



**INSTITUTO
FEDERAL**

Goiano

Campus
Rio Verde

MINISTÉRIO DA EDUCAÇÃO
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
GOIANO - CAMPUS RIO VERDE

MANOEL VICTOR RIBEIRO DA SILVA

**EXPLORAÇÃO INTERATIVA DE OPERADORES MORFOLÓGICOS
UTILIZANDO GALERIAS DE DESIGN**

RIO VERDE
MARÇO, 2024



**INSTITUTO
FEDERAL**

Goiano

Campus
Rio Verde

MANOEL VICTOR RIBEIRO DA SILVA

**EXPLORAÇÃO INTERATIVA DE OPERADORES MORFOLÓGICOS
UTILIZANDO GALERIAS DE DESIGN**

Trabalho de Conclusão de Curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia Goiano - Campus Rio Verde, como requisito parcial para a obtenção do Grau de Bacharelado em Ciência da Computação.

Orientador: Dr. Douglas Cedrim Oliveira
Instituto Federal Goiano

RIO VERDE
MARÇO, 2024

Sistema desenvolvido pelo ICMC/USP
Dados Internacionais de Catalogação na Publicação (CIP)
Sistema Integrado de Bibliotecas - Instituto Federal Goiano

SM285e Silva, Manoel Victor Ribeiro da
Exploração interativa de operadores morfológicos
utilizando galerias de design / Manoel Victor
Ribeiro da Silva; orientador Douglas Cedrim
Oliveira. -- Rio Verde, 2024.
70 p.

TCC (Graduação em Ciência da Computação) --
Instituto Federal Goiano, Campus Rio Verde, 2024.

1. Processamento digital de imagens. 2. Galerias
de design. 3. Operadores morfológicos. 4. Interface
interativa. I. Cedrim Oliveira, Douglas, orient. II.
Título.



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA GOIANO

**TERMO DE AUTORIZAÇÃO PARA DISPONIBILIZAR PRODUÇÃO TÉCNICA NO REPOSITÓRIO
INSTITUCIONAL DO IF GOIANO**

Repositório Institucional do IF Goiano - RIIF Goiano Sistema Integrado de Bibliotecas

- Profissional de Educação do IF Goiano -

Com base no disposto na Lei Federal nº 9.610/98, e manual sobre a Produção Técnica, publicado pela DAV/CAPES/MEC*, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia Goiano, a disponibilizar gratuitamente o documento no Repositório Institucional do IF Goiano (RIIF Goiano), sem ressarcimento de direitos autorais, conforme permissão assinada eletronicamente abaixo, em formato digital para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IF Goiano.

Identificação da Produção Técnica – DAV/CAPES

- Editoria Material Didático
 Curso de Formação Profissional Projetos de Extensão à Comunidade
 Relatório Técnico Conclusivo Atividade Técnica/Tecnológica
 Disseminação do Conhecimento Técnico/Tecnológico Produto Bibliográfico
[X] Outras Produções Técnicas - Tipo: TCC (Graduação)

Nome Completo do Autor/a: Manoel Victor Ribeiro da Silva

Matrícula: 2016102192010200

Título do Trabalho: EXPLORAÇÃO INTERATIVA DE OPERADORES MORFOLÓGICOS UTILIZANDO GALERIAS DE DESIGN

Restrições de Acesso ao Documento

Documento confidencial: Não Sim

Justifique: _____

Informe a data que poderá ser disponibilizado no RIIF Goiano: 07/ 03 / 2024

O documento está sujeito a registro de patente? Sim Não

O documento pode vir a ser publicado como livro e/ou artigo? Sim Não

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O/A referido/a docente e/ou autor/a declara que:

1 - o documento é seu trabalho original, detém os direitos autorais da produção técnica e não infringe os direitos de qualquer outra pessoa ou entidade;

2 - obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autor/a, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia Goiano os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;

3 - cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia Goiano.

Rio Verde, 7 de março de 2024.

(Assinado Eletronicamente)

Manoel Victor Ribeiro da Silva (Autor)

(Assinado Eletronicamente)

Douglas Cedrim Oliveira (Orientador)

1058004

(Assinatura do Docente, Autor e/ou Detentor dos Direitos Autorais)

Documento assinado eletronicamente por:

- Manoel Victor Ribeiro da Silva , 2016102192010200 - Discente, em 07/03/2024 21:43:49.
- Douglas Cedrim Oliveira, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 07/03/2024 21:26:33.

Este documento foi emitido pelo SUAP em 07/03/2024. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifgoiano.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 581184
Código de Autenticação: f7b3ecb807



INSTITUTO FEDERAL GOIANO

Campus Rio Verde

Rodovia Sul Goiana, Km 01, Zona Rural, 01, Zona Rural, RIO VERDE / GO, CEP 75901-970

(64) 3624-1000



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA GOIANO

Ata nº 9/2024 - GGRAD-RV/DE-RV/CMPRV/IFGOIANO

ATA DE DEFESA DE TRABALHO DE CURSO

Aos 28 dias do mês de fevereiro de dois mil e vinte e quatro, às dezoito horas, reuniu-se a banca examinadora composta pelos docentes: Dr. Douglas Cedrim Oliveira (orientador), Dr. Adriano Oliveira Barbosa (membro externo), Me. Fábio Montanha Ramos (membro interno), para examinar o Trabalho de Curso intitulado "EXPLORAÇÃO INTERATIVA DE OPERADORES MORFOLÓGICOS UTILIZANDO GALERIAS DE DESIGN" de Manoel Victor Ribeiro da Silva, estudante do Curso de Bacharelado em Ciência da Computação do IF Goiano – Campus Rio Verde, sob Matrícula nº 2016102192010200. A palavra foi concedida ao estudante para a apresentação oral do TC, em seguida houve arguição do candidato pelos membros da Banca Examinadora. Após tal etapa, a Banca Examinadora decidiu pela APROVAÇÃO do estudante. Ao final da sessão pública de defesa foi lavrada a presente ata, que segue assinada pelos membros da Banca Examinadora.

(Assinado Eletronicamente)

Douglas Cedrim Oliveira

Orientador(a)

(Assinado Eletronicamente)

Fábio Montanha Ramos

Membro

(Assinado Eletronicamente)

Adriano Oliveira Barbosa

Membro

Observação:

() O(a) estudante não compareceu à defesa do TC.

Documento assinado eletronicamente por:

- Fabio Montanha Ramos, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 04/03/2024 15:04:55.
- Douglas Cedrim Oliveira, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 28/02/2024 22:52:51.

Este documento foi emitido pelo SUAP em 28/02/2024. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifgoiano.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 577612
Código de Autenticação: bd0847d74a




Documento assinado digitalmente
gov.br ADRIANO OLIVEIRA BARBOSA
Data: 05/03/2024 19:58:51-0300
Verifique em <https://validar.iti.gov.br>


INSTITUTO FEDERAL GOIANO
Campus Rio Verde
Rodovia Sul Goiana, Km 01, Zona Rural, 01, Zona Rural, RIO VERDE / GO, CEP 75901-970
(64) 3624-1000

MANOEL VICTOR RIBEIRO DA SILVA

**EXPLORAÇÃO INTERATIVA DE OPERADORES MORFOLÓGICOS
UTILIZANDO GALERIAS DE DESIGN**


Trabalho de curso DEFENDIDO E APROVADO em 28 de fevereiro de 2024, pela Banca Examinadora constituída pelos membros:

Documento assinado digitalmente
 **FABIO MONTANHA RAMOS**
Data: 05/03/2024 11:28:07-0300
Verifique em <https://validar.iti.gov.br>

Documento assinado digitalmente
 **ADRIANO OLIVEIRA BARBOSA**
Data: 05/03/2024 16:32:32-0300
Verifique em <https://validar.iti.gov.br>

Me. Fábio Montanha Ramos
Instituto Federal Goiano

Dr. Adriano Oliveira Barbosa
Faculdade de Ciências Exatas e Tecnologia -
UFGD

Documento assinado digitalmente
 **DOUGLAS CEDRIM OLIVEIRA**
Data: 05/03/2024 09:34:52-0300
Verifique em <https://validar.iti.gov.br>

Dr. Douglas Cedrim Oliveira
Orientador

Rio Verde

2024

AGRADECIMENTOS

Gostaria de expressar minha sincera gratidão a todas as pessoas que tornaram possível a realização deste trabalho.

Ao meu orientador, Dr. Douglas Cedrim Oliveira, expresso profundo agradecimento pela orientação excepcional, apoio incansável e valiosas contribuições ao longo de todo o processo de pesquisa. Sua paciência, encorajamento e carinho foram fundamentais para o meu desenvolvimento acadêmico.

À minha querida família e amigos, agradeço pelo apoio incondicional, compreensão e encorajamento durante todo o percurso.

Aos membros da banca avaliadora, agradeço por dedicarem seu precioso tempo e expertise para contribuir com valiosas análises, enriquecendo assim a qualidade deste trabalho.

Cada um de vocês desempenhou um papel vital nessa jornada, e sou imensamente grato por todo o suporte e contribuições que tornaram este trabalho possível.

RESUMO

SILVA, Manoel Victor Ribeiro da. **Exploração interativa de operadores morfológicos utilizando galerias de design.** Março, 2024. 70 f. Monografia – (Curso de Bacharelado em Ciência da Computação), Instituto Federal de Educação, Ciência e Tecnologia Goiano - Campus Rio Verde. Rio Verde.

Este trabalho, intitulado “Exploração Interativa de Operadores Morfológicos Utilizando Galerias de Design”, aborda a criação e implementação de uma aplicação interativa de processamento de imagens. A proposta visa proporcionar aos usuários a aplicação intuitiva de operadores morfológicos, permitindo a exploração de diferentes combinações para aprimorar a qualidade visual de suas imagens através do uso de galerias de design. A aplicação registra todas as operações realizadas, promovendo uma experiência educativa. Mesmo para usuários não especialistas, a aplicação oferece uma plataforma para testar, combinar e aprender sobre os efeitos das operações morfológicas aplicadas em uma imagem, tornando o processo acessível e informativo.

Palavras-chave: Processamento digital de imagens. Galerias de design. Operadores morfológicos. Interface interativa.

ABSTRACT

SILVA, Manoel Victor Ribeiro da. **Interactive Exploration of Morphological Operators Using Design Galleries**. Março, 2024. 70 f. Monografia – (Bacharelado em Ciência da Computação), Instituto Federal de Educação, Ciência e Tecnologia Goiano - Campus Rio Verde. Rio Verde.

This work, entitled 'Interactive Exploration of Morphological Operators Using Design Galleries,' addresses the creation and implementation of an interactive image processing application. The proposal aims to provide users with an intuitive application of morphological operators, allowing the exploration of different combinations to enhance the visual quality of their images through the use of design galleries. The application records all performed operations, promoting an educational experience. Even for non-expert users, the application provides a platform to test, combine, and learn about the effects of morphological operations applied to an image, making the process accessible and informative.

Palavras-chave: Digital image processing. Design galleries. Morphological operators. Interactive user interface.

LISTA DE FIGURAS

Figura 1 – Imagem original (a) e seus resultados após aplicar os operadores Black Hat (b) e Gradiente (c).	1
Figura 2 – Galerias de Design no Adobe Photoshop CS4.	2
Figura 3 – Imagem Analógica relacionada com a Imagem Digital.	6
Figura 4 – Imagem Analógica relacionada com a Imagem Digital (Coordenadas).	6
Figura 5 – Exemplo de um processo de convolução, em que um <i>kernel</i> está sendo aplicado sobre um <i>pixel</i>	7
Figura 6 – Exemplo de uma imagem digital que apresenta ruído.	8
Figura 7 – Limiarização de uma imagem utilizando limiar T	10
Figura 8 – Modelos de arestas.	12
Figura 9 – Configuração de máscara de filtro passa-alta cujos pesos têm valores uniformes.	13
Figura 10 – Efeito de borda por um filtro passa-alta com dimensão de 3 x 3.	14
Figura 11 – Exemplos de elementos estruturantes.	16
Figura 12 – Ilustração da aplicação do operador morfológico de erosão.	17
Figura 13 – Ilustração da operação de erosão.	18
Figura 14 – Ilustração da aplicação do operador morfológico de dilatação.	19
Figura 15 – Ilustração da operação de dilatação.	20
Figura 16 – Ilustração da aplicação do operador morfológico de abertura.	21
Figura 17 – Ilustração da aplicação do operador morfológico de fechamento.	22
Figura 18 – Ilustração da aplicação do operador gradiente morfológico.	23
Figura 19 – Ilustração da aplicação do operador morfológico <i>Top Hat</i>	24
Figura 20 – Ilustração da aplicação do operador morfológico <i>Black Hat</i>	25
Figura 21 – Ilustração da aplicação da inversão.	27
Figura 22 – Detecção de curvas com a aplicação de operadores morfológicos. (a) Imagem inicial e (b) resultado após a aplicação.	28
Figura 23 – Comparação entre os operadores de detecção de bordas aplicados a uma imagem com ruídos.	29
Figura 24 – Exemplo de um esquema de galeria utilizado para seleção do jogo de luz em uma imagem.	30
Figura 25 – Exemplo de galerias geradas para seleção de isovalor em um dado volumétrico de um pé humano. Em verde os isovalores selecionados pelo usuário nas galerias.	31
Figura 26 – Ilustração do fluxo utilizado na aplicação.	31
Figura 27 – Imagem microscópica.	32
Figura 28 – Seleção da imagem.	33

Figura 29 – Seleção da imagem.	33
Figura 30 – Diagrama de casos de uso.	35
Figura 31 – Visualização gráfica de todos os <i>kernels</i> disponíveis, acompanhados de seus respectivos nomes.	37
Figura 32 – Interface da aplicação.	41
Figura 33 – Visualização instantânea na interface após a seleção da imagem.	41
Figura 34 – Representação de uma matriz 7x7 Diamante (a) e sua imagem correspondente (b).	44
Figura 35 – Estrutura da Interface.	44
Figura 36 – Posicionamento das imagens e seus respectivos <i>labels</i>	45
Figura 37 – Posicionamento dos botões.	46
Figura 38 – Imagem de entrada (a) e imagem de referencia (b).	47
Figura 39 – Imagens resultantes após aplicar os operadores morfológicos.	48
Figura 40 – Imagem de entrada (a) e imagem de referencia (b).	50
Figura 41 – Visão da interface ao carregar a imagem inicial.	51
Figura 42 – Imagem resultante após aplicar os operadores morfológicos.	51
Figura 43 – Imagem de entrada (a) e imagem de referencia (b).	53
Figura 44 – Visão da interface ao carregar a imagem inicial.	54
Figura 45 – Imagem resultante após aplicar os operadores morfológicos.	54
Figura 46 – Imagem de entrada (a) e imagem de referencia (b).	56
Figura 47 – Visão da interface ao carregar a imagem inicial.	56
Figura 48 – Imagem resultante após aplicar os operadores morfológicos.	57
Figura 49 – Imagem de entrada.	58
Figura 50 – Visão da interface ao carregar a imagem inicial.	59
Figura 51 – Imagem resultante após aplicar os operadores morfológicos.	59
Figura 52 – Imagem de entrada.	60
Figura 53 – Visão da interface ao carregar a imagem inicial.	61
Figura 54 – Imagem resultante após aplicar os operadores morfológicos.	62
Figura 55 – Imagem de entrada.	63
Figura 56 – Visão da interface ao carregar a imagem inicial	64
Figura 57 – Imagem resultante após aplicar os operadores morfológicos.	64

LISTA DE ABREVIATURAS E SIGLAS

GT	<i>Ground truth</i>
LOG	<i>Laplacian of Gaussians</i>
MP	<i>Megapixels</i>
PDI	Processamento digital de imagens;
RGB	Espaço de cor tricromático: <i>Red, Green, Blue</i>
SSIM	<i>Structural Similarity Image Metric</i>
TIFF	<i>Tagged Image File Format</i>

LISTA DE ALGORITMOS

Algoritmo 1 – Elemento estruturante.	38
Algoritmo 2 – Aplicação de operador morfológico.	38
Algoritmo 3 – Aplicação do operador morfológico de dilatação.	39
Algoritmo 4 – Aplicação do operador morfológico de erosão.	39
Algoritmo 5 – Operação de inversão.	39
Algoritmo 6 – Limiarização pelo método de Otsu (OTSU, 1979).	40

Sumário

1	–	INTRODUÇÃO	1
2	–	FUNDAMENTAÇÃO TEÓRICA	4
2.1		Imagem Digital	4
2.2		Processamento digital de imagens	5
2.2.1		Convolução	7
2.2.2		Filtros	8
2.2.3		Ruídos	8
2.2.4		Segmentação	9
2.2.4.1		Limiarização	9
2.2.5		Detecção de Arestas	11
2.2.6		Filtro passa-alta	12
2.2.7		Extração de Características	13
2.3		Operadores Morfológicos	15
2.3.1		Erosão	16
2.3.2		Dilatação	18
2.3.3		Abertura	20
2.3.4		Fechamento	21
2.3.5		Gradiente Morfológico	22
2.3.6		<i>Top Hat</i>	23
2.3.7		Black Hat	24
2.3.8		Inversão	26
3	–	REVISÃO BIBLIOGRÁFICA	28
4	–	MATERIAIS E MÉTODOS	32
4.1		Aquisição das Imagens	32
4.2		Requisitos do sistema	34
4.2.1		Requisitos funcionais	34
4.2.2		Requisitos não funcionais	34
4.2.3		Diagrama de Casos de Uso	34
4.3		Tecnologias utilizadas	35
4.4		Implementação dos Operadores Morfológicos	37
4.4.1		Elemento estruturante	37
4.4.2		Implementação da técnica de limiarização	40
4.5		Apresentação da Interface	40

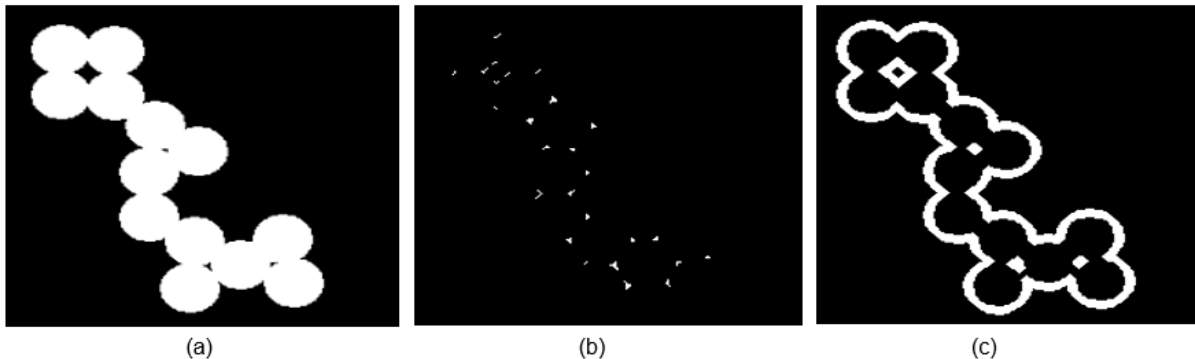
4.5.1	<i>Layout</i> dinâmico	43
4.6	Análise de Similaridade entre Imagens	45
4.6.1	Métrica de Similaridade Estrutural (SSIM)	46
4.6.2	Implementação do Processo	46
5	– RESULTADOS E DISCUSSÃO	47
5.1	Imagens sintéticas	47
5.1.1	Experimento 1 - Ruídos	47
5.1.2	Experimento 2 - Preenchimento	50
5.1.3	Experimento 3 - Palavras	53
5.1.4	Experimento 4 - Círculos	55
5.2	Imagens microscópicas	58
5.2.1	Experimento 1	58
5.2.2	Experimento 2	60
5.2.3	Experimento 3	63
6	– CONCLUSÃO	66
6.1	Considerações Finais	66
6.2	Trabalhos Futuros	66
	REFERÊNCIAS	68

1 INTRODUÇÃO

O universo visual expresso por imagens digitais é amplo e complexo, cheio de detalhes que frequentemente passam despercebidos. No âmbito do Processamento Digital de Imagens (PDI), a procura por técnicas que vão além da mera observação e que permitam uma análise mais profunda desses detalhes pode ser interessante. Nesse cenário, os operadores morfológicos se destacam como ferramentas poderosas, aptas a modelar e desvendar padrões escondidos, proporcionando uma compreensão mais refinada das estruturas nas imagens.

Entretanto, explorar os efeitos dos operadores morfológicos pode se tornar uma tarefa desafiadora. A aplicação de um operador morfológico em uma imagem pode causar um impacto significativo em sua aparência, e para um usuário não especialista, prever o resultado exato de uma operação pode ser difícil. Ilustrando isso, na figura 1 (a), temos uma imagem de entrada, e na figura 1 (b) e (c), é apresentado o resultado dessa mesma imagem com a aplicação de dois operadores morfológicos distintos.

Figura 1 – Imagem original (a) e seus resultados após aplicar os operadores Black Hat (b) e Gradiente (c).



Fonte: Autoria própria.

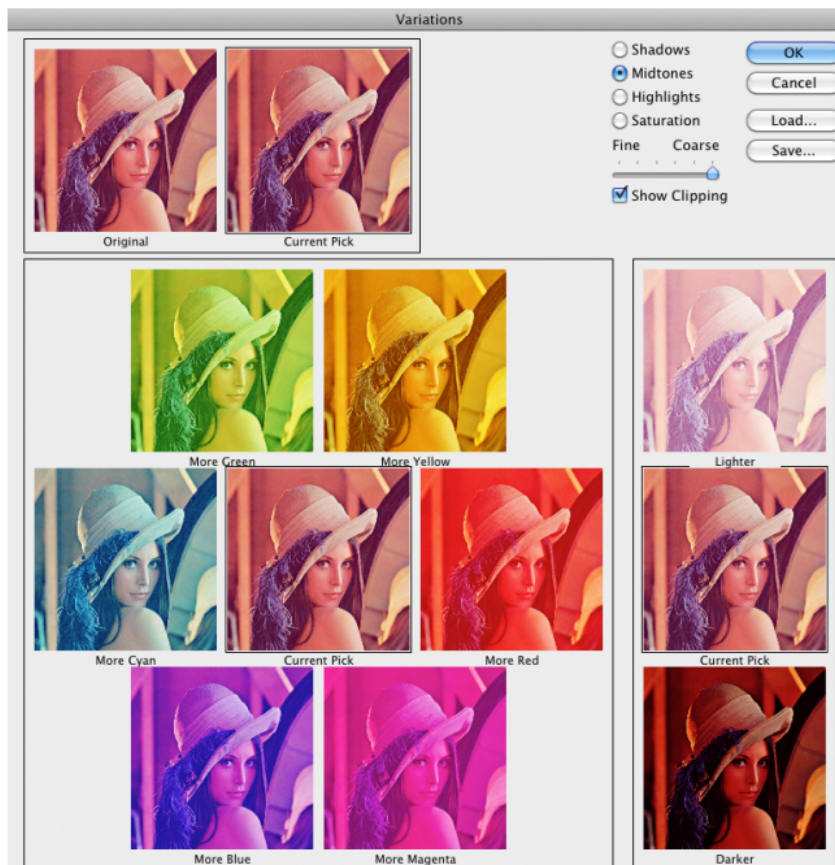
Nota-se que ambos os resultados (b) e (c) na figura 1, geraram imagens totalmente diferentes, destacando atributos diversos da imagem original. Em outras palavras, a aplicação desses operadores pode conduzir a resultados bem diferentes entre si, exigindo vários testes manuais para encontrar uma abordagem interessante, se tornando um processo desgastante e complexo.

E se fosse possível visualizar diversos resultados para uma mesma imagem simultaneamente? Escolher aquele que mais se aproxima do objetivo e traçar um caminho que se aproxima cada vez mais de um resultado ótimo?

Nesse contexto, este trabalho tem como objetivo desenvolver uma aplicação interativa que capacite usuários não especialistas em operadores morfológicos a explorar de forma detalhada os resultados dessas operações em imagens. A aplicação irá fornecer

uma visão abrangente dos efeitos dos operadores morfológicos, apresentando-os em um esquema de galeria, como exemplificado na Figura 2. Essa abordagem permite que os usuários moldem e refinem a visualização de maneira adaptativa, facilitando a aplicação combinada de diversos operadores de forma intuitiva.

Figura 2 – Galerias de Design no Adobe Photoshop CS4.



Fonte: Cunha, Lima e Vieira (2011).

O conceito de galeria já é amplamente difundido nos dias de hoje e é mais comum do que se pode pensar. Por exemplo, em aplicativos dedicados à edição de fotos e redes sociais, os usuários têm a opção de testar variações de efeitos aplicados em uma imagem, escolhendo aquela que mais os agrada. Esse processo não se limita apenas a alterações nas características de coloração, formas e estrutura da imagem, mas também é aplicado em contextos científicos. Em análises científicas, onde é crucial identificar padrões, segmentar imagens e realizar pré-processamentos para otimizar o trabalho posterior, o conceito de galeria pode ser igualmente valioso. Um exemplo prático desse conceito ocorre no pré-processamento de imagens coletadas por microscópio. Nessa situação, pode ser essencial eliminar os elementos que não são o foco da análise e realçar as regiões de interesse. A partir disso, é possível determinar alguns parâmetros importantes, como o diâmetro de algumas espécies ou determinar a morfologia de alguma estrutura. Em resumo, essa abordagem pode permitir conduzir análises tanto quantitativas quanto visuais, contribuindo para uma

compreensão mais completa e precisa do objeto de estudo.

Por fim, a proposta de utilização de galeria de design nesse trabalho, se torna particularmente interessante, pois irá possibilitar a exploração visual de todos os resultados obtidos para cada operador morfológico, permitindo a escolha do resultado ideal.

O restante desse trabalho foi dividido nos seguintes capítulos:

- **Capítulo 2:** Apresenta de maneira abrangente os conceitos e fundamentos que constituem a base teórica essencial para o desenvolvimento deste trabalho;
- **Capítulo 3:** Oferece uma análise de trabalhos recentes relacionados à proposta deste estudo. Examinando literaturas relevantes para contextualizar a pesquisa dentro do panorama atual da área de processamento de imagens;
- **Capítulo 4:** Detalha os materiais e métodos empregados durante o desenvolvimento do trabalho;
- **Capítulo 5:** Apresenta a análise detalhada dos resultados obtidos por meio da implementação proposta;
- **Capítulo 6:** Resume os resultados alcançados durante a pesquisa e destaca possíveis direções para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção, são apresentados os conceitos fundamentais relacionados a imagens, operadores morfológicos, e processamento de imagens. A explanação busca estabelecer uma base sólida para a compreensão dos métodos e técnicas abordados ao longo do trabalho.

2.1 Imagem Digital

Historicamente, as imagens eram frequentemente capturadas e armazenadas de forma analógica. Nesse contexto, a informação visual era representada de maneira contínua, em suportes físicos como fotografias impressas ou transparências. Cada imagem analógica carregava consigo a textura suave e contínua da cena capturada, proporcionando uma experiência visual única. No entanto, com o avanço da tecnologia, surgiu a era das imagens digitais, marcada pela representação discreta da informação visual em *pixels*, possibilitando novas formas de captura, manipulação e armazenamento de imagens.

Uma imagem pode ser vista como uma função bidimensional $f(x,y)$ que possui coordenadas espaciais x e y , e o valor f corresponde à intensidade medida por algum sensor naquele ponto (x,y) . Essa intensidade é um valor não negativo, e está relacionada com os comprimentos de onda presentes naquela região. Dependendo da natureza do sensor, essa imagem poderá ser colorida ou monocromática (tons de cinza). O conjunto \mathcal{U} desses pontos (x,y) define o domínio da imagem. Para formar uma imagem digital, é necessário efetuar duas operações: discretizar o domínio (amostragem) e discretizar a intensidade de f (quantização).

Matematicamente podemos escrever uma imagem como

$$f : \mathcal{U} \subset \mathbb{R}^2 \rightarrow \mathcal{C} \subset \mathbb{R}^m, \quad (1)$$

onde para $m = 3$, teremos imagens coloridas tricromáticas (ex.: RGB) e para $m = 1$, teremos imagens monocromáticas (ex.: tons de cinza). O conjunto \mathcal{C} pode ser chamado de espaço de cor da imagem f .

O domínio discretizado define em um *grid* bidimensional $(M, N) \in \mathbb{Z}^2$ que é representado como uma matriz, de M linhas e N colunas, onde cada índice de pares de inteiros (i,j) é capaz de identificar o ponto na imagem, e o nível de cinza é o valor do elemento dessa matriz naquele determinado ponto, ou seja, $f(i,j)$, como o ilustrado na Figura 3. Os elementos da matriz que representam essa imagem são conhecidos no inglês como *pixels*, uma abreviação de *picture elements* (GONZALEZ; WOODS, 2002).

O valor da intensidade também é discretizado, de forma que será representado por um conjunto finito de valores em uma região do espaço de cor \mathcal{C} . Valores mais baixos de intensidade correspondem a pontos com pouca luz refletida, já mais altos indicam que mais luz refletida foi capturada pelo sensor. Dessa forma, uma região preta

terá intensidade próxima de 0, ao passo que uma região muito clara (ex.: branco) terá intensidade máxima. Considerando uma imagem monocromática ($m = 1$), e a quantização para 8 bits, esses valores irão variar no intervalo $[0,255]$, sendo 0 a representação do preto e 255 a representação do branco, como segue ilustrado na Figura 3. Ao considerar 1 bit apenas para representar a intensidade, teremos uma imagem binária, onde 0 é o preto e 1 o branco.

$$f(x,y) = \begin{bmatrix} 255 & 234 & 241 \\ 128 & 140 & 190 \\ 56 & 69 & 90 \end{bmatrix} \quad (2)$$

A equação (2) apresenta uma matriz que retrata uma imagem digital, com $m=1$, e 8 bits de cor. Nesta matriz, por exemplo, o elemento (1,1) tem o nível de cinza igual a 140. Além da quantização do *pixel*, outro parâmetro que é importante na digitalização de uma imagem é a resolução espacial, que representa o número de amostras existentes em uma imagem (o número de *pixels*), ou seja, $M \times N$ (SCHNEIDER, 2005). A imagem correspondente à matriz mostrada na equação 2 tem uma resolução espacial de 9 *pixels*, ou seja, é uma matriz 3 x 3.

Para ilustrar a importância da resolução espacial em imagens digitais, consideremos o exemplo de uma imagem com resolução de 1920 x 1080. Nesse caso, a imagem resultante possui aproximadamente 2 *megapixels* (1,98MP). Esse valor representa a multiplicação da largura (1920 *pixels*) pela altura (1080 *pixels*) da imagem. Por exemplo, um celular que capture uma foto com 12MP irá gerar imagens digitais com resolução de 4624x2604 *pixels*, destacando a significativa diferença na quantidade de detalhes e nitidez quando comparado a imagens de menor resolução.

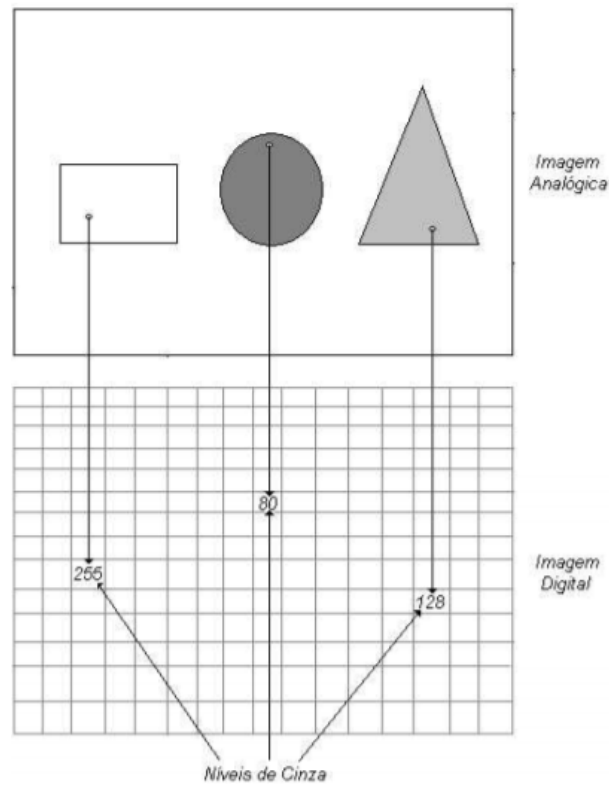
Assim, a resolução espacial não apenas determina a quantidade de *pixels* em uma imagem, mas também influencia diretamente na qualidade e riqueza de detalhes perceptíveis.

Ao representar uma imagem digital em um computador, o sistema de coordenadas utilizado pode ter diferentes origens: pode ter origem semelhante ao sistema cartesiano 2D que conhecemos; pode também ter origem como a ilustrada na Figura 4.

2.2 Processamento digital de imagens

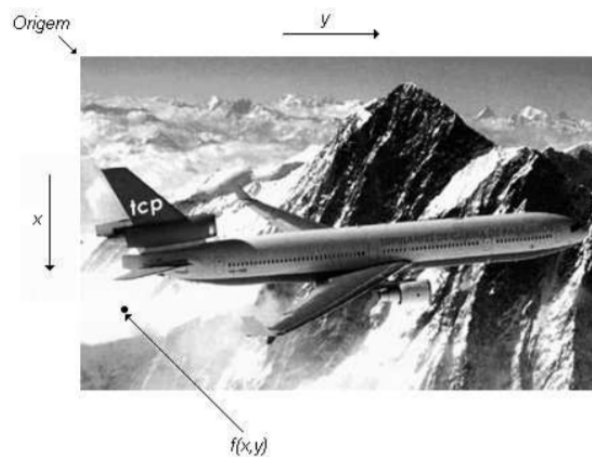
O processamento digital de imagens (PDI) é um campo da ciência da computação que se concentra na manipulação e análise de imagens digitais. É uma área ampla e diversificada, com aplicações em diversas áreas, como medicina, ciência, engenharia e indústria. Como já vimos anteriormente, imagem digital é uma representação de uma cena real ou virtual na forma de uma matriz de *pixels*. Cada *pixel* é um ponto na imagem que pode ser representado por uma combinação de valores de intensidade de cor. A partir disso o PDI pode ser dividido em duas categorias principais:

Figura 3 – Imagem Analógica relacionada com a Imagem Digital.



Fonte: Schneider (2005)

Figura 4 – Imagem Analógica relacionada com a Imagem Digital (Coordenadas).



Fonte: Schneider (2005).

- Manipulação de imagens: Esta categoria inclui operações que alteram a aparência ou o conteúdo de uma imagem.
- Análise de imagens: Esta categoria inclui operações que extraem informações de uma imagem.

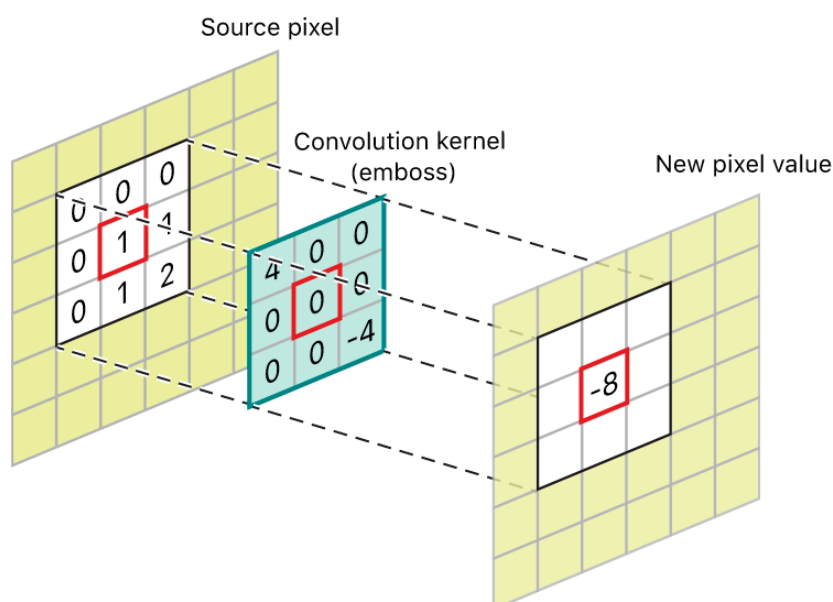
2.2.1 Convolução

A convolução é uma operação matemática de grande importância para diversos operadores comuns em processamento de imagens. Em termos simples, a convolução combina duas matrizes de números, geralmente de tamanhos diferentes, para gerar uma nova matriz. Essa técnica pode ser usada no processamento digital de imagens para implementar operadores cujos valores de *pixel* de saída são simples combinações lineares de certos valores de *pixel* de entrada.

Se tratando do contexto de processamento de uma imagem, uma das matrizes de entrada será uma imagem em nível de cinza, por exemplo. Já a segunda matriz, normalmente é muito menor (matriz de ordem 3, 5, ou 7), é conhecida como máscara (ou *kernel*).

A convolução é realizada deslizando o *kernel* sobre a imagem, geralmente começando no canto superior esquerdo, de modo a mover o *kernel* por todas as posições em que o *kernel* se encaixa totalmente dentro dos limites da imagem. Cada posição do *kernel* corresponde a um único *pixel* de saída, cujo valor é calculado multiplicando-se o valor do *kernel* e o valor subjacente do *pixel* da imagem para cada das células no *kernel* e, em seguida, somando todos esses números (FISHER et al., 2020). É possível observar na Figura 5 uma representação visual desse processo.

Figura 5 – Exemplo de um processo de convolução, em que um *kernel* está sendo aplicado sobre um *pixel*.



Fonte: Developer (2023)

Essa técnica tem semelhanças com a aplicação de operadores morfológicos, uma vez que também combina os valores do *kernel* com os da matriz, de forma deslizante, tendo o *kernel* um papel crucial na manipulação da forma e estrutura da imagem.

2.2.2 Filtros

Um filtro desempenha a função de realizar uma varredura para extrair informações de uma imagem, possibilitando a análise e, conseqüentemente, a correção de imperfeições ou criação de ruídos no momento da transmissão. Os filtros podem ser aplicados em dois cenários: domínio do espaço e domínio da frequência (Fourier).

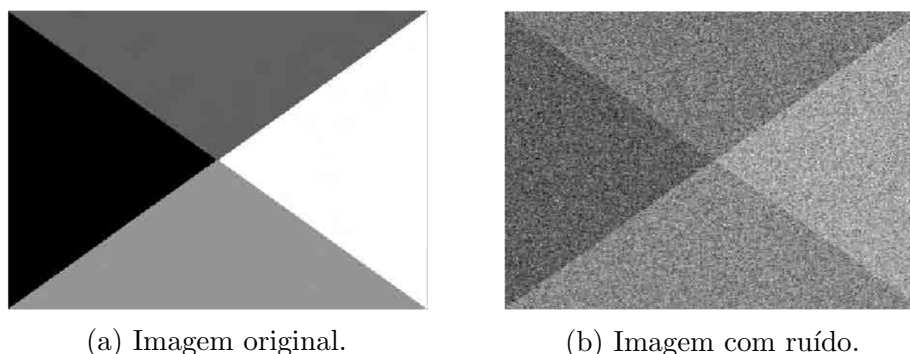
Os filtros de domínio do espaço operam diretamente na imagem, sendo considerados operações locais devido à sua forma de filtragem e à interação com os níveis de cinza em pontos específicos, dependendo dos valores originais e de seus vizinhos. Esses filtros utilizam máscaras de deslocamento compostas por matrizes, onde cada posição possui uma associação específica com outra posição.

A operação começa com a convolução da máscara, que atua como um meio de capturar informações da imagem. Posiciona-se a máscara sobre o *pixel* desejado para a operação de suavização, e o cálculo é realizado com base nos vizinhos do *pixel* escolhido. Esse processo é repetido para todos os *pixels* até obter uma nova imagem. A máscara ajusta o valor de cada *pixel* central com base nos resultados da filtragem, proporcionando a suavização desejada (SANCHES et al., 2015). Essa abordagem permite a modificação de um único *pixel* de cada vez, mantendo a associação com os *pixels* vizinhos durante o processo de filtragem.

2.2.3 Ruídos

Podem ser causados por informações falsas dentro de imagens ou até mesmo pela variação de brilho, resultando em informações indesejáveis em uma imagem. Conforme apresentado em Seara (1998), toda aquisição de imagens é propícia a algum tipo de ruído. É conhecido que ruídos sempre aconteceram no ambiente digital, não existindo uma maneira de prevêê-los, o que acaba se tornando algo inevitável. Existem dois tipos de ruídos específicos nas imagens: Gaussiano (*Gaussian Noise*) e Sal e Pimenta (*Salt and Pepper Noise*), ilustrados na Figura 6.

Figura 6 – Exemplo de uma imagem digital que apresenta ruído.



Fonte: D'Ippólito (2005).

O ruído Gaussiano apresenta um comportamento equivalente ao de se adicionar ao valor original de uma função f (imagem) uma função g (ruído) que possui distribuição Gaussiana. Ele pode ser formado pela má iluminação durante a captura da imagem, por altas temperaturas no sensor ou problemas na transmissão da imagem, podendo deixar a imagem extremamente danificada (KLEIN; GALLAGER, 2001). O ruído *Salt and Pepper* ocorre a partir de erros na transmissão de dados. Com a transmissão da imagem, algum *pixel* pode se corromper, alterando os tons de cinza de cada *pixel* vizinho, conforme Ribeiro (2006).

2.2.4 Segmentação

A segmentação de imagens desempenha um papel fundamental em diversas áreas do processamento de imagens e visão computacional, incluindo análise médica, reconhecimento de objetos e robótica. Essa etapa envolve a subdivisão de uma imagem em distintas regiões, cada uma representando um objeto ou componente significativo da cena (GONZALEZ; WOODS, 2002). A segmentação é caracterizada pelo processo de dividir uma imagem digital em subconjuntos de *pixels*, nos quais cada subconjunto representa uma região homogênea em termos de propriedades visuais, como cor, textura ou brilho.

Dentre os objetivos da segmentação, destacam-se:

- **Simplificar a Análise da Imagem:** Facilita a interpretação e o processamento da imagem, simplificando sua complexidade visual;
- **Extrair Características Relevantes:** Permite a extração de características significativas da imagem para aplicações específicas, como reconhecimento de objetos e classificação de texturas;
- **Melhorar a Qualidade da Imagem:** Contribui para aprimorar a qualidade visual, removendo ruídos e artefatos indesejados.

Existem diversas técnicas para realizar a segmentação de imagens, cada uma com suas vantagens e desvantagens (SERRA; SOILLE, 1994). As técnicas mais comuns podem ser categorizadas da seguinte forma:

- **Baseadas em Limiarização:** Definem um limiar para separar os *pixels* de diferentes regiões.
- **Baseadas em Regiões:** Agrupam *pixels* com propriedades visuais semelhantes em regiões homogêneas.
- **Baseadas em Bordas:** Identificam as bordas entre diferentes regiões da imagem.

Na Seção 2.2.4.1, será abordada com mais detalhes a técnica de limiarização, a qual foi escolhida como abordagem para a binarização das imagens neste trabalho.

2.2.4.1 Limiarização

A limiarização é comumente conhecida como uma importante abordagem da segmentação de imagens, sendo um caso específico de segmentação. Devido ao fato da

limiarização produzir uma imagem binária à saída, o processo também é denominado, muitas vezes, binarização. Essa técnica possui um princípio simples, que tem como objetivo separar as regiões de uma imagem em duas classes (o fundo e o objeto), portanto, as regiões restantes são classificadas como regiões não interessantes. Dessa maneira, na limiarização ou binarização, encontra-se em geral objetos pretos (0) sobre um fundo branco (255), resultando em uma imagem binária (preto e branca) com duas classes. Porém, a limiarização simples é eficiente apenas quando os objetos têm níveis de cinza bem distintos, ou seja, regiões com intensidade uniforme resultam em picos elevados no histograma. Logo, se os níveis de cinza dos *pixels* do objeto e do fundo apresentar distintamente duas classes na forma de dois picos a limiarização é trivial, pois o objetivo é encontrar o vale entre os dois picos, ou seja, encontrando um limiar T que separe as duas classes (NETA; DUTRA; ERTHAL, 2008).

Matematicamente, a limiarização pode ser definida como descrito na equação (3):

$$g(x,y) = \left\{ \begin{array}{l} 1, \text{ se } f(x,y) > T \\ 0, \text{ se } f(x,y) \leq T \end{array} \right\}, \quad (3)$$

onde: $f(x,y)$ = imagem de entrada, T = valor do limiar, $g(x,y)$ = imagem de saída (limiarizada). Na Figura 7, podemos observar um exemplo de uma imagem após usar a técnica de limiarização.

Figura 7 – Limiarização de uma imagem utilizando limiar T .



(a) Imagem em escala de cinza.



(b) Imagem limiarizada $T = 64$.

Fonte: Souza e Correia (2007).

Portanto, *pixels* rotulados com 1 correspondem aos objetos (região de interesse), enquanto que aqueles rotulados com 0 correspondem ao fundo (tudo que não seja região de interesse).

Entretanto, a segmentação simples por limiarização, se torna inviável em imagens que possuem luminosidade irregular, visto que a imagem é composta pelo produto de uma

componente de iluminação e uma componente de reflectância. Quando se tem a iluminação uniforme, os picos e vales do histograma da imagem são bem definidos, porém o mesmo não ocorre com a iluminação não uniforme, onde são apresentados diversos ruídos no histograma, também conhecidos como picos e vales pouco profundos.

2.2.5 Detecção de Arestas

Antes de falarmos sobre detecção de arestas, devemos entender melhor o que é a mesma. Uma aresta pode ser considerada um conjunto de *pixels* que estão conectados, e se encontram na borda ou no contorno entre duas regiões. Comumente é confundido bordas com arestas, porém a borda de uma região finita, resulta em um caminho fechado e é, portanto, um conceito global. Já as arestas são constituídas por *pixels* onde ocorre uma grande variação de intensidade de cor na imagem, e é, portanto um conceito local (OLIVEIRA, 2014).

O processo de detecção de arestas ao longo de uma imagem, acontece quando se verifica a existência de uma grande variação nos tons de cinza, e quanto maior a intensidade dessa variação, melhor será a detecção, a menos que seja um caso ideal, onde independente da variação, a detecção será realizada. De forma matemática, a taxa da variação instantânea de uma grandeza, é conhecida por ser a primeira derivada de uma função, sendo assim, é admissível o uso das derivadas por grande parte dos detectores de arestas.

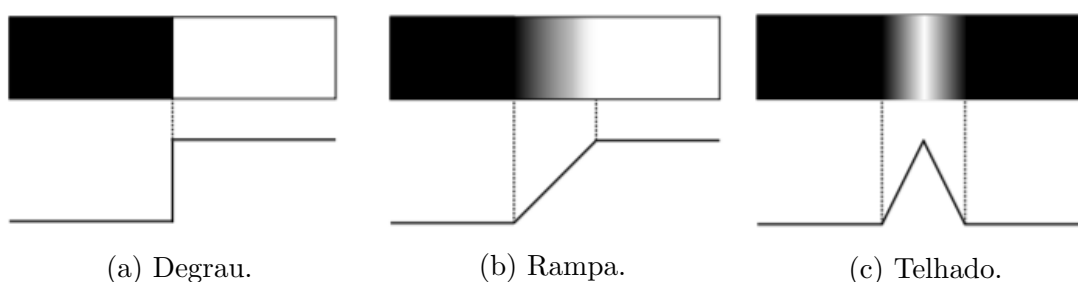
Podemos considerar que a detecção de arestas é, indiscutivelmente uma das operações básicas suscetível a ser realizada em uma imagem. Isso acontece em função do fato de que através das bordas de um objeto, é possível identificar sua forma, sendo algo muito útil para resolução de diversos problemas relacionados a área de visão computacional e processamento de imagens, como por exemplo a detecção de círculos ao longo de uma imagem. Nesse contexto, a qualidade dos resultados obtidos através de um detector de arestas impacta diretamente no sucesso de aplicações que tem foco em imagens. A detecção de arestas pode ser complementada e aprimorada por operações morfológicas, como erosão e dilatação. Essas operações podem realçar ou suavizar bordas, contribuindo para uma segmentação mais precisa e melhorando a qualidade geral da análise de imagem.

Como esse procedimento identifica grupos de *pixels* que definem uma borda, também pode ser visto como uma linha em que a intensidade dos *pixels* em ambos os lados é muito superior/inferior em relação ao outro. As principais etapas relacionadas a esta classe de algoritmos segundo Burger et al. (2009) são:

- **Suavização:** realizada com a finalidade eliminar alguns ruídos que a imagem possa conter, de modo a facilitar o processo de detecção;
- **Detecção dos pontos da aresta:** operação local que determina os pontos candidatos a serem pontos da borda;
- **Localização da aresta:** retira os pontos (falsos candidatos) que não pertencem à borda, deixando apenas os que realmente interessam.

Os modelos de detecção de arestas podem ser classificados de acordo com a variação de intensidade e distância. A aresta em degrau, é o modelo onde ocorre variação com 1 *pixel* de distância, como em imagens criadas para serem utilizadas em modelagem de animações e sólidos. Pela mudança de intensidade com 1 *pixel* de distância ocorrer de modo natural, pode ser considerado um modelo limpo e natural, sem a necessidade de qualquer etapa de processamento, como por exemplo a suavização, para que as arestas apareçam. No mundo real, as imagens digitais possuem arestas com ruídos e desfoque devido a uma serie fatores externos, por conta disso, as arestas acabam apresentando um perfil de rampa na sua variação de intensidade. A inclinação da rampa é inversamente proporcional ao grau de definição da aresta (GONZALEZ; WOODS, 2002). Como podemos notar, nesse modelo não existe mais uma aresta fina onde a sua variação possui uma distância de 1 *pixel*. Ao invés disso, temos uma distância bem maior, e um ponto de aresta é qualquer ponto contido na rampa, e um segmento de aresta será o conjunto conectado dos pontos que estão na rampa. Um outro modelo de aresta é o que chamamos de aresta em forma de telhado também conhecido como *roof edge*. Ela é composta basicamente de apenas uma linha que se destaca ao cortar uma determinada região da imagem. Se esta aresta possui uma distância (essa distância é o comprimento entre os dois lados que a cercam) mínima (1 *pixel*), esta, nada mais é do que uma linha com 1 *pixel* de espessura que atravessa determinada região da imagem (OLIVEIRA, 2014). Todos esses modelos estão ilustrados na Figura 8.

Figura 8 – Modelos de arestas.



Fonte: Oliveira (2014).

2.2.6 Filtro passa-alta

Os filtros passa-alta tem o objetivo de realçar os detalhes de uma imagem, ou seja, realçar os componentes de alta frequência espacial, portanto, a técnica consiste na remoção dos componentes de baixa frequência enquanto preserva as variações locais (alta frequência) de uma imagem. Esse procedimento pode tanto realçar bordas de determinadas áreas quanto elevar as variações de brilho em uma imagem. Um *kernel* é utilizado, com um valor alto no centro, e com valores negativos circundantes. Quando se trata de filtros passa-alta, as suas dimensões são somente impares (3 x 3, 5 x 5 ...). O tamanho da mascara irá influenciar no

resultado, pois quanto maior, mais altas frequências espaciais serão enfatizadas, ou seja, os detalhes geométricos da imagem. Os filtros com tamanho maior que 100×100 *pixels*, podem ser utilizados para eliminar de maneira seletiva determinados tipos de baixa frequência, como por exemplo sombreamentos. Enquanto esses filtros atuam nas intensidades de *pixel*, existem operações morfológicas que influenciam na estrutura da imagem, destacando ou removendo regiões com base em padrões geométricos. Combinar ambas as abordagens oferece uma maneira abrangente de aprimorar e evidenciar características específicas na imagem.

Na Figura 9 pode-se observar duas máscaras, uma com tamanho 3×3 , e a outra com tamanho 5×5 . Pode-se notar que a soma de todos os pesos tem resultado igual a zero, isso faz com que seja removida a média local, produzindo uma imagem a qual mede o desvio de entrada em relação a média local. Se a máscara for aplicada em uma região homogênea da imagem, o resultado será um valor digital muito baixo ou próximo de zero, ao mesmo tempo que nas regiões não homogêneas, o resultado será um realçamento. Um filtro passa-alta aplicado com esses valores de peso, pode resultar em números digitais negativos. Isso é corrigido de forma automática pelos algoritmos, que realizam um processo de normalização dos valores digitais para o intervalo de quantização da imagem após a convolução.

Figura 9 – Configuração de máscara de filtro passa-alta cujos pesos têm valores uniformes.

1/9	<table style="border-collapse: collapse; text-align: center;"> <tr><td style="border: 1px solid black; padding: 2px;">-1</td><td style="border: 1px solid black; padding: 2px;">-1</td><td style="border: 1px solid black; padding: 2px;">-1</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">-1</td><td style="border: 1px solid black; padding: 2px;">+8</td><td style="border: 1px solid black; padding: 2px;">-1</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">-1</td><td style="border: 1px solid black; padding: 2px;">-1</td><td style="border: 1px solid black; padding: 2px;">-1</td></tr> </table>	-1	-1	-1	-1	+8	-1	-1	-1	-1
-1	-1	-1								
-1	+8	-1								
-1	-1	-1								

1/25	<table style="border-collapse: collapse; text-align: center;"> <tr><td style="border: 1px solid black; padding: 2px;">-1</td><td style="border: 1px solid black; padding: 2px;">-1</td><td style="border: 1px solid black; padding: 2px;">-1</td><td style="border: 1px solid black; padding: 2px;">-1</td><td style="border: 1px solid black; padding: 2px;">-1</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">-1</td><td style="border: 1px solid black; padding: 2px;">-1</td><td style="border: 1px solid black; padding: 2px;">-1</td><td style="border: 1px solid black; padding: 2px;">-1</td><td style="border: 1px solid black; padding: 2px;">-1</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">-1</td><td style="border: 1px solid black; padding: 2px;">-1</td><td style="border: 1px solid black; padding: 2px;">+25</td><td style="border: 1px solid black; padding: 2px;">-1</td><td style="border: 1px solid black; padding: 2px;">-1</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">-1</td><td style="border: 1px solid black; padding: 2px;">-1</td><td style="border: 1px solid black; padding: 2px;">-1</td><td style="border: 1px solid black; padding: 2px;">-1</td><td style="border: 1px solid black; padding: 2px;">-1</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">-1</td><td style="border: 1px solid black; padding: 2px;">-1</td><td style="border: 1px solid black; padding: 2px;">-1</td><td style="border: 1px solid black; padding: 2px;">-1</td><td style="border: 1px solid black; padding: 2px;">-1</td></tr> </table>	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	+25	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1																						
-1	-1	-1	-1	-1																						
-1	-1	+25	-1	-1																						
-1	-1	-1	-1	-1																						
-1	-1	-1	-1	-1																						

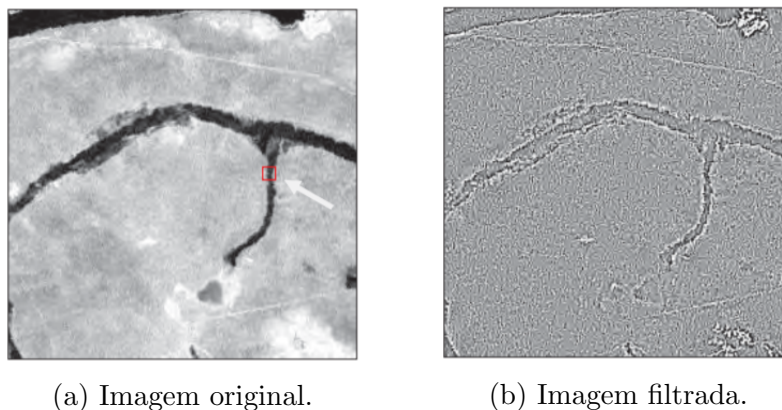
Fonte: Meneses e Almeida (2012).

Na Figura 10 foi aplicado um filtro passa-alta 3×3 com o objetivo de ressaltar o limite ou efeito de borda. Entretanto, quando o objetivo é fazer uma interpretação visual temática da imagem, é desejável reforçar os componentes de alta frequência que representam os detalhes da imagem, sem eliminar totalmente os componentes de baixa frequência (MENESES; ALMEIDA, 2012).

2.2.7 Extração de Características

Os recursos de uma imagem, também conhecidos como atributos ou características, são fundamentais no processo de identificação de uma imagem. O objetivo da extração de características é extrair as propriedades de uma imagem, podendo então diferenciá-la de outras. Existem alguns tipos de características de imagem que são frequentemente usados, sendo eles: características de cor, textura, forma e relações espaciais. Através da extração

Figura 10 – Efeito de borda por um filtro passa-alta com dimensão de 3 x 3.



Fonte: Meneses e Almeida (2012).

dos recursos de imagem, eles devem ser capazes de descrever objetos de maneira abstrata e concreta, de modo a fornecer uma base confiável para a classificação subsequente. Segundo Xiangang (2008) os bons recursos devem ter as seguintes características

- Exclusividade: cada objeto individual deve ter sua própria representação única, que pode ser usada para distingui-lo de outros objetos;
- Integridade: O algoritmo pode descrever as características do objeto como um todo;
- Invariância sob a estrutura geométrica: os recursos gerados não devem mudar com as mudanças de posição, tamanho e direção da rotação, para evitar que os resultados sejam afetados;
- Agilidade: métodos descritivos podem refletir facilmente as diferenças entre objetos semelhantes;
- Abstração: As informações úteis sobre as características podem ser extraídas de muitos detalhes da imagem. A descrição dos objetos é simples e eficaz.

A representação e extração de recursos de imagem são a etapa básica na análise de imagens. Boas características podem identificar completamente o objeto que precisa ser identificado.

Quando falamos de características de forma, elas podem ser divididas em duas categorias, uma baseada nas características da borda e a outra baseada nas características regionais. Por conseguinte, os métodos de extração de recursos de forma de imagem também são divididos em métodos de extração de recursos baseados em limites e métodos de extração de recursos baseados em região. Como a intuitividade e a compreensão dos recursos de forma, os alvos nas imagens podem ser identificados muito bem. No entanto, essa abordagem necessita de um modelo matemático sólido. Quando o alvo está fora de forma, o resultado não é confiável. A precisão do resultado da extração do recurso de forma depende do efeito de pré-segmentação (LIU; SHI, 2011).

2.3 Operadores Morfológicos

A morfologia matemática é um ramo da matemática aplicada que se concentra no estudo da forma e da estrutura de objetos e imagens. Ela foi desenvolvida na década de 1980 por Serra (1982). Os operadores morfológicos, fundamentais nessa abordagem, constituem um conjunto de ferramentas matemáticas empregadas na manipulação de imagens. Esses operadores partem do princípio de que uma imagem pode ser considerada como uma superfície topológica.

Inicialmente aplicados em contextos médicos, como na identificação de estruturas ósseas e órgãos, os operadores morfológicos ganharam destaque em diversas áreas, como visão computacional, análise de imagens e cartografia. Essa versatilidade ampliou sua aplicação para além do campo inicial, tornando-os uma ferramenta valiosa em várias disciplinas. As seções seguintes detalharão os operadores morfológicos específicos utilizados neste trabalho, proporcionando uma visão mais aprofundada sobre sua aplicação e contribuições.

As operações entre conjuntos desempenham um papel fundamental na morfologia matemática, desempenhando um papel crucial no processamento de imagens. Essas operações estão intrinsecamente ligadas aos operadores morfológicos, fornecendo a base teórica necessária para entender as transformações aplicadas às imagens binárias. A seguir, abordaremos as definições dessas operações, fundamentadas em Pedrini e Schwartz (2008):

Sejam A e B duas imagens binárias representadas pelos conjuntos no espaço \mathbb{Z}^2 com componentes $a = (a_1, a_2)$ e $b = (b_1, b_2)$, respectivamente, ou seja, pares ordenados formados pelas coordenadas dos *pixels* dos objetos em A e B . Um conjunto A de coordenadas de *pixels* que satisfazem uma determinada condição é expresso como $A = \{a \mid \text{condição}\}$.

A união de A e B , denotada $A \cup B$, é o conjunto dos elementos que pertencem a A , B ou ambos, expressa como

$$A \cup B = \{c \mid c \in A \text{ ou } c \in B\} \quad (4)$$

A *intersecção* de dois conjuntos A e B , denotada $A \cap B$, é o conjunto de todos os elementos que pertencem a ambos os conjuntos, denotada por

$$A \cap B = \{c \mid c \in A \text{ e } c \in B\} \quad (5)$$

A *translação* de A pelo elemento p , denotada $A + p$, é definida como

$$A + p = \{a + p \mid a \in A\} \quad (6)$$

A *reflexão* de A , denotada \hat{A} , é definida como

$$\hat{A} = \{-a \mid a \in A\} \quad (7)$$

O *complemento* de A , denotado A^c , é o conjunto de todos os *pixels* que não pertencem a A , dado por

$$A^c = \{p \mid p \notin A\} \quad (8)$$

A *diferença* entre A e B , denotada $A - B$, é o conjunto de todos os elementos que pertencem a A , mas não pertencem a B , expressa por

$$A - B = \{p \mid p \in (A \cup B^c)\} \quad (9)$$

Um operador morfológico é um mapeamento entre o conjunto A que define a imagem e um conjunto B , chamado elemento estruturante, também definido em \mathbb{Z}^2 . O elemento estruturante pode ser pensado como uma matriz. O elemento estruturante é expresso com respeito a uma origem local. Alguns exemplos típicos de elementos estruturantes são mostrados na Figura 11, a origem está marcada com uma cruz (+). Os pontos pertencentes ao objeto são marcados com um círculo preenchido. É interessante notar que a origem não é necessariamente um elemento de B , como é o caso do exemplo mais a direita da Figura 11.

Figura 11 – Exemplos de elementos estruturantes.



Fonte: Pedrini e Schwartz (2008).

2.3.1 Erosão

A erosão em processamento digital de imagens é uma operação que consiste em remover *pixels* da imagem original com base na forma e tamanho do elemento estruturante, conhecido como *máscara* ou *kernel*. Essa operação cria um efeito semelhante a um desgaste na imagem, onde regiões pequenas e detalhes finos são eliminados (SERRA; SOILLE, 1994). Ela é uma operação não linear, o que significa que ela não preserva as propriedades geométricas da imagem original. Por exemplo, a erosão pode reduzir o tamanho dos objetos na imagem, ou até mesmo removê-los completamente. O processo de erosão pode ser descrito da seguinte forma:

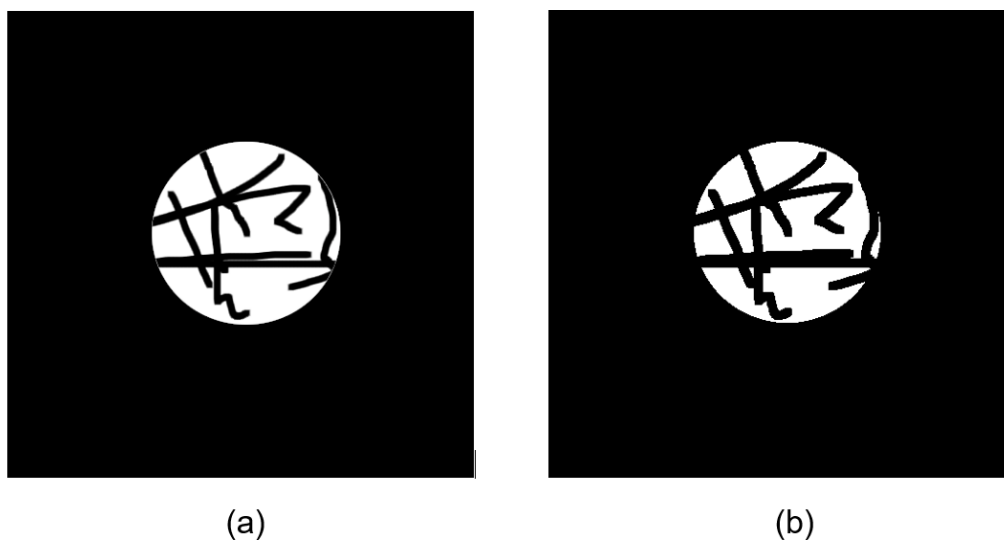
1. O *kernel* é deslizado sobre a imagem.
2. Em cada posição do *kernel*, verifica-se se todos os *pixels* da máscara coincidem com os *pixels* da imagem sob o *kernel*.
3. Se todos os *pixels* coincidirem, o *pixel* no centro do *kernel* permanece na imagem resultante; caso contrário, é removido.

A erosão é uma operação morfológica versátil que pode ser usada em uma variedade de aplicações, incluindo:

- Remoção de ruído: A erosão pode ser usada para remover ruído de uma imagem, como *pixels* isolados ou linhas finas (SERRA; SOILLE, 1994); .
- Segmentação: A erosão pode ser usada para segmentar uma imagem, identificando os componentes conectados (GONZALEZ; WOODS, 2002);
- Detecção de bordas: A erosão pode ser usada para detectar bordas em uma imagem, tornando-as mais visíveis (ROSIN, 2014).
- Simplificação de objetos: A erosão pode ser usada para simplificar objetos em uma imagem, removendo detalhes desnecessários (WANG; LIU; WANG, 2022).

Na Figura 12 é apresentado um exemplo de como ficaria uma imagem após a aplicação do operador morfológico de erosão em uma imagem, onde a imagem original está representada à esquerda e a imagem resultante da erosão representada à direita.

Figura 12 – Ilustração da aplicação do operador morfológico de erosão.



Fonte: Autoria própria.

De maneira simples e objetiva, podemos expressar a erosão da seguinte forma : Dada uma imagem binária A e um elemento estruturante B (ou *kernel*), a erosão de A por B , denotada por $\varepsilon(A,B)$, é definida como:

$$\varepsilon(A,B) = \{z \mid (B)_z \subseteq A\} \quad (10)$$

Onde:

- $(B)_z$ representa a translação do elemento estruturante B pelo vetor z .
- \subseteq denota a operação de inclusão de conjuntos.

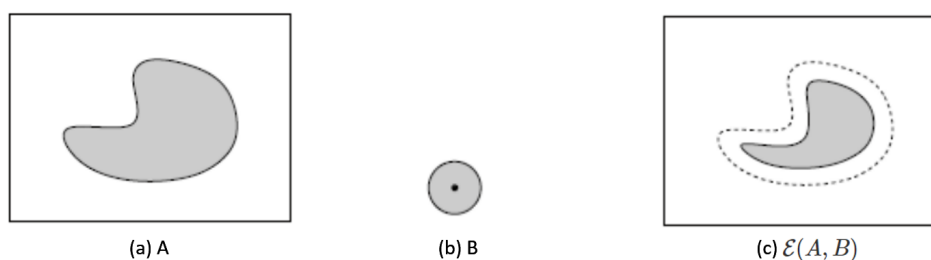
A operação de erosão basicamente analisa cada *pixel* da imagem A e verifica se, ao posicionar o elemento estruturante B centrado nesse *pixel*, todos os *pixels* do elemento estruturante B estão contidos em A . Se sim, o *pixel* central é mantido em $\varepsilon(A,B)$, caso contrário, é removido.

De forma detalhada :

- Para cada ponto z na imagem, coloque o elemento estruturante B centrado em z .
- Se todos os *pixels* de B estiverem contidos em A para essa posição, o ponto z é mantido em $\varepsilon(A,B)$.
- Caso contrário, o ponto z é removido.

Na figura 13 é possível observar uma ilustração do efeito da operação de erosão.

Figura 13 – Ilustração da operação de erosão.



Fonte: Pedrini e Schwartz (2008).

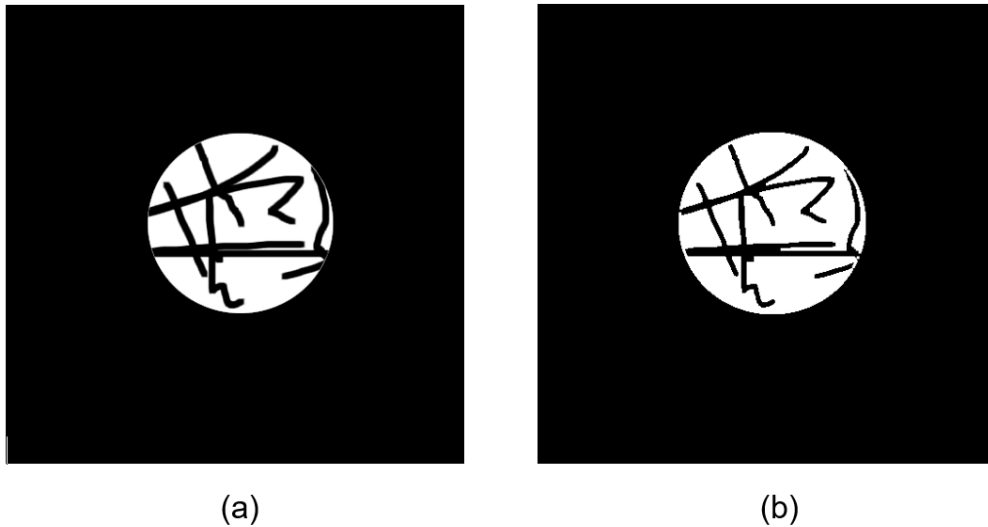
O elemento estruturante B utilizado para alterar a imagem é um disco circular, cuja origem localiza-se no centro do círculo. As linhas tracejadas nas figura delimitam o contorno original do objeto na imagem e as linhas sólidas mostram o resultado após a erosão (c). O elemento estruturante é que determina quais porções da forma são afetadas pela operação.

2.3.2 Dilatação

O operador morfológico de dilatação é uma operação não linear que consiste em aumentar o tamanho dos objetos em uma imagem. Ela é definida como a operação que, para cada *pixel* da imagem, substitui seu valor pelo maior valor dos *pixels* da vizinhança do *kernel* (SERRA; SOILLE, 1994).

Na Figura 14 é apresentado um exemplo de como ficaria uma imagem após a aplicação do operador morfológico de dilatação em uma imagem, onde a imagem original está representada à esquerda e a imagem resultante da dilatação representada à direita.

Figura 14 – Ilustração da aplicação do operador morfológico de dilatação.



Fonte: Autoria própria.

De maneira simples e objetiva, podemos expressar a dilatação da seguinte forma: Dada uma imagem binária A e um elemento estruturante (ou *kernel*) B , a dilatação de A por B , denotada por $\delta(A,B)$, é definida como:

$$\delta(A,B) = \{z \mid (B)_z \cap A \neq \emptyset\} \quad (11)$$

onde:

- $(B)_z$ representa a translação do elemento estruturante B pelo vetor z .
- \cap denota a operação de interseção de conjuntos.
- \emptyset representa o conjunto vazio.

A operação de dilatação basicamente analisa cada *pixel* da imagem A e verifica se, ao posicionar o elemento estruturante B centrado nesse *pixel*, há interseção de *pixels* entre o elemento estruturante B e A . Se sim, o *pixel* central é mantido em $\delta(A,B)$, caso contrário, é removido.

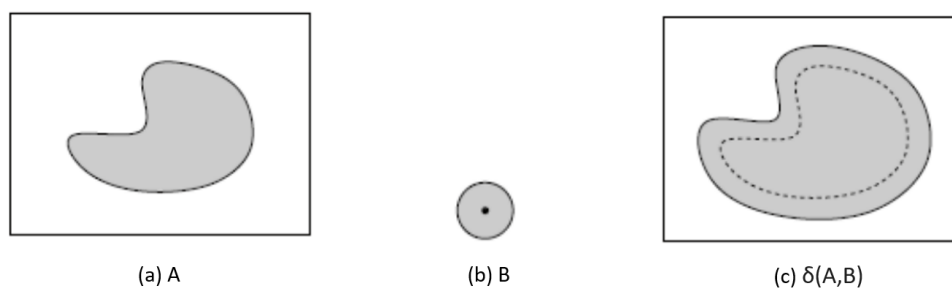
Ou de forma detalhada :

- Para cada ponto z na imagem, coloque o elemento estruturante B centrado em z .
- Se houver interseção de *pixels* entre B e A para essa posição, o ponto z é mantido em $\delta(A,B)$.
- Caso contrário, o ponto z é removido.

Na figura 15 é possível observar uma ilustração do efeito da operação de erosão.

O elemento estruturante B utilizado para alterar a imagem é um disco circular, cuja origem localiza-se no centro do círculo. As linhas tracejadas nas figura delimitam o contorno original do objeto na imagem e as linhas sólidas mostram o resultado após

Figura 15 – Ilustração da operação de dilatação.



Fonte: Pedrini e Schwartz (2008).

a dilatação (c). O elemento estruturante é que determina quais porções da forma são afetadas pela operação.

2.3.3 Abertura

No âmbito do processamento digital de imagens, o operador morfológico de abertura desempenha um papel crucial, especialmente na fase de pré-processamento, contribuindo para aprimorar a qualidade das imagens e extrair características relevantes. Este operador, derivado da teoria dos conjuntos e fundamentado na morfologia matemática, oferece uma abordagem única para suavizar bordas, remover pequenos detalhes indesejados e separar objetos previamente conectados (SERRA; SOILLE, 1994). Essa operação morfológica é composta por duas etapas fundamentais: a erosão, seguida pela dilatação. Essa sequência de operações permite que o operador atenuar pequenos detalhes e realce características estruturais mais significativas.

A abertura é eficaz na eliminação de pequenos detalhes, ruídos e irregularidades que podem comprometer a precisão em análises posteriores. Ela é uma operação idempotente, ou seja, aplicar a abertura repetidamente não altera o resultado após a primeira aplicação (SERRA; SOILLE, 1994). A escolha do tamanho do *kernel* na erosão e dilatação afeta diretamente o impacto da abertura na imagem, sendo fundamental selecionar parâmetros adequados para cada aplicação específica.

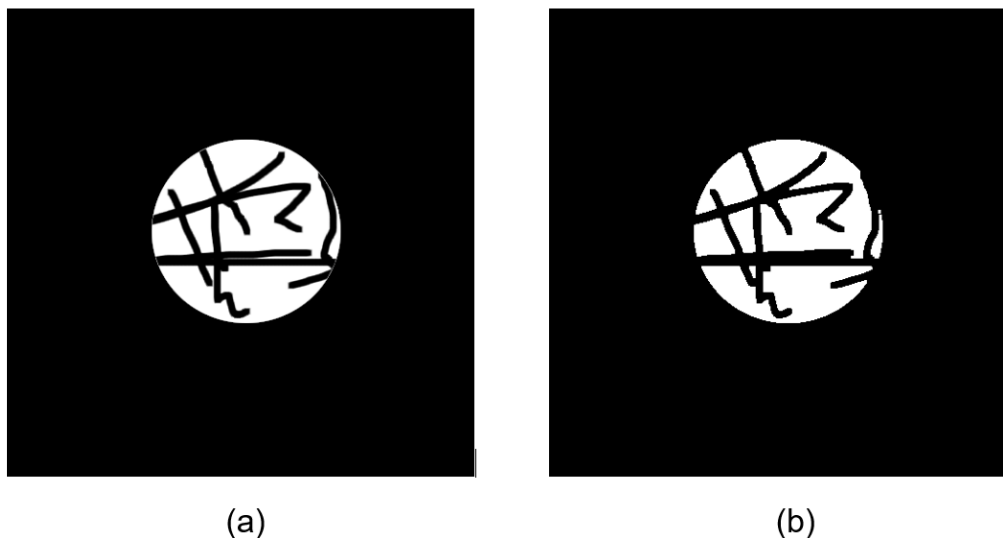
Na Figura 16 é apresentado um exemplo de como ficaria uma imagem após a aplicação do operador morfológico de abertura em uma imagem, onde a imagem original está representada à esquerda e a imagem resultante da abertura representada à direita.

Essa operador é definido como uma erosão seguida de uma dilatação e é expressa matematicamente da seguinte forma:

$$\text{Abertura}(A, B) = \delta(\varepsilon(A, B), B) \quad (12)$$

Essa fórmula indica que a abertura de um conjunto A com um elemento estruturante B é obtida primeiro aplicando uma erosão (ε) em A usando B e, em seguida, aplicando uma dilatação (δ) ao resultado usando o mesmo elemento estruturante B . Em termos mais

Figura 16 – Ilustração da aplicação do operador morfológico de abertura.



Fonte: Autoria própria.

simples, a abertura morfológica remove pequenos detalhes e, em um certo sentido, abre espaços entre objetos, preservando as características gerais das regiões maiores.

2.3.4 Fechamento

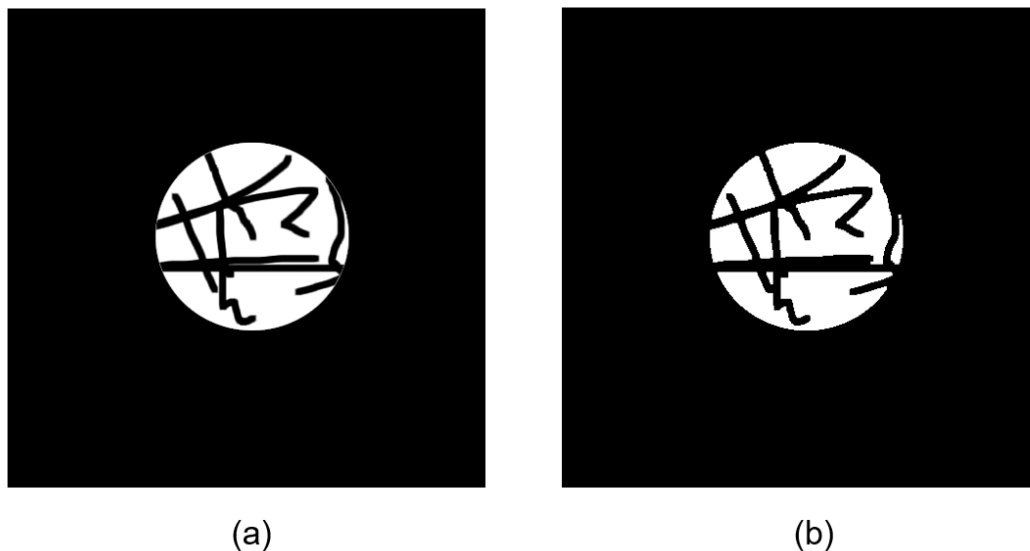
O operador morfológico de fechamento é uma técnica fundamental na morfologia matemática, desempenhando um papel crucial no processamento digital de imagens. Sua aplicação visa suavizar bordas, preencher lacunas e aprimorar a integridade de objetos, tornando-se uma ferramenta versátil para melhorar a qualidade visual e facilitar análises subsequentes. Ele é composto por duas etapas principais: a dilatação seguida pela erosão. Essa sequência de operações é projetada para eliminar pequenas aberturas e suavizar as bordas dos objetos na imagem. O operador de fechamento é utilizado em diversas aplicações práticas, incluindo:

- Remoção de Ruídos e Detalhes Pequenos: Ao preencher pequenas aberturas e suavizar bordas, o fechamento é eficaz na eliminação de ruídos e detalhes insignificantes, resultando em uma imagem mais limpa e coesa (LI; WANG, 2014a); .
- Integridade de Objetos: Contribui para a manutenção da integridade estrutural de objetos na imagem, prevenindo a fragmentação e conectando componentes desconexos;
- Preparação para Análises Subsequentes: O fechamento é frequentemente aplicado como uma etapa preliminar antes de análises mais avançadas, como segmentação de imagem ou extração de características, proporcionando uma base mais sólida para essas análises.

Na Figura 17 é apresentado um exemplo de como ficaria uma imagem após a

aplicação do operador morfológico de fechamento em uma imagem, onde a imagem original está representada à esquerda e a imagem resultante da abertura representada à direita.

Figura 17 – Ilustração da aplicação do operador morfológico de fechamento.



Fonte: Autoria própria.

Sua expressão matemática é dada pela dilatação seguida de uma erosão. Matematicamente, o fechamento (F) de um conjunto A com um elemento estruturante B é definido como:

$$F(A, B) = \varepsilon(\delta(A, B), B) \quad (13)$$

Nessa formulação, $\delta(A, B)$ representa a dilatação de A por B , e $\varepsilon(*, B)$ representa a erosão pelo mesmo elemento estruturante B . Dessa forma, o operador vai fechar pequenos buracos ou desconexões em uma região.

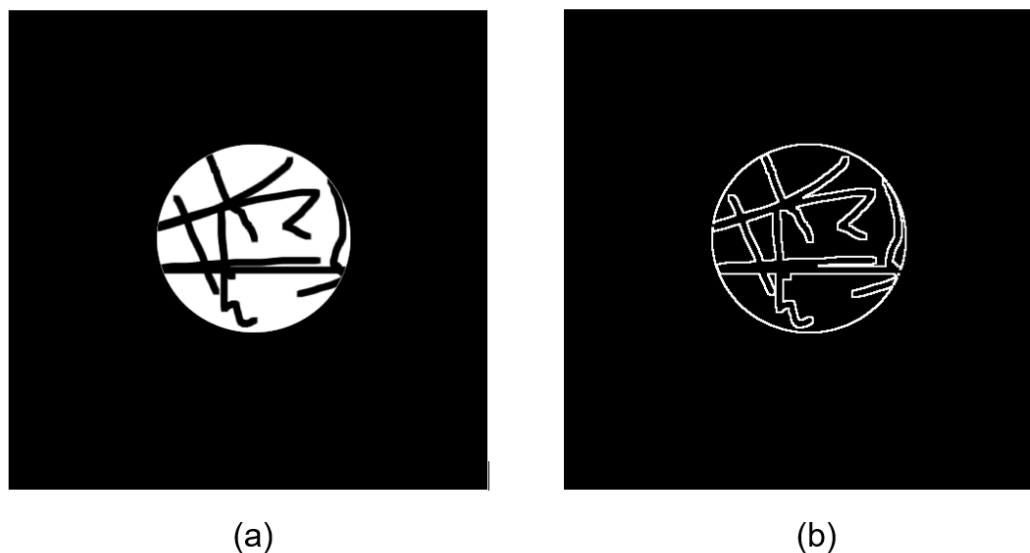
2.3.5 Gradiente Morfológico

Destaca-se por sua capacidade de evidenciar variações significativas na intensidade de borda de objetos em uma imagem, proporcionando uma abordagem valiosa para a análise de detalhes e contornos. Ele foi introduzido pela primeira vez por Serra (1982), que o definiu como “a diferença entre a imagem dilatada e a imagem erodida”. Essa operação resulta em uma imagem que destaca as variações de intensidade de borda (visto na Subseção 2.2.5). Um exemplo de aplicação prática desse operador, seria na extração de características relevantes em uma imagem (conceito explicado na Subseção 2.2.7), como texturas e padrões, sendo útil em reconhecimento de padrões e visão computacional.

A aplicação do operador gradiente morfológico é especialmente eficaz em imagens onde as bordas e contornos são características cruciais para análises específicas. Na

Figura 18 é apresentado um exemplo de como fica uma imagem após a aplicação do operador gradiente morfológico em uma imagem, onde a imagem original está representada à esquerda e a imagem resultante da abertura representada à direita.

Figura 18 – Ilustração da aplicação do operador gradiente morfológico.



Fonte: Autoria própria.

Sua expressão matemática é dada pela diferença entre a dilatação e a erosão do conjunto original. Matematicamente, o gradiente morfológico (G) de um conjunto A com um elemento estruturante B é definido como:

$$G(A, B) = \delta(A, B) - \varepsilon(A, B) \quad (14)$$

Nessa formulação, $\delta(A, B)$ representa a dilatação de A por B , e $\varepsilon(A, B)$ representa a erosão pelo mesmo elemento estruturante B . Após essas operações, teremos dois conjuntos resultantes, e a expressão $\delta(A, B) - \varepsilon(A, B)$ representa a subtração entre esses conjuntos.

2.3.6 Top Hat

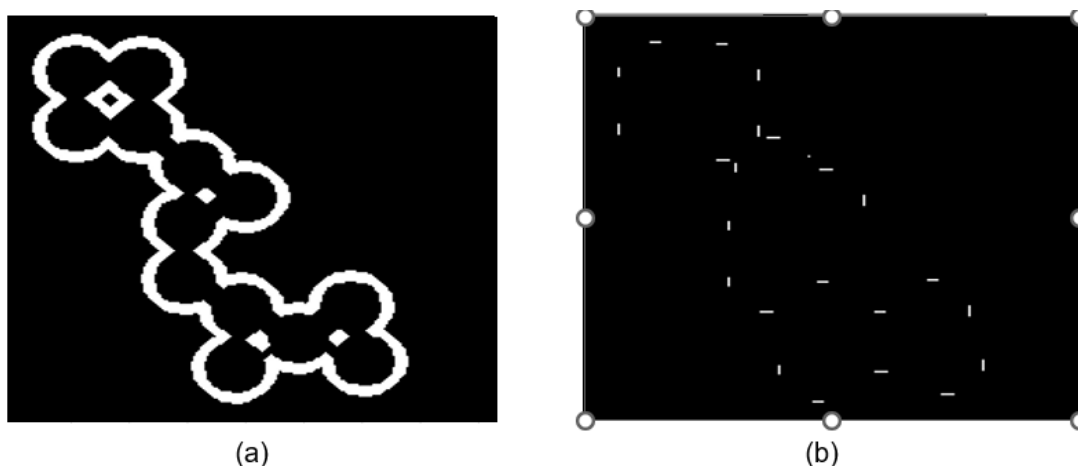
O operador *Top Hat*, também conhecido como *operador Cartola*, destaca-se por sua capacidade de revelar informações sutis e nuances em uma imagem, especialmente em áreas onde detalhes finos ou pequenas características são de interesse, e também realça regiões claras em uma imagem em relação ao seu entorno mais escuro. Essa operação é caracterizada pela diferença entre a imagem original e sua abertura, onde a abertura é obtida pela erosão seguida pela dilatação. Isso resulta em uma imagem que destaca as variações locais, realçando pequenos detalhes que podem ter sido perdidos ou suavizados em operações anteriores (GONZALEZ; WOODS, 2002).

O operador *Top Hat* é empregado em diversas situações práticas, incluindo:

- Ao realçar detalhes finos e pequenas estruturas: O operador *Top Hat* é essencial na detecção de características que podem ser cruciais em áreas como biomedicina, onde estruturas celulares delicadas precisam ser identificadas (LI; WANG, 2014b); .
- Inspeção de Superfícies e Texturas: Utilizado para a inspeção de superfícies, especialmente em imagens texturizadas, o operador *Top Hat* realça imperfeições e pequenas variações, contribuindo para análises detalhadas de texturas;
- Remoção de Componentes de Fundo: Pode ser aplicado para remover componentes de fundo indesejados, destacando apenas os detalhes de interesse em primeiro plano.

A aplicação do operador *Top Hat* é particularmente valiosa em imagens onde detalhes finos são de importância crítica, auxiliando na análise precisa e na interpretação de características sutis (LI; WANG, 2014b). Na Figura 19 é apresentado um exemplo de como ficaria uma imagem após a aplicação do operador morfológico *Top Hat* em uma imagem, onde a imagem original está representada à esquerda e a imagem resultante da abertura representada à direita.

Figura 19 – Ilustração da aplicação do operador morfológico *Top Hat*.



Fonte: Autoria própria.

Matematicamente, o *Top Hat* (T) de um conjunto A com um elemento estruturante B é definido como a diferença entre a imagem original e sua abertura:

$$T(A, B) = A - \text{Abertura}(A, B) \quad (15)$$

Nessa formulação, $\text{Abertura}(A, B)$ representa a abertura de A por B , que consiste em realizar uma erosão seguida de uma dilatação.

2.3.7 Black Hat

O operador Black Hat, também conhecido como operador *Chapéu Preto*, destaca-se por sua capacidade de revelar características sutis, especialmente em áreas onde sombras e regiões mais escuras são de interesse. Essa operação é obtida pela diferença entre a imagem

original e sua operação de fechamento, onde o fechamento é obtido pela dilatação seguida pela erosão. Isso resulta em uma imagem que destaca as variações locais em regiões mais escuras, evidenciando detalhes que podem ter sido perdidos ou suavizados em operações anteriores (GONZALEZ; WOODS, 2002).

O operador *Black Hat* é utilizado em diversas situações práticas, incluindo:

- Detecção de Anomalias em Imagens Escuras: Ao realçar variações sutis em regiões escuras, o operador *Black Hat* é valioso na detecção de anomalias; .
- Ênfase em Detalhes em Ambientes com Sombras: Utilizado para destacar detalhes em regiões sombreadas, o operador *Black Hat* é eficaz em ambientes onde a presença de sombras pode obscurecer características importantes (KULKARNI; SHEKHAR, 2013);
- Segmentação de Componentes Escuros: Pode ser aplicado para segmentar componentes mais escuros em primeiro plano, isolando-os do fundo e facilitando análises específicas.

A aplicação do operador *Black Hat* é particularmente valiosa em imagens onde sombras e regiões mais escuras contêm informações importantes, contribuindo para análises e interpretações mais detalhadas. Na Figura 20 é apresentado um exemplo de como ficaria uma imagem após a aplicação do operador morfológico *Black Hat* em uma imagem, onde a imagem original está representada à esquerda e a imagem resultante da abertura representada à direita.

Figura 20 – Ilustração da aplicação do operador morfológico *Black Hat*.



Fonte: Autoria própria.

Matematicamente, o *Black Hat* (B) de um conjunto A com um elemento estruturante B é definido como a diferença entre a abertura de A e a imagem original:

$$B(A, B) = F(A, B) - A \quad (16)$$

Nessa formulação, $F(A, B)$ representa o fechamento de A por B , que consiste em realizar uma erosão seguida de uma dilatação.

2.3.8 Inversão

O processo de inversão é fundamentalmente baseado na operação de subtração do valor máximo da intensidade de *pixel* em uma imagem pelo valor do *pixel* original, ou seja, inversão é uma operação morfológica que troca os níveis de intensidade de uma imagem, tornando as regiões mais escuras mais claras e as regiões mais claras mais escuras. (GONZALEZ; WOODS, 2002). A primeira etapa envolve a identificação do valor máximo de intensidade presente na imagem original e este valor servirá como referência para a inversão. Posteriormente cada *pixel* na imagem original é subtraído do valor máximo identificado. O resultado é uma nova imagem onde as intensidades são invertidas, criando uma representação visual que realça detalhes distintos.

O processo de inversão encontra aplicação em diversas áreas, oferecendo novas perspectivas visuais. Segue alguns exemplos :

- Realce de Detalhes em Regiões Escuras: Ao inverter as intensidades, detalhes em regiões mais escuras tornam-se mais evidentes, sendo útil em situações onde a visibilidade é comprometida. .
- Ênfase em Componentes de Interesse: Pode ser aplicado para destacar componentes específicos em uma imagem, ressaltando áreas de interesse e simplificando a análise visual (LI; WANG, 2014a).
- Correção de Exposição: Utilizado para corrigir problemas de exposição, o processo de inversão pode equilibrar a distribuição de intensidades na imagem, melhorando a representação visual.

A aplicação do processo de inversão pode variar conforme a natureza da imagem e o contexto da análise. É essencial considerar a finalidade específica da inversão em cada aplicação.

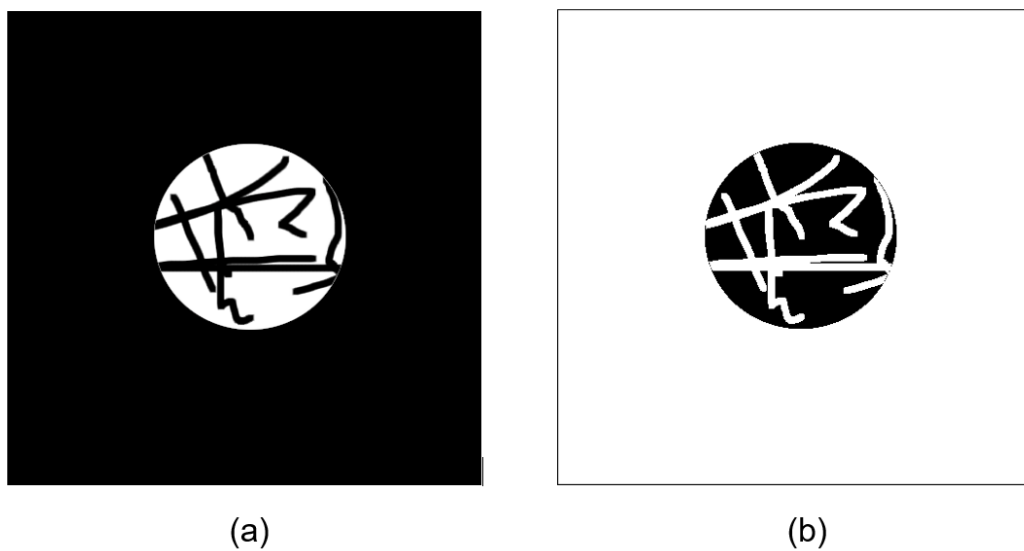
Na Figura 21 é apresentado um exemplo de como ficaria uma imagem após a aplicação da inversão em uma imagem, onde a imagem original está representada à esquerda e a imagem resultante da abertura representada à direita.

Matematicamente, a inversão (I) de um conjunto A pode ser definida como o complemento do conjunto original:

$$I(A) = A^c \quad (17)$$

Nessa formulação, A^c representa o complemento em relação ao conjunto de valores que A pode assumir, ou seja, onde era 0 passa a ser 1, e vice versa. Cabe salientar que as operações de complemento são definidas em relação ao conjunto definido por A .

Figura 21 – Ilustração da aplicação da inversão.



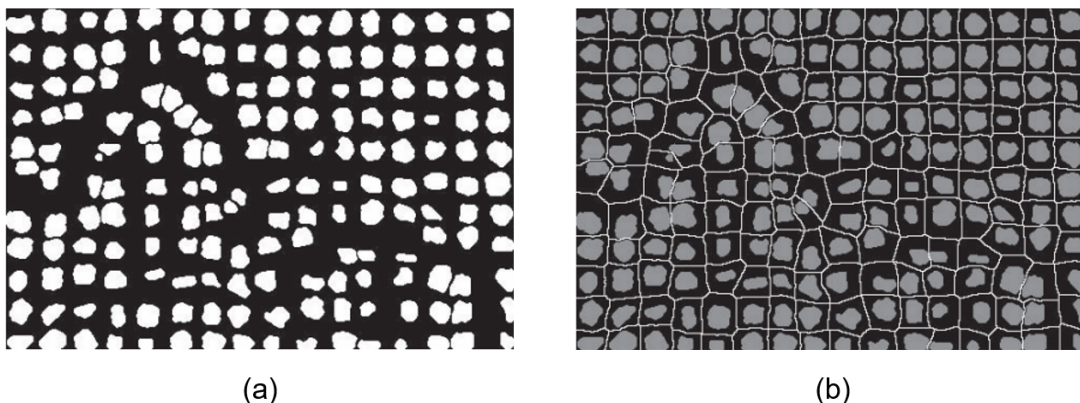
Fonte: Autoria própria.

3 REVISÃO BIBLIOGRÁFICA

Diversos pesquisadores têm se dedicado no estudo dos operadores morfológicos. No trabalho de Statella e Silva (2008) encontra-se um exemplo do uso de operadores morfológicos para remoção de ruído, onde o autor usou uma combinação de operadores morfológicos (abertura e fechamento) para remover ruído de imagens de satélites, o que melhorou a qualidade das imagens.

Já Marquez-Neila, Baumela e Alvarez (2013) apresentam uma nova abordagem para a evolução de contornos em imagens, utilizando um operador morfológico baseado em curvatura para a evolução de curvas e superfícies. Os autores argumentam que essa abordagem é mais estável e rápida do que as soluções padrão baseadas em equações diferenciais parciais e *level-sets*. Eles demonstram a eficácia do operador morfológico proposto em várias aplicações de visão computacional, incluindo detecção de bordas e rastreamento de contornos, ou seja, apresenta uma nova técnica que pode ser facilmente implementada e visualizada em galerias de imagens. O operador morfológico proposto pode ser aplicado a imagens de diferentes dimensões, o que amplia as possibilidades de aplicação em diversas áreas da visão computacional. A Figura 22 mostra como as curvas evoluem usando operadores morfológicos. Na imagem (a), você vê várias curvas em movimento. Na imagem (b), após aplicar um processo chamado afinamento homotópico, cada curva é mantida dentro de sua própria área delimitada, evitando que as curvas se sobreponham ou saiam de seus limites. Isso ilustra como os operadores morfológicos ajudam a controlar a evolução das curvas de forma precisa e organizada.

Figura 22 – Detecção de curvas com a aplicação de operadores morfológicos. (a) Imagem inicial e (b) resultado após a aplicação.

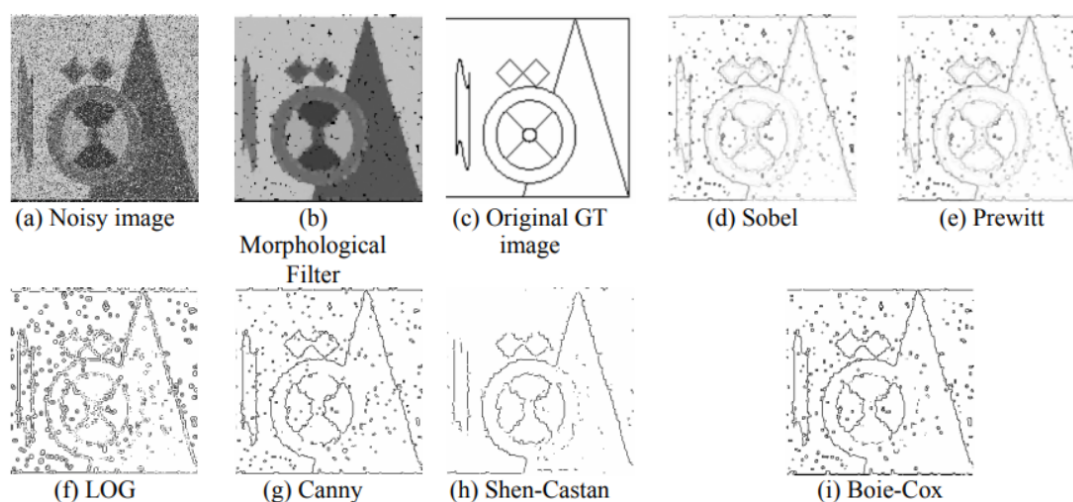


Fonte: Marquez-Neila, Baumela e Alvarez (2013).

Roushdy (2006) apresenta uma análise comparativa de diferentes algoritmos de detecção de bordas em imagens em escala de cinza com ruído, utilizando filtros morfológicos. O artigo discute as vantagens e desvantagens de cada algoritmo e apresenta uma avaliação

objetiva e subjetiva do desempenho dos operadores de borda. Além disso, o trabalho destaca a importância do filtro morfológico como um processo inicial na detecção de bordas em imagens com ruído, o que permite entender como diferentes algoritmos de detecção de bordas podem ser aplicados em imagens com ruído e como o filtro morfológico pode ser usado para melhorar a qualidade da imagem antes da detecção de bordas. A Figura 23 traz uma comparação entre diferentes operadores de detecção de bordas (*Sobel*, *Prewitt*, *LOG*, *Canny*, *Shen-Castan*, *Boie-Cox*) aplicados a uma imagem com ruído de sal e pimenta. O ruído de sal e pimenta é um tipo de distorção comum em imagens que se assemelha a pontos brancos e pretos aleatórios, afetando a qualidade visual da imagem.

Figura 23 – Comparação entre os operadores de detecção de bordas aplicados a uma imagem com ruídos.



Fonte: Roushdy (2006).

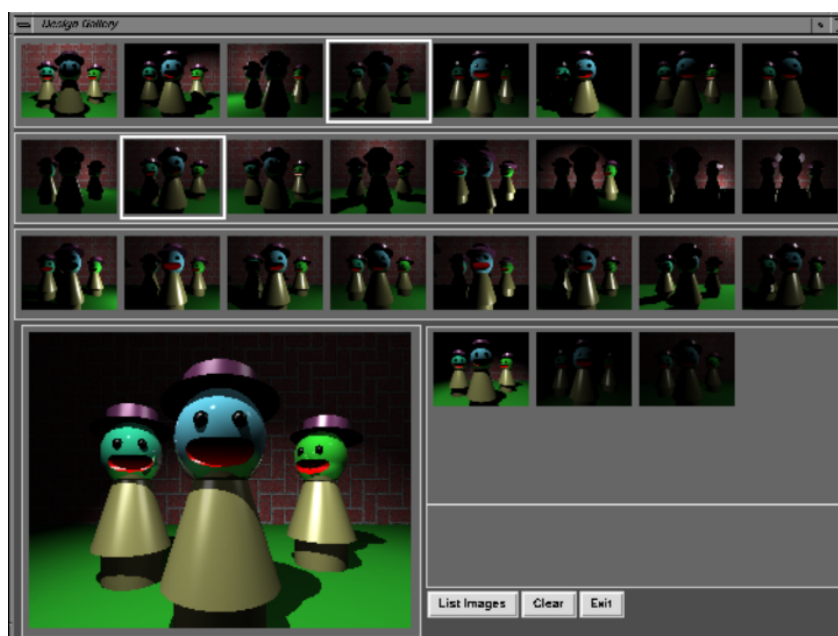
O termo *ground truth* (*GT*) se refere à imagem original sem nenhum tipo de distorção ou ruído, sendo considerada a referência verdadeira para comparação. Neste contexto, a imagem *ground truth* é usada como base para avaliar a eficácia dos operadores de detecção de bordas após a aplicação de um filtro morfológico, que tem o objetivo de melhorar a qualidade da imagem removendo o ruído de sal e pimenta antes da detecção de bordas. Assim, a Figura 23 mostra como os diferentes operadores de detecção de bordas se comportam ao identificar as bordas em uma imagem com ruído de sal e pimenta após a aplicação do filtro morfológico, comparando os resultados com a imagem original sem ruído (*ground truth*).

No contexto da manipulação de imagens usando operadores morfológicos, as *galerias de design* podem ser utilizadas para visualizar os resultados da aplicação de diferentes parâmetros de um operador morfológico. Por exemplo, uma galeria de design para o operador de erosão poderia permitir ao usuário visualizar os resultados da aplicação do operador para diferentes tamanhos do núcleo erosionista.

O trabalho de Marks et al. (2023), propõe uma abordagem generalista para o ajuste

de parâmetros em computação gráfica e animação. Essa abordagem, conhecida como *design galleries*, apresenta ao usuário uma ampla seleção de gráficos ou animações perceptualmente diferentes que podem ser produzidas por meio da variação de um determinado vetor de parâmetros de entrada. As *design galleries* são compostas por uma coleção de exemplos de resultados de renderização, cada um representando um conjunto diferente de valores de parâmetros. Os exemplos são organizados de forma a facilitar a exploração do espaço de parâmetros e a identificação de resultados de interesse, ilustrado na Figura 24.

Figura 24 – Exemplo de um esquema de galeria utilizado para seleção do jogo de luz em uma imagem.



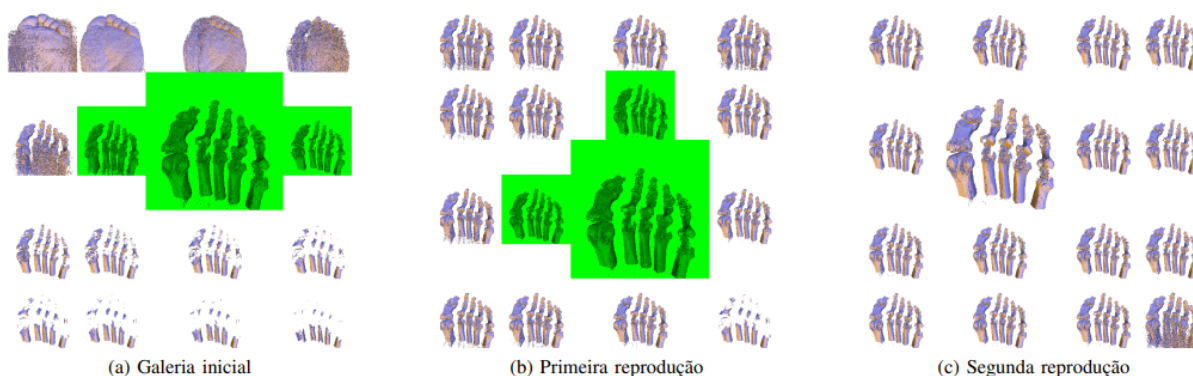
Fonte: Marks et al. (2023).

A abordagem proposta por Marks et al. (2023) tem várias vantagens em relação a outras abordagens de ajuste de parâmetros. Em primeiro lugar, ela é generalista, podendo ser aplicada a uma ampla variedade de problemas de computação gráfica e animação. Em segundo lugar, ela é eficiente, pois permite ao usuário explorar o espaço de parâmetros de forma rápida e fácil. Em terceiro lugar, ela é interativa, permitindo ao usuário interagir com os exemplos de forma a obter resultados mais refinados.

Já no trabalho de Cunha, Lima e Vieira (2011), as Galerias de Design são utilizadas para auxiliar o usuário na seleção do parâmetro ideal para a extração de isosuperfícies de qualidade em dados volumétricos. Através da interface gráfica baseada em Galerias de Design, o usuário explora o espaço de isovalores através de representações visuais das isosuperfícies correspondentes, assistindo o usuário na seleção do parâmetro ótimo. Um exemplo dessa interação pode ser observado na Figura 25. Além disso, o trabalho propõe o uso de um algoritmo genético para gerar novas galerias a partir das seleções do usuário em galerias anteriores, refinando a seleção inicial e permitindo ao usuário obter resultados

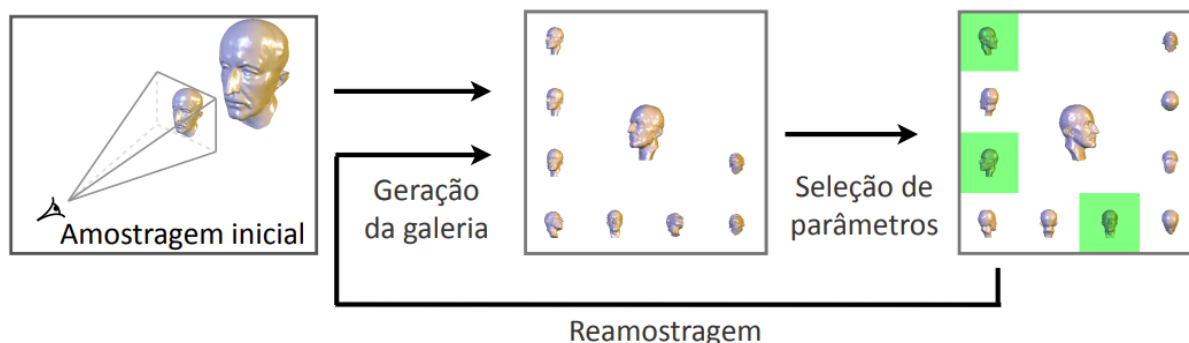
mais precisos. A Figura 26 ilustra o esquema do fluxo da aplicação na prática. Portanto, as Galerias de Design são úteis nesse trabalho, pois permitem ao usuário explorar de forma eficiente e interativa o espaço de parâmetros para a extração de isosuperfícies de qualidade em dados volumétricos.

Figura 25 – Exemplo de galerias geradas para seleção de isovalor em um dado volumétrico de um pé humano. Em verde os isovalores selecionados pelo usuário nas galerias.



Fonte: Cunha, Lima e Vieira (2011).

Figura 26 – Ilustração do fluxo utilizado na aplicação.



Fonte: Cunha, Lima e Vieira (2011).

Além de sua aplicação no contexto acadêmico, o estudo destaca a presença das Galerias de Design em softwares amplamente utilizados na indústria, como o Adobe After Effects CS4 e o Adobe Photoshop CS4, como pôde ser visto no Capítulo 1, na Figura 2). Nestas aplicações, as Galerias de Design são empregadas para ajustar parâmetros de animação e recoloração de imagens, evidenciando sua relevância e aplicabilidade prática em diferentes contextos.

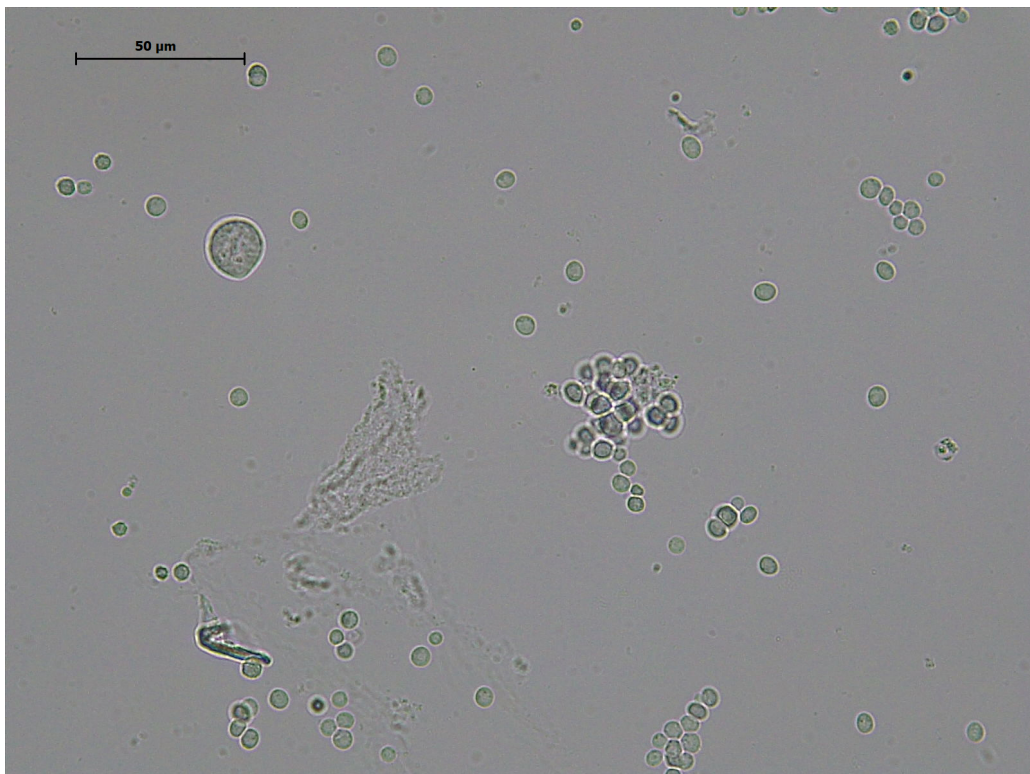
4 MATERIAIS E MÉTODOS

Nesta seção, abordaremos as bibliotecas fundamentais adotadas na implementação do trabalho, a seleção da linguagem de programação, uma análise detalhada da interface desenvolvida para a aplicação e a aquisição de algumas das imagens utilizadas nos experimentos deste trabalho.

4.1 Aquisição das Imagens

As etapas aqui descritas foram realizadas em colaboração com Dra. Natália Karla Bellini e Dra. Maria Reyes-Batlle, laboratório de amebas de vida livre, Universidade de La Laguna, Espanha. As Figuras 27, 28 e 29 apresentam alguns dos resultados obtidos a partir das imagens coletadas.

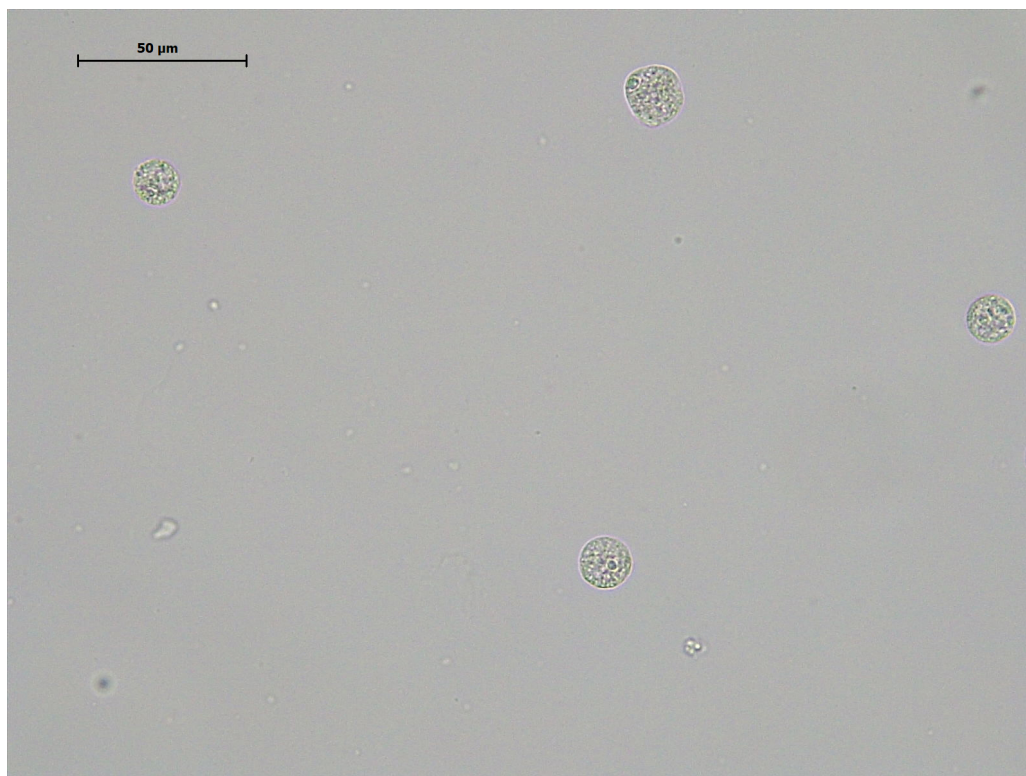
Figura 27 – Imagem microscópica.



Fonte: Autoria própria.

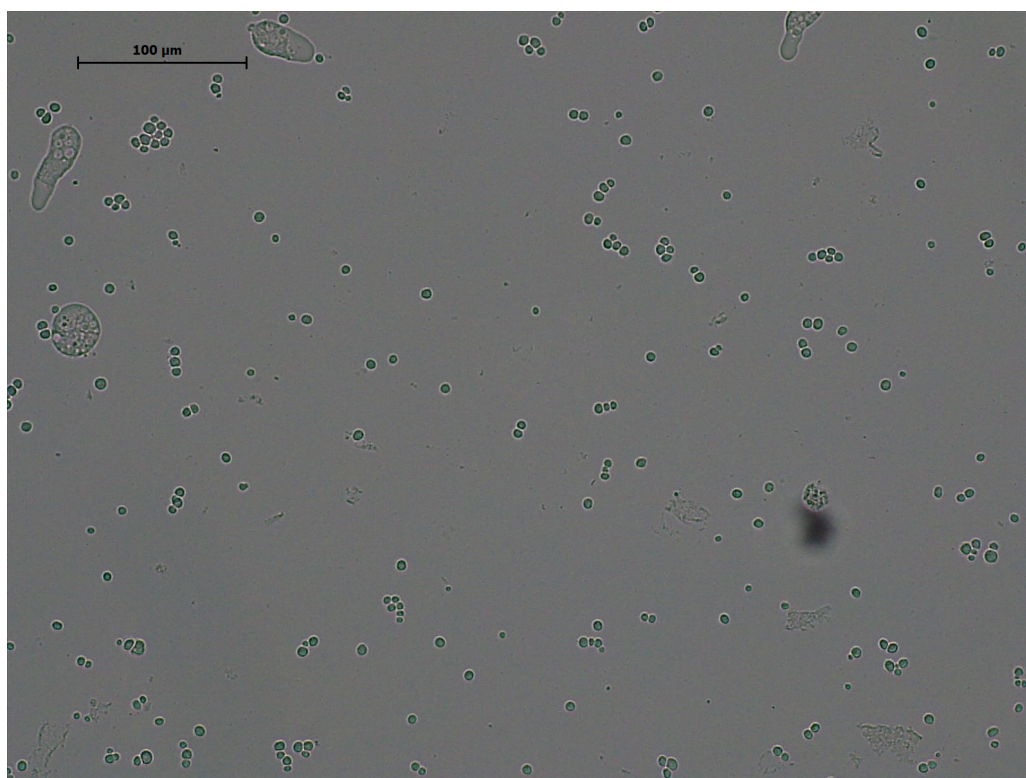
As culturas de amebas de vida livre dos gêneros *Naegleria* e *Vermamoeba* foram mantidas em placas de ágar não nutritivo, temperatura ambiente, conforme protocolos de cultura celular (BELLINI et al., 2020). Por sua vez as culturas de *Acanthamoeba* foram mantidas em meio líquido (ALFIERI et al., 2000). Os trofozoítos (1×10^6 células) foram centrifugados diretamente de cultura líquida, ou ressuspensos da superfície do

Figura 28 – Seleção da imagem.



Fonte: Autoria própria.

Figura 29 – Seleção da imagem.



Fonte: Autoria própria.

ágar com solução salina. Na sequência foram induzidos à diferenciação para estágio de cisto utilizando meios líquidos específicos para cada gênero de ameba (SOHN et al., 2017). Lâminas de vidro foram preparadas com as culturas a fresco. Aproximadamente 1x10⁴ cistos foram depositados na lâmina, sobrepostos com lamínula, e diretamente visualizados em microscópio Leica DM1000 LED, magnificação 200x, 400x e 1000x. As imagens foram adquiridas em câmera Leica ICC50W, acoplada ao microscópio, e processadas em software LAS EZ versão 3.4, cujos parâmetros de aquisição da imagem (ex.: intensidade, exposição e saturação da luz) foram ajustados manualmente e utilizados como padrão para as demais aquisições. Os arquivos foram exportados em formato TIFF e utilizados para análise morfológica computacional.

4.2 Requisitos do sistema

Nesta seção são discutidos os requisitos elicitados para a elaboração do sistema, bem como é apresentado um diagrama de casos de uso da aplicação proposta.

4.2.1 Requisitos funcionais

- **RF01:** Efetuar carregamento da imagem em mais de um formato comum (ex.: jpg e png);
- **RF02:** Permitir aplicação de operadores morfológicos mais comuns;
- **RF03:** Oferecer aos usuários a escolha entre diferentes elementos estruturantes;
- **RF04:** Desenvolver um sistema de registro que mantenha histórico dos operadores morfológicos aplicados;
- **RF05:** Permitir aos usuários acessar e revisar o histórico a qualquer momento;
- **RF06:** Possibilitar reversão de operações efetuadas pelo usuário;
- **RF07:** Exportar a imagem resultante do processamento na aplicação em formato comum (ex.: jpg e png);
- **RF08:** Exportar em formato textual o histórico de operações efetuadas pelo usuário;
- **RF09:** Exportar em formato textual o elemento estruturante utilizado pelo usuário.

4.2.2 Requisitos não funcionais

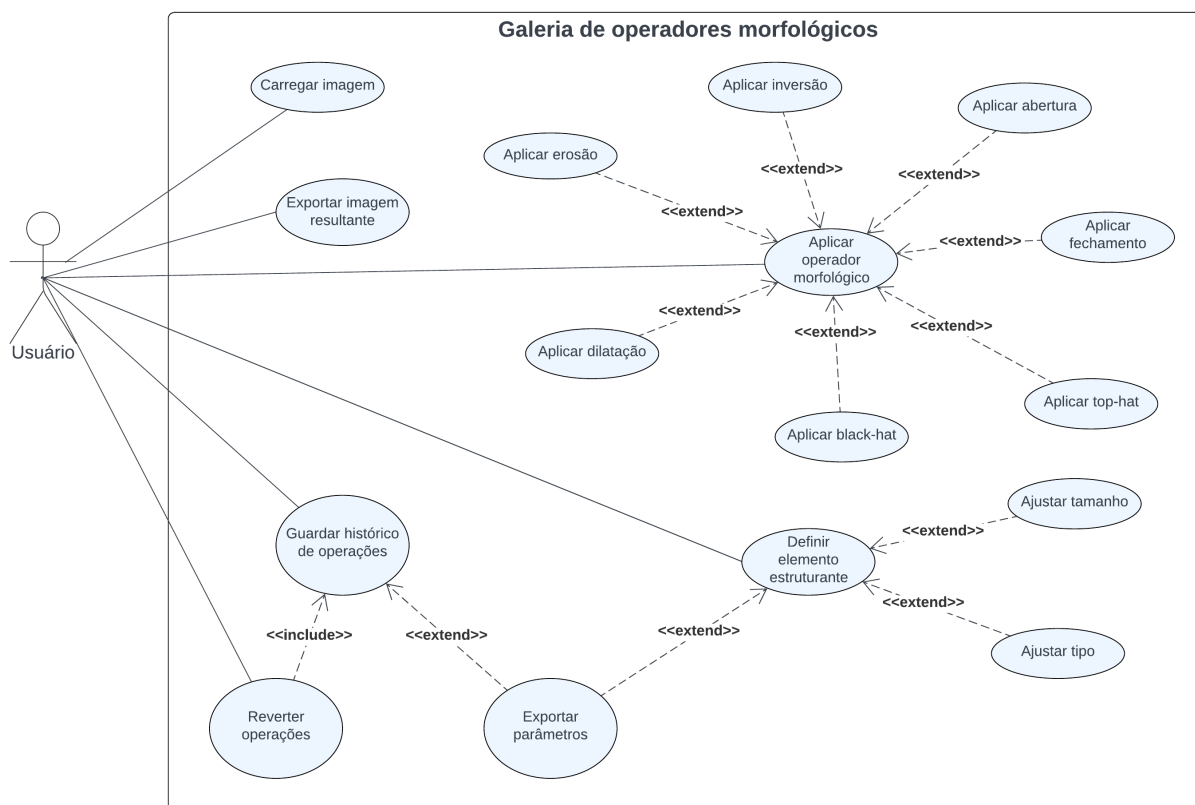
- **RNF01:** Interface de usuário intuitiva e amigável;
- **RNF02:** Aplicação desenvolvida deve rodar em ambiente desktop;
- **RNF03:** Aplicação desenvolvida deve ser multiplataforma.

4.2.3 Diagrama de Casos de Uso

A representação visual dos requisitos e interações entre usuários e o sistema é interessante para uma compreensão abrangente do escopo da aplicação. O diagrama de

casos de uso apresentado na Figura 30 destaca as principais funcionalidades da aplicação, delineando como os usuários interagem com o sistema e quais são as operações disponíveis.

Figura 30 – Diagrama de casos de uso.



Fonte: Autoria própria.

4.3 Tecnologias utilizadas

Python

Em relação a linguagem de programação escolhida para o desenvolvimento desse trabalho, esta sendo utilizada a linguagem Python (versão 3.7.3), que é conhecida por sua sintaxe clara e legível, fator que facilita a compreensão do código, especialmente em projetos complexos como os envolvidos no processamento digital de imagens. A simplicidade da linguagem reduz a probabilidade de erros, promovendo um ambiente de desenvolvimento mais robusto. A comunidade Python é vasta e ativa, proporcionando acesso a uma ampla gama de bibliotecas e frameworks especializados em processamento de imagens. Ferramentas como OpenCV, Pillow e scikit-image oferecem funcionalidades robustas para manipulação e análise de imagens, enquanto a comunidade fornece suporte ativo e recursos educacionais. Python é uma linguagem de programação versátil, permitindo a integração fácil com outras tecnologias e a adaptação a diferentes requisitos de projeto.

Em resumo, a escolha do Python 3.7.3 como linguagem de programação para o desenvolvimento dessa aplicação baseada em processamento de imagens, não apenas oferece uma sintaxe amigável e clara, mas também aproveita uma comunidade robusta, uma ampla gama de bibliotecas especializadas e a flexibilidade necessária para enfrentar os desafios complexos que possam surgir.

Foi utilizado também a biblioteca NumPy na versão 1.21.6.

OpenCV

No âmbito do processamento de imagens, a biblioteca OpenCV (Open Source Computer Vision) destaca-se como uma ferramenta essencial, proporcionando um conjunto abrangente de funcionalidades para manipulação, análise e processamento de imagens. Dentro desse contexto, a utilização de operadores morfológicos, operações fundamentais em visão computacional, se beneficia significativamente da implementação eficiente presente no OpenCV. Ela simplifica o processo de carregamento e visualização de imagens, oferecendo funcionalidades que tornam essas operações diretas e acessíveis, contém uma implementação eficiente dos principais operadores morfológicos, permitindo a aplicação direta em imagens com poucas linhas de código. Além da implementação direta dos operadores, o OpenCV oferece a flexibilidade de definir elementos estruturantes personalizados. Isso possibilita a adaptação dos operadores morfológicos de acordo com as características específicas de cada imagem e aplicação. A documentação detalhada e a comunidade ativa também tornam o processo de aprendizado e implementação mais acessível, tornando a OpenCV uma ferramenta indispensável no desenvolvimento desse trabalho, que tem como foco o processamento digital de imagens. Está sendo utilizada a versão 4.8.1 da biblioteca.

Qt Designer

A ferramenta QT Designer (versão 5.11.1) desempenha um papel essencial na criação e design da interface gráfica da aplicação, que está sendo desenvolvida com Python no *backend*. Essa ferramenta, parte integrante do framework Qt, oferece uma abordagem visual e intuitiva para a construção de interfaces de usuário dinâmicas e interativas. O QT Designer simplifica significativamente o processo de criação de interfaces gráficas, permitindo a criação dos *layouts* de forma visual, arrastando e soltando elementos, sem a necessidade de codificação manual, acelerando bastante o processo de desenvolvimento.

O uso do QT Designer 5.11.1 na construção da interface da aplicação representa uma escolha estratégica para otimizar o desenvolvimento, permitindo uma abordagem visual e eficiente. Sua integração bastante funcional com o Python oferece um fluxo de trabalho harmônico entre o design da interface e a implementação da lógica de *backend*.

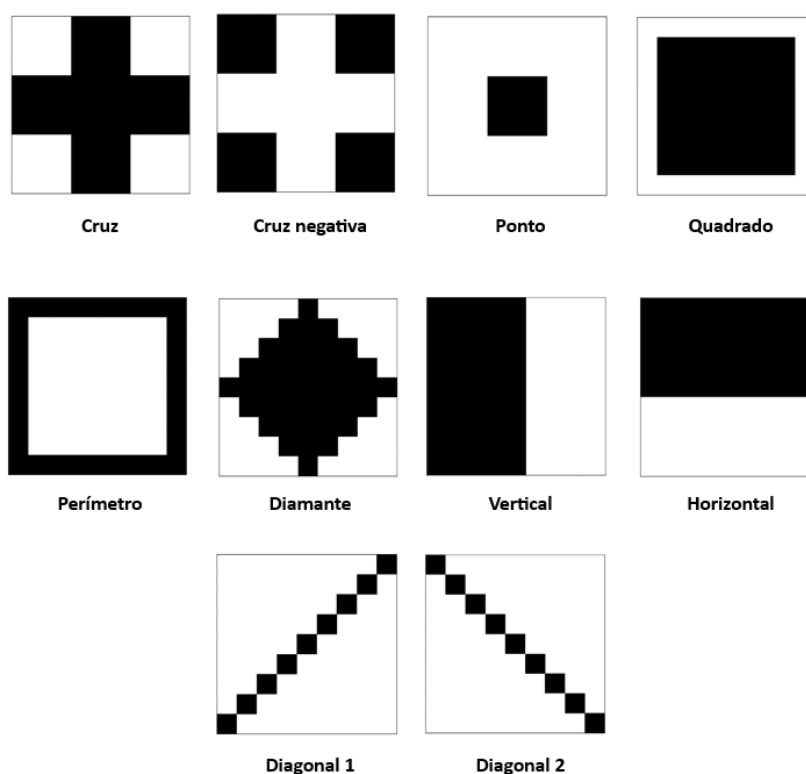
4.4 Implementação dos Operadores Morfológicos

A seguir, serão apresentadas as principais funções e implementações utilizadas para aplicar os operadores morfológicos nas imagens, destacando seus parâmetros e funcionamento.

4.4.1 Elemento estruturante

Na implementação dos operadores morfológicos, a definição do elemento estruturante, muitas vezes representado por um *kernel*, desempenha um papel crucial. A aplicação disponibiliza 10 opções de configuração para o *kernel*, que será utilizado na aplicação dos operadores morfológicos. Na Figura 31, é possível visualizar graficamente cada uma dessas opções.

Figura 31 – Visualização gráfica de todos os *kernels* disponíveis, acompanhados de seus respectivos nomes.



Fonte: Autoria própria.

Considere o exemplo de atribuição de valores a um *kernel*, utilizando a biblioteca NumPy no contexto da OpenCV mostrado no Algoritmo 1.

Neste código, o elemento_estruturante representa um *kernel* 3x3 com um padrão específico. Cada valor na matriz indica a contribuição do *pixel* correspondente no processo de operação morfológica. A manipulação desses valores pode resultar em diferentes efeitos ao aplicar o operador morfológico sobre a imagem. Assumir um *kernel* como esse ao

Algoritmo 1: Elemento estruturante.

```
1 elemento_estruturante = np.array([
2     [0, 1, 0],
3     [1, 1, 1],
4     [0, 1, 0]
5 ], dtype=np.uint8)
```

mencionar o elemento estruturante ao longo do trabalho proporcionará uma compreensão prática das operações discutidas.

Função `morphologyEx`

A biblioteca OpenCV oferece a função `cv2.morphologyEx()` para a aplicação de diversas operações morfológicas em imagens. Esta função é extremamente versátil, permitindo moldar a imagem de diferentes maneiras com base em um elemento estruturante. No código abaixo, é possível ver o exemplo da estrutura genérica para o uso dessa função.

```
1 cv2.morphologyEx(imagem_cv, operacao, elemento_estruturante)
```

- **imagem_cv:** A imagem de entrada na qual a operação morfológica será aplicada.
- **operacao:** O tipo de operação morfológica desejada. Abaixo estão os exemplos de operações utilizadas nesse trabalho:
 - `cv2.MORPH_TOPHAT`: Operador Top Hat;
 - `cv2.MORPH_GRADIENT`: Operador de Gradiente;
 - `cv2.MORPH_CLOSE`: Operação de Fechamento;
 - `cv2.MORPH_BLACKHAT`: Operador Black Hat;
 - `cv2.MORPH_OPEN`: Operação de Abertura.

É possível ver no Algoritmo 2 um exemplo de como ficaria a chamada dessa função em código.

Algoritmo 2: Aplicação de operador morfológico.

```
1 imagem_transformada = cv2.morphologyEx(imagem_original,
   ↪ cv2.MORPH_GRADIENT, elemento_estruturante)
```

Neste exemplo, o operador morfológico de gradiente foi aplicado a “`imagem_original`”, usando um determinado “`elemento_estruturante`”, resultando em uma nova imagem que foi atribuída a variável “`imagem_transformada`”.

Função dilate

A função `cv2.dilate` da biblioteca OpenCV é empregada para realizar a operação morfológica de dilatação em imagens. No exemplo mostrado no Algoritmo 3, apresentamos uma estrutura genérica para a utilização da função `cv2.dilate`.

Algoritmo 3: Aplicação do operador morfológico de dilatação.

```
1 cv2.dilate(imagem_original, elemento_estruturante,  
  ↪ iterations=1)
```

-
- **imagem_original:** A imagem de entrada na qual a operação de dilatação será aplicada.
 - **iterations:** controla quantas vezes o operador morfológico será aplicado consecutivamente à imagem. Quanto maior o valor, mais intensa será a dilatação.

Função erode

A função `cv2.erode` na biblioteca OpenCV é usada para realizar a operação de erosão em uma imagem. A estrutura básica para utilizar a função é exemplificada no Algoritmo 4.

Algoritmo 4: Aplicação do operador morfológico de erosão.

```
1 cv2.erode(imagem_cv, elemento_estruturante, iterations=1)
```

-
- **imagem_original:** A imagem de entrada na qual a operação de erosão será aplicada.
 - **iterations:** controla quantas vezes o operador morfológico será aplicado consecutivamente à imagem. Quanto maior o valor, mais intensa será a dilatação..

Função bitwise_not

A função `cv2.bitwise_not` é utilizada para realizar a inversão de *pixels* em uma imagem. A estrutura básica para utilizar a função é exemplificada no Algoritmo 5.

Algoritmo 5: Operação de inversão.

```
1 cv2.bitwise_not(imagem_cv)
```

-
- **imagem_cv:** A imagem de entrada na qual a operação de inversão será aplicada.

Essa função basicamente troca os *pixels* pretos pelos brancos e os brancos pelos pretos na imagem, criando uma versão invertida da original.

4.4.2 Implementação da técnica de limiarização

A função `cv2.threshold` é comumente usada para aplicar a técnica de limiarização em imagens. Essa técnica é fundamental para segmentar objetos ou regiões de interesse com base nos valores de intensidade dos *pixels*. No código mostrado no Algoritmo 6, é possível ver um exemplo da estrutura que foi utilizada nesse trabalho, para o uso dessa função.

Algoritmo 6: Limiarização pelo método de Otsu (OTSU, 1979).

```
1 cv2.threshold(imagem_cv, 0, 255, cv2.THRESH_BINARY +  
  ↪ cv2.THRESH_OTSU)
```

- **imagem_cv:** A imagem de entrada na qual a limiarização será aplicada.
- **0:** Representa o valor do limiar (ponto de corte). Neste contexto, o valor é definido como zero, pois o método de Otsu será responsável por determinar essa informação;
- **255:** Valor atribuído aos *pixels* que superam o limiar;
- **cv2.THRESH_BINARY + cv2.THRESH_OTSU:** Método de limiarização escolhido. No caso, utiliza-se uma combinação de limiarização binária e o método de Otsu (OTSU, 1979) para determinar automaticamente um valor de limiar ideal.

4.5 Apresentação da Interface

A interface da aplicação desenvolvida neste trabalho oferece diversas funcionalidades, cada uma destacada através dos 6 botões apresentados na Figura 32:

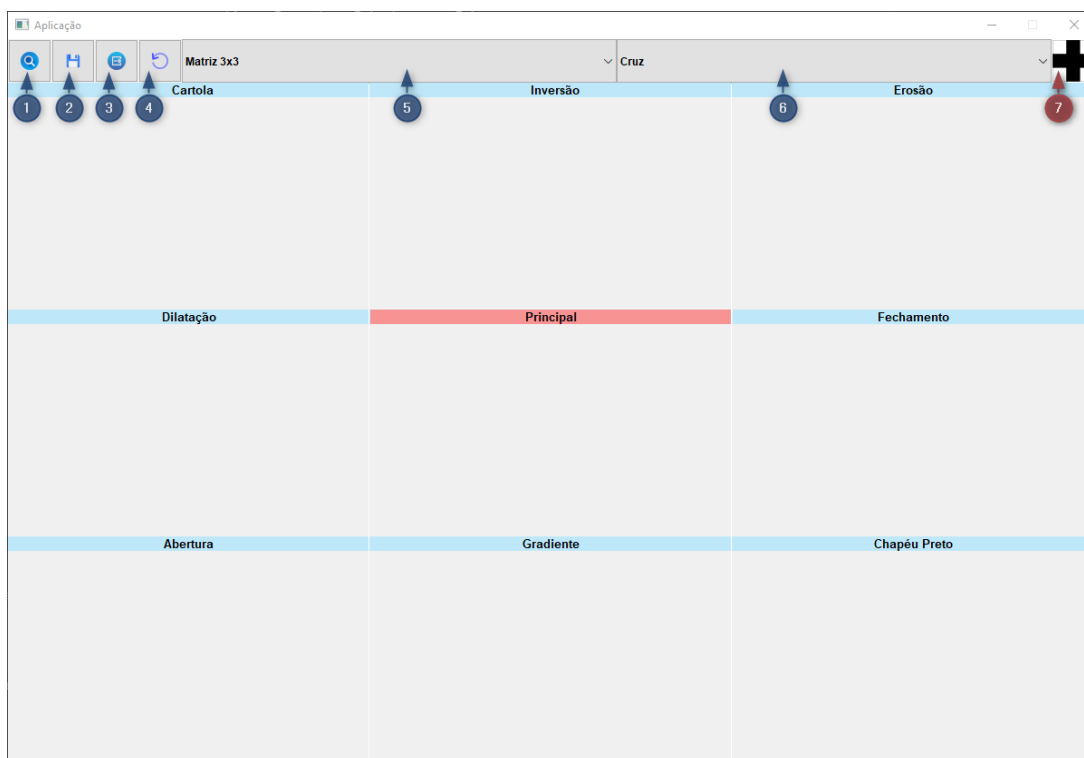
Carregar imagem

Ao clicar no Botão 1, uma janela de busca padrão será aberta, permitindo que o usuário navegue até o diretório do arquivo desejado. A imagem selecionada será carregada automaticamente na interface, com todos os operadores morfológicos aplicados, ilustrado na Figura 33, sendo a imagem **Principal** (título destacado na cor vermelha) a imagem carregada originalmente ou a mais recente escolhida pelo usuário. É importante observar que a interface efetua a binarização automaticamente, caso a imagem de entrada não esteja inicialmente nesse formato, facilitando assim o processo de aplicação dos operadores.

Salvar imagem principal

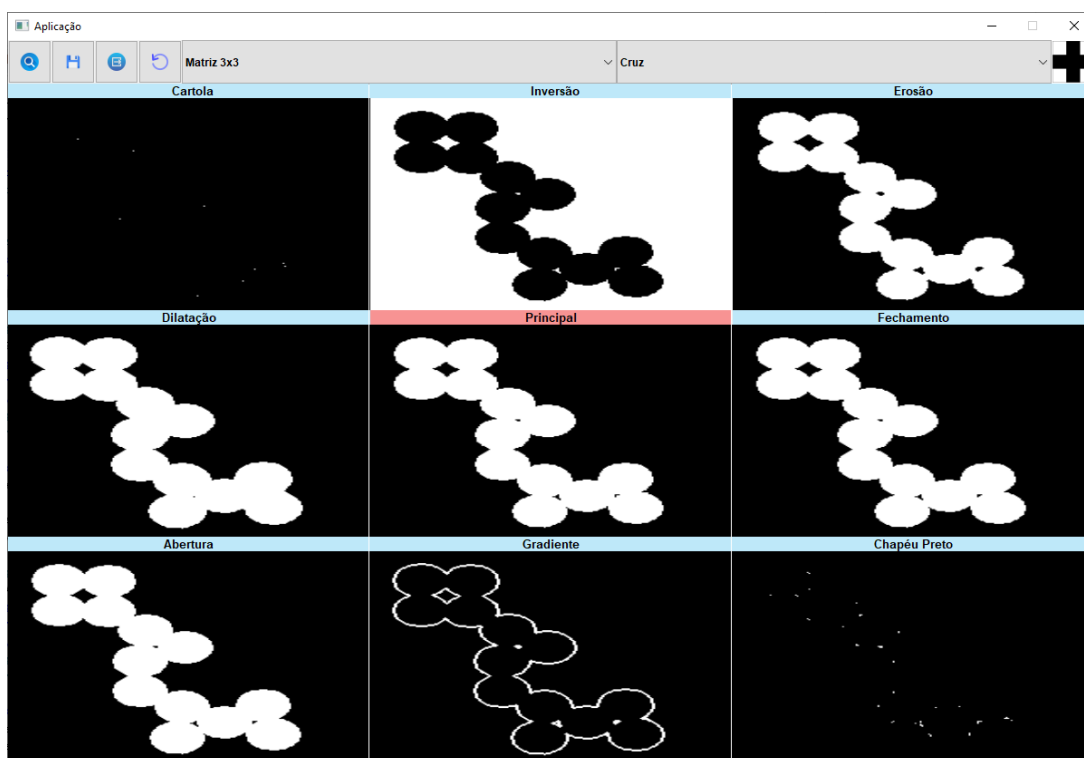
O Botão 2 possibilita salvar a imagem “Principal” a qualquer momento da interação do usuário. Ao ser acionado, uma janela de busca padrão permitirá ao usuário determinar o diretório desejado para salvar a imagem. Automaticamente, será gerado um arquivo de

Figura 32 – Interface da aplicação.



Fonte: Autoria própria.

Figura 33 – Visualização instantânea na interface após a seleção da imagem.



Fonte: Autoria própria.

texto (.txt) com o mesmo nome, adicionando “_log” como prefixo, representando o histórico de operações realizadas.

Exportar log

O Botão 3 desempenha um papel fundamental ao permitir que o usuário exporte de forma personalizada o registro de todas as operações aplicadas à imagem até o momento. O log inclui detalhes essenciais, tais como a sequência de operadores morfológicos, o tamanho do *kernel* e sua estrutura. Ao exportar, o usuário tem a flexibilidade de escolher um nome para o arquivo .txt resultante, além de determinar o diretório de destino.

A estrutura do log gerado é idêntica àquela exportada automaticamente ao salvar uma imagem na aplicação. Contudo, esta funcionalidade proporciona ao usuário a liberdade de realizar análises específicas, sem a necessidade de salvar uma imagem. Ao clicar no botão log (Botão 3), será aberta uma janela que permite ao usuário escolher o diretório desejado e definir o nome do arquivo. Após o arquivo ser salvo ele vai conter a seguinte estrutura:

```
1 Nome do arquivo de imagem original: V1
2
3 Opção selecionada no kernel: V2
4
5 Matriz:
6
7 V3
8
9 Sequência de operadores aplicados na imagem:
10
11 V4
```

Onde :

- **V1:** Apresenta o nome do arquivo de imagem carregado na interface, pronto para o processamento.
- **V2:** Mostra a combinação escolhida nos combobox da interface, determinando a estrutura e tamanho da matriz.
- **V3:** Exibe a matriz real utilizada para a aplicação dos operadores.
- **V4:** Detalha a sequência de operações morfológicas aplicadas à imagem inicial, resultando na versão final obtida.

Reverter operação

O botão 4 possibilita reverter o último filtro aplicado pelo usuário. Se o resultado não for satisfatório, a imagem retornará ao estado anterior, permitindo que o usuário escolha um novo filtro. É importante destacar que essa funcionalidade reverte apenas um passo anterior.

Seleção do elemento estruturante

Essa etapa é composta por dois elementos. O primeiro (Botão 5) é um combobox de simples seleção que permite ao usuário escolher o tamanho do *kernel* a ser utilizado para aplicar os operadores. As opções disponíveis são: Matriz 3x3, Matriz 5x5 e Matriz 7x7.

O segundo (Botão 6) é um combobox de simples seleção que permite ao usuário escolher variações de preenchimentos da matriz (*kernel*) que vai ser utilizada para aplicar os operadores morfológicos. Este botão está relacionado com o botão 5, já que um determina o tamanho e o outro a forma da matriz (*kernel*). As opções incluem Cruz, Cruz Negativa, Ponto, Quadrado, Perímetro, Diamante, Vertical, Horizontal, Diagonal 1 e Diagonal 2.

Adicionalmente, ao escolher uma nova opção nesse combobox (Botão 6), a interface apresentará uma representação visual de como seria essa matriz. A imagem será exibida nas cores preto e branco, sendo a cor branca representada pelo valor 0 e a cor preta pelo valor 1. Essa visualização proporciona ao usuário uma prévia imediata da combinação escolhida, e está localizada logo a direita desse combo, na Figura 32, indicada pela seta de número 7.

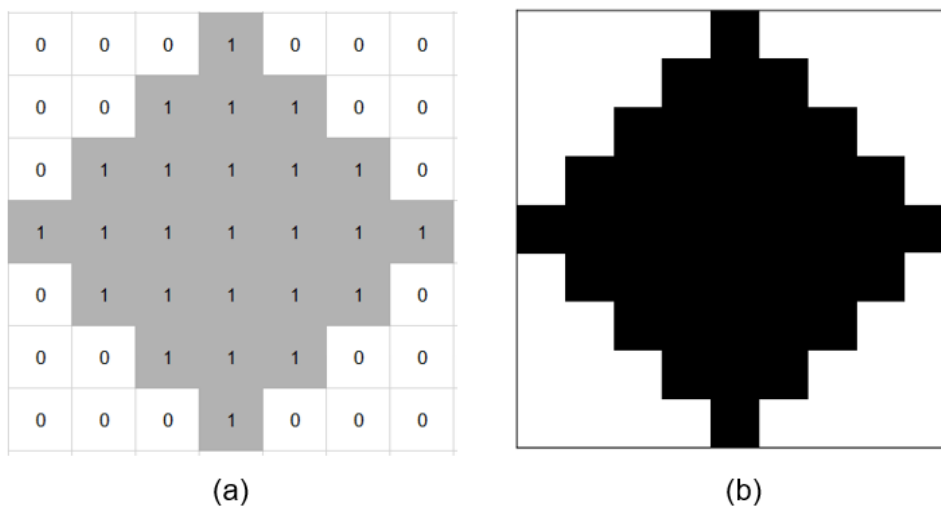
Para tornar mais claro como a matriz é influenciada pelo tamanho e configuração do *kernel* na Figura 34 é apresentado um exemplo da matriz preenchida com os valores 0 e 1, acompanhada de sua imagem correspondente.

4.5.1 *Layout* dinâmico

Para desenvolver o layout da interface, optou-se por uma abordagem dinâmica, que possibilita a construção em tempo de execução. Em vez de posicionar manualmente cada componente e mantê-lo estático, as coordenadas e tamanhos de cada elemento foram determinados por cálculos predefinidos no código. Essas coordenadas são então utilizadas na função disponível no Qt, no formato `setGeometry(x, y, largura, altura)`, em que cada parâmetro representa:

- x: a coordenada horizontal do componente;
- y: a coordenada vertical do componente;
- largura: a largura do componente;
- altura: a altura do componente.

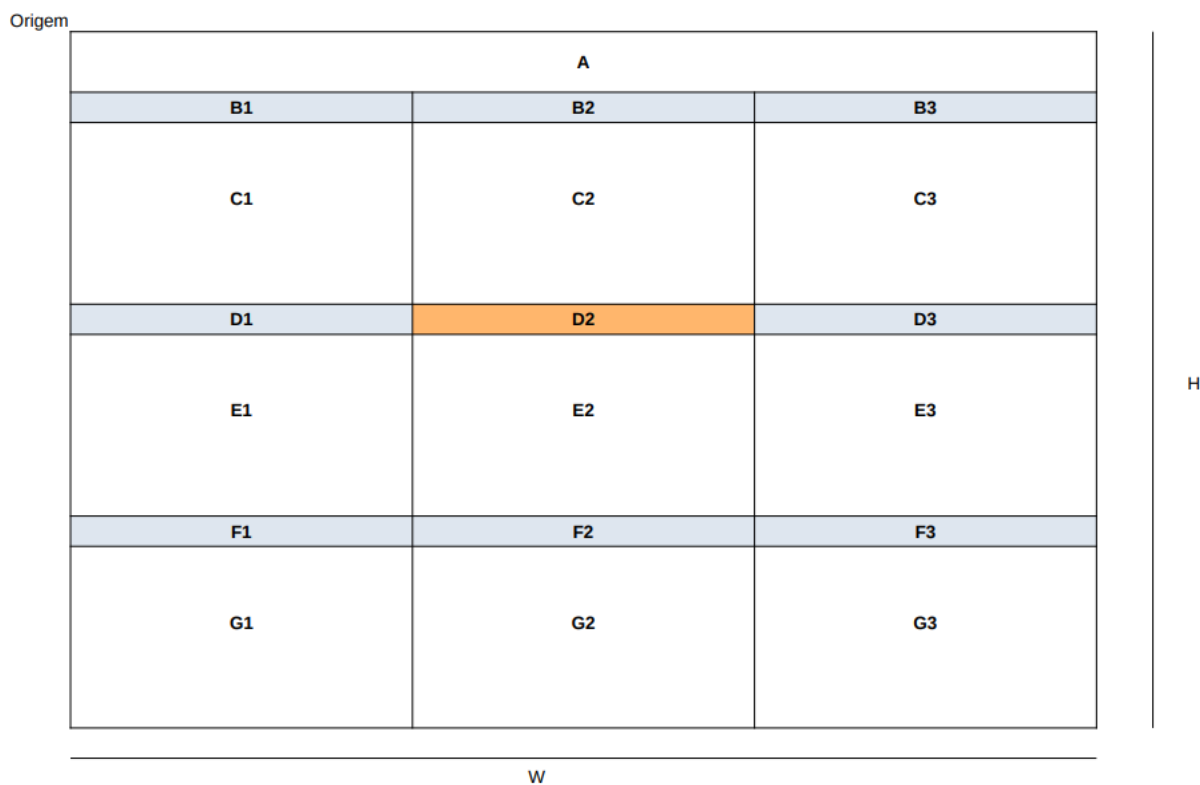
Figura 34 – Representação de uma matriz 7x7 Diamante (a) e sua imagem correspondente (b).



Fonte: Autoria própria.

Dessa maneira, todos os componentes são ajustados dinamicamente quando a aplicação é iniciada. Para uma compreensão mais detalhada desse processo, a Figura 35 apresenta a estrutura do design proposto.

Figura 35 – Estrutura da Interface.



Fonte: Autoria própria.

Na Figura 36, apresentam-se os cálculos elaborados para montar a parte de exibição

das imagens na tela, cada uma com os operadores aplicados. Os labels B1, B2, B3, D1, D2, D3, F1, F2 e F3 representam os títulos de cada imagem, indicando o operador aplicado, enquanto os cálculos C1, C2, C3, E1, E2, E3, G1, G2 e G3 referem-se ao posicionamento dessas imagens em que cada operador foi aplicado. É importante observar que alguns valores foram pré-estabelecidos para que os cálculos funcionassem (destacados em vermelho na Figura 36), atribuindo esses valores às suas respectivas variáveis. Portanto, caso seja necessário adaptar a estrutura da interface, basta ajustar os valores das variáveis de entrada, e todo o resto se adaptará conforme o que já está estabelecido. Essa abordagem simplifica a adaptação da interface a diferentes tamanhos de monitores, proporcionando uma experiência mais flexível para os usuários, independentemente da resolução de seus monitores.

Figura 36 – Posicionamento das imagens e seus respectivos *labels*.

Tamanhos	
W = 1200	Label_W = Img_W
H = 800	Label_H = 16
Ico_H=48	Img_W = floor(W/3)
Ico_W=48	Img_H = floor((H - (A.height - 3*B1.height))/3)

A		
A.origem = (0,0)		
A.width = W		
A.height = Ico_H		

B1	B2	B3
B1.origem = (0, Ico_H)	B2.origem = (Label_W, Ico_H)	B3.origem = (2*Label_W, Ico_H)
B1.width = Label_W	B2.width = Label_W	B3.width = Label_W
B1.height = Label_H	B2.height = Label_H	B3.height = Label_H
C1	C2	C3
C1.origem = (0, Ico_H+Label_H)	C2.origem = (Label_W, Ico_H+Label_H)	C3.origem = (2*Label_W, Ico_H+Label_H)
C1.width = Img_W	C2.width = Img_W	C3.width = Img_W
C1.height = Img_H	C2.height = Img_H	C3.height = Img_H
D1	D2	D3
D1.origem = (0, Ico_H+Label_H+Img_H)	D2.origem = (Label_W, Ico_H+Label_H+Img_H)	D3.origem = (2*Label_W, Ico_H+Label_H+Img_H)
E1	E2	E3
E1.origem = (0, Ico_H+2*Label_H+Img_H)	E2.origem = (Label_W, Ico_H+2*Label_H+Img_H)	E3.origem = (2*Label_W, Ico_H+2*Label_H+Img_H)
F1	F2	F3
F1.origem = (0, Ico_H+2*Label_H+2*Img_H)	F2.origem = (Label_W, Ico_H+2*Label_H+2*Img_H)	F3.origem = (2*Label_W, Ico_H+2*Label_H+2*Img_H)
G1	G2	G3
G1.origem = (0, Ico_H+3*Label_H+2*Img_H)	G2.origem = (Label_W, Ico_H+3*Label_H+2*Img_H)	G3.origem = (2*Label_W, Ico_H+3*Label_H+2*Img_H)

D1, D2, D3, F1, F2 e F3 têm o mesmo width e height de B1
 E1, E2, E3, G1, G2 e G3 têm o mesmo width e height de C1

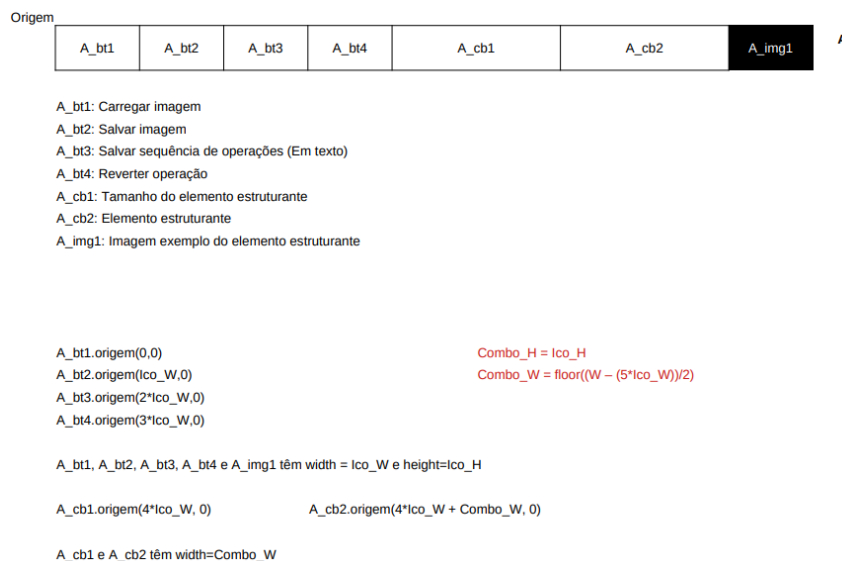
Fonte: Autoria própria.

A Figura 37 representa os cálculos para o 'cabeçalho' da aplicação, onde estão posicionados todos os botões e suas funcionalidades.

4.6 Análise de Similaridade entre Imagens

Nesta seção, é apresentado o processo de análise de similaridade entre a imagem de referência e as imagens resultantes do processamento. O objetivo é realizar uma avaliação quantitativa que indique o quão próxima cada imagem resultado está da imagem

Figura 37 – Posicionamento dos botões.



Fonte: Autoria própria.

de referência. Para essa análise, utilizou-se a biblioteca scikit-image para métricas de similaridade e a OpenCV para manipulação de imagens.

4.6.1 Métrica de Similaridade Estrutural (SSIM)

A métrica de Similaridade Estrutural (CARDOSO et al., 2012) é uma medida amplamente empregada para avaliar a semelhança entre duas imagens. Ela fornece um valor entre -1 e 1, onde 1 indica uma correspondência perfeita, enquanto -1 sugere uma dissimilaridade total. O cálculo é realizado comparando diferentes aspectos da estrutura da imagem, incluindo luminância, contraste e estrutura.

4.6.2 Implementação do Processo

O processo de análise implementado neste trabalho, consiste nas seguintes etapas:

- **Carregamento das Imagens:** As imagens de referência e resultado são carregadas a partir de arquivos. A imagem de referência é convertida para tons de cinza para simplificar a comparação.
- **Cálculo da Métrica SSIM:** Para cada imagem de resultado, a métrica SSIM é calculada em relação à imagem de referência. A biblioteca scikit-image fornece a função `metrics.structural_similarity` para realizar esse cálculo.

Os resultados da análise são exibidos, indicando a pontuação SSIM para cada imagem resultado em comparação com a imagem de referência. Essa pontuação oferece uma visão quantitativa da similaridade estrutural entre as imagens.

5 RESULTADOS E DISCUSSÃO

Os resultados apresentados nesta seção incluem dois tipos de imagens: Sintéticas e Microscópicas. As imagens microscópicas são obtidas diretamente do mundo real, conforme detalhado na Seção 4.1. Por outro lado, as imagens sintéticas são geradas digitalmente por meio de ferramentas específicas, não sendo capturadas de situações reais. Estas últimas são produzidas de maneira controlada e manipulada digitalmente, oferecendo um contexto virtual e controlado para a análise e experimentação, e foram retiradas do trabalho Banon e Barrera (1994).

5.1 Imagens sintéticas

Para os experimentos realizados nas imagens sintéticas, foi adotada a seguinte abordagem: escolhemos uma imagem inicial como ponto de partida e estabelecemos uma imagem de referência como o resultado desejado. A imagem inicial é carregada na aplicação, e os operadores morfológicos são aplicados com o intuito de aproximar o resultado ao máximo da imagem de referência. Durante o processo, a sequência de operadores morfológicos é determinada de maneira visual, permitindo que o usuário, a cada interação, escolha a opção que mais se aproxima do resultado desejado. Essa abordagem proporciona uma manipulação morfológica otimizada da imagem inicial, permitindo uma adaptação flexível às preferências do usuário e uma busca eficaz pelo estado desejado representado pela imagem de referência.

5.1.1 Experimento 1 - Ruídos

Para o primeiro teste, foi utilizada uma imagem de entrada contendo alguns ruídos. A imagem de referência para este teste será a própria imagem de entrada, porém com menos ruídos e formas mais nitidamente definidas, veja Figura 38.

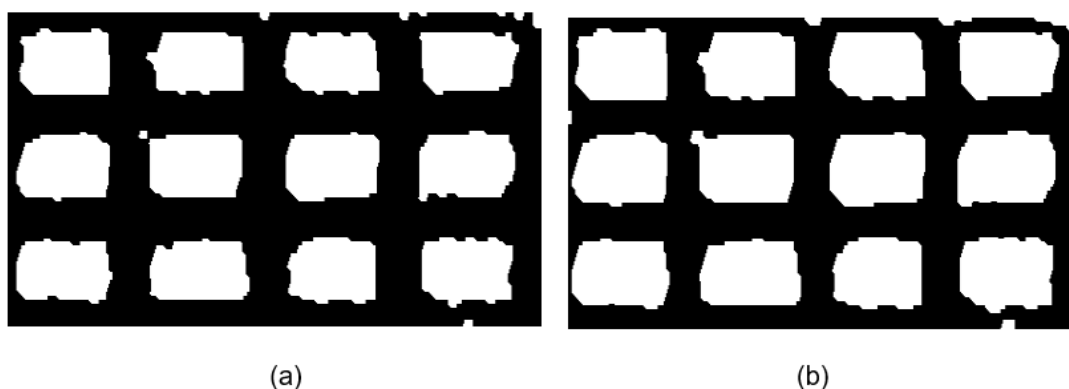
Figura 38 – Imagem de entrada (a) e imagem de referencia (b).



Fonte: Autoria própria.

Ao carregar a imagem de entrada no programa e aplicar uma combinação de operadores morfológicos, foi possível alcançar dois resultados próximos apresentados na Figura 39. Embora não sejam uma correspondência perfeita com a imagem de referência, é evidente uma melhora significativa em comparação à imagem inicial. A maioria dos ruídos presentes na imagem foram eliminados, e as formas ganharam maior precisão. Estes resultados representam um avanço notável no processo de manipulação morfológica aplicado à imagem de entrada.

Figura 39 – Imagens resultantes após aplicar os operadores morfológicos.



Fonte: Autoria própria.

A seguir, é apresentado a estrutura de log gerada durante o processamento que levou ao resultado ilustrado na Figura 39(a).

```
1 Nome do arquivo de imagem original: QuadradoOriginal.png
2
3 Opção selecionada no Kernel: Matriz 7x7 - Quadrado
4
5 Matriz:
6 0 0 0 0 0 0 0
7 0 1 1 1 1 1 0
8 0 1 1 1 1 1 1
9 0 1 1 1 1 1 0
10 0 1 1 1 1 1 0
11 0 1 1 1 1 1 0
12 0 0 0 0 0 0 0
13
14 Sequência de operadores aplicados na imagem:
15 Erosão
16 Dilatação
17 Fechamento
```

```
18 Abertura
19 Abertura
20 Abertura
21 Abertura
22 Fechamento
23 Fechamento
24 Dilatação
25 Fechamento
26 Fechamento
27 Abertura
28 Abertura
29 Dilatação
```

A seguir, é apresentado a estrutura de log gerada durante o processamento que levou ao resultado ilustrado na Figura 39(b).

```
1 Nome do arquivo de imagem original: QuadradoOriginal.png
2
3 Opção selecionada no Kernel: Matriz 7x7 - Quadrado
4
5 Matriz:
6 0 0 0 0 0 0 0
7 0 1 1 1 1 1 0
8 0 1 1 1 1 1 1
9 0 1 1 1 1 1 0
10 0 1 1 1 1 1 0
11 0 1 1 1 1 1 0
12 0 0 0 0 0 0 0
13
14 Sequência de operadores aplicados na imagem:
15 Fechamento
16 Fechamento
17 Abertura
18 Dilatação
19 Fechamento
20 Fechamento
21 Dilatação
22 Fechamento
23 Fechamento
```

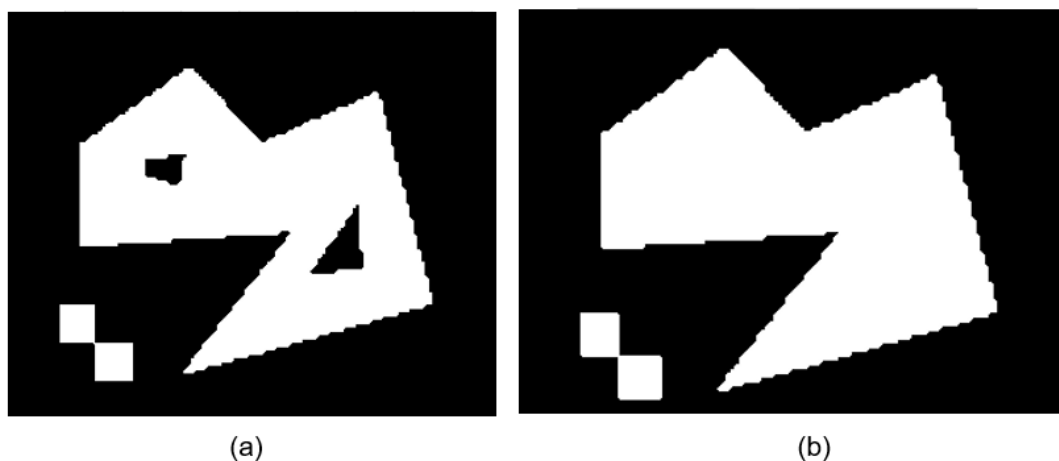
Ao empregar a métrica de similaridade estrutural (SSIM) nos resultados destacados na Figura 39, conseguimos quantificar a proximidade de cada resultado em relação à imagem de referência. A imagem de entrada, ilustrada na Figura 38 (a), inicialmente apresentou um SSIM de 0.4438. Após o processamento na interface, os dois resultados obtidos, indicados como (a) e (b) na Figura 39, alcançaram SSIM de 0.7321 e 0.6952, respectivamente. Isso significa que ambos superaram a imagem de entrada em fidelidade, mas é notável que o resultado (a) obteve uma pontuação mais próxima da fidelidade desejada.

É importante ressaltar que essa análise, embora não seja o foco principal deste trabalho, foi incorporada como exemplo específico apenas para esse teste. Ela destaca como essas imagens finais podem ser valiosas em outros tipos de análises após seu processamento.

5.1.2 Experimento 2 - Preenchimento

Para o segundo teste, foi utilizada uma imagem de entrada, contendo algumas formas. A imagem de referência para esse teste, será a imagem original, preservando suas características e formas principais, porém preenchendo a parte interna que continha alguns furos, (veja Figura 40).

Figura 40 – Imagem de entrada (a) e imagem de referência (b).

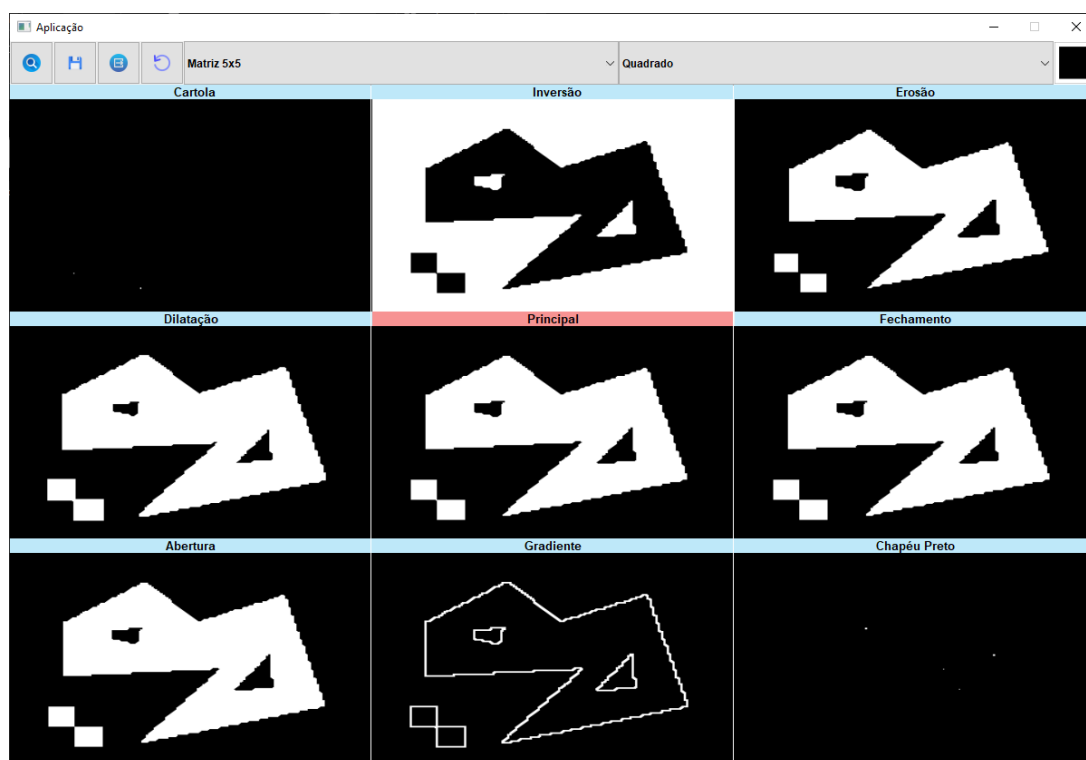


Fonte: Autoria própria.

Ao carregar a imagem de entrada na aplicação, os primeiros filtros aplicados foram apresentados como opções, conforme ilustrado na Figura 41.

Após aplicar diversas combinações de operadores morfológicos, alcançamos o resultado apresentado na Figura 42. Nesse caso, foi possível obter um resultado bastante satisfatório em comparação com a imagem de referência. Os espaços foram preenchidos, e a forma original foi mantida, preservando as características de interesse.

Figura 41 – Visão da interface ao carregar a imagem inicial.



Fonte: Autoria própria.

Figura 42 – Imagem resultante após aplicar os operadores morfológicos.



Fonte: Autoria própria.

A seguir, é apresentado a estrutura de log gerada durante o processamento que levou ao resultado ilustrado na Figura 42.

- 1 Nome do arquivo de imagem original: PrenchimentoOriginal.png
- 2
- 3 Opção selecionada no Kernel: Matriz 5x5 - Quadrado

```
4
5  Matriz:
6  0 0 0 0 0
7  0 1 1 1 0
8  0 1 1 1 0
9  0 1 1 1 0
10 0 0 0 0 0
11
12 Sequência de operadores aplicados na imagem:
13 Erosão
14 Abertura
15 Dilatação
16 Dilatação
17 Fechamento
18 Erosão
19 Fechamento
20 Fechamento
21 Erosão
22 Fechamento
23 Erosão
24 Abertura
25 Dilatação
26 Dilatação
27 Dilatação
28 Dilatação
29 Dilatação
30 Dilatação
31 Dilatação
32 Fechamento
33 Dilatação
34 Fechamento
35 Erosão
36 Erosão
37 Erosão
38 Erosão
39 Erosão
40 Erosão
41 Erosão
```

5.1.3 Experimento 3 - Palavras

No terceiro teste, utilizamos uma imagem de entrada contendo algumas letras com traços finos e formas não muito bem estruturadas. A imagem de referência para esse teste é uma versão da imagem de entrada, porém com as letras melhor desenhadas e com espessura maior, veja Figura 43.

Figura 43 – Imagem de entrada (a) e imagem de referencia (b).



Fonte: Autoria própria.

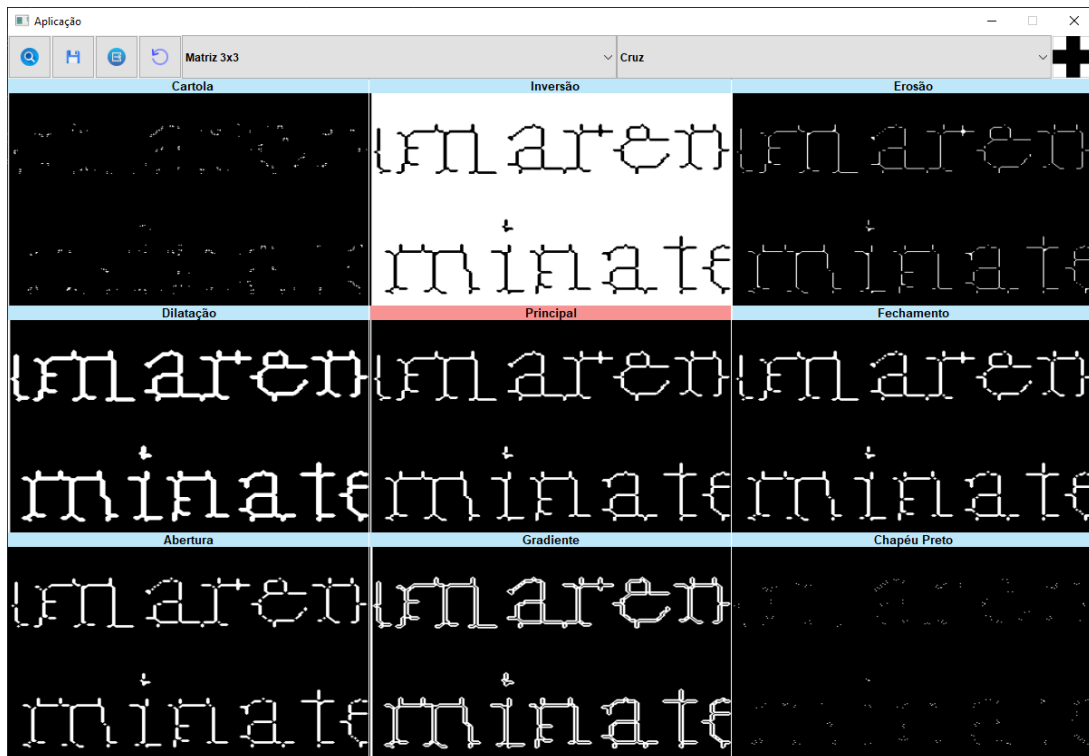
Ao carregar a imagem de entrada na aplicação, os primeiros filtros aplicados foram apresentados como opções, conforme ilustrado na Figura 41.

Após a aplicação de várias combinações de operadores morfológicos, obtivemos o resultado apresentado na Figura 45. Embora o resultado não tenha coincidido exatamente com a imagem de referência, observamos melhorias significativas. As letras na imagem ficaram mais nítidas e suas espessuras foram aprimoradas, contribuindo para uma representação visual mais refinada.

A seguir, é apresentado a estrutura de log gerada durante o processamento que levou ao resultado ilustrado na Figura 45.

```
1 Nome do arquivo de imagem original: EscritaReferencia.png
2
3 Opção selecionada no Kernel: Matriz 3x3 - Cruz
4
5 Matriz:
6 0 1 0
7 1 1 1
```

Figura 44 – Visão da interface ao carregar a imagem inicial.



Fonte: Autoria própria.

Figura 45 – Imagem resultante após aplicar os operadores morfológicos.



Fonte: Autoria própria.

```

8  0 1 0
9
10 Sequência de operadores aplicados na imagem:
11 Gradiente
12 Dilatação
13 Fechamento

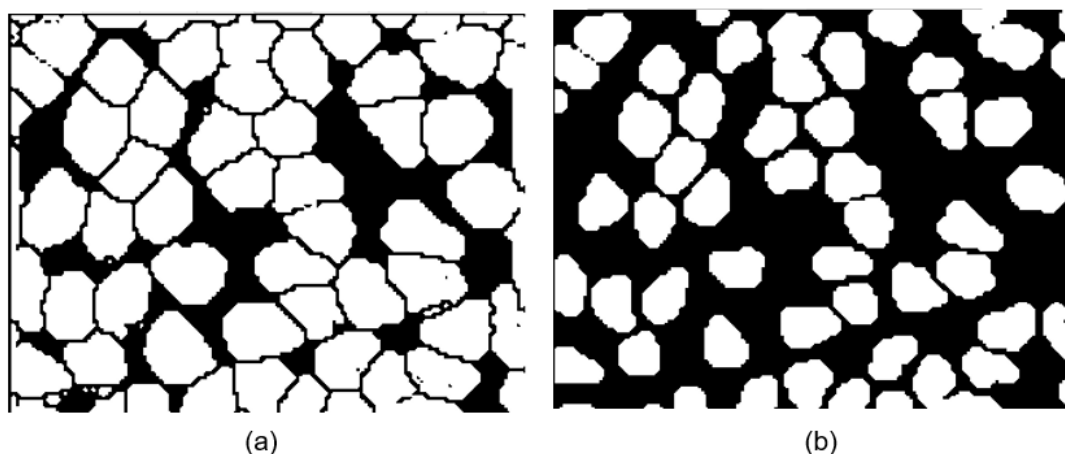
```

14 Dilatação
15 Dilatação
16 Fechamento
17 Abertura
18 Dilatação
19 Dilatação
20 Erosão
21 Abertura
22 Dilatação
23 Dilatação
24 Erosão
25 Dilatação
26 Dilatação
27 Dilatação
28 Dilatação
29 Fechamento
30 Fechamento
31 Erosão
32 Fechamento
33 Dilatação
34 Fechamento
35 Dilatação
36 Dilatação
37 Dilatação
38 Fechamento
39 Erosão
40 Dilatação
41 Fechamento
42 Erosão
43 Fechamento

5.1.4 Experimento 4 - Círculos

No quarto teste, empregamos uma imagem de entrada contendo formas semelhantes a círculos, porém bastante dilatados e sem marcações bem definidas, quase fundindo-se umas às outras. A imagem de referência para este teste é uma versão da imagem de entrada, porém com as formas menos dilatadas, apresentando maior espaço entre elas e definições mais nítidas, veja a Figura 46.

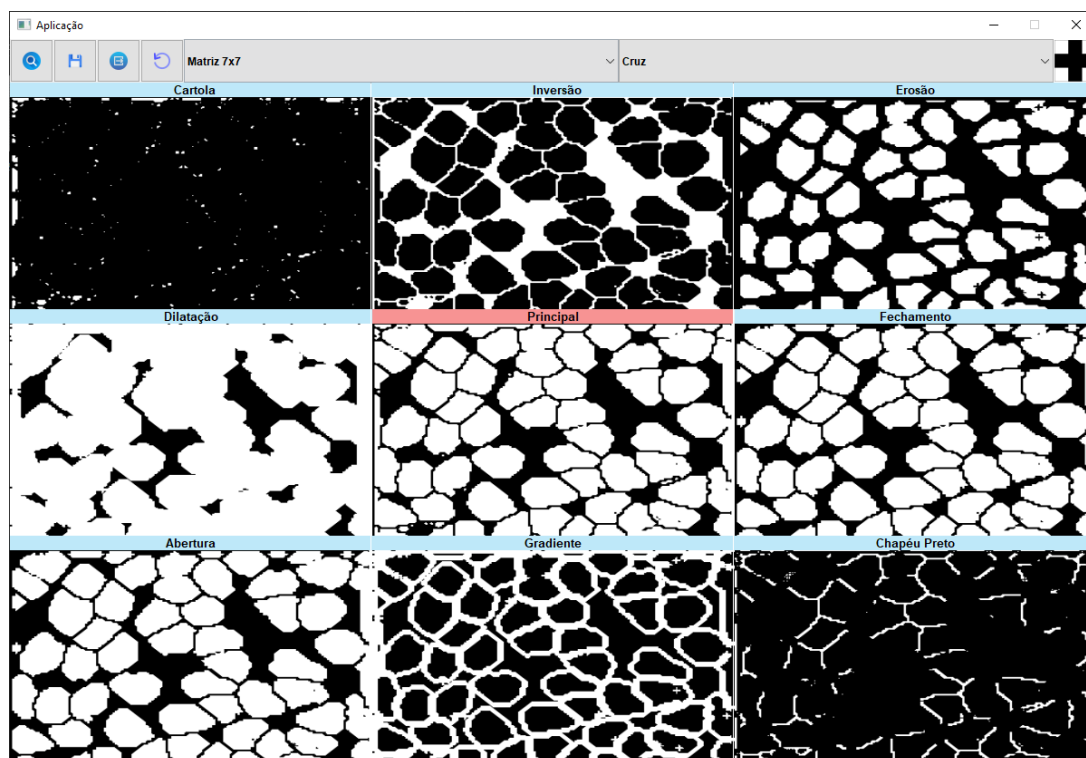
Figura 46 – Imagem de entrada (a) e imagem de referência (b).



Fonte: Autoria própria.

Ao carregar a imagem de entrada na aplicação, os primeiros filtros aplicados foram apresentados como opções, conforme ilustrado na Figura 47.

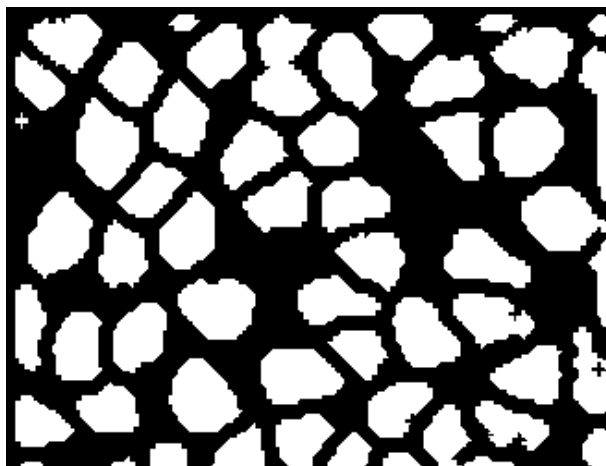
Figura 47 – Visão da interface ao carregar a imagem inicial.



Fonte: Autoria própria.

Após a aplicação de diversas combinações de operadores morfológicos, alcançamos o resultado retratado na Figura 48. Nota-se uma aproximação em relação à imagem de referência, com as formas menos dilatadas, resultando em contornos mais nítidos e de fácil identificação

Figura 48 – Imagem resultante após aplicar os operadores morfológicos.



Fonte: Autoria própria.

A seguir, é apresentado a estrutura de log gerada durante o processamento que levou ao resultado ilustrado na Figura 48.

```
1 Nome do arquivo de imagem original: CirculoOriginal.png
2
3 Opção selecionada no Kernel: Matriz 7x7 - Cruz
4
5 Matriz:
6 0 0 0 1 0 0 0
7 0 0 0 1 0 0 0
8 0 0 0 1 0 0 0
9 1 1 1 1 1 1 1
10 0 0 0 1 0 0 0
11 0 0 0 1 0 0 0
12 0 0 0 1 0 0 0
13
14 Sequência de operadores aplicados na imagem:
15 Erosão
16 Fechamento
17 Dilatação
18 Erosão
19 Abertura
20 Fechamento
21 Erosão
22 Dilatação
```

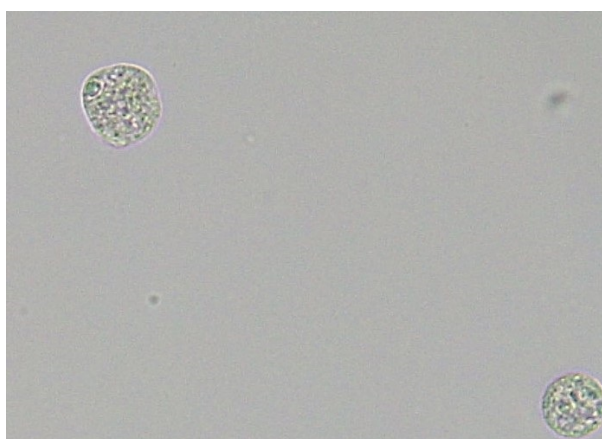
5.2 Imagens microscópicas

Ao contrário dos testes em imagens sintéticas, nos experimentos com imagens reais (microscópio), não será estabelecida uma imagem de referência. A avaliação do resultado final será baseada na imagem de entrada, e o processo será considerado concluído quando a imagem alcançar um nível satisfatório de detalhamento, realçando e destacando de forma eficaz as características presentes na imagem original. A abordagem consiste em explorar as funcionalidades da interface, utilizando o que ela oferece, e com base na análise visual, tentar alcançar um resultado satisfatório sem a obrigatoriedade de atingir um padrão *ótimo* predefinido. Ao término desse processo, obtém-se uma imagem segmentada. Isso é particularmente interessante, pois essas imagens podem ser utilizadas para a extração de características, possibilitando determinar informações importantes sobre os objetos presentes na imagem final, como a distância entre eles, sua forma, área, entre outros. É importante ressaltar que a extração de características não é o foco deste trabalho; no entanto, isso demonstra o potencial das imagens processadas para aplicações futuras. Todas as imagens utilizadas aqui, foram coletadas utilizando a abordagem descrita na Seção 4.1

5.2.1 Experimento 1

Neste teste, empregamos a imagem de entrada, conforme ilustrado na Figura 49. A imagem contém duas formas circulares, e o objetivo será separar essas formas do restante da imagem, destacando as formas com a cor branca e o restante (fundo) na cor preta.

Figura 49 – Imagem de entrada.

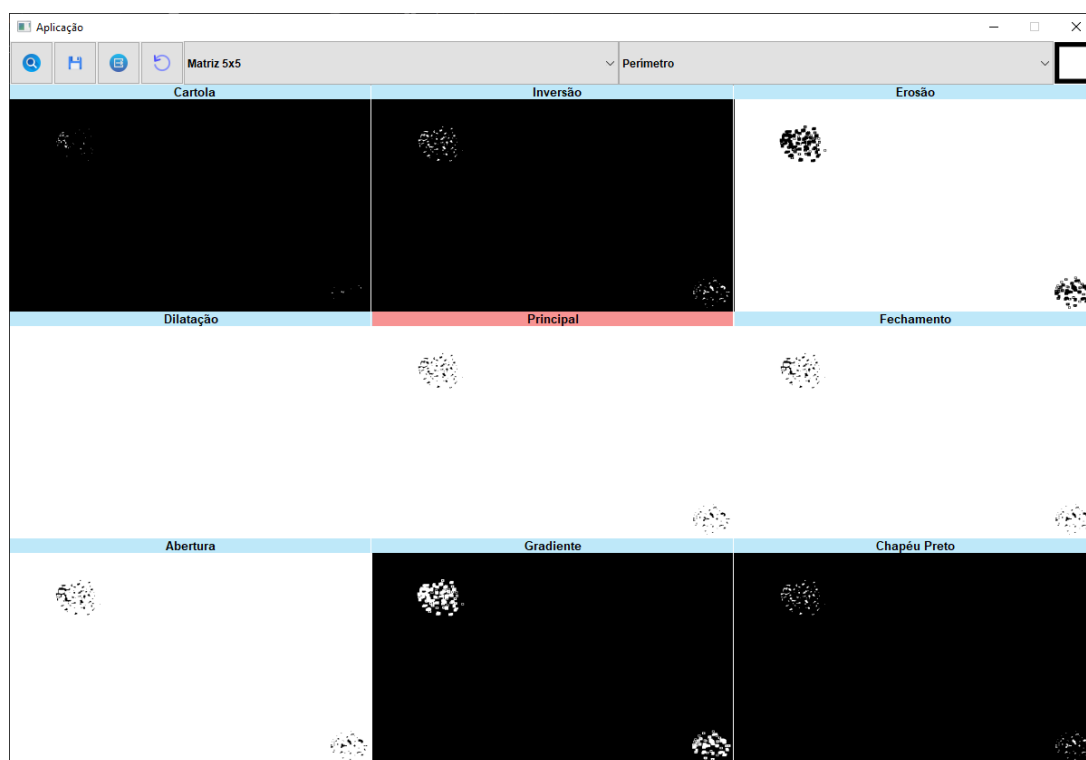


Fonte: Autoria própria.

Ao carregar a imagem de entrada na aplicação, os primeiros filtros aplicados foram apresentados como opções, conforme ilustrado na Figura 50.

Após aplicar algumas operações morfológicas na imagem de entrada, chegou-se no resultado mostrado na Figura 51

Figura 50 – Visão da interface ao carregar a imagem inicial.



Fonte: Autoria própria.

Figura 51 – Imagem resultante após aplicar os operadores morfológicos.



Fonte: Autoria própria.

A seguir, é apresentado a estrutura de log gerada durante o processamento que levou ao resultado ilustrado na Figura 51.

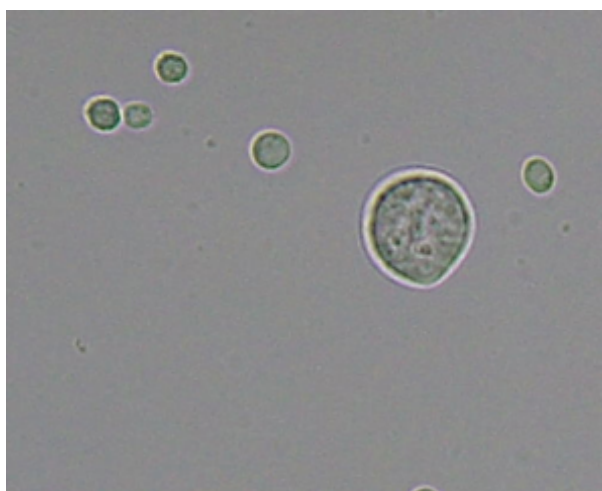
```
1 Nome do arquivo de imagem original: SimplesOriginal.jpg
2
3 Opção selecionada no Kernel: Matriz 5x5 - Perímetro
4
```

```
5  Matriz:
6  1 1 1 1 1
7  1 0 0 0 1
8  1 0 0 0 1
9  1 0 0 0 1
10 1 1 1 1 1
11
12 Sequência de operadores aplicados na imagem:
13 Gradiente
14 Gradiente
15 Dilatação
16 Dilatação
17 Dilatação
18 Abertura
```

5.2.2 Experimento 2

Neste teste, utilizamos a imagem de entrada, conforme apresentado na Figura 52. A imagem contém diversas formas circulares em tamanhos variados. O objetivo será separar essas formas do restante da imagem, realçando-as com a cor branca, enquanto o restante (fundo) é preenchido com a cor preta, preservando ao máximo as características das formas principais.

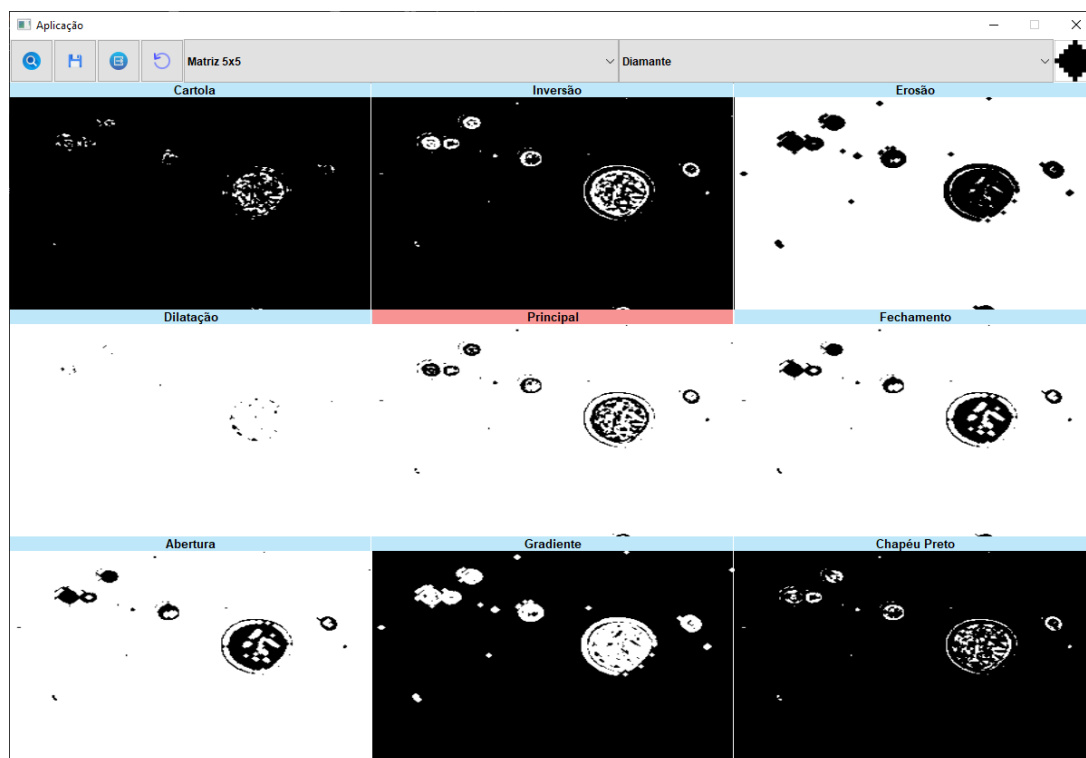
Figura 52 – Imagem de entrada.



Fonte: Autoria própria.

Ao carregar a imagem de entrada na aplicação, os primeiros filtros aplicados foram apresentados como opções, conforme ilustrado na Figura 53.

Figura 53 – Visão da interface ao carregar a imagem inicial.



Fonte: Autoria própria.

Após a aplicação de operações morfológicas na imagem de entrada, foi possível alcançar o resultado apresentado na Figura 54. É crucial observar que, para alcançar esse resultado específico, alguns objetos foram perdidos durante o processamento. Inicialmente, havia seis regiões circulares, mas, ao final, restaram apenas quatro. Isso evidencia que os operadores morfológicos podem gerar resultados interessantes, mas é necessário cautela, pois podem ocorrer perdas de informações durante o processo. Portanto, a escolha do caminho adequado é essencial para preservar todas as regiões de interesse.

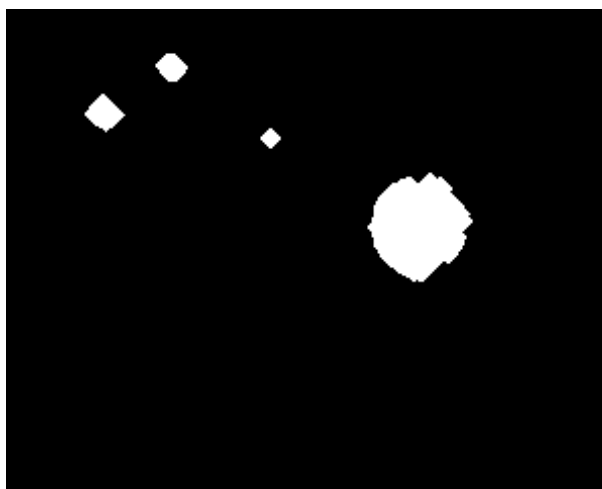
A seguir, é apresentado a estrutura de log gerada durante o processamento que levou ao resultado ilustrado na Figura 54.

```

1 Nome do arquivo de imagem original: MistaOriginal.png
2
3 Opção selecionada no Kernel: Matriz 5x5 - Diamante
4
5 Matriz:
6 0 0 1 0 0
7 0 1 1 1 0
8 1 1 1 1 1
9 0 1 1 1 0
10 0 0 1 0 0

```

Figura 54 – Imagem resultante após aplicar os operadores morfológicos.



Fonte: Autoria própria.

11

12 Sequência de operadores aplicados na imagem:

13 Erosão

14 Inversão

15 Erosão

16 Fechamento

17 Abertura

18 Dilatação

19 Erosão

20 Erosão

21 Abertura

22 Dilatação

23 Dilatação

24 Dilatação

25 Dilatação

26 Dilatação

27 Dilatação

28 Erosão

29 Erosão

30 Dilatação

31 Dilatação

32 Fechamento

33 Erosão

34 Erosão

35 Erosão

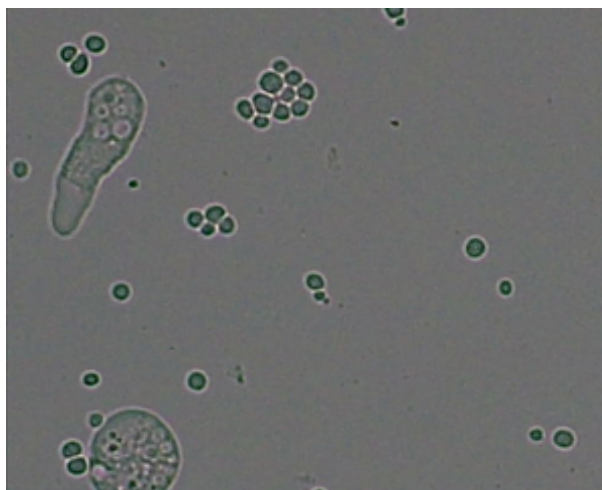
36 Erosão

37 Erosão

5.2.3 Experimento 3

Neste teste, utilizamos a imagem de entrada, conforme apresentado na Figura 55. Diversas formas estão presentes na imagem, sem seguir necessariamente o mesmo padrão de tamanho e formato. O objetivo é separar essas formas do restante da imagem, realçando-as com a cor branca, enquanto o fundo é preenchido com a cor preta, preservando ao máximo as características das formas principais.

Figura 55 – Imagem de entrada.



Fonte: Autoria própria.

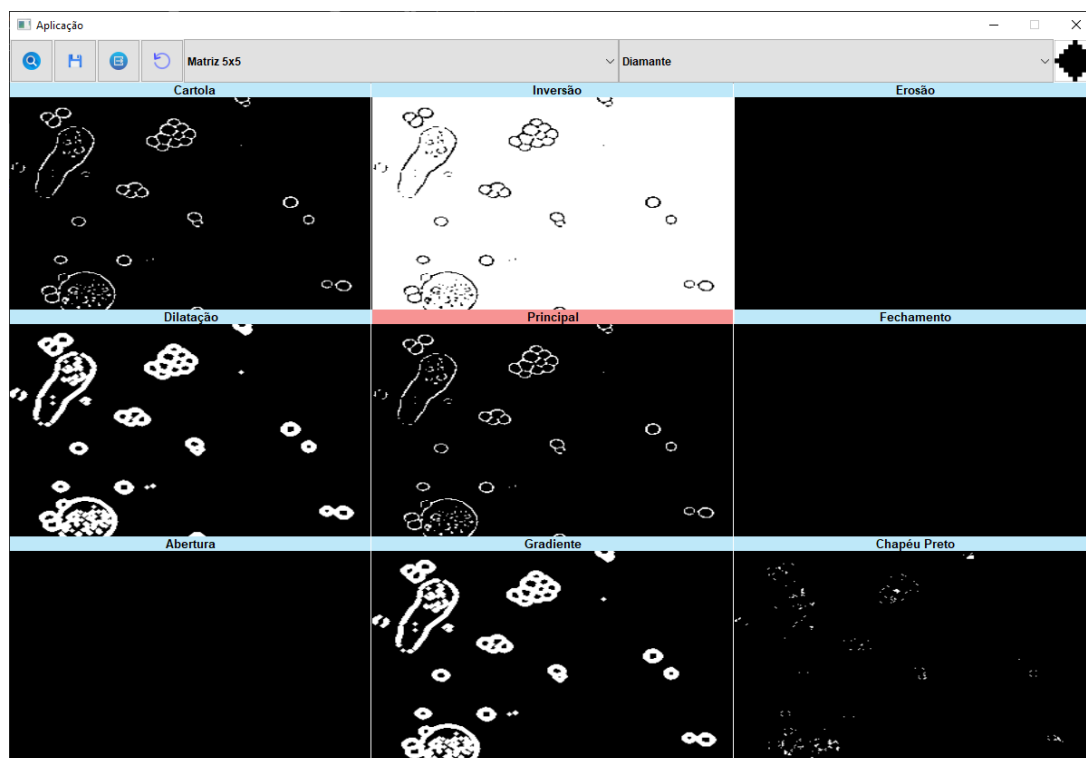
Ao carregar a imagem de entrada na aplicação, os primeiros filtros aplicados foram apresentados como opções, conforme ilustrado na Figura 56.

Após aplicar algumas operações morfológicas na imagem de entrada, chegou-se no resultado mostrado na Figura 57

Neste caso, o resultado visual mais satisfatório foi alcançado ao separar as formas, evitando preenchê-las completamente. Vale ressaltar que, embora essa tenha sido a melhor configuração visual encontrada durante a exploração realizada na interface para essa imagem específica, isso não implica que seja a melhor solução possível. A interface oferece diversas combinações de *kernels* e operadores, e o resultado pode variar dependendo das escolhas feitas durante a exploração.

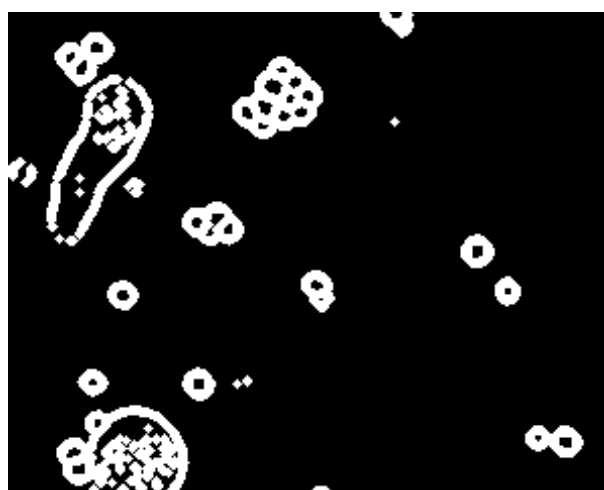
A seguir, é apresentado a estrutura de log gerada durante o processamento que levou ao resultado ilustrado na Figura 57.

Figura 56 – Visão da interface ao carregar a imagem inicial



Fonte: Autoria própria.

Figura 57 – Imagem resultante após aplicar os operadores morfológicos.



Fonte: Autoria própria.

```

1 Nome do arquivo de imagem original: ComplexaOriginal.png
2
3 Opção selecionada no Kernel: Matriz 5x5 - Diamante
4
5 Matriz:
6 0 0 1 0 0
    
```

```
7 0 1 1 1 0
8 1 1 1 1 1
9 0 1 1 1 0
10 0 0 1 0 0
11
12 Sequência de operadores aplicados na imagem:
13 Dilatação
14 Dilatação
15 Gradiente
16 Dilatação
17 Gradiente
18 Dilatação
19 Abertura
```

6 CONCLUSÃO

6.1 Considerações Finais

A amplitude de possibilidades oferecidas aos usuários pela aplicação desenvolvida, permite ao alcançar resultados de forma interativa e exploratória. Dessa forma, entende-se que a aplicação consegue atingir o objetivo proposto nesse trabalho, pois permite que o usuário trabalhe com tais operações ainda que tenha pouco ou nenhum conhecimento de operadores morfológicos.

Destacamos a inclusão de funcionalidades cruciais, como a capacidade de reverter passos, facilitando o processo de utilização. A possibilidade de salvar a imagem a qualquer momento, aliada à opção de exportar o log das operações aplicadas, oferece aos usuários um controle significativo sobre o processo, permitindo o armazenamento de informações valiosas para usos futuros.

A aplicação demonstrou um bom desempenho durante a aplicação dos filtros, mantendo-se livre de lentidões e inconsistências. A consistência observada ao longo dos testes reforça a fidelidade da aplicação à sua proposta original, garantindo uma experiência confiável para os usuários.

É relevante salientar que, embora nem sempre tenha sido possível alcançar resultados ótimos, a natureza exploratória da interface abre amplas oportunidades para experimentação. Quanto mais a aplicação é utilizada e compreendida, maiores são as chances de se obter resultados satisfatórios, destacando a importância do aprendizado contínuo e da exploração das diversas funcionalidades oferecidas. Além disso, cabe destacar que a aplicação possui um -, uma vez que possibilita a usuários que estejam estudando tais operadores explorar uma galeria de resultados para uma imagem de entrada qualquer de seu interesse.

6.2 Trabalhos Futuros

Realizar a implementação de operadores morfológicos mais complexos, tais como: *hit-miss*, *thinning*, *thickening* e *skeletons*, conforme abordado por Gonzalez e Woods (2002).

Para aprimorar a usabilidade da aplicação, é considerada a possibilidade de simplificar o arquivo de log gerado. Uma abordagem para essa simplificação envolve otimizar a sequência de operadores escolhida pelo usuário. Por exemplo, a aplicação da dilatação seguida da erosão, ou vice-versa, pode ser simplificada para a aplicação direta de uma abertura ou fechamento. Essa otimização não apenas reduz a complexidade do arquivo de log, mas também facilita a leitura e reprodução dos testes, proporcionando uma abordagem mais eficiente para alcançar os mesmos resultados.

Outra proposta para a continuidade deste trabalho, seria a criação de uma base

de dados dinâmica, gerada a partir das análises e processamentos realizados em imagens dentro da aplicação proposta, visando transformar o histórico de manipulações de imagem, registrado no log da aplicação, em uma fonte de informações para treinamento de algoritmos de Machine Learning. Ao carregar a imagem original e aplicar uma série de operadores morfológicos, a aplicação gera uma trilha detalhada de todas as etapas realizadas, registrando cada filtro aplicado e a sequência exata dessas manipulações. Este log vai revelar as preferências do usuário, escolhas de filtros mais frequentes e as variações de como diferentes imagens respondem a diferentes processamentos, ou seja, cada imagem processada torna-se uma amostra, e as operações morfológicas aplicadas, juntamente com a sequência utilizada, constituem características valiosas, com o potencial de ensinar um algoritmo a reconhecer padrões, preferências e a prever quais operadores morfológicos podem otimizar uma imagem específica.

Referências

- ALFIERI, S. C. et al. Proteinase activities in total extracts and in medium conditioned by *acanthamoeba polyphaga* trophozoites. **Journal of Parasitology**, v. 86, n. 2, p. 220–227, 2000. Citado na página 32.
- BANON, G. J. F.; BARRERA, J. **Bases da Morfologia Matemática para a análise de imagens binárias**. [S.l.]: UFPE-DI, 1994. Citado na página 47.
- BELLINI, N. K. et al. Isolation of *naegleria* spp. from a brazilian water source. **Pathogens**, Multidisciplinary Digital Publishing Institute, v. 9, n. 2, p. 90, 2020. Citado na página 32.
- BURGER, W. et al. **Principles of digital image processing**. [S.l.]: Springer, 2009. Citado na página 11.
- CARDOSO, J. V. d. M. et al. Comparação das métricas objetivas de qualidade de vídeos baseadas na similaridade estrutural e na sensibilidade ao erro. **Revista de tecnologia da informação e comunicação**, v. 1, n. 2, p. 33–40, 2012. Citado na página 46.
- CUNHA, A.; LIMA, L.; VIEIRA, T. Extração de isosuperfícies usando galerias de design. In: **Workshop de Trabalho de Graduação do Sibgrapi**. [s.n.], 2011. Disponível em: <<https://ic.ufal.br/professor/thales/>>. Citado 3 vezes nas páginas 2, 30 e 31.
- DEVELOPER, A. **Blurring an Image**. 2023. Acesso em: 17 de fevereiro 2023. Disponível em: <https://developer.apple.com/documentation/accelerate/blurring_an_image>. Citado na página 7.
- D'IPPÓLITO, K. M. **Estudo de métodos numéricos para eliminação de ruídos em imagens digitais**. Dissertação de Mestrado — Universidade Estadual Paulista (UNESP), 2005. Citado na página 8.
- FISHER, R. et al. **Convolucao**. 2020. Acesso em: 10 de fevereiro 2023. Disponível em: <<https://homepages.inf.ed.ac.uk/rbf/HIPR2/convolve.htm/>>. Citado na página 7.
- GONZALEZ, R. C.; WOODS, R. E. **Digital image processing**. [S.l.]: Prentice Hall, 2002. Citado 8 vezes nas páginas 4, 9, 12, 17, 23, 25, 26 e 66.
- KLEIN, T. E.; GALLAGER, R. G. Power control for the additive white gaussian noise channel under channel estimation errors. In: IEEE. **International Symposium on Information Theory**. [S.l.], 2001. p. 304. Citado na página 9.
- KULKARNI, R. R.; SHEKHAR, S. Black hat transform for shadow detection and removal. **Signal Processing Letters**, IEEE, v. 20, n. 1, p. 26–29, 2013. Citado na página 25.
- LI, C.; WANG, Y. **Mathematical morphology in image processing and pattern recognition**. [S.l.]: CRC Press, 2014. Citado 2 vezes nas páginas 21 e 26.
- LI, X.; WANG, Y. Top-hat transform for small-object detection in medical images. **IEEE Transactions on Image Processing**, IEEE, v. 23, n. 10, p. 4233–4244, 2014. Citado na página 24.

- LIU, J.; SHI, Y. Image feature extraction method based on shape characteristics and its application in medical image analysis. In: SPRINGER. **International Conference on Applied Informatics and Communication**. [S.l.], 2011. p. 172–178. Citado na página 14.
- MARKS, J. et al. Design galleries: A general approach to setting parameters for computer graphics and animation. In: **Seminal Graphics Papers: Pushing the Boundaries, Volume 2**. [S.l.]: ACM, 2023. p. 73–84. Citado 2 vezes nas páginas 29 e 30.
- MARQUEZ-NEILA, P.; BAUMELA, L.; ALVAREZ, L. A morphological approach to curvature-based evolution of curves and surfaces. **Transactions on Pattern Analysis and Machine Intelligence**, IEEE, v. 36, n. 1, p. 2–17, 2013. Citado na página 28.
- MENESES, P. R.; ALMEIDA, T. d. **Introdução ao processamento de imagens de sensoriamento remoto**. [S.l.]: Universidade de Brasília, Brasília, 2012. Citado 2 vezes nas páginas 13 e 14.
- NETA, S. A.; DUTRA, L. V.; ERTHAL, G. J. Limiarização automática em histogramas multimodais. In: **Proceedings of the 7th Brazilian Conference on Dynamics, Control and Applications**. [S.l.: s.n.], 2008. Citado na página 10.
- OLIVEIRA, H. **Deteção de Bordas em Imagens Digitais**. figshare, 2014. Disponível em: <https://figshare.com/articles/Detec_o_de_Bordas_em_Imagens_Digitais/1053117/1>. Citado 2 vezes nas páginas 11 e 12.
- OTSU, N. A threshold selection method from gray-level histograms. **Transactions on systems, man, and cybernetics**, IEEE, v. 9, n. 1, p. 62–66, 1979. Citado 2 vezes nas páginas e 40.
- PEDRINI, H.; SCHWARTZ, W. R. **Análise de imagens digitais: princípios, algoritmos e aplicações**. [S.l.]: Thomson Learning, 2008. Citado 4 vezes nas páginas 15, 16, 18 e 20.
- RIBEIRO, B. **Suavização de Imagens**. 2006. Acesso em: 10 de fevereiro 2023. Disponível em: <<http://profs.ic.uff.br/~aconci/suavizacao.pdf>>. Citado na página 9.
- ROSIN, P. L. **Mathematical morphology in image processing**. [S.l.]: Springer-Verlag, 2014. Citado na página 17.
- ROUSHDY, M. Comparative study of edge detection algorithms applying on the grayscale noisy image using morphological filter. **International Journal of Graphics, Vision and Image Processing (GVIP)**, v. 6, n. 4, p. 17–23, 2006. Citado 2 vezes nas páginas 28 e 29.
- SANCHES, C. H. et al. Técnicas de suavização de imagens e eliminação de ruídos. In: **Anais do EATI-Encontro Anual de Tecnologia da Informação**. [S.l.: s.n.], 2015. p. 21–30. Citado na página 8.
- SCHNEIDER, G. A. **Segmentação e extração das características de defeitos em imagens radiográficas de juntas soldadas e tubulações**. Dissertação (Mestrado) — Centro Federal de Educação Tecnológica do Paraná, 2005. Citado 2 vezes nas páginas 5 e 6.
- SEARA, D. M. **Algoritmos para detecção de bordas**. 1998. Acesso em: 10 de fevereiro 2023. Disponível em: <<https://www.inf.ufsc.br/~aldo.vw/visao/1998/seara/index.html>>. Citado na página 8.

- SERRA, J. **Image analysis and mathematical morphology**. [S.l.]: Academic press, 1982. Citado 2 vezes nas páginas 15 e 22.
- SERRA, J.; SOILLE, P. **Mathematical morphology and image analysis**. [S.l.]: Springer-Verlag, 1994. Citado 5 vezes nas páginas 9, 16, 17, 18 e 20.
- SOHN, H.-J. et al. Efficient liquid media for encystation of pathogenic free-living amoebae. **The Korean journal of parasitology**, Korean Society for Parasitology, v. 55, n. 3, p. 233, 2017. Citado na página 34.
- SOUZA, T.; CORREIA, S. Estudo de técnicas de realce de imagens digitais e suas aplicações. In: **II Congresso de Pesquisa e Inovação da Rede Norte Nordeste de Educação Tecnológica**. [S.l.: s.n.], 2007. Citado na página 10.
- STATELLA, T.; SILVA, E. A. da. Shadows and clouds detection in high resolution images using mathematical morphology. In: **Pecora 17-The Future of Land Imaging**. [S.l.: s.n.], 2008. p. 18–20. Citado na página 28.
- WANG, X.; LIU, L.; WANG, Y. A novel erosion-based image simplification algorithm for remote sensing images. **Transactions on Geoscience and Remote Sensing**, IEEE, v. 60, n. 5, p. 4284–4296, 2022. Citado na página 17.
- XIANGANG, J. **Digital image pattern recognition engineering software design**. [S.l.]: Chian WaterPower Press, Beijing, 2008. Citado na página 14.