

**UNIVERSIDADE DO ESTADO DO AMAZONAS  
ESCOLA SUPERIOR DE TECNOLOGIA**

**JOÃO VITOR DA SILVA CAMPOS**

**DESENVOLVIMENTO DE UM SOFTWARE PARA AUXILIAR O  
APRENDIZADO DE CONCEITOS E APLICAÇÕES ABORDADOS NA  
DISCIPLINA DE PROCESSAMENTO DIGITAL DE IMAGENS.**

Manaus  
2023

**JOÃO VITOR DA SILVA CAMPOS**

**DESENVOLVIMENTO DE UM SOFTWARE PARA AUXILIAR O  
APRENDIZADO DE CONCEITOS E APLICAÇÕES ABORDADOS NA  
DISCIPLINA DE PROCESSAMENTO DIGITAL DE IMAGENS.**

Projeto de pesquisa desenvolvido durante a disciplina de Trabalho de Conclusão de Curso II e apresentada à banca avaliadora do Curso de Engenharia Elétrica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para obtenção do título de Engenheiro Eletricista.

Orientador: Jozias Parente de Oliveira, Dr.

Manaus

2023

**Universidade do Estado do Amazonas – UEA**  
**Escola Superior de Tecnologia - EST**

*Reitor:*

**André Luiz Nunes Zogahib**

*Vice-Reitor:*

**Kátia do Nascimento Coureiro**

*Diretora da Escola Superior de Tecnologia:*

**Ingrid Sammyne Gadelha Figueiredo**

*Coordenador do Curso de Engenharia Elétrica*

**Israel Gondres Torné**

*Banca Avaliadora composta por: Data da defesa: 31/08/2023.*

**Prof. Jozias Parente de Oliveira (Orientador)**

**Prof. Ingrid Sammyne Gadelha Figueiredo**

**Prof. Antônio Luiz Alencar Pantoja**

## **CIP – Catalogação na Publicação**

Campos, João Vitor da Silva

Desenvolvimento de um software para auxiliar o aprendizado de conceitos e aplicações abordados na disciplina de processamento digital de imagens. / João Vitor da Silva Campos; [orientado por] Jozias Parente de Oliveira. – Manaus: 2023. 61 p.: il.

Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica). Universidade do Estado do Amazonas, 2023.

1. PDI. 2. Software de Aprendizado. 3. Desenvolvimento.
- I. Oliveira, Jozias Parente.

JOÃO VITOR DA SILVA CAMPOS

DESENVOLVIMENTO DE UM SOFTWARE PARA AUXILIAR O APRENDIZADO DE  
CONCEITOS E APLICAÇÕES ABORDADOS NA DISCIPLINA DE PROCESSAMENTO  
DIGITAL DE IMAGENS

Pesquisa desenvolvida durante a disciplina de Trabalho de Conclusão de Curso II e apresentada à banca avaliadora do Curso de Engenharia Elétrica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para a obtenção do título de Engenheiro Eletricista

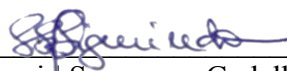
Nota obtida: 10,0 ( Dez )

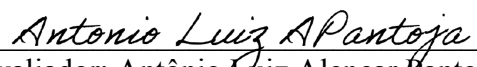
Aprovada em 31 / 08 / 2023.

Área de concentração: Processamento de Sinais

BANCA EXAMINADORA

  
Orientador: Joazias Parente de Oliveira, Dr.

  
Avaliador: Ingrid Sammyne Gadelha Figueiredo, Ma.

  
Avaliador: Antônio Luiz Alencar Pantoja

Manaus 2023

## DEDICATÓRIA

*Á toda minha família que ao longo da minha vida sempre me inspiraram com suas próprias histórias de vida, em especial a minha mãe e meu pai que sempre me incentivaram ao estudo e realização dos meus sonhos. Dedico a vocês como demonstração da minha gratidão.*

## AGRADECIMENTOS

A minha mãe que sempre esteve ao meu lado em todos os momentos me apoiando e incentivando em diversas situações, e eu tenho imensa gratidão por tudo, ao meu pai que me sempre me incentivou aos estudos e nunca deixou de medir esforços para que isso acontecesse, a vocês serei eternamente grato pelos ensinamentos, educação e bons momentos que vocês me proporcionaram.

Aos meus irmãos mais velhos que me inspiram com suas histórias e conquistas, pois cada um deles fizeram e continuarão fazendo uma grande diferença na minha vida. A minha namorada e grande companheira Sabrina que esteve ao meu lado nesses anos e que sempre me ajuda e apoia nos momentos mais importantes da minha vida.

Aos meus amigos da faculdade que durante esses anos difíceis deixaram tudo mais leve, vou guardar pra sempre cada momento compartilhado com vocês, em especial aos meus amigos do trio elétrico Gabriela Lima e Lucas Bayma que me acompanharam desde os primeiros momentos na faculdade e foram as pessoas mais inteligentes, confiáveis e divertidas na qual eu tive o prazer de ter conhecido, e ao meu grande amigo Diego Lucena que sempre acreditou mais em mim do que eu acreditava, e muito das minhas conquistas eu devo a ele.

## RESUMO

O trabalho apresenta o desenvolvimento de um software de aprendizado de conceitos da disciplina de processamento digital de imagens, visando auxiliar professores que ministram esses assuntos e exemplificar de forma prática as técnicas ensinadas dentro de sala de aula facilitando e contribuindo com o entendimento por parte do aluno. As técnicas desenvolvidas na ferramenta são as de limiarização onde a imagem é binarizada a partir de um valor de intensidade de pixel, alteração do brilho e contraste na imagem com visualização de seus respectivos histogramas, aplicações de filtros passa baixa com filtros gaussiano e média que permitem suavizar a imagem reduzindo ruídos como sal e pimenta, aplicações de filtros passa altas com detecção de bordas pelos operadores Canny e Sobel, e por fim a visualização dos espaços de cores RGB e YCbCr e em escala de cinzas. Para validar os resultados e o funcionamento da ferramenta uma pesquisa de satisfação foi realizada com alunos que cursam ou cursaram a matéria de PDI utilizando o cálculo de ranking médio.

**Palavras-chave:** PDI, Software de aprendizado, Desenvolvimento.

## ABSTRACT

The work presents the development of a software for learning concepts of the discipline of digital image processing, bringing in assistant professors who teach these subjects and exemplifying in a practical way the techniques taught in the classroom, facilitating and contributing to the understanding by the student. . The techniques developed in the tool are the limits of where the image is binarized from a pixel intensity value, changing the brightness and contrast in the image with visualization of their respective histograms, applications of low pass filters with Gaussian and average filters that allow smoothing the image with noise like salt and pepper, applying high pass filters with edge detection by the Canny and Sobel operators, and finally the visualization of RGB and YCbCr color spaces and in grayscale. To validate the results and the functioning of the tool, a satisfaction survey was carried out with students who attended or attended the PDI subject using the average ranking calculation.

**Keywords:** PDI, Learning software, Development.



## Lista de Figuras

1	Diagrama de etapas seguidas em um processamento digital de imagens . . .	19
2	Exemplo resolução temporal . . . . .	21
3	Exemplo resolução espacial . . . . .	21
4	Espaço de cores . . . . .	22
5	Exemplo de aumento e diminuição do nível de brilho em uma imagem. a) Imagem de entrada. b) imagem de saída com brilho alto. c) imagem de saída com brilho baixo . . . . .	23
6	Exemplo de aumento e diminuição do nível de contraste em uma imagem. a) Imagem de entrada. b) imagem de saída com contraste alto. c) imagem de saída com contraste baixo . . . . .	24
7	Exemplo de um histograma do contraste de uma imagem . . . . .	24
8	Exemplo de máscara utilizada para filtro passa-baixa . . . . .	25
9	Exemplo de filtro passa-baixa. a) imagem original. b) imagem filtrada utilizando filtro 3x3 . . . . .	26
10	Exemplo de uma máscara para filtros passa-altas] . . . . .	27
11	Exemplo aplicação de um filtro passa-altas. a) imagem original. b) imagem filtrada utilizando filtro 3x3 de exemplo . . . . .	27
12	Filtro de média 5x5 . . . . .	28
13	Curva gaussiana . . . . .	29
14	Exemplo de um filtro gaussino. a) imagem original. b) filtro gaussino com desvio padrão = 1, máscara 3x3 . . . . .	29
15	Exemplo de limiarização. a) imagem de entrada original. b) e c) resultados da aplicação de uma limiarização . . . . .	30
16	Detecção de bordas utilizando Sobel. a) imagem de entrada e b) imagem de saída . . . . .	32
17	Detecção de bordas utilizando Canny. a) imagem de entrada e b) imagem de saída . . . . .	33
18	Exemplo de uma aplicação Streamlit . . . . .	35
19	Layout da tela de início do software App PDI . . . . .	39
20	Fluxo de funcionamento App PDI . . . . .	41
21	Filtro de média aplicando Kernel 3x3 . . . . .	42
22	Filtro de Gaussiano aplicando Kernel 5x5 . . . . .	42
23	Filtro de Limiarização aplicado com valor 180 . . . . .	43
24	Aumento do brilho da imagem . . . . .	44
25	Aumento do contraste da imagem . . . . .	45
26	Diminuição do contraste da imagem . . . . .	46
27	Detecção de bordas pelo operador Sobel . . . . .	47

28	Detecção de bordas pelo operador Canny . . . . .	48
29	Exemplo da função visualizar RGB . . . . .	49
30	Exemplo da função visualizar YCbCr . . . . .	50
31	Tela inicial do PDI App . . . . .	54
32	Menu de aquisição de imagens . . . . .	54
33	Menu de processamentos . . . . .	55
34	Menu de upload de imagem . . . . .	56
35	Exemplo de resultado do upload de imagem . . . . .	56
36	Exemplo de resultado pela webcam . . . . .	57

## Lista de Tabelas

1	Tabela de Ranking Médio calculado . . . . .	51
2	Tabela de resultados I . . . . .	52
3	Tabela de resultados II . . . . .	52

## LISTA DE SIGLAS E ABREVIATURAS

**PDI** Processamento Digital de Imagens

**OpenCV** *Open Source Computer Vision Library*

**RGB** (*Red, Green, Blue*)

**YCbCr** Luminância, Crominância Azul, Crominância Vermelha

**RPC** Acrônimo de Remote Procedure Call

**RM** Ranking Médio

**MP** Média Ponderada

**$f_i$**  Frequência observada de cada resposta para cada item

**$V_i$**  Valor ou peso de cada resposta

**NS** Número de sujeitos ou total de respostas

**CSS** *Cascading Style Sheet*

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>15</b>
1.1	JUSTIFICATIVA	16
1.2	OBJETIVO	16
1.2.1	Geral	16
1.2.2	Específicos	16
1.3	RESUMO DOS CAPÍTULOS	17
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>18</b>
2.1	O PROCESSAMENTO DIGITAL DE IMAGENS	18
2.1.1	Etapas do processamento digital de imagens	18
2.1.2	Resolução de imagens	20
2.1.3	Espaços de cores	21
2.2	BRILHO E CONTRASTE	23
2.3	FILTROS ESPACIAIS	24
2.3.1	Filtro passa-baixa	24
2.3.2	Filtro passa-alta	26
2.4	SUAVIZAÇÃO DE IMAGENS	27
2.4.1	Filtro de média	27
2.4.2	Filtro Gaussiano	28
2.5	LIMIARIZAÇÃO	29
2.6	DETECÇÃO DE BORDAS	30
2.6.1	Sobel	30
2.6.2	Canny	32
2.7	LINGUAGEM PYTHON	33
2.8	OPENCV	34
2.9	STREAMLIT	34
2.10	FERRAMENTAS COMPUTACIONAIS PARA APRENDIZADO	35
2.10.1	EasYmage	35
2.10.2	Pixelation	36
2.10.3	Visnode	36
<b>3</b>	<b>MATERIAIS E MÉTODOS</b>	<b>37</b>
3.1	AMBIENTE DE DESENVOLVIMENTO	37
3.2	MÉTODOS	37
3.2.1	Definição dos requisitos	37
3.2.2	Definição da interface gráfica de usuário	37

3.2.3	Desenvolvimento da ferramenta . . . . .	37
3.2.4	Validação e testes da ferramenta . . . . .	38
3.2.5	Pesquisa de aceitação . . . . .	38
<b>4</b>	<b>DESENVOLVIMENTO DO SOFTWARE . . . . .</b>	<b>39</b>
4.1	ARQUITETURA E FUNCIONALIDADE DO SOFTWARE . . . . .	39
4.1.1	Requisitos funcionais e não funcionais . . . . .	39
4.1.2	Fluxo de funcionalidade . . . . .	40
4.2	AQUISIÇÃO DE IMAGENS ESTÁTICAS E DINÂMICAS . . . . .	41
4.3	TÉCNICAS DE PROCESSAMENTO DE IMAGEM . . . . .	41
4.3.1	Filtro Gaussiano e Filtro de Média . . . . .	41
4.3.2	Limiarização . . . . .	43
4.3.3	Brilho e Contraste . . . . .	43
4.3.4	Operador Sobel . . . . .	46
4.3.5	Operador Canny . . . . .	47
4.3.6	Visualização dos Espaços de Cores . . . . .	48
<b>5</b>	<b>AVALIAÇÃO E RESULTADOS . . . . .</b>	<b>51</b>
5.1	PESQUISA DE SATISFAÇÃO . . . . .	51
5.2	COMPARAÇÃO DE LITERATURA . . . . .	52
5.3	APLICAÇÕES E PRÁTICAS DE USO . . . . .	53
5.3.1	Tutorial de uso . . . . .	53
<b>6</b>	<b>CONCLUSÕES . . . . .</b>	<b>58</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>60</b>

## 1 INTRODUÇÃO

Desenvolver novas metodologias de ensino sempre foi um desafio de anos de estudo e que abrange várias áreas do conhecimento, ministrar aulas, cursos e conquistar a atenção do ouvinte a ponto de esclarecer e ensinar novos assuntos são situações recorrentes em diversas instituições de ensino. Com o avanço tecnológico novas ferramentas surgem possibilitando que novas metodologias que envolvem o desenvolvimento de programas que auxiliam o aprendizado melhorando a qualidade do ensino.(ZEM-LOPES; PEDRO; ISOTANI, 2014)

Nos últimos anos diversas aplicações computacionais foram desenvolvidas onde possibilitam melhorar o entendimento em algumas etapas de aprendizado, tanto em conceitos e aplicações dos assuntos abordados, ferramentas como matlab, multisim, protheus, que permitem a simulação de circuitos e assim concatenar as ideias, bem como ferramentas são desenvolvidas nas próprias universidades através de projetos de pesquisas. Este trabalho tem como objetivo desenvolver um software de aprendizado que permita auxiliar o aluno no entendimento de temas e assuntos abordados na disciplina de processamento digital de imagens, permitindo que seja inserido imagens de modo que ele possa alterar alguns de seus parâmetros como brilho, contraste e visualizar a imagem de saída junto do seu histograma, para isso será utilizada a linguagem de programação python junto de algumas de suas bibliotecas como OpenCV, Matplotlib para visualização de gráficos e Numpy para manipulação de alguns parâmetros de dados.

Com o aprendizado baseado em software que permitem a visualização dos conceitos da disciplina, pretende-se facilitar o ensino e esclarecer o assunto, aumentando assim a reflexão do que foi aprendido em sala.

No contexto acadêmico, é muito comum que em determinadas situações ocorra uma certa dificuldade do discente entender assuntos relacionados ao seu curso, seja por conta da aquisição de novos desafios e responsabilidades que universidade propõe, sobrecarga de trabalho ocasionando falta de tempo e conseqüentemente perda do foco no horário de aula, ou até mesmo metodologias de ensino tradicionais. Portanto é possível notar que há uma necessidade de não só tentar despertar o interesse dos alunos no aprendizado dentro da uni verdade, mas também de melhorar algumas metodologias de ensino auxiliando assim o estudo e o ensino de assuntos abordados ao longo do curso. Portanto, observou-se a possibilidade de utilizar novas tecnologias de programação que permita auxiliar e melhorar a relação entre o aluno e professor, mais precisamente um software de ensino para exemplificar alguns dos assuntos abordados na disciplina de processamento digital de imagem.

## 1.1 JUSTIFICATIVA

A implementação de um software de ensino para a disciplina de processamento digital de imagens possibilita que diversas técnicas que são ensinadas nesta disciplina possam ser exemplificadas de forma prática, realizando aquisição de imagens que serão fornecidas pelos próprios usuários, proporcionando uma visão ampla do conteúdo, facilitando o aprendizado e trazendo novas metodologias de ensino. Através do uso de novas tecnologias despertar o interesse do discente pela disciplina, além de facilitar a forma com que o assunto é abordado dentro da sala de aula.

Para o desenvolvimento desse projeto foram utilizados os conhecimentos adquiridos durante a formação do curso de Engenharia Elétrica, principalmente das disciplinas de Processamento Digital de Imagens, Processamento Digital de Sinais, Sistemas Microprocessados, Linguagem de Programação.

## 1.2 OBJETIVO

### 1.2.1 Geral

Desenvolver um software com interfaces gráficas que realize demonstrações e aplicações de técnicas de processamento digital de imagens para auxiliar o aprendizado de conceitos abordados nesta disciplina. O software realiza a aquisição de imagens fornecendo exemplos que abrangem as suas transformações além de plotar o histograma da imagem para visualização do usuário. O software visa aprimorar metodologias de ensino, auxiliar o aprendizado e através de exemplos visuais possibilitar concatenar ideias relacionadas a disciplina de Processamento Digital de Imagens (PDI) aprimorando o pensamento criativo do aluno.

### 1.2.2 Específicos

- a) Desenvolver um software de processamento de imagens com interface amigável e funcionalidades que permitam a aquisição de imagens estáticas e dinâmicas.
- b) Implementar os algoritmos de limiarização, alteração de brilho e contraste, detecção de bordas usando os operadores Canny e Sobel, e filtros de suavização (média e gaussiano) no software.
- c) Plotar o histograma da imagem ao lado da imagem de saída para os processamentos de brilho e contraste.
- d) Integrar a capacidade de visualização das imagens nos espaços de cores RGB e Ycbcr no software.
- e) Criar uma interface de usuário intuitiva que permita aos usuários interagir facilmente com as funcionalidades do software, incluindo a configuração de parâmetros através



de controles deslizantes.

- f) Realizar testes de desempenho e realizar comparações com outros softwares de aprendizado de PDI.
- g) Realizar testes de validação para garantir que as funcionalidades do software atendam aos requisitos estabelecidos.

### **1.3 RESUMO DOS CAPÍTULOS**

No tópico 2 é descrito os conceitos que serão desenvolvidos na ferramenta e o referencial teórico utilizado como base deste trabalho além de apresentar trabalhos relacionados. No no tópico 3 é comentado sobre o ambiente de desenvolvimento e as metodologias aplicadas ao trabalho na parte de definição de requisitos, desenvolvimento da ferramenta e as métricas que foram utilizadas para validar o resultado obtido com o software. Com o tópico 4 será apresentada as tecnologias utilizadas pra sua viabilização, a arquitetura e funcionalidades do software além de exemplos dos resultados após a plicação dos processamentos. Por fim no tópico 5, será apresentado resultados de uma pesquisa de satisfação realizada com alunos que cursam ou cursaram a disciplina e como esses resultados contribuíram de forma positiva para o melhoramento da ferramenta, juntamente com uma discussão de como o software App PDI pode ser utilizado dentro da sala de aula para maior proveito da ferramenta.

## 2 REFERENCIAL TEÓRICO

Neste tópico, serão abordados os aspectos teóricos dos conceitos relacionados ao projeto. Primeiramente, será realizada uma introdução sobre o processamento digital de imagens e como se dão as suas devidas etapas de processamento. Em seguida, serão abordados conceitos introdutórios referentes as características de uma imagem digital, além de técnicas utilizadas em PDI, como filtros digitais, limiarização, detecção de bordas e por fim uma descrição das tecnologias que serão utilizadas para o seu desenvolvimento.

### 2.1 O PROCESSAMENTO DIGITAL DE IMAGENS

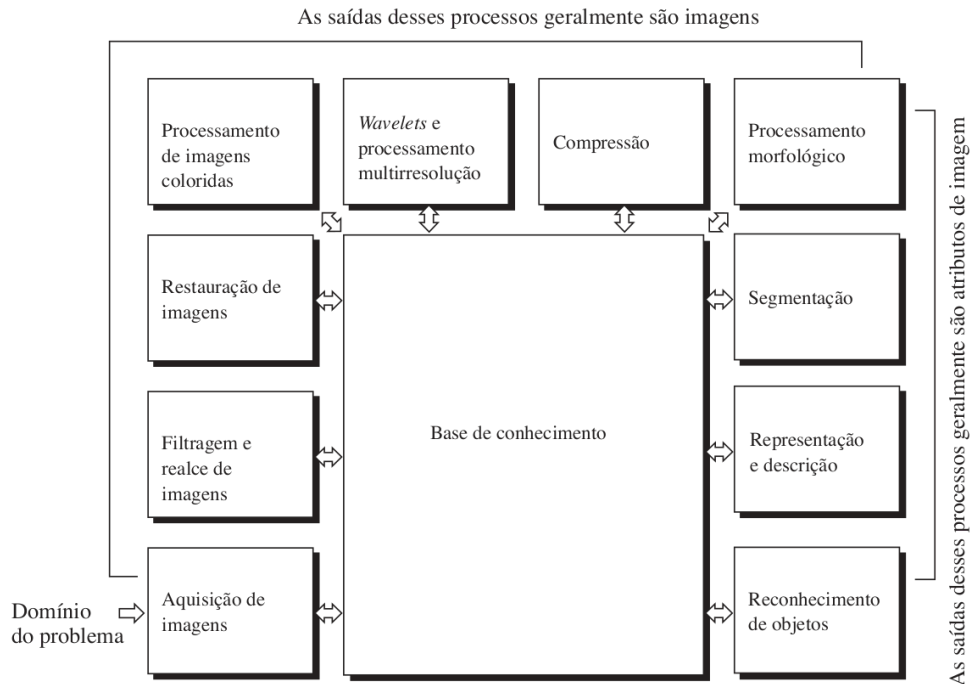
O processamento digital de imagens é uma área da ciência da computação que envolve o processamento de imagens digitais por meio de algoritmos e técnicas de processamento de sinais para melhorar a qualidade ou extrair informações úteis da imagem. O processo geralmente começa com a captura da imagem por meio de uma câmera digital ou de outro dispositivo de captura. Em seguida, a imagem é digitalizada e convertida em um formato que possa ser manipulado e processado por um computador.(GOMES, 2001)

O processamento de imagens pode incluir uma série de técnicas, como filtragem para reduzir o ruído, correção de cores para ajustar o equilíbrio de cores e melhorar a aparência geral da imagem, segmentação para identificar objetos na imagem, reconhecimento de padrões para detectar padrões específicos na imagem, entre outras técnicas. O processamento digital de imagens tem aplicações em uma ampla variedade de áreas, incluindo medicina, ciência forense, engenharia, automação industrial, entre outras. Por exemplo, ele pode ser usado para identificar tumores em imagens médicas, analisar imagens de vigilância para detectar atividades suspeitas, ou para processar imagens de satélite para monitorar condições meteorológicas e ambientais.(GOMES, 2001)

#### 2.1.1 Etapas do processamento digital de imagens

Para realizar o processamento digital de imagens algumas técnicas são aplicadas seguidas em uma determinada ordem para melhorar ou modificar imagens digitais, como correção de cores, remoção de ruído, realce de certos pixels e outros, por esse motivo algumas etapas são bem conhecidas e abrangem de forma geral em diversas aplicações de processamentos, como mostrado no diagrama da figura 1. A seguir é descrita cada uma das suas etapas. (GONZALEZ, 2010)

Figura 1 – Diagrama de etapas seguidas em um processamento digital de imagens



Fonte: (GONZALEZ, 2010)

- a) Aquisição de imagens: essa etapa consiste em realizar a captura de uma imagem, seja ela colorida ou em escala de cinzas, através de um equipamento, transformando-a para o domínio digital, em seguida realizará um redimensionamento em seu tamanho para ser utilizada nos processos seguintes.
- b) Filtragem e realce de imagens: nessa etapa são utilizadas técnicas de filtragem espaciais, ou no domínio da frequência para destacar ou suavizar determinadas regiões da imagem, visando aprimorar a imagem de modo a destacar regiões de interesse, como bordas, rosto, ou um objeto específico.
- c) Restauração de imagens: somente é utilizada caso a imagem apresente alguma degradação por conta de não ter sido capturada de forma adequada ou quando o próprio objeto da imagem possuir defeito físico, essa etapa se baseia em modelos matemáticos e probabilísticos para restabelecer características que haviam sido perdidas, como por exemplo a digitalização de textos de livros antigos.
- d) Processamento de imagens coloridas: Essa etapa utiliza a característica que uma imagem de cores possui para extrair um determinado objeto ou região de interesse, bem como converte-la para um formato mais adequando a outra aplicação, como uma escala de tons de cinza.
- e) Compressão: essa etapa lida com a proporção de capacidade necessária para processamento da imagem, onde possui dois parâmetros importantes para suas métricas, a taxa de compressão que seria a razão entre o tamanho final da imagem compri-

mida e o tamanho da imagem original, e a velocidade de processamento que indica o tempo em que a imagem leva para ser transmitida.

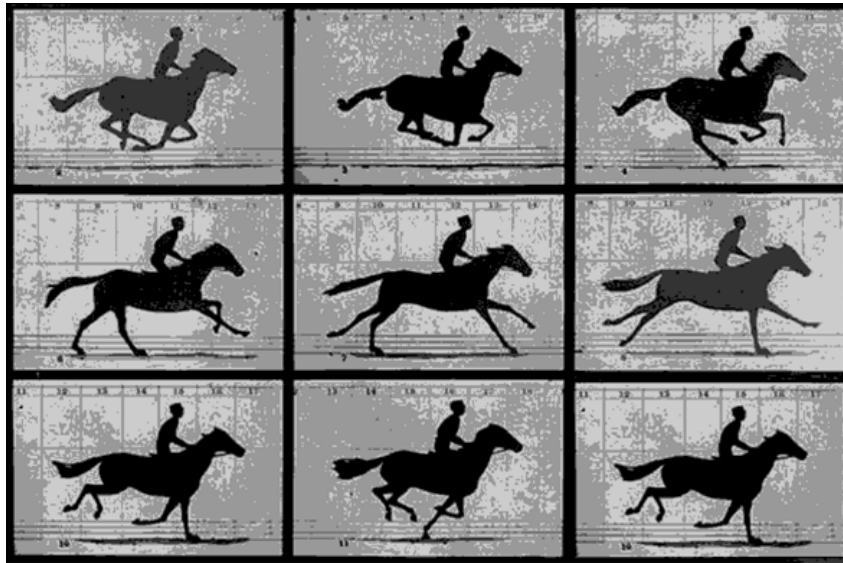
- f) Processamento morfológico: trabalha com ferramentas que extraem características da imagem úteis na descrição e representação de formas, como a simplificação de formas complexas em padrões mais geométricos e lineares, dois operadores básicos são utilizados na maior parte das técnicas de morfologia que são a erosão e a dilatação.
- g) Segmentação: divide a imagem como um todo nas partes que a constituem e que se diferenciam entre si, desse modo ela permite simplificar a busca por um determinado grupo de pixels da imagem.
- h) Representação e descrição: parte do resultado obtido na segmentação, correspondendo a fronteira de uma região ou dos pontos de dentro dela, e a descrição lida com a extração de atributos podendo assim distinguir objetos em suas respectivas regiões de interesse.
- i) Reconhecimento de objetos: o objetivo do reconhecimento é realizar, de forma automática, a “identificação” dos objetos segmentados na imagem. Existem duas etapas no processo de classificação de formas: o aprendizado utilizando técnicas de inteligência artificial e o reconhecimento do objeto a partir de suas características.

### 2.1.2 Resolução de imagens

A resolução de uma imagem digital é uma medida que indica o número de pontos, chamados de pixels, que compõem uma imagem numa tela digital, ela é uma característica importante pois afeta diretamente a qualidade e a nitidez de imagens virtuais, quanto mais pixels, maior é a resolução da imagens e mais detalhes podem ser exibidos. (MARTINS, 2014)

A resolução pode ser expressa em termos de largura pela altura. Existem diversas medidas de resolução de imagens, onde duas delas são classificadas como temporal e espacial. A temporal pode ser exemplificada como os vídeos, onde uma sequência de imagens é mostrada em determinado período de tempo que são determinados pela taxa de quadros em uma sequência de imagens, como mostrado na figura 2.

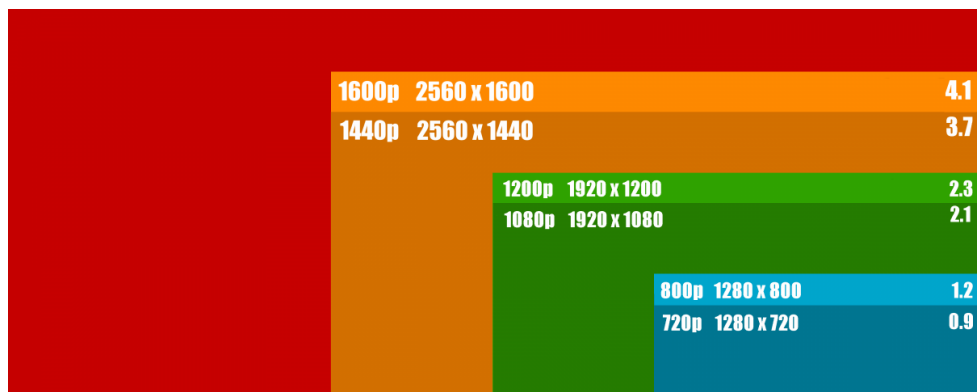
Figura 2 – Exemplo resolução temporal



Fonte: (RENE.E, 2023)

A resolução espacial é determinada pela quantidade de detalhes em um determinado momento, ou seja pelo número de pixels, quanto maior o número de pixels maior será a resolução espacial da imagem, figura 3.

Figura 3 – Exemplo resolução espacial



Fonte: (André Rodrigues, 2023)

### 2.1.3 Espaços de cores

Espaço de cores é um sistema utilizado para representar as cores em um modelo matemático. Existem diferentes espaços de cores, cada um com sua própria forma de descrever as cores.(FERREIRA; RODRIGUES, 2021)

Um dos espaços de cores mais comuns é o (*Red, Green, Blue*) (RGB), que significa em português, respectivamente, Vermelho, Verde e Azul. As cores nesse sistema são obtidas através das misturas das três cores primárias, em quantidades determinadas. Cada uma

das cores obtidas está enquadrada numa escala que varia de 0 a 255. (FERREIRA; RODRIGUES, 2021)

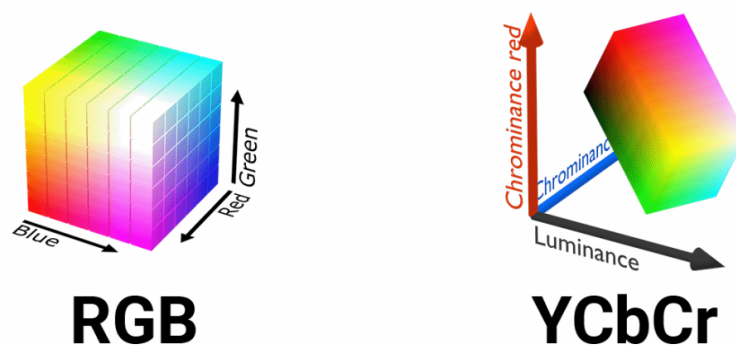
O RGB é um espaço de cores aditivo, que é comumente utilizado em dispositivos de exibição, como monitores de computador e televisores. Nesse espaço de cores, cada cor é formada pela combinação de diferentes intensidades de vermelho, verde e azul. Cada cor é representada por um valor numérico entre 0 e 255, que indica a intensidade de cada cor primária. (CORRÊA<sup>1</sup> et al., 2014)

É possível visualizar a estrutura do espaço de cor RGB como sendo um cubo, figura 4, onde as três cores mencionadas estão equidistantes entre si.

O espaço de cor formado pelas componentes Luminância, Crominância Azul, Crominância Vermelha (YCbCr) (COSTA; GUIMARÃES et al., 2021) é um espaço de cores subtrativo que também pode ser utilizado em aplicações de vídeo e transmissão digital de sinais. Nesse espaço de cores, a luminância (Y) é separada das informações de crominância (CbCr), que são divididas em duas componentes de cor: azul e vermelha respectivamente. O YCbCr é usado para comprimir sinais de vídeo, pois a separação da luminância e crominância permite que a cor seja representada com menos dados, sem perda significativa de qualidade.

Para o YCbCr a figura 4 mostra como os parâmetros de crominância azul, vermelha e a luminância são dispostas em eixos cartesianos formando o espaço de seu cor.

Figura 4 – Espaço de cores



Fonte: (ANDRÉI I., 2022)

É importante notar que, embora o RGB e o YCbCr sejam diferentes, eles são relacionados. A maioria dos dispositivos de exibição modernos convertem sinais YCbCr em RGB antes de exibir as imagens, e muitos formatos de vídeo comprimido (como o MPEG) usam o espaço de cores YCbCr como base.

## 2.2 BRILHO E CONTRASTE

O brilho é uma medida da quantidade de luz emitida ou refletida pela imagem e pode ser alterado por meio da adição ou subtração de uma constante em todos os valores de intensidade dos pixels da imagem. Aumentar o brilho de uma imagem resulta em uma imagem mais clara e diminuir o brilho resulta em uma imagem mais escura.(AGUIAR, 2020)

A intensidade de brilho de uma imagem se dá através de uma função bidimensional contínua  $f(x,y)$ , na qual  $x$  e  $y$  são coordenadas espaciais e o valor de  $f$  em qualquer ponto  $(x,y)$  é proporcional à intensidade luminosa aplicada nas imagens em nível de cinza.(AGUIAR, 2020). Na figura 5 é ilustrado um exemplo de aumento e diminuição do nível de brilho em uma imagem.

Figura 5 – Exemplo de aumento e diminuição do nível de brilho em uma imagem. a) Imagem de entrada. b) imagem de saída com brilho alto. c) imagem de saída com brilho baixo

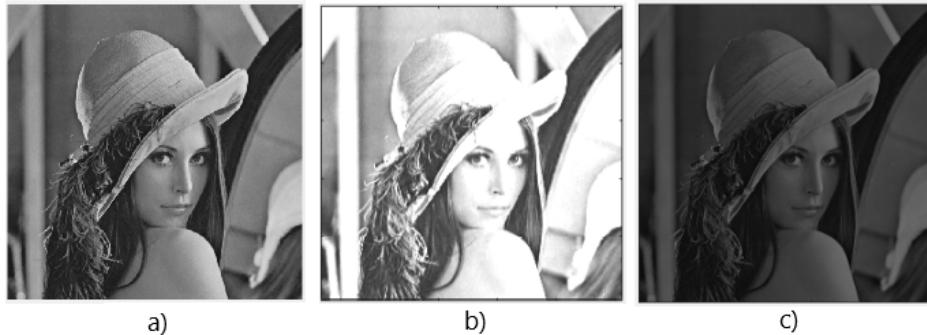


Fonte: Elaborado pelo autor

O contraste, por sua vez, é uma medida da diferença entre as intensidades mais claras e mais escuras da imagem. O contraste pode ser alterado pela expansão ou compressão da faixa de valores de intensidade da imagem. Aumentar o contraste de uma imagem aumenta a diferença entre as áreas claras e escuras da imagem, enquanto diminuir o contraste reduz a diferença.

O conceito de contraste da imagem é a diferença entre os níveis superiores e inferiores de intensidade presentes em uma imagem. Quando um número significativo de pixels em uma imagem possui uma alta faixa dinâmica, podemos esperar que a imagem tenha um alto contraste. Por outro lado, uma imagem com baixa faixa dinâmica normalmente tem uma aparência embaçada.(AGUIAR, 2020), Na figura 6 é ilustrado um exemplo de aumento e diminuição do nível de contraste em uma imagem.

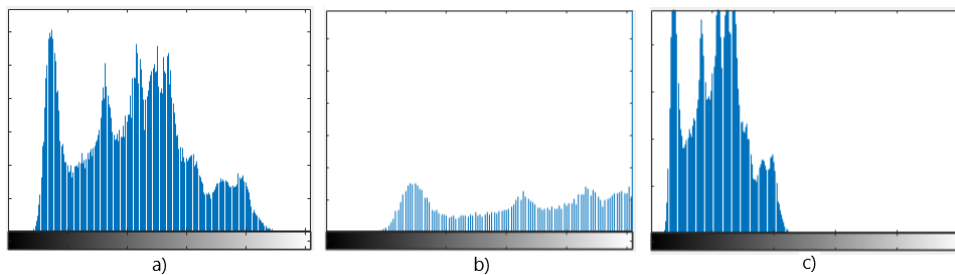
Figura 6 – Exemplo de aumento e diminuição do nível de contraste em uma imagem. a) Imagem de entrada. b) imagem de saída com contraste alto. c) imagem de saída com contraste baixo



Fonte: Elaborato pelo autor

A figura 7 demonstra o histograma das respectivas imagens da figura 6, neste gráfico é possível notar que quanto menor o contraste aplicado a imagem, o histograma diminui e se concentra em determinada região da escala de cinza, por outro lado, com o aumento do contraste, ele se espalha ao longo do eixo.

Figura 7 – Exemplo de um histograma do contraste de uma imagem



Fonte: Elaborado pelo autor

Existem várias técnicas para ajustar o brilho e o contraste da imagem no processamento digital de imagens. Alguns exemplos incluem transformação de intensidade linear, que ajusta a escala de cinza de uma imagem para um novo intervalo de valores e equalização de histograma, que redistribui as intensidades de pixel para melhorar o contraste local.

## 2.3 FILTROS ESPACIAIS

### 2.3.1 Filtro passa-baixa

Um filtro passa-baixas é um tipo de filtro que permite a passagem de frequências mais baixas e atenua ou elimina as frequências mais altas em um sinal de imagem. No



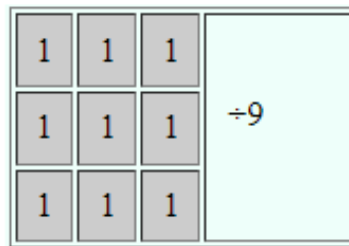
processamento digital de imagens, um filtro passa-baixas é usado para suavizar uma imagem, eliminando o ruído de alta frequência e realçando as áreas de baixa frequência, como os gradientes de cor e a textura.(FLORENCIO, 2009)

Segundo (Divisão de Processamento de Imagens, 2022) os filtros lineares passa-baixas, possuem as seguintes características:

- a) Eliminar informações de alta frequência, ou seja, extrair informações muito discrepantes (em geral ruídos) da imagem digital.
- b) Suavizar a informação da imagem: o resultado da aplicação desse filtro é a geração de uma imagem de saída mais homogênea, ou mais suavizada, em comparação com a imagem original.

A figura 8 apresenta um exemplo de máscara 3x3 utilizada na aplicação de filtros passa-baixas.

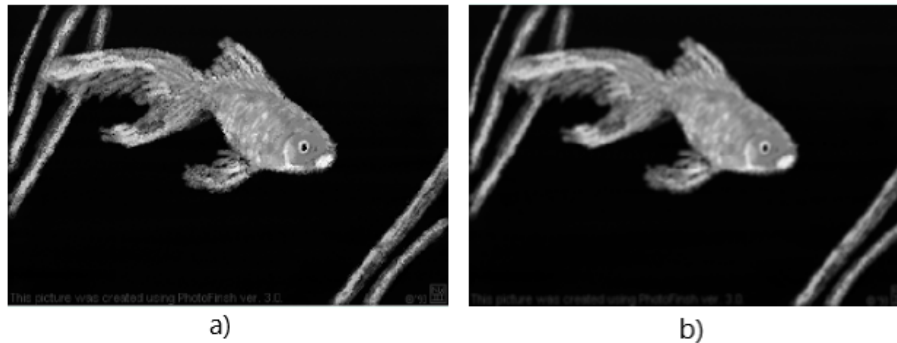
Figura 8 – Exemplo de máscara utilizada para filtro passa-baixa



Fonte: (Divisão de Processamento de Imagens, 2022)

O filtro passa-baixas pode ser utilizado em aplicações como redução no nível de ruído, suavizando bordas e realce de detalhes em imagens. No entanto, assim como o filtro passa-alta, é importante ter cuidado ao utilizá-lo, pois ele pode causar a perda de detalhes importantes da imagem dependendo do tamanho da máscara. A figura 9 apresenta um exemplo desse filtro.

Figura 9 – Exemplo de filtro passa-baixa. a) imagem original. b) imagem filtrada utilizando filtro 3x3



Fonte: (Divisão de Processamento de Imagens, 2022)

### 2.3.2 Filtro passa-alta

Um filtro passa-alta é um tipo de filtro que permite a passagem de frequências mais altas e atenua ou elimina as frequências mais baixas em um sinal de imagem. No processamento digital de imagens, um filtro passa-alta é usado para realçar os detalhes de alta frequência em uma imagem, como bordas, texturas e outros padrões finos. (FLORENCIO, 2009)

Segundo (Divisão de Processamento de Imagens, 2022) as características dos filtros passa altas são:

- a) Realçam a imagem ou alvos presentes na imagem digital. item Aumenta a nitidez das transições entre diferentes regiões de uma imagem digital.
- b) Podem funcionar como detectores de pontos isolados, de linhas e de bordas presentes na imagem.

Um filtro passa-alta é implementado como uma convolução entre a imagem original e um kernel ou máscara de filtro com coeficientes positivos e negativos. Os coeficientes positivos correspondem aos pixels da imagem que são amplificados pelo filtro, enquanto os coeficientes negativos correspondem aos pixels que são atenuados ou suprimidos, a figura 10 demonstra um exemplo de máscara utilizada no filtro passa-alta.

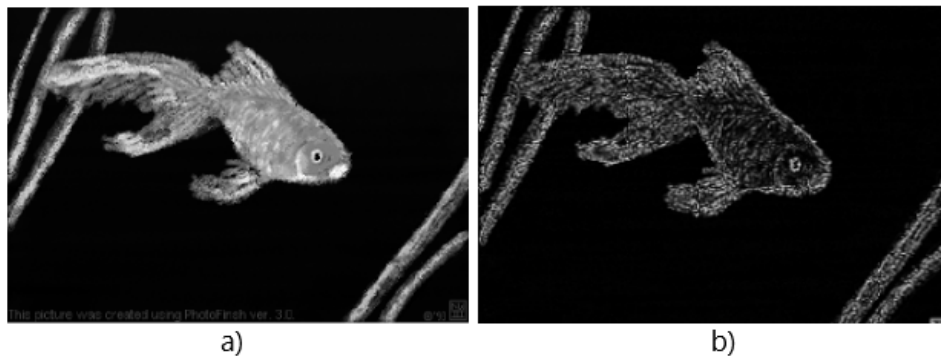
Figura 10 – Exemplo de uma máscara para filtros passa-altas]

-1	-1	-1
-1	8	-1
-1	-1	-1

Fonte: (Divisão de Processamento de Imagens, 2022)

Filtros passa-alta são comumente usados em aplicações como detecção de borda, remoção de ruído e aprimoramento de detalhes de imagem, como mostrado na figura 11.

Figura 11 – Exemplo aplicação de um filtro passa-altas. a) imagem original. b) imagem filtrada utilizando filtro 3x3 de exemplo



Fonte: (Divisão de Processamento de Imagens, 2022)

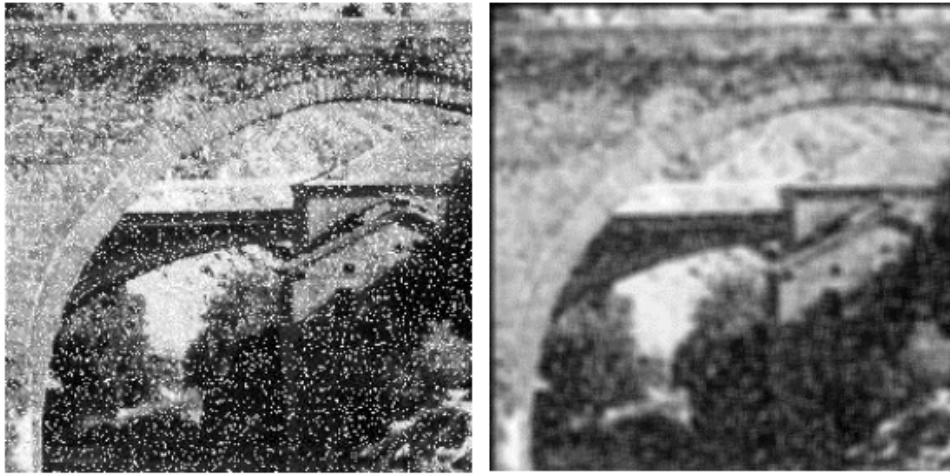
## 2.4 SUAVIZAÇÃO DE IMAGENS

### 2.4.1 Filtro de média

O filtro de média é do tipo passa-baixa que reduz a quantidade de variação de intensidade entre um pixel e seus vizinhos, eliminando ruídos. A ideia é substituir cada valor de pixel em uma imagem com o valor médio de seus vizinhos, incluindo ele mesmo, o que produz o efeito de eliminar os valores de pixels que são representativos de seus arredores. O filtro de Média é um filtro Não-Linear, sendo assim é baseado em torno de uma janela.(SANCHES et al., 2015)

Os filtros de média podem ser implementados calculando a média dos valores de pixel em uma janela de convolução, onde a saída do filtro é a média dos valores dos de pixel a pixel na janela, como mostrado na figura 9. O filtro de média pode efetivamente remover o ruído de baixa frequência e suavizar a imagem. Porém, também pode causar perda de detalhes finos, especialmente principalmente nas bordas da imagem. A figura 22 mostra um exemplo de como essa perda nos detalhes pode ocorrer.

Figura 12 – Filtro de média 5x5



Fonte: (BRUNO RIBEIRO, 2015)

Portanto, é importante escolher um tamanho de janela apropriado e ajustar os parâmetros do filtro para cada aplicação.

#### 2.4.2 Filtro Gaussiano

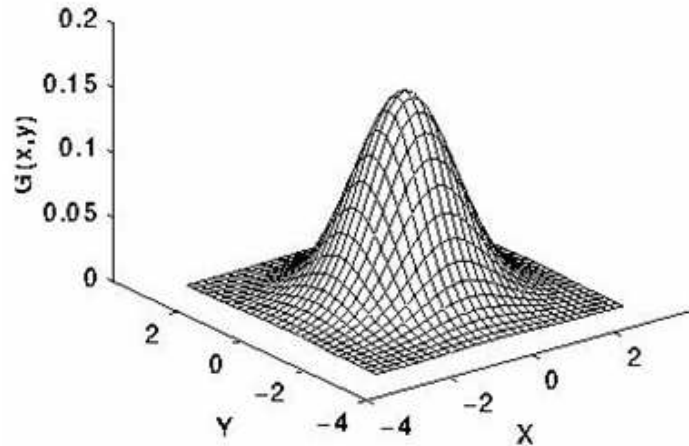
Um filtro gaussiano é utilizado para borrar ou desfocar a imagem na qual ele é aplicado com o objetivo de reduzir os ruídos. O resultado desta operação é a suavização da imagem, lembrando a visualização da mesma através de uma tela translúcida ou como se tivesse sendo vista através de uma lente fora de foco. A suavização gaussiana é largamente utilizada no estágio de pré-processamento da imagem a fim de enaltecer a estrutura da imagem em diferentes escalas.(JESUS; JR, 2015)

Matematicamente, a aplicação do filtro gaussiano é realizada da mesma forma que a convolução da imagem com uma função gaussiana, como um filtro passa baixa. Esta função gaussiana expressa a distribuição normal em estatística e é mostrada na equação 1, utilizada para definir a curva gaussiana de uma dimensão. Como uma imagem é definida em duas dimensões, a função gaussiana utilizada no filtro gaussiano aplicado a ela, também deve ser definido em duas dimensões.(JESUS; JR, 2015)

$$G(x, y) = G(x).G(y)^t = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} = 0 \quad (1)$$

Essa distribuição tem o formato mostrado na Figura 13, onde x é a distância da origem no eixo horizontal, y é a distância da origem no eixo vertical e o sigma é o desvio padrão da distribuição Gaussiana.

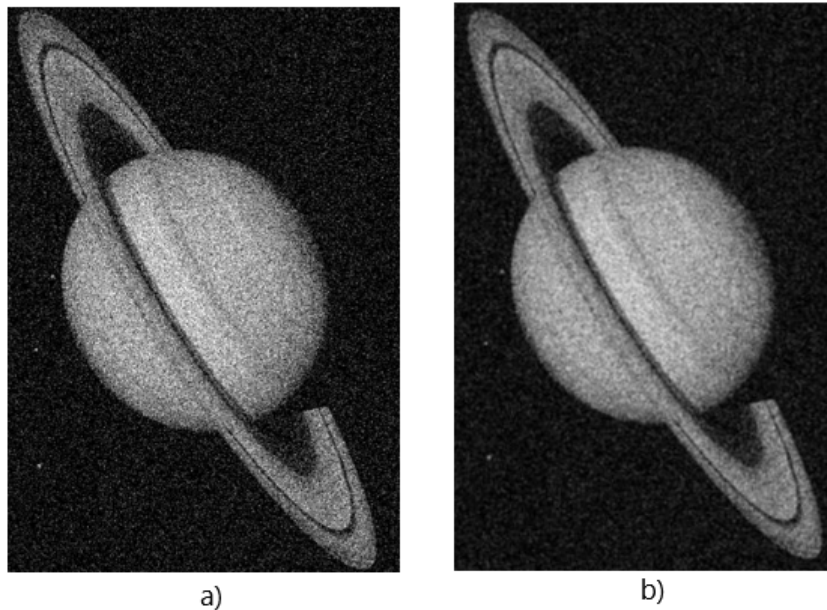
Figura 13 – Curva gaussiana



Fonte: (JESUS; JR, 2015)

A filtragem gaussiana geralmente é aplicada quando você deseja reduzir o ruído em uma imagem, conforme mostrado na Figura 14

Figura 14 – Exemplo de um filtro gaussiano. a) imagem original. b) filtro gaussiano com desvio padrão = 1, máscara 3x3



a)

b)

Fonte: (JESUS; JR, 2015)

## 2.5 LIMIAZIZAÇÃO

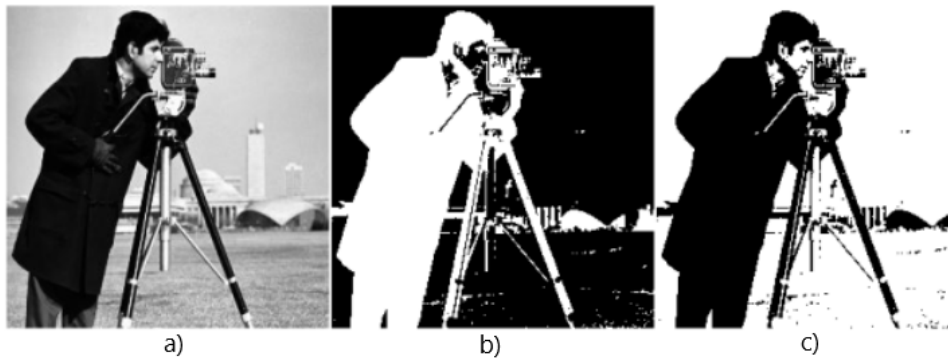
A limiarização de imagens é uma ferramenta de segmentação de baixa complexidade e computacionalmente rápida. É indicada para aplicações que exijam a identificação e extração do objeto da imagem, cujo tempo computacional despendido seja um fator relevante. Existem diferentes métodos de limiarização que podem ser usados, porém

todos eles envolvem a definição de um valor de limiar e a aplicação de uma regra para classificar os pixels da imagem com base nesse valor.(LOPES, 2003)

**Limiarização global:** Nesse método, um único valor de limiar é escolhido para toda a imagem. Os pixels com valores de intensidade acima do limiar são considerados como pertencentes à classe de objeto, enquanto os pixels abaixo do limiar são considerados como pertencentes à classe de fundo.

**Limiarização local:** Esse método envolve a escolha de valores de limiar diferentes para diferentes regiões da imagem. Isso é útil em imagens que apresentam variações de iluminação ou contraste em diferentes partes da imagem. Na figura 15 é ilustrado um exemplo de limiarização aplicada a uma imagem.

Figura 15 – Exemplo de limiarização. a) imagem de entrada original. b) e c) resultados da aplicação de uma limiarização



Fonte: (ANDRE C. MENDES; MAYCOL d. A., 2015)

A limiarização é uma técnica comum no processamento de imagens e é usada em muitas aplicações, tais como: segmentação de imagens como detecção de bordas, reconhecimento de objetos entre outros. No entanto, a escolha do método de limiarização e limiar adequado depende das características da imagem em questão e das necessidades específicas da aplicação.

## 2.6 DETECÇÃO DE BORDAS

### 2.6.1 Sobel

O operador de Sobel tem por objetivo calcular o gradiente da intensidade da imagem em cada ponto, fornecendo assim a direção da maior variação de claro para escuro e a magnitude de variação nessa direção. Assim, torna-se possível estimar a presença de uma transição claro para escuro e qual a orientação da transição. (GADIOLI; SILVA; NASCIMENTO, )

O gradiente de uma imagem é definido como mostrado na equação 2

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

É muito importante utilizar o gradiente pois ele aponta para a direção em que ocorre a mudança mais rápida de intensidade. As equações 3 e 4 mostram as máscaras aplicadas para cada um dos eixos x e y da imagem.

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * I \quad (3)$$

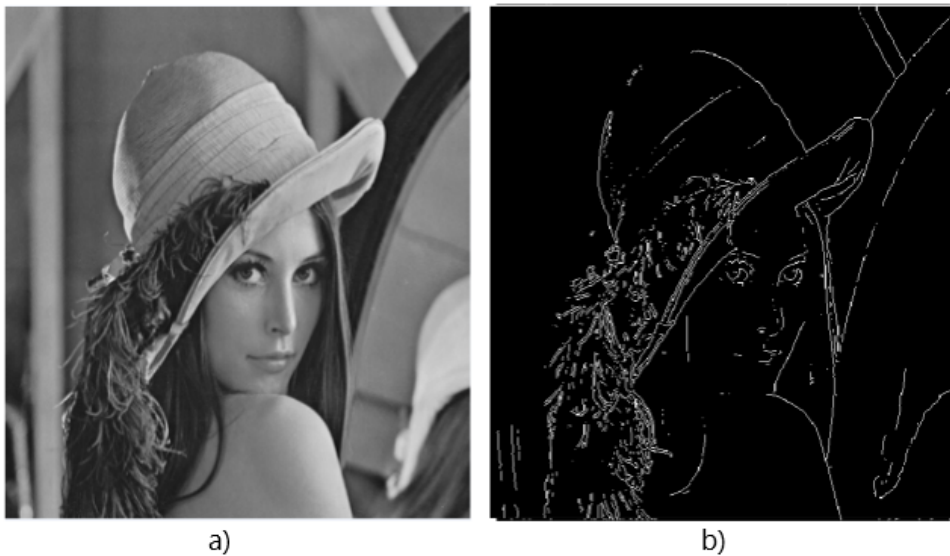
$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * I \quad (4)$$

O operador Sobel é aplicado nas direções x e y separadamente, gerando dois gradientes locais, um para a direção horizontal (x) e outro para a direção vertical (y). O gradiente resultante é então calculado como a raiz quadrada da soma dos quadrados dos gradientes parciais em ambas as direções, equação 5. (GADIOLI; SILVA; NASCIMENTO, )

$$G = \sqrt{G_x^2 + G_y^2} \quad (5)$$

A detecção de bordas de Sobel é amplamente utilizada em processamento de imagens e visão computacional para detecção de arestas em imagens médicas, reconhecimento de objetos, segmentação de imagens e outras aplicações. Na figura 16 é ilustrado um exemplo do operador Sobel aplicado em uma imagem.

Figura 16 – Detecção de bordas utilizando Sobel. a) imagem de entrada e b) imagem de saída



Fonte: Elaborado pelo autor

A técnica de detecção de bordas por Sobel é sensível ao ruído da imagem e pode levar a criação de bordas falsas. Nesse sentido, é recomendável aplicar técnicas de pré-processamento para reduzir o ruído antes de aplicar a detecção de borda.

### 2.6.2 Canny

A detecção de bordas por Canny é uma técnica de processamento de imagem que também realiza a detecção de arestas em uma imagem, porém é uma técnica que utiliza um algoritmo diferente do operador Sobel. O operador Canny pode ser considerada uma das técnicas mais precisas e populares para essa aplicação.

Como explicado no artigo (NETO; FILHO; NORTE, ) algoritmo do Canny foi projetado para possuir estas três propriedades:

- a) Minimização do erro na detecção. O algoritmo deve marcar maior número de bordas reais possíveis na imagem;
- b) Localização das bordas. A distância entre as bordas reais da imagem e as bordas encontradas pelo algoritmo devem ser minimizadas;
- c) Resposta mínima: uma borda dada na imagem deve ser marcada uma única vez, e uma imagem com ruído não deve criar bordas falsas.

Ao aplicar os conceitos de detecção de borda por Canny o resultado é uma imagem binarizada que destaca as bordas da imagem original.

O algoritmo de Canny funciona em 6 passos, são eles:



- a) Ler a imagem a ser processada e suaviza-la
- b) Realizar a busca por gradientes, onde ele intensifica regiões com maiores derivadas
- c) Encontrar a direção das bordas utilizando as mascaras
- d) Aproximar as direções para 0, 45, 90 ou 135 graus, convolucionando a imagem percorrendo as direções x e y
- e) Supressão de não máximos, onde somente os máximos locais são considerados como bordas
- f) Realizar a limiarização por histerese, onde as bordas finas são determinadas, sendo assim elas suprimem as bordas determinadas como não sendo fortes.

A detecção de bordas por Canny é útil em muitas aplicações de visão computacional, como detecção de objetos, reconhecimento de faces, segmentação de imagem e rastreamento de objetos em movimento. Na figura 17 é ilustrado um exemplo do operador Canny aplicado em uma imagem.

Figura 17 – Detecção de bordas utilizando Canny. a) imagem de entrada e b) imagem de saída



Fonte: Elaborado pelo autor

## 2.7 LINGUAGEM PYTHON

Python é uma linguagem de programação de alto nível, interpretada, orientada a objetos e com tipagem dinâmica. Foi criada no início dos anos 90 por Guido van Rossum e tem se tornado uma das linguagens mais populares e utilizadas no mundo da programação. Python é amplamente utilizado em diversas áreas, incluindo desenvolvimento web, ciência de dados, inteligência artificial, automação de tarefas, jogos, entre outros.

Alguns dos frameworks e bibliotecas mais populares em Python incluem Flask, Django, Pandas, NumPy, SciPy e TensorFlow.(PYTHON, 2021)

Devido à sua sintaxe simples e legível, Python é frequentemente considerada uma das melhores linguagens de programação para iniciantes. Além disso, a grande comunidade de usuários e desenvolvedores em todo o mundo significa que há uma enorme quantidade de recursos disponíveis para aprendizado e suporte.(DUQUE, 2011)

## 2.8 OPENCV

OpenCV (*Open Source Computer Vision Library*) é uma biblioteca de software de visão computacional e aprendizado de máquina de código aberto. O OpenCV foi desenvolvido para fornecer uma infraestrutura comum para aplicativos de visão computacional e para acelerar o uso da percepção da máquina nos produtos comerciais, os quais incluem um conjunto de algoritmos clássicos das áreas supracitadas (opencv.org, 2023), sendo estes utilizados para a detecção e reconhecimento de faces, identificação de objetos, classificação de ações humanas em vídeos, rastreamento de movimentos de objetos, dentre outros.

Segundo (GOLLAPUDI; GOLLAPUDI, 2019) a biblioteca OpenCV consiste em dois tipos de módulos, quais sejam módulos principais e adicionais. Os módulos principais do OpenCV vêm por padrão como versões embutidas. Eles formam módulos centrais porque eles fornecem as principais funcionalidades, como tarefas de processamento de imagens, filtragem, transformação e outras.

Os módulos extras não vêm por padrão com a distribuição OpenCV. Esses módulos são relacionados a funcionalidades adicionais de visão computacional como reconhecimento de texto.

## 2.9 STREAMLIT

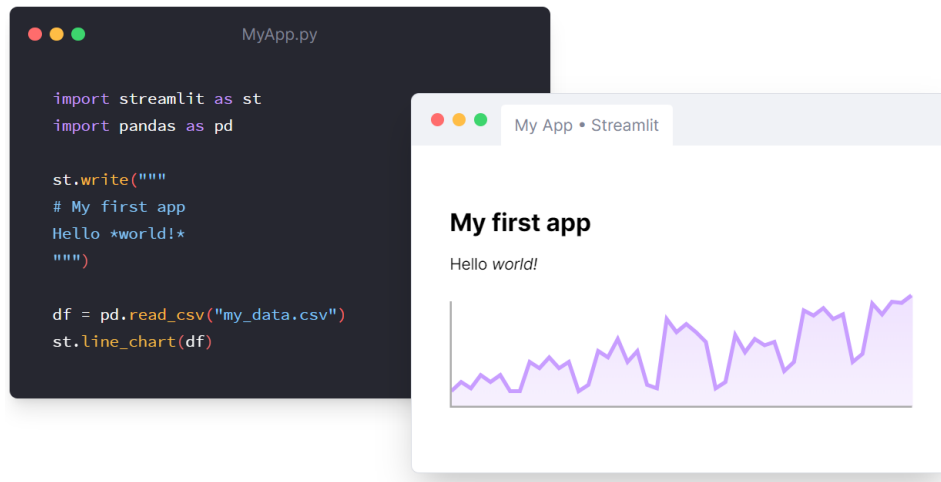
O Streamlit é a biblioteca da linguagem Python que permite utilizar a estrutura e linguagem python para o desenvolvimento web sem a necessidade de se utilizar as linguagens convencionais como JavaScript, *Cascading Style Sheet* (CSS) e marcação pelo Markdown, realizando essa adaptação através do da biblioteca React. O streamalit utiliza widgets ou 'ferramentas' que são funções prontas para diversas aplicações possíveis, e esses widgets são divididos nos seguintes tipos:

- a) Exibição de dados.
- b) Texto
- c) Gráficos
- d) Inputs
- e) Status / Progresso

- f) Mídia
- g) Layouts

Como uma biblioteca que permite facilitar o desenvolvimento web, ela possui compatibilidade com diversas outras bibliotecas python, como OpenCV, TensorFlow, Matplotlib e outros.

Figura 18 – Exemplo de uma aplicação Streamlit



Fonte: (SNOWFLAKE, 2023)

## 2.10 FERRAMENTAS COMPUTACIONAIS PARA APRENDIZADO

Mesmo com o esforço e a dedicação de educadores, estudantes frustram-se com disciplinas de PDI devido à grande complexidade (GARCIA; GUZMÁN-RAMÍREZ; PACHECO, 2015). Na maioria dos casos, estudantes descobrem que a teoria não é atrativa, difícil e pouco aplicável (ZIN; SALLEH; BAKRI, 2015). Segundo a proposta de (BEHAR.P.B., 1993), a produção de um software educativo deve “satisfazer as intenções do professor e as características dos estudantes; possibilitar vários estilos e tipos de aprendizagem e aproveitar as qualidades educativas que oferece o computador – em particular, a interatividade; e o controle do usuário sobre o que se aprende e como se aprende.”

### 2.10.1 EasYmage

O EasYmage é uma arquitetura para modelagem e simulação de técnicas utilizadas no ensino de PDI, sendo a modelagem baseada no uso de Acrônimo de Remote Procedure Call (RPC). Operações e cenários podem ser reconfigurados alterando parâmetros do simulador, sem necessidade de modificar a estrutura do modelo. O modelo RPC do EasYmage é organizado em uma hierarquia de sub-redes (ou páginas). Uma sub página é um modelo RPC, cuja semântica depende de outras páginas e/ou super páginas. Essa organização permite a adição de novas operações e, se necessário, a alteração das operações nativamente incorporadas ao modelo.

### 2.10.2 Pixelation

Um dos principais diferenciais da ferramenta em relação às outras, é que nela é possível aprender programação através da OpenCV, uma biblioteca para Visão Computacional amplamente utilizada, tanto no meio acadêmico, quanto no de mercado. As funções da ferramenta podem ser acessadas por meio da barra de menus disposta na janela principal. As funções mais utilizadas também podem ser acessadas por meio dos botões de um menu vertical localizado à esquerda da janela principal do software. (SILVA; CHAVES; AQUINO, 2012)

As abas do menu principal no canto superior da janela principal oferecem as opções: Arquivo, PDI, OpenCV, e Ajuda. O menu Arquivo deve possuir a função de selecionar as imagens que o usuário deseja aplicar algum código de PDI. Através desse menu também é possível sair do programa, encerrando a execução do software. O menu PDI abre dois submenus: Conceitos e Aplicações. Em Conceitos o usuário tem acesso a um menu de conteúdos teóricos sobre fundamentos de processamento digital de imagens, já em Aplicações o usuário encontra uma listagem de aplicações dos métodos de análise de PDI e pode acessar cada item da listagem para visualizar os detalhes sobre cada aplicação.(SILVA; CHAVES; AQUINO, 2012)

### 2.10.3 Visnode

É uma ferramenta de aprendizado para técnicas de PDI, tendo como objetivo dar assistência ao aluno no entendimento dos algoritmos, a ferramenta provê informações detalhadas destes, sendo possível visualizar uma explicação textual de seu funcionamento, além do código de sua implementação. Essa funcionalidade também pode servir como material didático, auxiliando na etapa de aprendizado de usuários com pouco conhecimento na área. Para a gamificação, o software conta com elementos como: Desafios, Missões, Narrativas, Recompensas e Rankings. Uma missão consiste em uma tarefa que o usuário deverá realizar. Cada missão possui uma narrativa, na qual deve contextualizar o problema visando motivar o usuário à resolução do mesmo.(BECKER; POHREN; BEZ, )

### 3 MATERIAIS E MÉTODOS

A seguir são abordados os aspectos metodológicos da pesquisa realizada, escrevendo-se os procedimentos necessários para desenvolver um software com interfaces gráficas que realize demonstrações de aplicações de conceitos de processamento digital de imagens. O trabalho em questão é uma pesquisa aplicada, e tem como objetivo a realização de pesquisa exploratória relacionada ao material bibliográfico e de laboratório. São utilizados os procedimentos técnicos de pesquisa bibliográfica e experimental, onde também é utilizado o método de abordagem hipotético-dedutivo e método de procedimento monográfico para realizar a sua elaboração. Em relação a coleta de dados, foi utilizada a documentação indireta e a análise e interpretação de dados, ocorreu de forma qualitativa de maneira global.

#### 3.1 AMBIENTE DE DESENVOLVIMENTO

O hardware utilizado nesta pesquisa tem a seguinte configuração: Processador Intel(R) Core(TM) i3-9100F, 3.60 GHz; RAM 16 GB; Sistema operacional 64 bits.

#### 3.2 MÉTODOS

As etapas da metodológicas que foram realizadas neste projeto são as seguintes: definição dos requisitos, definição da interface gráfica de usuário, desenvolvimento da ferramenta, validação e testes da ferramenta, pesquisa de aceitação.

##### 3.2.1 Definição dos requisitos

Nesta etapa foi definido todos os requisitos iniciais para o desenvolvimento do software como por exemplo: quais bibliotecas foram utilizadas para realizar a aquisição de imagens, quais técnicas de processamento de imagens serão exemplificadas pela ferramenta. Além disso, foi definido o que se espera para cada conceito demonstrado pela ferramenta em termos de imagem resultante, parâmetros de controle, entre outros aspectos.

##### 3.2.2 Definição da interface gráfica de usuário

Nesta etapa foi realizada a definição da estrutura da interface gráfica de usuário, qual seja: o formato do layout do software, menus, botões, controles deslizantes, histogramas entre outros, bem como essas funcionalidades irão interagir com as técnicas escolhidas.

##### 3.2.3 Desenvolvimento da ferramenta

Após definição dos requisitos e especificações técnicas, foi realizado o processo de desenvolvimento da ferramenta, utilizando técnicas de desenvolvimento ágil relacionada a programação orientada a objetos para garantir um melhor desempenho. Para realizar

a aquisição das imagens foi utilizado uma *web cam* que tem uma resolução máxima de 720p, e o ambiente de desenvolvimento será o *visual studio code* que possui realce de sintaxe, complementação inteligente de código, snippets e refatoração de código, bem como a utilização do controle de versionamento git.

### 3.2.4 Validação e testes da ferramenta

Nesta etapa se iniciou a fase de teste, onde todas as funções implementadas foram ser revisadas e corrigidas quando necessário. Esta fase foi essencial para evitar que a ferramenta fosse disponibilizada com algum erro e garantir que os resultados obtidos estejam coerentes com a teoria ensinada em sala de aula.

### 3.2.5 Pesquisa de aceitação

Nesta etapa foi realizada uma pesquisa com os alunos da disciplina de processamento digital de imagem em que a ferramenta desenvolvida foi disponibilizada aos alunos para coletar, por meio de um questionário, informações pertinentes sobre usabilidade e facilidade para aprendizagem. Para analisar os resultados será utilizado o cálculo do Ranking Médio (RM) proposto por (OLIVEIRA, 2005). Neste modelo atribui-se um valor de 1 a 5 para cada resposta a partir da qual é calculada a média ponderada para cada item, baseando-se na frequência das respostas. Desta forma será obtido o RM da seguinte maneira:

$$MP = \sum_{i=1}^n (f_i * V_i) \quad (6)$$

$$RM = \frac{MP}{NS} \quad (7)$$

Onde,

MP - Média Ponderada

$f_i$  - frequência observada de cada resposta para cada item

$V_i$  - valor ou peso de cada resposta

RM - Ranking Médio

NS - número de sujeitos ou total de respostas

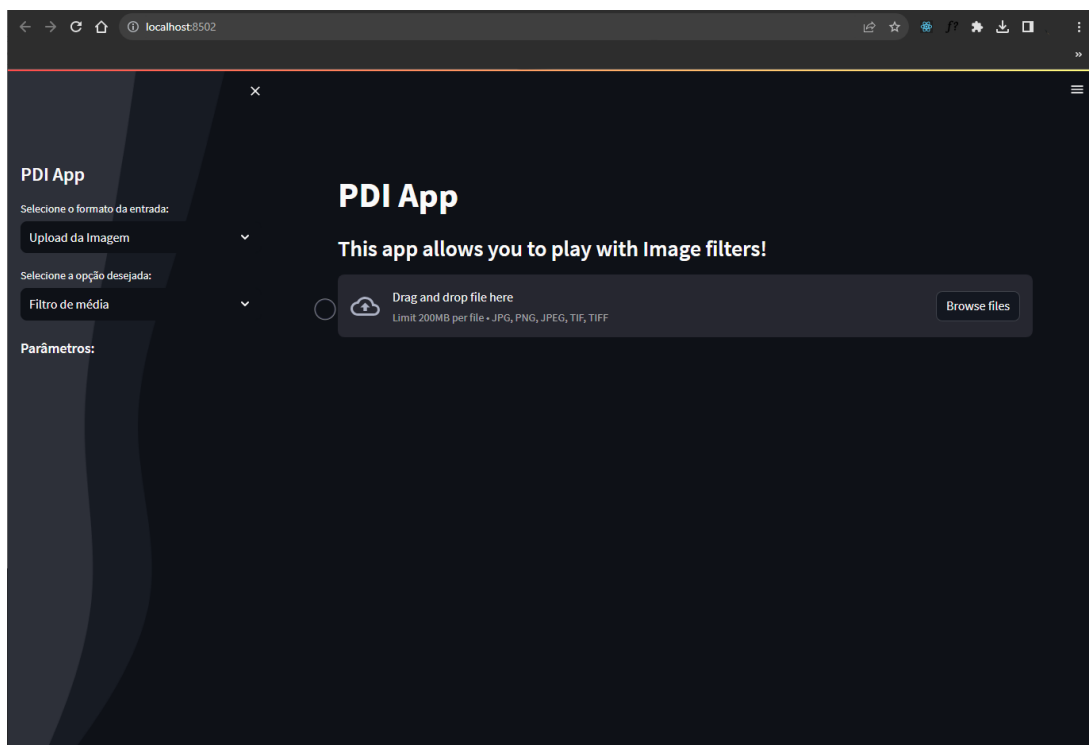
Quanto mais próximo de 5 o RM estiver maior será o nível de satisfação dos estudantes e quanto mais próximo de 1 menor será o nível de satisfação deles.

## 4 DESENVOLVIMENTO DO SOFTWARE

### 4.1 ARQUITETURA E FUNCIONALIDADE DO SOFTWARE

O software foi desenvolvido utilizando principalmente duas bibliotecas do python, o streamlit para criação da interface gráfica e o OpenCv para realizar o processamento das imagens. O software recebeu o nome de App PDI e visa possibilitar a sua utilização em sala de aula para o melhor entendimento de conceitos de PDI através de aplicações práticas. Para estruturar o desenvolvimento da ferramenta, foi definido requisitos funcionais e não funcionais para que não houvesse nenhum conflito entre os processamentos que pudessem gerar *bugs* ou falhas durante a execução do programa.

Figura 19 – Layout da tela de início do software App PDI



Fonte: Elaborado pelo autor

A figura 19 apresenta o *layout* da tela de início do programa, nela contém o menu lateral que possui as opções de aquisição de imagens, sendo elas por imagens estáticas ou dinâmicas, os processamentos disponíveis e por fim, quando uma imagem estiver, os controles deslizantes que alteram os parâmetros do processamento escolhido.

#### 4.1.1 Requisitos funcionais e não funcionais

Os requisitos funcionais e não funcionais da ferramenta foram baseados nos principais processamentos que podem ser estudados na disciplina de PDI integrada junto das diversas funcionalidades que as bibliotecas python permitem.

Requisitos funcionais:

- a) Realizar aquisição de imagens estáticas e dinâmicas (videos)
- b) Imagens estáticas nos formatos JPG, PNG, JPEG, TIF, TIFF
- c) Possibilitar o processamentos do filtro Gaussiano e filtro de média.
- d) Possibilitar aplicação de técnicas de limiarização e alteração das informações de brilho e contraste.
- e) Detecção de bordas pelos operadores de Canny e Sobel.
- f) Visualizar a imagem em escala de cinzas e os espaços de cores no formato RGB e YCbCr.
- g) O programa deve rodar como um executável.

Requisitos não funcionais:

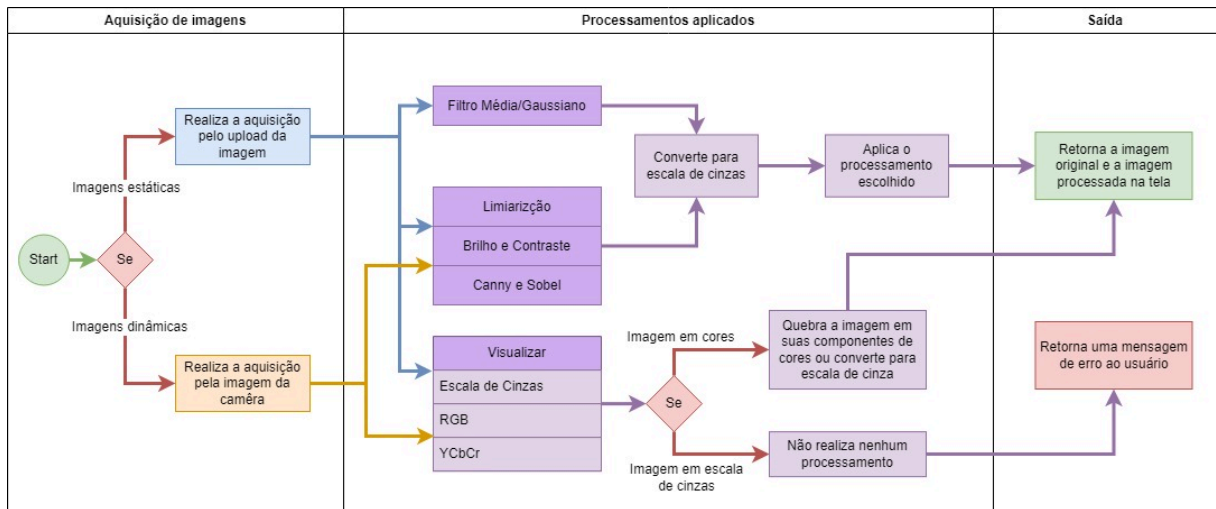
- a) Imagens que não estejam nos formatos JPG, PNG, JPEG, TIF, TIFF
- b) Os filtros gaussiano e de média não são implementados pra imagens dinâmicas
- c) Opção de visualizar não é disponível para imagens em escala de cinzas.

#### 4.1.2 Fluxo de funcionalidade

Para o desenvolvimento do software os requisitos funcionais e não funcionais são de extrema importância, a figura 20 demonstra o fluxo de funcionamento da ferramenta que foi baseado nos requisitos mencionados, deste modo é possível dividir o seu funcionamento em três momentos, sendo eles a aquisição de imagens, a aplicação dos processamentos e o retorno das informações de saída ao usuário. Primeiramente a ferramenta realiza a aquisição das imagens pelo seu *upload* ou pela câmera que esteve disponível, em ambos os casos quando se é escolhido qualquer opção que não seja referente a função de 'visualizar' as imagens são convertidas em escala de cinzas e o processamento selecionado é aplicado retornando a imagem original e o seu resultado, isso ocorre pois em sua grande maioria os processamentos ocorrem em escala de cinza e possibilita um melhor entendimento daquelas técnicas. quando a opção de visualizar é selecionada primeiramente a ferramenta verifica se a imagem de entrada está em escala de cinzas, pois caso ela esteja não essa opção se torna inutilizável e retorna uma mensagem de erro para o usuário, caso seja uma imagem em cores ela retorna o resultado conforme o espaço de cor selecionado.



Figura 20 – Fluxo de funcionamento App PDI



Fonte: Elaborado pelo autor

## 4.2 AQUISIÇÃO DE IMAGENS ESTÁTICAS E DINÂMICAS

O PDI App pode realizar a aquisição de imagens de duas formas, onde cada uma possui suas vantagens em visualização, sendo elas:

- Imagens estáticas: Onde para realizar a aquisição é utilizada o *widget* de *upload* da biblioteca *streamlit*, ele permite que imagens em terminados formados possam ser anexados na ferramenta e assim utilizadas para aplicar os processamentos disponíveis.
- Imagens dinâmicas ou video: Onde é aplicado uma função própria da biblioteca OpenCv que realizar a aquisição de imagens a partir de um *driver* de câmera que esteja instalado e disponível no computador, a função *VideoCapture* funciona de forma constante e pega as informações da câmera e aplica o processamento em cada *frame*.

## 4.3 TÉCNICAS DE PROCESSAMENTO DE IMAGEM

### 4.3.1 Filtro Gaussiano e Filtro de Média

Para a funcionalidade de filtro gaussiano e de média é utilizada, respectivamente as funções 'Blur' e 'GaussianBlur' do Opencv. A função 'Blur' se baseia na técnica explicada no tópico 2.4.1 onde se espera dois parâmetros de entrada, o primeiro sendo a imagem original, e o segundo sendo Kernel que representa o valor da dimensão da matriz que será utilizada para o processamento do filtro de média. A função 'GaussianBlur' também utiliza um kernel como parâmetro, porém dessa vez o kernel se comporta como na forma gaussiana explicada no tópico 2.4.2 além de possuir o parâmetro de sigmaX que permite

configurar o desvio padrão na direção do eixo X, porém esse parâmetro não será explorado neste software.

Figura 21 – Filtro de média aplicando Kernel 3x3



Fonte: Elaborado pelo autor

Figura 22 – Filtro de Gaussiano aplicando Kernel 5x5



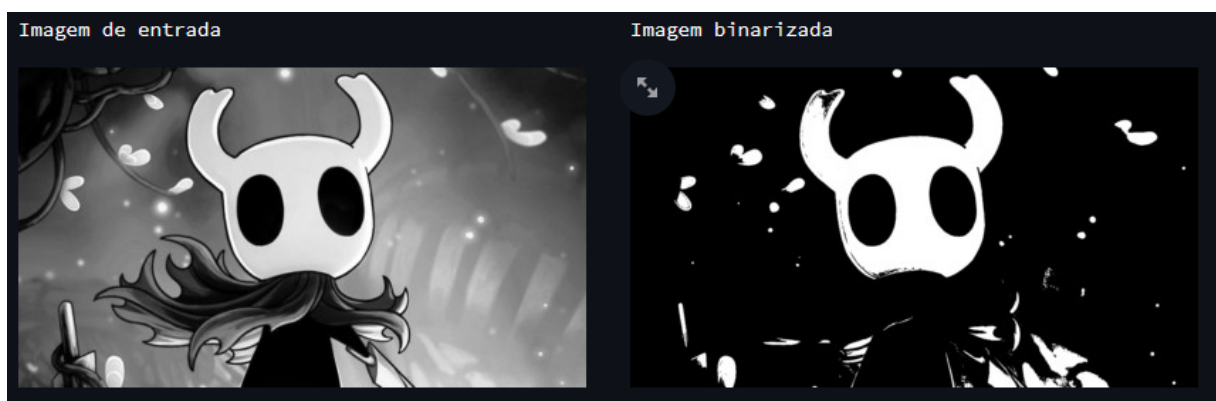
Fonte: Elaborado pelo autor

Cada um dos filtros utilizados possui um controle deslizante que permite modificar os valores de *kernel's* utilizado, possibilitando aumentar a matriz e assim suavizando ainda mais a imagem, para o filtro de média o controle deslizante possui o intervalo de 0 a 25 com incremento 1 e para o filtro gaussiano é de 0 a 25 porém somente com os valores ímpares.

### 4.3.2 Limiarização

A técnica de limiarização utiliza a função 'cv.threshold' com o parâmetro '*THRESH\_BINARY*' para transformar a imagem de entrada em uma imagem binarizada, de acordo com o valor de intensidade do pixel. Essa função recebe como parâmetros a imagem de entrada, o valor do nível que será utilizado como referencia para a limiarização, o valor de intensidade máxima da imagem (caso seja uma imagem em 8 bits 255) e o '*THRESH\_BINARY*', porém o software permite somente o ter o controle deslizante referente ao valor de referência.

Figura 23 – Filtro de Limiarização aplicado com valor 180



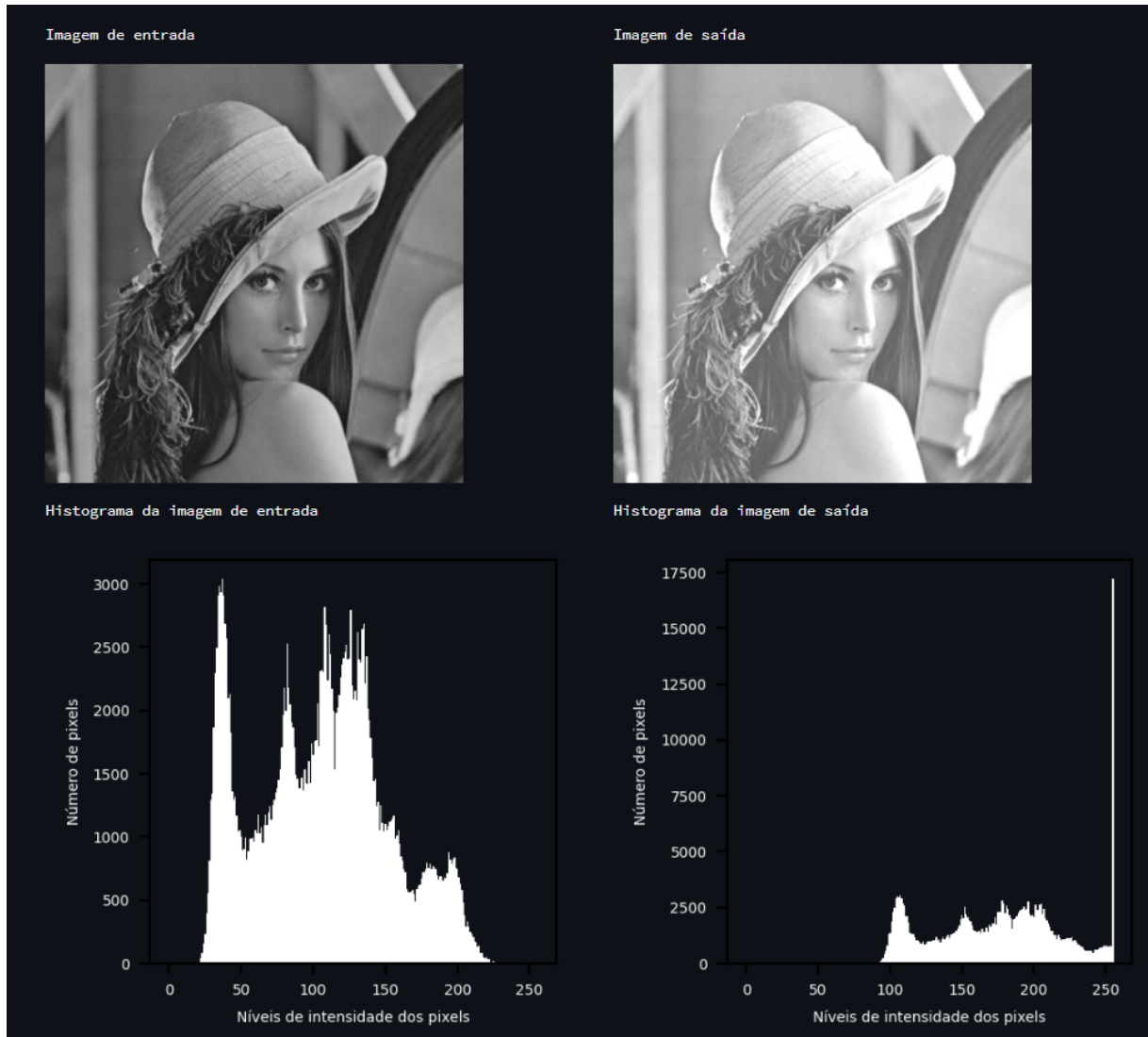
Fonte: Elaborado pelo autor

Na figura 23 foi escolhido o valor 180, onde qualquer pixel com valor menor ou igual tiveram sua intensidade reduzida a 0 e para os pixel com intensidade maior que 180, suas intensidades modificadas para 255.

### 4.3.3 Brilho e Contraste

As funcionalidades de brilho e contraste possuem a mesma função de aplicação '*convertScaleAbs*' da biblioteca opencv, porém possuem parâmetros diferentes aplicados, para que seja possível a imagem sofrer alteração somente no brilho o parâmetro alpha possui valor constante igual 1 e o parâmetro beta o valor escolhi pelo usuário através do controle deslizante com intervalo de 0 a 255, enquanto no contraste o valor de beta se torna igual a 0 e o alpha recebe os valores do controle deslizante de intervalo 0 a 2.

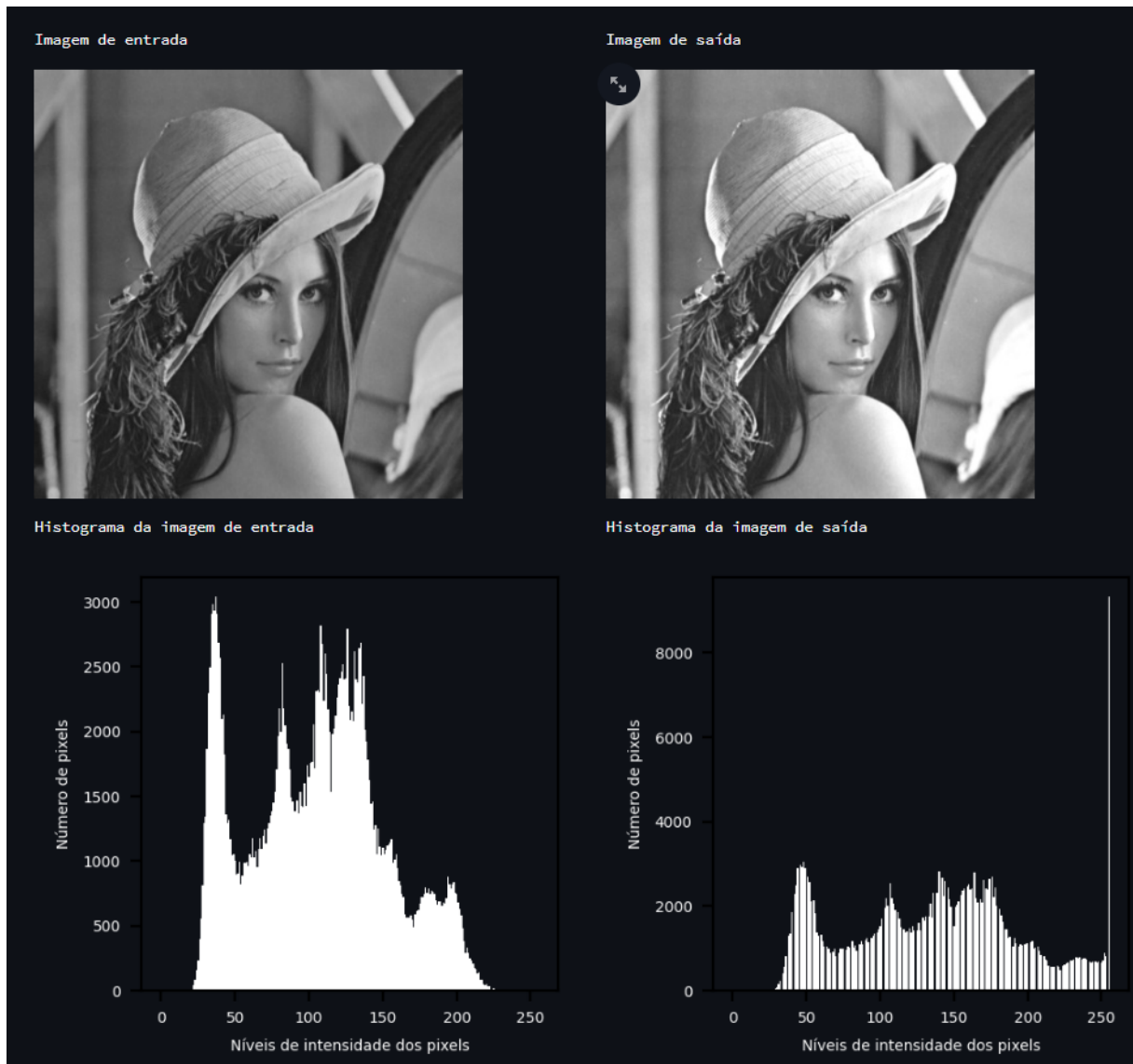
Figura 24 – Aumento do brilho da imagem



Fonte: Elaborado pelo autor

Essas duas aplicações possuem um diferencial em relação a outras técnicas pois é possível visualizar o histograma das imagens estáticas, utilizando a biblioteca `matplotlib` com a função `'histogram'`, os histogramas podem mostrar efeitos essenciais para o entendimento desse processamento em questão.

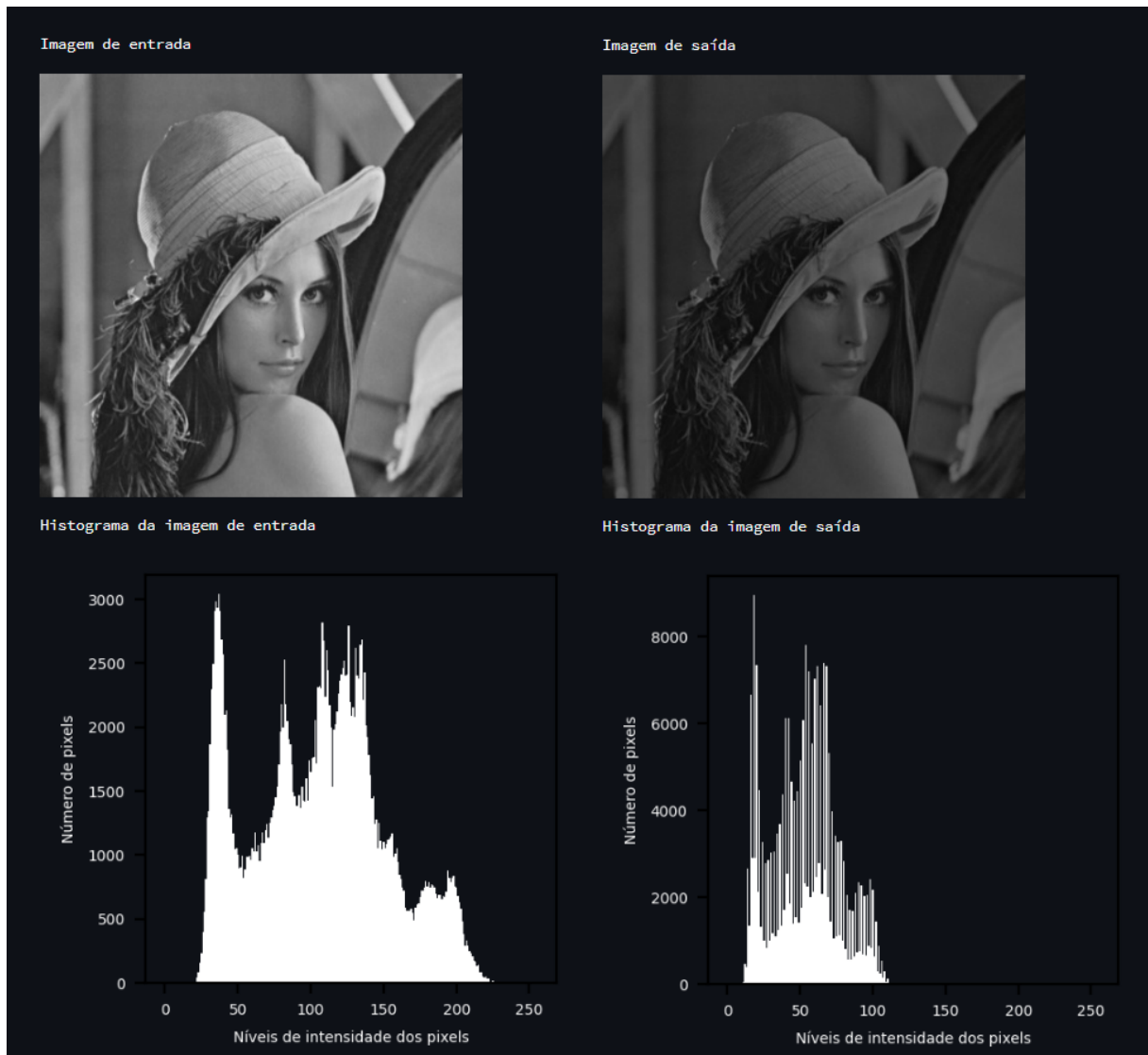
Figura 25 – Aumento do contraste da imagem



Fonte: Elaborado pelo autor

Na figura 25 foi aplicado o valor de 1.30 para o alpha, no histograma da imagem de saída é possível observar que os valores tendem a se espalhar pelo espectro.

Figura 26 – Diminuição do contraste da imagem



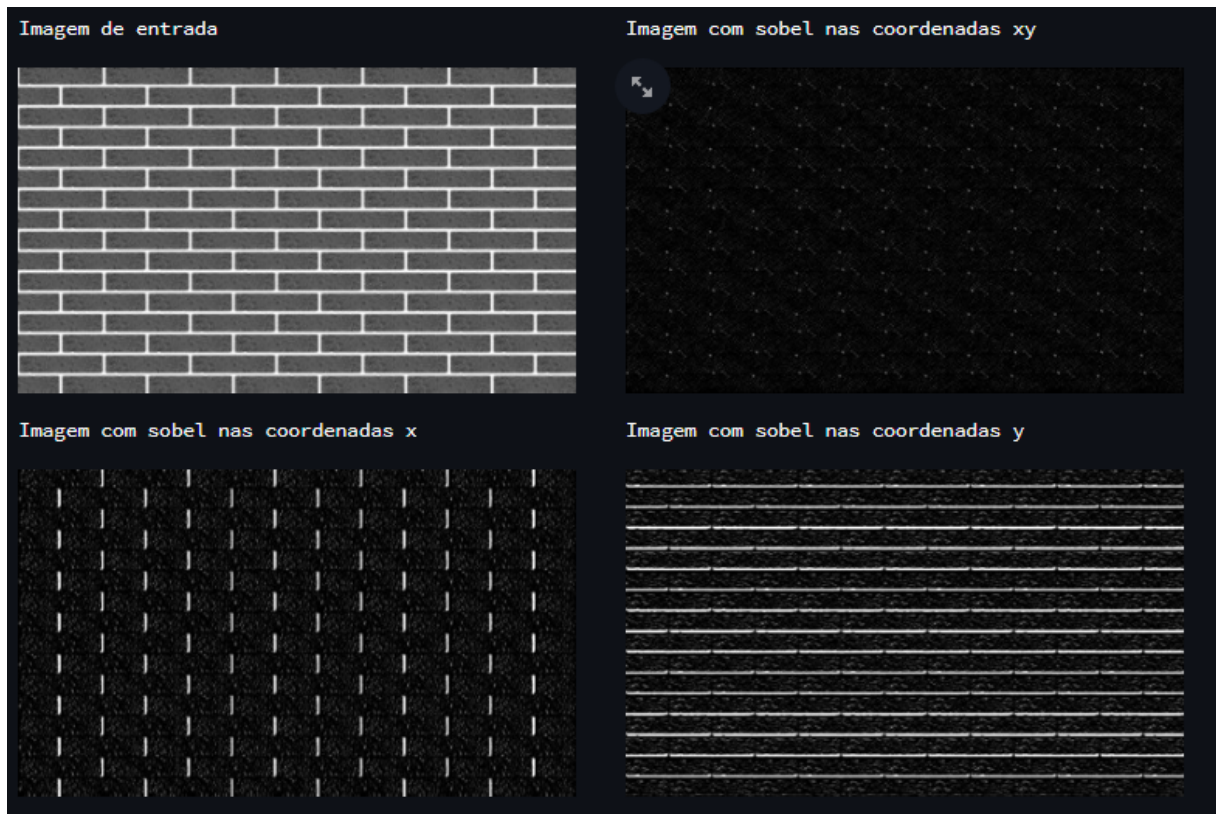
Fonte: Elaborado pelo autor

Na figura 26 o valor de alpha foi setado para 0.60 de modo a diminuir o contraste e assim os valores tenderam a se agrupar ao lado esquerdo do histograma, onde contém os valores de pixels mais escuros.

#### 4.3.4 Operador Sobel

Ao utilizar a funcionalidade do operador Sobel, quatro imagens de saída serão apresentadas, sendo elas a imagem original, a imagem com detecção de bordas nos eixos xy, uma outra somente no eixo x e a ultima somente no eixo y, isso ocorre pois a função 'sobel' do Opencv permite utilizar a ideia dos gradientes e obter a imagem com a detecção em um determinado eixo, deste modo se torna uma ótima forma de analisar o o efeito de processamento em determinadas imagens.

Figura 27 – Detecção de bordas pelo operador Sobel



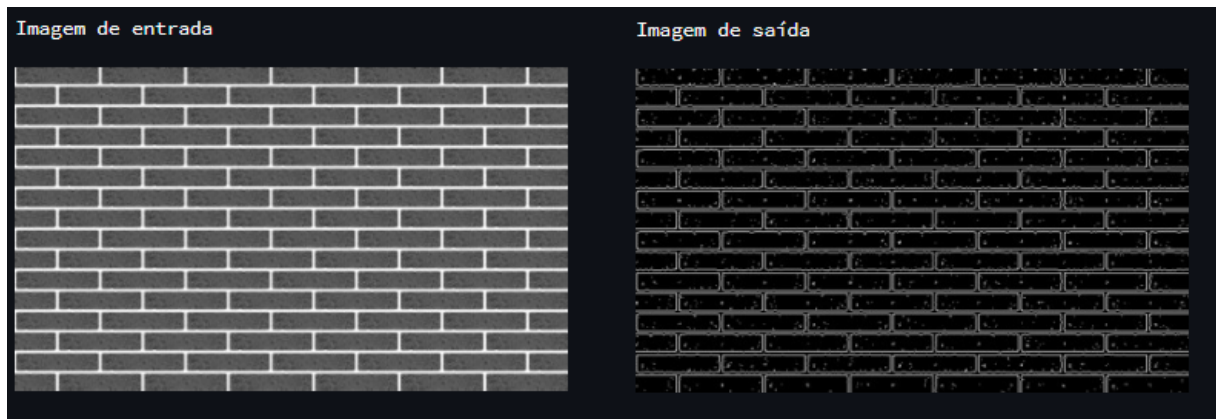
Fonte: Elaborado pelo autor

Na figura 27 nota-se que na imagem correspondente ao eixo x as linhas verticais foram marcadas pois é aonde ocorre a transição dos valores de intensidade dos pixels da imagem na horizontal (a verificação ocorre da esquerda para a direita), de forma análoga no eixo y as linhas horizontais é aonde ocorre as transições verticais nos valores de intensidade(onde a verificação ocorre de cima para baixo).

#### 4.3.5 Operador Canny

Com operador Canny utilizando a função 'canny' do opencv é possível modificar 2 parâmetros da função, sendo eles os valores de pixel máximo e mínimo, de modo que a combinação desses números podem tornar o processamento de detecção de bordas mais preciso, e esses valores podem ser alterados através de dois controles deslizantes com intervalos de 0 a 255.

Figura 28 – Detecção de bordas pelo operador Canny



Fonte: Elaborado pelo autor

#### 4.3.6 Visualização dos Espaços de Cores

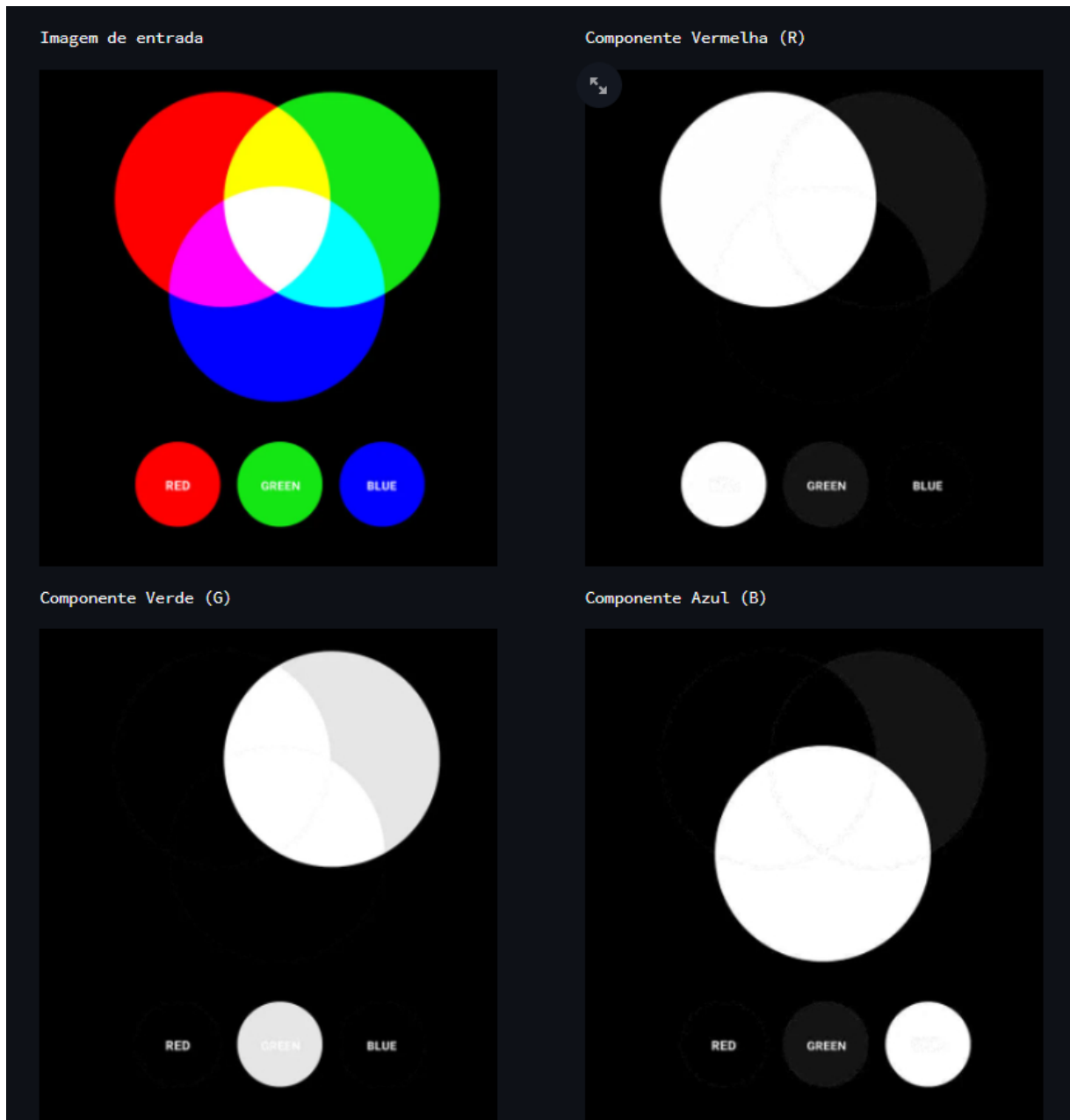
Esta funcionalidade explora principalmente o conceito dos espaços de cores RGB e YCbCr, de modo que quando uma imagem é anexada no software ou quando a imagem é adquirida direto de uma webcam as componentes dela são separadas e mostradas na tela para serem estudadas. Na figura 29 nota-se que como somente as cores verde, vermelho e azul são destacadas de branco em suas respectivas componentes, além de escurecerem as cores que não fazem parte daquele componente.

No espaço de cor Ycbr um resultado similar acontece, onde é possível observar a componente Y responsável pela informação de brilho da imagem, além de nos outros dois componentes Cb e Cr destacarem as cores azul e vermelha respectivamente como mostrado na figura 30 da bandeira do Amazonas que contém as cores azul e vermelho.

Nessa opção caso uma imagem que já esteja em escala de cinzas seja adicionada, a ferramenta retorna uma mensagem de erro ao usuário, pois neste caso em específico não há nenhum processamento que possa ser realizado, a imagem estando em escala de cinzas e sem os dados iniciais do processamento não é possível recuperar as informações de cores para que seria disponibilizado no RGB ou YCbCr da imagem.

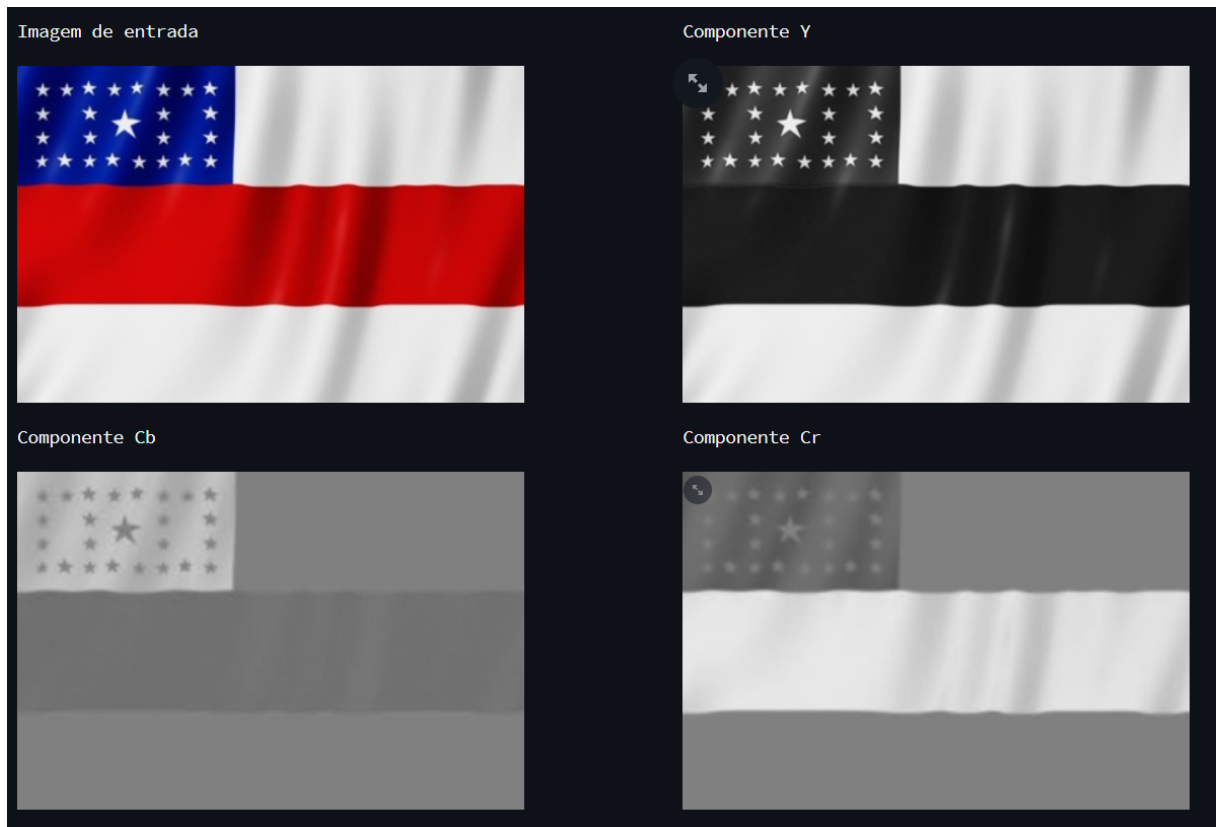


Figura 29 – Exemplo da função visualizar RGB



Fonte: Elaborado pelo autor

Figura 30 – Exemplo da função visualizar YCbCr



Fonte: Elaborado pelo autor

## 5 AVALIAÇÃO E RESULTADOS

### 5.1 PESQUISA DE SATISFAÇÃO

Para que fosse possível identificar uma métrica de avaliação de resultados obtidos com a utilização do software App PDI foi criado um formulário que continha questões referentes a cada técnica desenvolvida no software, onde elas iriam desde uma avaliação de 1 a 5 sendo 1 não atendeu ao objetivo e 5 atendeu ao objetivo, além de uma questão final onde era possível deixar um comentário de possível melhoria, assim foi possível receber um *feedback* para corrigir possíveis problemas e melhorar a experiência de uso da ferramenta.

As tabelas 2 e 3 apresentam os resultados que foram obtidos com o formulário de pesquisa, como a avaliação se referia a diversas técnicas optou-se por dividir em duas tabelas para uma melhor visualização. A tabela 2 possui os resultados a respeito da usabilidade e facilidade de navegação, além de filtro gaussiano, média e a técnica de limiarização, por sua vez a tabela 3 apresenta os resultados das técnicas de brilho e contraste, detecção de bordas pelos operadores de Canny e Sobel e a opção de visualização dos espaços de cores.

A tabela 1 demonstra os resultados calculados do método matemático do Ranking Médio, através dessa métrica pode-se observar a satisfação do usuário com a ferramenta e onde pode ser um bom ponto de melhoria, como por exemplo a detecção de bordas pelo operador Canny obteve a menor nota de RM por sua vez a técnica de limiarização obteve a maior nota.

Tabela 1 – Tabela de Ranking Médio calculado

<b>Técnicas</b>	<b>RM</b>
<b>F. Gaussiano</b>	4,61
<b>F. Média</b>	4,46
<b>Limiarização</b>	4,92
<b>Brilho e Contraste</b>	4,53
<b>Canny</b>	4,84
<b>Sobel</b>	4,30
<b>Visualizar</b>	4,73
<b>Usabilidade</b>	4,69

Tabela 2 – Tabela de resultados I

N. Participantes	Notas avaliadas			
	Usabilidade	F. Gaussiano	F. Média	Limiarização
Participante 1	4	3	3	5
Participante 2	5	5	5	5
Participante 3	5	3	5	5
Participante 4	5	5	5	5
Participante 5	4	5	4	4
Participante 6	5	5	5	5
Participante 7	5	5	3	5
Participante 8	5	5	5	5
Participante 9	5	5	4	5
Participante 10	5	5	5	5
Participante 11	4	4	5	5
Participante 12	4	5	5	5
Participante 13	5	5	4	4

Tabela 3 – Tabela de resultados II

N. Participantes	Notas avaliadas			
	Brilho e Contraste	Canny	Sobel	Visualizar
Participante 1	4	5	5	5
Participante 2	5	5	5	5
Participante 3	5	4	3	5
Participante 4	3	5	5	4
Participante 5	5	5	4	5
Participante 6	5	5	5	4
Participante 7	5	4	5	5
Participante 8	3	5	5	4
Participante 9	5	5	4	5
Participante 10	5	5	5	5
Participante 11	5	5	5	5
Participante 12	3	5	3	4
Participante 13	5	5	4	3

## 5.2 COMPARAÇÃO DE LITERATURA

No tópico 2.10 da metodologia foi apresentado outras literaturas e ferramentas que possuem um objetivo muito parecido com o deste trabalho, criação de um software de aprendizado para conceitos abordados na disciplina de PDI, desta forma o App PDI possui vantagens em relação a outros trabalhos.

A ferramenta EasYmage possui uma interface muito parecida com a do App PDI onde nela pode-se adicionar uma imagem e obter o processamento na saída, porém somente podendo aplicar a limiarização e rotação da imagem em 90° e 180°, por sua vez a o APP PDI possuem mais funcionalidades além de aquisição de imagens estáticas e dinâmicas.

Em relação as outras duas ferramentas elas trazem um diferencial que seria a possibilidade de mostrar o conteúdo teórico do processamento que esta sendo aplicado, apresentando um breve resumo ou até explicações mais detalhadas onde no App PDI não é possível.

### 5.3 APLICAÇÕES E PRÁTICAS DE USO

Como mencionado nos tópicos anteriores o App PDI possui o objetivo de facilitar o entendimento do assunto de processamento digital de imagens, por se tratar de um software com exemplos práticos algumas medidas de uso podem ser tomadas para uma melhor experiência de uso, pois um usuário que não conheça os conceitos iniciais de PDI pode acabar não usufruindo dessa aplicação da maneira correta.

O App PDI foi desenvolvido para que auxilie no aprendizado, ou seja ele pode ser utilizado preferencialmente após explicações teóricas do assunto em questão, em uma sala de aula o professor deve passar o conteúdo e explicar os principais conceitos envolvendo as técnicas de PDI após isso o aluno pode estudar aquela técnica de forma livre e modificar seus parâmetros de forma livre entendendo como ela pode ser aplicada em uma imagem. Outra alternativa pode ser em resolução de atividades, onde o professor pode solicitar para que o aluno desenvolva um código em python, ou matlab que replique o funcionamento de determinada técnica presente no software e utilizando o a ferramenta para que ele saiba qual o resultado que ele deve alcançar.

Um ponto importante é enfatizar que o App PDI foi desenvolvido para fins didáticos, logo o seu objetivo principal não é realizar uma performance superior a softwares comerciais já utilizados para o estudo de PDI, como matlab e suas bibliotecas, e sim fornecer uma alternativa mais acessível e dinâmica de exemplos dos assuntos abordados.

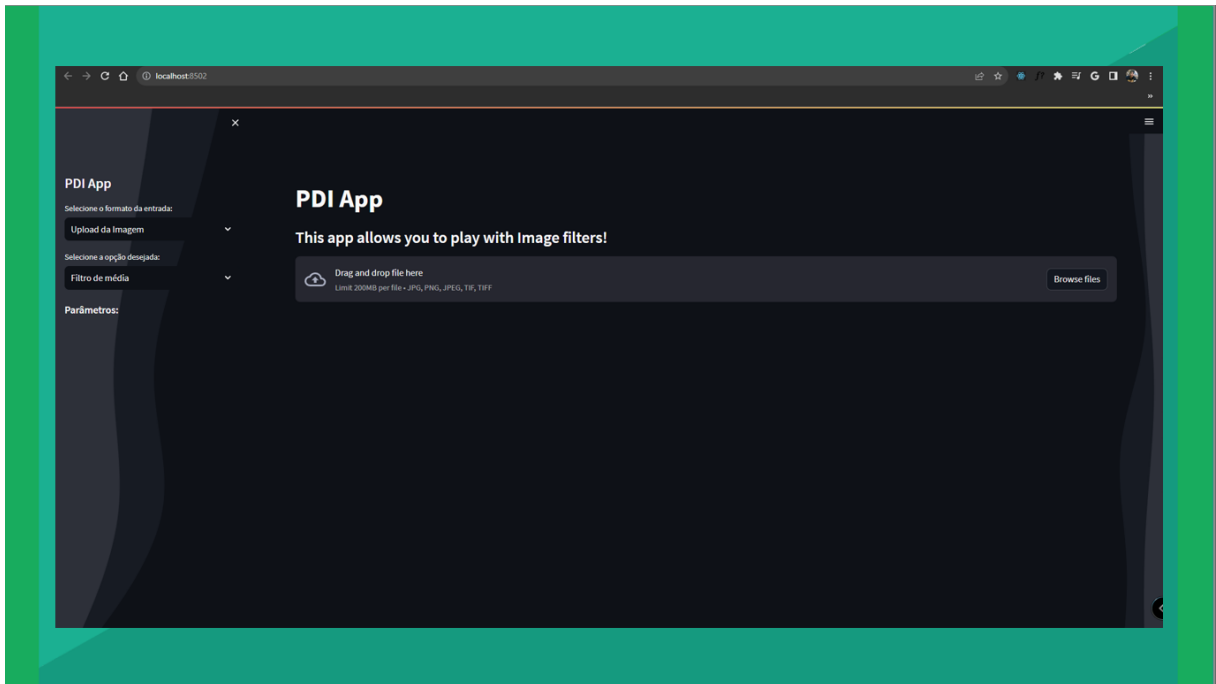
#### 5.3.1 Tutorial de uso

Este tópico apresenta um tutorial de como utilizar a ferramenta, uma breve explicação sobre cada um dos menus disponíveis na ferramenta, além de como os resultados são apresentados na tela.

Ao executar o arquivo do programa ele gera um link para ser executado em localhost além de abrir este link de forma automática. Deste modo a figura 31 demonstra a página inicial do software.

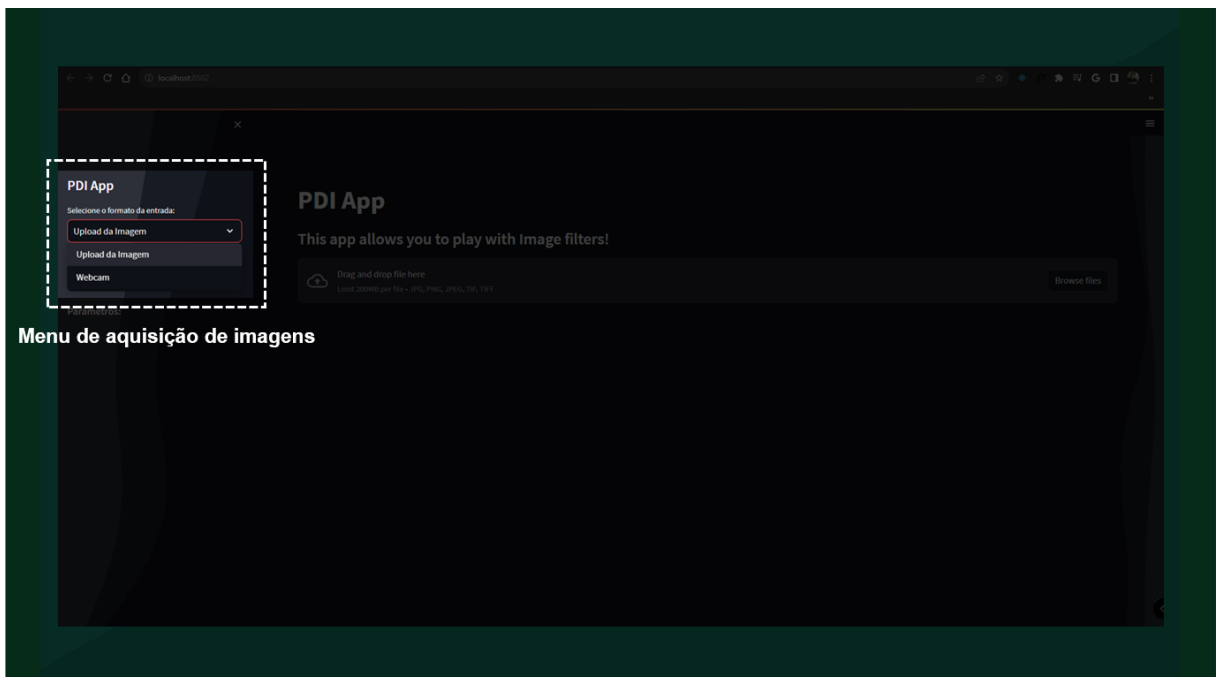
A figura 32 mostra o menu de aquisição de imagens onde nele contém as opções que podem ser escolhidas pelo usuário, sendo elas "upload de imagem" para imagens estáticas e "webcam" para imagens dinâmicas.

Figura 31 – Tela inicial do PDI App



Fonte: Elaborado pelo autor

Figura 32 – Menu de aquisição de imagens

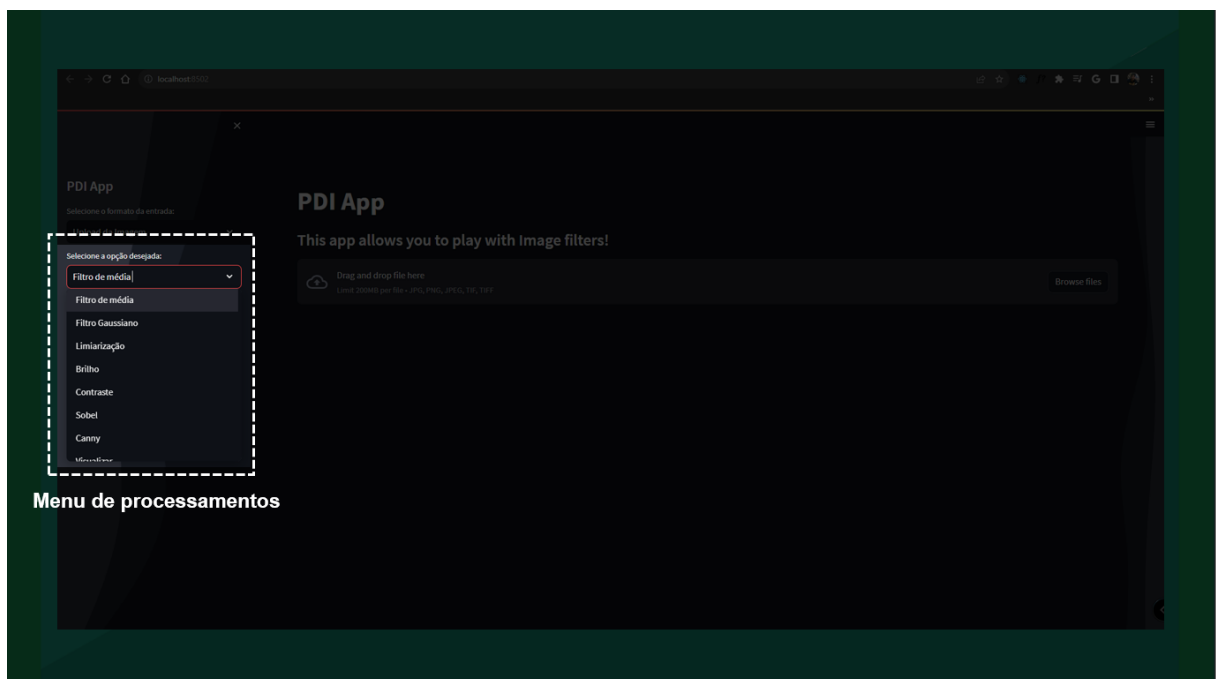


Fonte: Elaborado pelo autor

Após a definição do método de aquisição, pode-se escolher qual processamento aplicar através do menu de processamentos, que oferece uma lista de processamentos disponíveis representado na figura 33. A figura 34 apresenta o caso em que foi escolhida a aquisição por meio do *upload*, nele é possível pesquisar uma imagem do computador do usuário e anexa-la na ferramenta, quando se escolhe a opção de imagens dinâmicas o PDI App já busca um *driver* de câmera disponível na maquina em que esta rodando e tenta executa-lo.

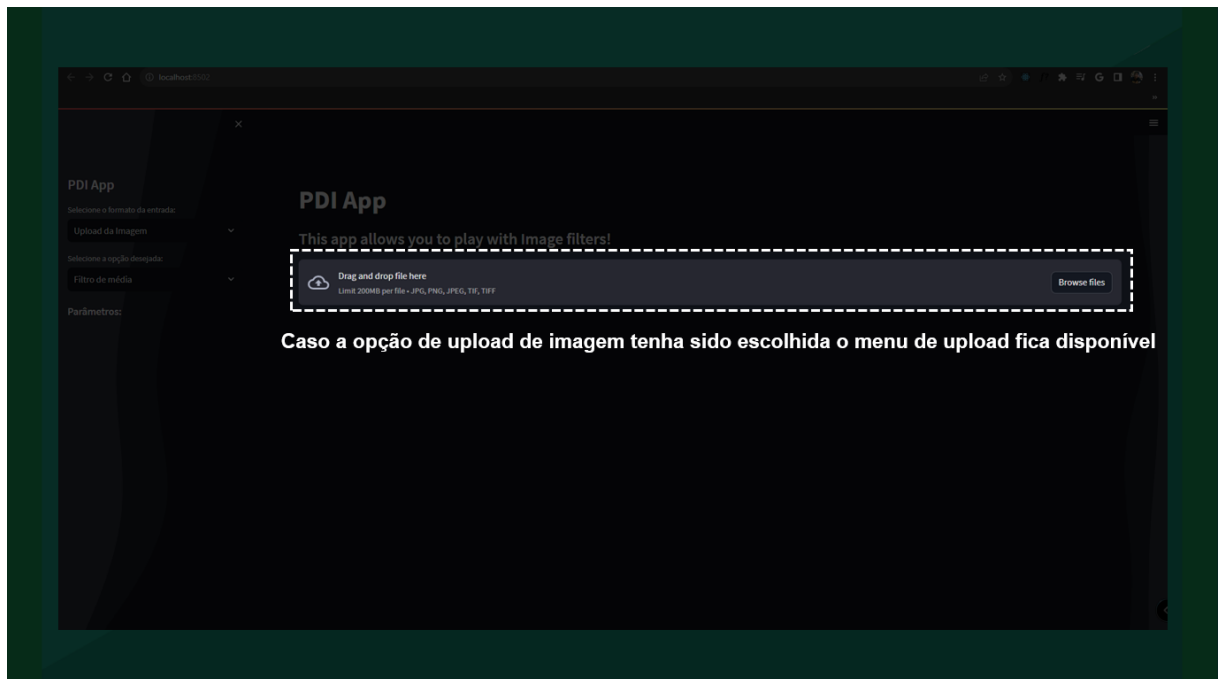
Por fim as figuras 35 e 36 mostram como o PDI App retorna as imagens de saída em ambos os casos de aquisição, onde pode-se observar que o programa sempre mostra a imagem original aplicado em escala de cinzas e a imagem processada.

Figura 33 – Menu de processamentos



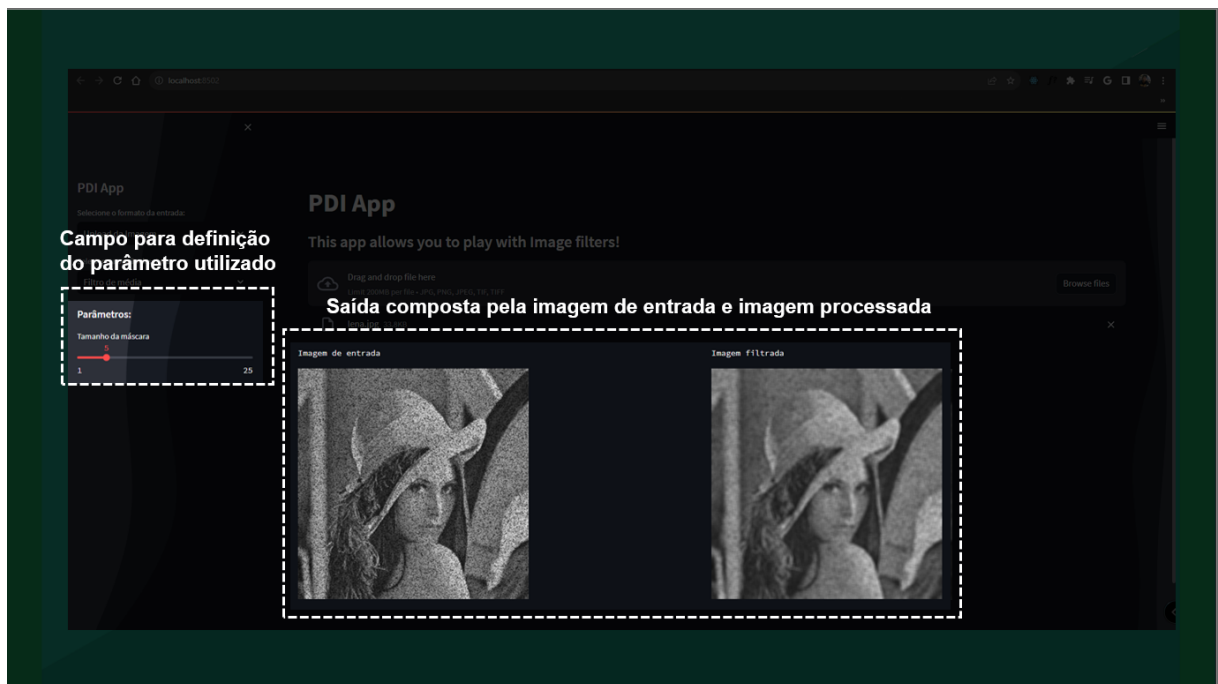
Fonte: Elaborado pelo autor

Figura 34 – Menu de upload de imagem



Fonte: Elaborado pelo autor

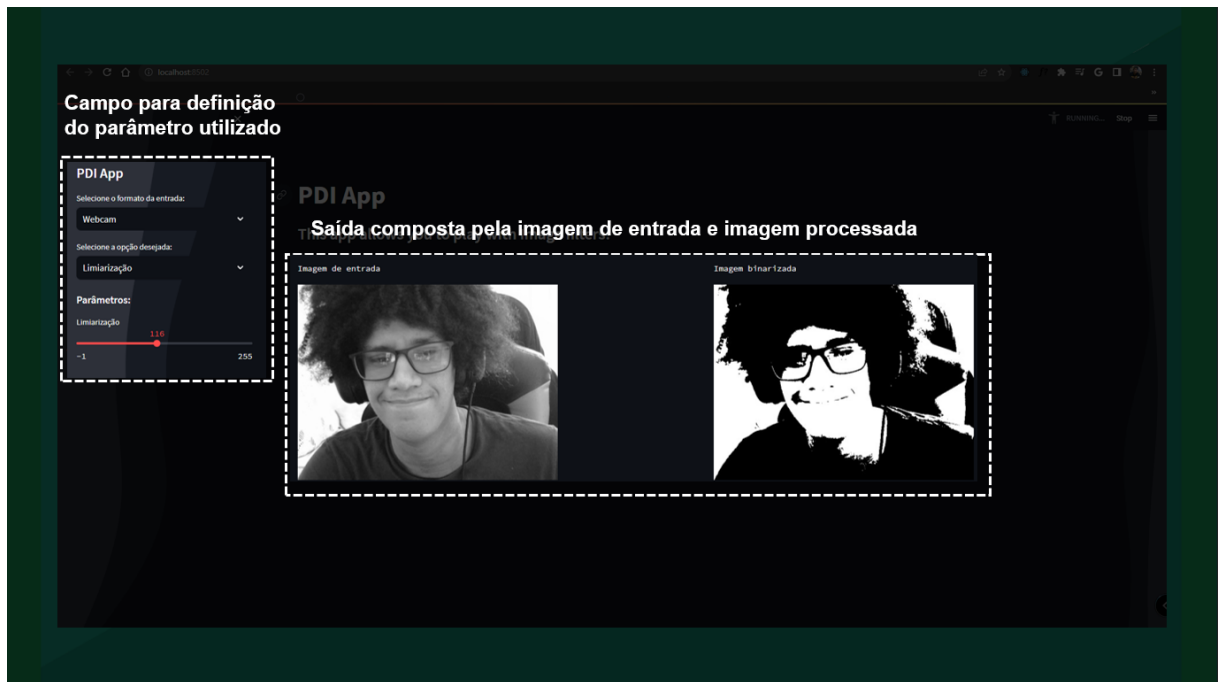
Figura 35 – Exemplo de resultado do upload de imagem



Fonte: Elaborado pelo autor



Figura 36 – Exemplo de resultado pela webcam



Fonte: Elaborado pelo autor

## 6 CONCLUSÕES

Através de pesquisas e planejamentos foi possível definir as tecnologias que melhor se aplicavam para o desenvolvimento desse projeto como o uso linguagem python, e as bibliotecas OpenCV e streamlit, de forma a possibilitar uma boa experiência ao usuário ao mesmo tempo que atende ao objetivo de facilitar o aprendizado dos conceitos abordados na disciplina de Processamento Digital de Imagens.

Com relação aos objetivos a serem atingidos neste trabalho, todos os algoritmos planejados como limiarização, alteração de contraste e brilho, filtros de suavização, detecção de bordas e visualização dos espaços de cores foram implementados onde o operador Sobel apresenta uma limiarização quando aplicado para os eixos x e y ao mesmo tempo, o que refletiu nos resultados da pesquisa de satisfação.

Com os resultados obtidos através dos formulários de pesquisa disponibilizado a alunos que cursam ou já cursaram a disciplina em questão, obteve-se um importante conteúdo para que houvessem melhorias na ferramenta, como na mudança de cores no texto presente nas opções de seleção das técnicas que possuía uma dificuldade em sua leitura e correção nos intervalos que apresentavam problemas na detecção de bordas pelo operador Sobel. A interface criada foi desenvolvida de forma escalável o que permite a integração de novas funcionalidades além de evitar que erros possam acontecer.

Outro ponto que pode ser observado é nos resultados da tabela 3 onde a opção de detecção de bordas pelo operador Sobel possui o menor valor de RM, isso pode ser devido a sua complexidade de aplicação, por se tratar de um processamento que pode ser aplicado em diferentes coordenadas (X e Y) e junção das duas nem sempre apresentar um bom resultado para todas as imagens testadas pode acabar gerando dificuldade em sua visualização, por outro lado a técnica de limiarização que possui o maior valor de RM é uma técnica de baixa complexidade e funciona muito bem para diversas imagens tanto estáticas quanto dinâmicas.

Com relação a usabilidade e funcionalidade de navegação o software App PDI obteve um valor de RM de 4,69 o que torna muito próximo do valor 5, tendo um total de 13 participantes na pesquisa, com isso observa-se que a fase de testes e a pesquisa de opinião e satisfação do usuário é uma métrica muito importante para o desenvolvimento de software refinando o seu funcionamento de acordo com o perfil de seus usuários.

Portanto, ao considerar a opinião dos usuários a partir da pesquisa realizada, o App PDI obteve uma ótima aceitação porém muitas funcionalidades novas ainda podem ser implementadas, como uma opção que fornece uma descrição teórico da técnica que esta sendo utilizada tanto como um breve texto ou informações detalhadas além de ser possível adicionar assuntos relacionados ao temas padrões de compressão de áudio e vídeo, abordando as normas JPEG e MPEG, como a estrutura da ferramenta foi desenvolvida de

forma escalável adicionar outras funcionalidades não causam problemas e nem afetam as funções existentes no programa.

Outro ponto a ser considerado para trabalhos futuros pode ser utilizar outras formas de métricas de avaliação, podendo ter resultados tanto de alunos participantes na pesquisa quanto de professores que já ministraram a disciplina, deste modo obtendo resultados ainda mais confiáveis considerando profissionais que atuam na área.

## REFERÊNCIAS BIBLIOGRÁFICAS

AGUIAR, G. M. d. Técnicas de processamento digital de imagens aplicadas em imagens de sensores rps. Universidade Federal Rural do Semi-Árido, 2020.

ANDRE C. MENDES; MAYCOL d. A. *limiarização aplicada ao cameraman*. 2015. Disponível em: <https://maalencar.wordpress.com/2012/06/14/image-segmentation/>. Acesso em: 05 de maio 2023.

ANDRÉI I. *Espaco de cores*. 2022. Disponível em: <https://www.ivanandrei.com/rgb-vs-ybcr-cual-es-mejor-en-una-conexion-hdmi/>. Acesso em: 03 de maio 2023.

André Rodrigues. *especial*. 2023. Disponível em: <https://https://www.mundodigital.art.br/resolucao-para-site-qual-e-a-melhor/>. Acesso em: 24 de Maio 2023.

BECKER, J. D.; POHREN, N.; BEZ, M. R. Visnode: uma ferramenta gamificada para o aprendizado de processamento digital de imagens. *Educação & Linguagem*, v. 21, n. 2, p. 5–24.

BEHAR.P.B.

*Avaliação de softwares educacionais no processo ensinoaprendizagem computadorizado: estudo de caso*. — Universidade Federal do Rio Grande do Sul, 1993.

BRUNO RIBEIRO. *Filtro de média 5x5*. 2015. Disponível em: <http://www.ic.uff.br/~aconci/suavizacao.pdf>. Acesso em: 15 de maio 2023.

CORRÊA<sup>1</sup>, J. H. G. et al. Um modelo simples e parametrizável para classificação de cores no sistema rgb. 2014.

COSTA, C. M. da; GUIMARÃES, L. V. et al. Análise experimental da conversão rgb/ycbcr e codificação run-length para compressão de imagens em sistemas de tempo real baseados em vants/experimental analysis of rgb/ycbcr conversion and run-length encoding for image compression in uav-based real-time systems. *Brazilian Journal of Development*, v. 7, n. 7, p. 73063–73075, 2021.

Divisão de Processamento de Imagens. *Filtros gerais*. 2022. Disponível em: [http://www.dpi.inpe.br/~carlos/Academicos/Cursos/Pdi/pdi\\_filtros.htm#s1\\_1](http://www.dpi.inpe.br/~carlos/Academicos/Cursos/Pdi/pdi_filtros.htm#s1_1) Acesso em: 17 de maio 2023.

DUQUE, R. G. *Python para todos*. [S.l.]: Creative Commons Reconocimiento, 2011.

FERREIRA, L. O.; RODRIGUES, W. M. Funções lineares no estudo do sistema de cores rgb. *Dados Internacionais de Catalogação na Publicação (CIP)*, p. 55, 2021.

FLORENCIO, R. d. B. Processamento de imagens subaquáticas. Universidade Federal do Rio de Janeiro, 2009.

GADIOLI, L. B. T.; SILVA, V. L. G. da; NASCIMENTO, L. A. Comparação de métodos de detecção de bordas em imagens de satélite de florestas de eucalipto no vale do jequitinhonha.

GOLLAPUDI, S.; GOLLAPUDI, S. OpenCV with python. *Learn Computer Vision Using OpenCV: With Deep Learning CNNs and RNNs*, Springer, p. 31–50, 2019.

- GOMES, O. d. F. M. Processamento e análise de imagens aplicados à caracterização automática de materiais. **Rio de Janeiro: Dissertação (Mestrado em Ciências da Engenharia Metalúrgica). Pontifícia Universidade Católica do Rio de Janeiro**, 2001.
- GONZALEZ, R. C. W. R. C. **Processamento Digital de Imagens**. [S.l.]: Pearson Prentice Hall., 2010. v. 3<sup>a</sup> Edição.
- JESUS, E. O.; JR, R. C. A utilização de filtros gaussianos na análise de imagens digitais. **Proceeding Series of the Brazilian Society of Computational and Applied Mathematics**, v. 3, n. 1, 2015.
- LOPES, F. M. Um modelo perceptivo de limiarização de imagens digitais. **Universidade Federal do Paraná**, 2003.
- MARTINS, S. B. **Introdução ao Processamento Digital de Imagens**. [S.l.]: Instituto de Computação Universidade Estadual de Campinas (UNICAMP), 2014.
- NETO, E. C.; FILHO, P. P. R.; NORTE, A. C. **TÉCNICAS DE PDI E INTELIGÊNCIA ARTIFICIAL APLICADAS AO RECONHECIMENTO DE PLACAS DE CARRO NOS PADROES BRASILEIROS**. [S.l.]: IFCE-CE.
- OLIVEIRA, L. H.  
**Exemplo de cálculo de Ranking Médio para Likert. Notas de Aula. Metodologia Científica e Técnicas de Pesquisa em Administração** — PPGA CNEC/FACECA, 2005.
- opencv.org. **Site OpenCV**. 2023. Disponível em: <<https://opencv.org/about/>>. Acesso em: 15 de fevereiro 2023.
- PYTHON, W. Python. **Python Releases Wind**, Citeseer, v. 24, 2021.
- RENE.E. **temporal**. 2023. Disponível em: <<https://www.reneelab.biz/frames-por-segundo-ou-quadros-por-segundo.html>>. Acesso em: 24 de Maio 2023.
- SANCHES, C. H. et al. Técnicas de suavização de imagens e eliminação de ruídos. **Anais do EATI–Encontro Anual de Tecnologia da Informação. Frederico Westphalen–RS**, p. 21–30, 2015.
- SILVA, M. A. M.; CHAVES, A.; AQUINO, F. Projeto e desenvolvimento de uma ferramenta educativa para ensino de processamento de imagens baseado na biblioteca opencv. In: **XL Congresso Brasileiro de Educação em Engenharia, Belém**. [S.l.: s.n.], 2012.
- SNOWFLAKE. **Streamlit**. 2023. Disponível em: <<https://blog.streamlit.io/>>. Acesso em: 16 de Julho 2023.
- ZEM-LOPES, A. M.; PEDRO, L. Z.; ISOTANI, S. Qualidade de softwares educacionais baseados na web (semântica): Um mapeamento sistemático. **RENOTE**, v. 12, n. 1, 2014.