Lee R. Measures of the amount of ecologic association between species. **Ecology**, Wiley Online Library, v. 26, n. 3, p. 297–302, 1945.

EIBEN, Agoston E; RAUE, P-E; RUTTKAY, Zs. Genetic algorithms with multi-parent recombination. In: SPRINGER. **International Conference on Parallel Problem Solving from Nature**. [S.l.], 1994. p. 78–87.

FASSEG. **The FAcE Semantic SEGmentation repository**. 2018. Disponível em: [<http://massimomauro.github.io/FASSEG-repository/>](http://massimomauro.github.io/FASSEG-repository/).

GONZALEZ, Rafael C; WOODS, Richard E. **Digital image processing**. [S.l.]: Upper Saddle River, NJ: Prentice Hall, 2012.

GUILHERME, LUIS RENATO WOISKI. Uma abordagem de afinamento por aprendizagem através de exemplos. **PUCPR- Brasil**, v. 10, 2007. Disponível em: <http://www.ppgia.pucpr.br/teses/DissertacaoPPGIa-LuisRenato-2007.pdf>.

HAMMOUCHE, Kamal; DIAF, Moussa; SIARRY, Patrick. A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation. **Computer Vision and Image Understanding**, Elsevier, v. 109, n. 2, p. 163–175, 2008.

HOLLAND, John H. Genetic algorithms. **Scientific american**, JSTOR, v. 267, n. 1, p. 66–73, 1992.

HOLT, Christopher M et al. An improved parallel thinning algorithm. **Communications of the ACM**, ACM, v. 30, n. 2, p. 156–160, 1987.

HUANG, Chien-nan; LIM, Chin-Choon; MING, C. Comparison of image processing algorithms and neural networks in machine vision inspection. **Computers & industrial engineering**, Elsevier, v. 23, n. 1-4, p. 105–108, 1992.

JAYARAM, Sriram et al. Effect of colorspace transformation, the illuminance component, and color modeling on skin detection. In: IEEE. **Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on**. [S.l.], 2004. v. 2, p. II–II.

JEDYNAK, Bruno; ZHENG, Huicheng; DAOUDI, Mohamed. Maximum entropy models for skin detection. In: SPRINGER. **International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition**. [S.l.], 2003. p. 180–193.

JONES, Michael J; REHG, James M. Statistical color models with application to skin detection. **International Journal of Computer Vision**, Springer, v. 46, n. 1, p. 81–96, 2002.

KAKUMANU, Praveen; MAKROGIANNIS, Sokratis; BOURBAKIS, Nikolaos. A survey of skin-color modeling and detection methods. **Pattern recognition**, Elsevier, v. 40, n. 3, p. 1106–1122, 2007.

KASS, Michael; WITKIN, Andrew; TERZOPOULOS, Demetri. Snakes: Active contour models. **International journal of computer vision**, Springer, v. 1, n. 4, p. 321–331, 1988.

KIM, Byung-Gyu; SHIM, Jae-Ick; PARK, Dong-Jo. Fast image segmentation based on multi-resolution analysis and wavelets. **Pattern Recognition Letters**, Elsevier, v. 24, n. 16, p. 2995–3006, 2003.

LEE, Jae Y; YOO, Suk I. An elliptical boundary model for skin color detection. In: **Proc. of the 2002 International Conference on Imaging Science, Systems, and Technology**. [S.l.: s.n.], 2002.

LI, Jian-Ping et al. A species conserving genetic algorithm for multimodal function optimization. **Evolutionary computation**, MIT Press, v. 10, n. 3, p. 207–234, 2002.

LIKERT, Rensis. A technique for the measurement of attitudes. **Archives of psychology**, 1932.

LINDEN, Ricardo. **Algoritmos genéticos (2a edição)**. [S.l.]: Brasport, 2008.

MATIAS, Ítalo de Oliveira. **O uso do algoritmo genético em segmentação de imagens digitais**. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, 2007.

MINKOWSKI, Hermann. Theorie der konvexen körper, insbesondere begründung ihres oberflächenbegriffs. **Gesammelte Abhandlungen**, v. 2, p. 131–229, 1911.

NOH, Hyeonwoo; HONG, Seunghoon; HAN, Bohyung. Learning deconvolution network for semantic segmentation. In: **Proceedings of the IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2015. p. 1520–1528.

PAL, Nikhil R; PAL, Sankar K. A review on image segmentation techniques. **Pattern recognition**, Elsevier, v. 26, n. 9, p. 1277–1294, 1993.

PEDRINI, Hélio; SCHWARTZ, William Robson. **Análise de imagens digitais: princípios, algoritmos e aplicações**. [S.l.]: Thomson Learning, 2008.

PHUNG, Son Lam; CHAI, Douglas; BOUZERDOUM, Abdesselam. A universal and robust human skin color model using neural networks. In: IEEE. **Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on**. [S.l.], 2001. v. 4, p. 2844–2849.

ROLI, Fabio. **Image Analysis And Processing Iciap 2005: 13th International Conference Cagliari, Italy, September 6-8, 2005 Proceedings**. [S.l.]: Springer Science & Business Media, 2005.

RÖNNAU, Rodrigo F.

Segmentação Automática de Órgãos em Imagens de Tomografia Computadorizada do Tórax — Universidade Feevale, 2015.

SEBE, Nicu et al. Skin detection: A bayesian network approach. In: IEEE. **Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on**. [S.l.], 2004. v. 2, p. 903–906.

SOBEL, Irwin; FELDMAN, Gary. A 3x3 isotropic gradient operator for image processing. **a talk at the Stanford Artificial Project in**, p. 271–272, 1968.

SØRENSEN, Thorvald. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons. **Biol. Skr.**, v. 5, p. 1–34, 1948.

STENTIFORD, FWM; MORTIMER, RG. Some new heuristics for thinning binary handprinted characters for ocr. **IEEE transactions on systems, man, and cybernetics**, IEEE, n. 1, p. 81–84, 1983.

TING, Chuan-Kang. On the mean convergence time of multi-parent genetic algorithms without selection. In: SPRINGER. **European Conference on Artificial Life**. [S.l.], 2005. p. 403–412.

VISNODE. **VISNode**. 2018. Disponível em: <<http://github.com/VISNode/VISNode>>.

YEN, Jui-Cheng; CHANG, Fu-Juay; CHANG, Shyang. A new criterion for automatic multilevel thresholding. **IEEE Transactions on Image Processing**, IEEE, v. 4, n. 3, p. 370–378, 1995.

ZEILER, Matthew D; FERGUS, Rob. Visualizing and understanding convolutional networks. In: SPRINGER. **European conference on computer vision**. [S.l.], 2014. p. 818–833.

ZHANG, TY; SUEN, Ching Y. A fast parallel algorithm for thinning digital patterns. **Communications of the ACM**, ACM, v. 27, n. 3, p. 236–239, 1984.