

# **UNIP**

**UNIVERSIDADE PAULISTA**

## **Processamento de Imagem e Visão Computacional**

**Autora:** Profa. Vanessa Santos Lessa

**Colaboradora:** Profa. Larissa Rodrigues Damiani

## Professora conteudista: Vanessa Santos Lessa

Doutora em Ciências e Aplicações Geoespaciais pelo Instituto Presbiteriano Mackenzie, mestre em Engenharia Elétrica com ênfase em Inteligência Artificial Aplicada à Automação pelo Centro Universitário da Fundação Educacional Inaciana (FEI) e bacharel em Engenharia da Computação pela Universidade São Judas Tadeu (USJT). Coordenadora do curso de Bacharelado em Ciência da Computação na Universidade Paulista (UNIP) na modalidade educação a distância (EaD). Atua há mais de 18 anos em cargos técnicos e gerenciais na área de computação.

### Dados Internacionais de Catalogação na Publicação (CIP)

L638p Lessa, Vanessa Santos.

Processamento de Imagem e Visão Computacional / Vanessa Santos Lessa. – São Paulo: Editora Sol, 2023.

154 p., il.

Nota: este volume está publicado nos Cadernos de Estudos e Pesquisas da UNIP, Série Didática, ISSN 1517-9230.

1. Processamento. 2. Imagem. 3. Operações. I. Título.

CDU 681.335.8

U518.67 – 23

Profa. Sandra Miessa  
**Reitora**

Profa. Dra. Marília Ancona Lopez  
**Vice-Reitora de Graduação**

Profa. Dra. Marina Ancona Lopez Soligo  
**Vice-Reitora de Pós-Graduação e Pesquisa**

Profa. Dra. Claudia Meucci Andreatini  
**Vice-Reitora de Administração e Finanças**

Prof. Dr. Paschoal Laercio Armonia  
**Vice-Reitor de Extensão**

Prof. Fábio Romeu de Carvalho  
**Vice-Reitor de Planejamento**

Profa. Melânia Dalla Torre  
**Vice-Reitora das Unidades Universitárias**

Profa. Silvia Gomes Miessa  
**Vice-Reitora de Recursos Humanos e de Pessoal**

Profa. Laura Ancona Lee  
**Vice-Reitora de Relações Internacionais**

Prof. Marcus Vinícius Mathias  
**Vice-Reitor de Assuntos da Comunidade Universitária**

## **UNIP EaD**

Profa. Elisabete Brihy  
Profa. M. Isabel Cristina Satie Yoshida Tonetto  
Prof. M. Ivan Daliberto Frugoli  
Prof. Dr. Luiz Felipe Scabar

### **Material Didático**

Comissão editorial:

Profa. Dra. Christiane Mazur Doi  
Profa. Dra. Ronilda Ribeiro

Apoio:

Profa. Cláudia Regina Baptista  
Profa. M. Deise Alcantara Carreiro  
Profa. Ana Paula Tôrres de Novaes Menezes

Projeto gráfico:

Prof. Alexandre Ponzetto

Revisão:

Lucas Ricardi  
Kleber Souza



# Sumário

## Processamento de Imagem e Visão Computacional

APRESENTAÇÃO .....	7
INTRODUÇÃO .....	8

### Unidade I

1 PROCESSAMENTO DE IMAGEM .....	9
1.1 Terminologia e conceitos básicos .....	12
1.2 Percepção visual (sistema biológico <i>versus</i> computacional) .....	13
1.3 Fundamentos de imagens digitais e suas características .....	16
1.4 Fontes de dados .....	22
1.5 Canais de imagem (espaço RGB, CMY, CMYK e HSV) .....	27
1.6 Variação de canais .....	31
1.7 Resolução, cores e formatos .....	32
1.8 Manipulação matricial e representação em matriz .....	33
1.9 Sistema de coordenadas e manipulação de pixels .....	35
1.10 Região de interesse (ROI, do inglês region of interest) .....	36
2 TÉCNICAS DE PRÉ-PROCESSAMENTO .....	37
2.1 Rotação .....	39
2.2 Histograma de cores .....	41
2.3 Operações aritméticas .....	45
2.4 Transformações geométricas .....	46
2.5 Ruído em imagens .....	47
2.6 Escala de cinza .....	49
2.7 Métodos de filtragem .....	52
3 SUAUIZAÇÃO DE IMAGENS .....	54
3.1 Suavização por cálculo da média .....	55
3.2 Suavização pela gaussiana .....	58
3.3 Suavização pela mediana .....	61
3.4 Suavização com filtro bilateral .....	64
4 SEGMENTAÇÃO DE IMAGENS .....	67
4.1 Detecção de bordas .....	71
4.2 Sobel .....	71
4.3 Operador laplaciano .....	74
4.4 Filtro máscara de desaguçamento .....	76
4.5 Detector de bordas de Canny .....	78

## Unidade II

5 OPERAÇÕES MORFOLÓGICAS.....	85
5.1 Elemento estruturante.....	86
5.2 Erosão e dilatação.....	87
5.3 Abertura e fechamento.....	90
5.4 Gradiente morfológico.....	93
5.5 Top Hat.....	95
5.6 Detecção de descontinuidades.....	97
5.7 Detecção de similaridades.....	98
6 EXTRAÇÃO DE CARACTERÍSTICAS.....	101
6.1 Redução de dimensionalidade.....	103
6.2 Maldição da dimensionalidade.....	104
6.3 Segmentação no espaço de atributos.....	106
6.4 Análise de componentes principais (PCA).....	108
6.5 Análise discriminante linear (LDA).....	109
7 RECONHECIMENTO DE PADRÕES.....	111
7.1 Classificador de K-vizinhos mais próximos.....	112
7.2 Método de classificação em cascata (Haar cascade).....	115
7.3 Estado da arte.....	118
7.4 Abordagens baseadas em aprendizado profundo.....	120
7.5 Redes neurais artificiais convolucionais (CNNs).....	121
7.6 Operação de convolução.....	123
7.7 Pooling.....	125
7.8 Flatten layer.....	126
7.9 Funções de ativação.....	127
8 VISÃO COMPUTACIONAL.....	129
8.1 Tecnologias e projetos de visão computacional.....	130
8.2 Planejamento de projeto de visão computacional.....	131
8.3 Principais desafios.....	133
8.4 Organização das etapas de desenvolvimento.....	135
8.5 Implantação do projeto.....	137
8.6 Aplicações de reconhecimento facial e segmentação de objetos.....	138
8.7 Arcabouço tecnológico.....	140

## APRESENTAÇÃO

Como futuro cientista da computação, você encontrará desafios envolvendo a extração de informações numéricas ou simbólicas a partir de imagens. O maior objetivo na visão computacional é fazer com que os computadores possam ter uma compreensão visual do mundo; dessa forma, o computador poderia ver, identificar e processar imagens da mesma forma que a visão humana.

O objetivo desta disciplina é fornecer conceitos e técnicas básicas em processamento de imagem, propiciar a compreensão dos principais fundamentos e suas aplicações nos diferentes estágios de um projeto de visão computacional, evoluindo desde a manipulação básica e inicial de arquivos até o reconhecimento avançado de padrões existente em imagens.

Neste livro-texto, você terá a oportunidade de compreender os fundamentos de processamento de imagens, suas técnicas de pré-processamento, métodos de filtro, detecção e reconhecimento de padrões em imagens para resolução de problemas em visão computacional.

Vale acrescentar que este material é escrito em linguagem simples e direta, como se houvesse uma conversa entre a autora e o leitor. Adicionalmente, são inseridas muitas figuras, que auxiliam no entendimento dos tópicos desenvolvidos. Os itens chamados de observação e de lembrete são oportunidades para que você solucione eventuais dúvidas. Os itens chamados de saiba mais possibilitam que você amplie seus conhecimentos. Há, ainda, muitos exemplos de aplicação, resolvidos em detalhes, o que implica a fixação dos assuntos abordados.

## INTRODUÇÃO

O processamento de imagens digitais permite a preparação de imagens para possibilitar resultados melhores na aplicação dos métodos de visão computacional. Diversas áreas já perceberam o poder da aplicação de visão computacional em alguns processos. Podemos citar, por exemplo, a linha de produção de uma fábrica, pensando em um processo de atividade visual como o diagnóstico de falhas (peças fora do padrão), pois um operador humano poderia ficar sobrecarregado e comprometer a qualidade do seu trabalho, ao contrário do computador, que pode ver várias peças ao mesmo tempo sem ficar "cansado".

O processamento de imagem e visão computacional tem desempenhado um importante papel na transformação digital, possibilitando a melhoria nos sistemas de monitoramento e segurança, sistemas de reconhecimento (trajetos, pessoas, animais e objetos) em veículos autônomos, processamento de imagens de exames médicos, como radiografia, tomografia e ultrassonografia, para auxiliar em diagnósticos, entre outras contribuições. Esses sistemas podem fornecer um grau de precisão e eficiência para identificar detalhes que um ser humano poderia não desempenhar.

O conteúdo deste livro-texto foi dividido em duas unidades.

Na unidade I, apresentaremos inicialmente os conceitos básicos do processamento de imagem, evoluindo para as técnicas de pré-processamento, suavização e segmentação de imagens.

Na unidade II, abordaremos as operações morfológicas, as extrações de características, o reconhecimento de padrões e a visão computacional.

Esperamos que você tenha uma boa leitura e se sinta motivado a ler e conhecer mais sobre a disciplina.

Bons estudos.



# Unidade I

## 1 PROCESSAMENTO DE IMAGEM

O campo do processamento de imagem surgiu como resultado do avanço da tecnologia da computação e da necessidade de lidar com imagens digitais geradas por diversos dispositivos, como câmeras e scanners. A história do processamento de imagem pode ser dividida em algumas fases chave:

- **Décadas de 1950–1960:** os primeiros passos na digitalização e processamento de imagens ocorreram nessa época. Pesquisadores começaram a desenvolver técnicas para converter imagens em formatos digitais e, em seguida, a aplicar operações simples, como filtragem e realce, usando computadores muito primitivos.
- **Década de 1970:** durante essa década, houve um aumento significativo na pesquisa em processamento de imagem, impulsionado pelo desenvolvimento de hardware mais poderoso e pela disponibilidade de imagens digitais de várias fontes, incluindo imagens médicas, imagens de satélite e imagens de documentos impressos.
- **Década de 1980:** nesta década, a pesquisa em processamento de imagem começou a se consolidar como um campo interdisciplinar, com contribuições de áreas como ciência da computação, engenharia elétrica, matemática e ciências da informação. A criação de conferências e periódicos dedicados ao processamento de imagem ajudou a formalizar o campo.
- **Década de 1990:** com o aumento do poder computacional e o desenvolvimento de algoritmos mais avançados, o processamento de imagem tornou-se mais sofisticado. Foram desenvolvidas técnicas de reconhecimento de padrões, análise de textura, processamento de vídeo em tempo real e sistemas de visão computacional para aplicações industriais e de automação.
- **Década de 2000 em diante:** o processamento de imagem continuou a evoluir rapidamente com o advento de técnicas de aprendizado de máquina e inteligência artificial. Algoritmos de deep learning, como redes neurais convolucionais (CNNs), revolucionaram a capacidade de realizar tarefas complexas de análise de imagem, como reconhecimento de objetos e detecção de padrões em grandes conjuntos de dados.



### Saiba mais

Algoritmos de deep learning são um conjunto de técnicas e métodos usados no campo de aprendizado de máquina que se concentram na construção e treinamento de redes neurais artificiais profundas, também conhecidas como redes neurais profundas ou DNN (Deep Neural Networks). Essas redes neurais são compostas de várias camadas de neurônios artificiais interconectados, o que lhes permite aprender e representar padrões e características complexas nos dados.

Para saber mais sobre redes neurais artificiais, recomendamos a leitura do capítulo 18.7 da obra a seguir:

NORVIG, P.; RUSSEL, S. *Inteligência artificial*. 3. ed. São Paulo: Grupo GEN, 2013.

Atualmente, o processamento de imagem desempenha um papel fundamental em diversas áreas, incluindo medicina, indústria, entretenimento, automação e muito mais. Com a crescente disponibilidade de dados de imagem e o contínuo avanço da tecnologia, o campo continua a se expandir e a se tornar cada vez mais sofisticado, permitindo uma variedade de aplicações úteis e impactantes em nossa sociedade.

O processamento de imagem é um campo interdisciplinar que se concentra na manipulação e análise de imagens digitais com o objetivo de melhorar sua qualidade, extrair informações úteis e tomar decisões com base nelas. Esse campo desempenha um papel fundamental em uma variedade de aplicações, desde a medicina até a indústria, passando pela ciência e entretenimento. Vamos explorar alguns dos aspectos mais importantes do processamento de imagem:

**1. Aquisição de imagem:** o processo começa com a aquisição de imagens, que pode ser feito por meio de câmeras digitais, scanners, sensores remotos e outras tecnologias. A qualidade da imagem inicial é crucial, pois afeta diretamente as etapas subsequentes de processamento.

**2. Pré-processamento:** nesta etapa, a imagem é ajustada e aprimorada para a correção de problemas comuns, como ruído, iluminação inadequada e distorções. Isso inclui filtragem, normalização de brilho e contraste, remoção de artefatos e calibração.

**3. Segmentação:** a segmentação é o processo de dividir uma imagem em regiões ou objetos de interesse. Isso pode ser feito com base em características como cor, intensidade, textura ou forma. A segmentação é essencial para isolar objetos de fundo e facilitar análises posteriores.

**4. Extração de características:** uma vez segmentados, os objetos de interesse podem ser descritos por meio da extração de características relevantes, como bordas, texturas, formas geométricas e cores. Essas características são usadas em tarefas como reconhecimento de padrões e classificação de objetos.

**5. Filtragem e realce:** para destacar informações importantes em uma imagem, técnicas de filtragem e realce são aplicadas. Isso pode incluir realces de borda, realces de contraste ou filtragem de suavização para reduzir o ruído.

**6. Transformações espaciais e de frequência:** transformações como a transformada de Fourier ou a transformada Wavelet permitem representar a informação de uma imagem em domínios alternativos, facilitando a detecção de padrões e a compressão de dados.

**7. Reconhecimento de padrões:** o processamento de imagem desempenha um papel crítico no reconhecimento de padrões, em que algoritmos são treinados para identificar objetos, rostos, caracteres e outras informações em imagens.

**8. Fusão de imagens:** em aplicações como visão computacional e sensoriamento remoto, é comum combinar informações de diferentes fontes de imagem para criar uma imagem mais rica e informativa. Isso é conhecido como fusão de imagens.

**9. Análise e tomada de decisão:** as informações extraídas das imagens são usadas para análises qualitativas e quantitativas. Isso pode incluir diagnósticos médicos, controle de qualidade na indústria e detecção de anomalias em sistemas de vigilância.

**10. Aplicações:** o processamento de imagem é utilizado em aplicações, incluindo medicina (diagnóstico por imagem, cirurgia assistida por computador), automação industrial, visão computacional (veículos autônomos, reconhecimento facial), entretenimento (animação, efeitos especiais), sensoriamento remoto (monitoramento ambiental, agrícola e meteorológico), segurança (reconhecimento de padrões em câmeras de vigilância) e muito mais.

O processamento de imagem continua a evoluir com o avanço da tecnologia, incluindo a aplicação de técnicas de aprendizado de máquina e inteligência artificial para automatizar tarefas complexas de análise e interpretação de imagens. É um campo fascinante que desempenha um papel fundamental em nossa vida cotidiana e em uma variedade de setores, tornando possível a extração de informações valiosas a partir de imagens digitais.



### Lembrete

O processamento de imagem é um campo interdisciplinar que se concentra na manipulação e análise de imagens digitais com o objetivo de melhorar sua qualidade, extrair informações úteis e tomar decisões com base nelas. Esse campo desempenha um papel fundamental em uma variedade de aplicações, desde a medicina até a indústria, passando pela ciência e entretenimento.

## 1.1 Terminologia e conceitos básicos

O processamento de imagem é um campo repleto de terminologia e conceitos fundamentais que são essenciais para compreender como as imagens digitais são processadas e analisadas. Aqui estão alguns dos termos e conceitos básicos mais importantes:

- **Pixel:** o pixel (abreviação de "picture element", ou "elemento de imagem") é a menor unidade de uma imagem digital. Cada pixel representa um ponto ou elemento de cor na imagem e é a unidade básica na qual as imagens são divididas para análise.
- **Resolução:** a resolução de uma imagem refere-se ao número de pixels em largura e altura que a compõem. Ela determina a nitidez e detalhes da imagem. Uma resolução mais alta significa mais pixels, resultando em imagens mais nítidas.
- **Formato de imagem:** o formato de imagem descreve a estrutura de armazenamento de uma imagem digital. Alguns formatos populares incluem JPEG, PNG, GIF e TIFF. Cada formato tem suas próprias características e aplicações.
- **Matriz de pixels:** uma imagem digital pode ser vista como uma matriz bidimensional de pixels, em que cada pixel é representado por um valor numérico que indica sua cor ou intensidade. Em imagens coloridas, geralmente existem três canais (vermelho, verde e azul) para representar cores.
- **Escala de cinza:** uma imagem em escala de cinza é uma imagem em que cada pixel é representado por um valor de intensidade que varia de preto (valor mínimo) a branco (valor máximo). É frequentemente usada para representar imagens em preto e branco.
- **Histograma:** um histograma é uma representação gráfica da distribuição das intensidades de pixels em uma imagem. É útil para entender a distribuição de cores ou intensidades na imagem e pode ser usado para ajustar o contraste e o brilho.
- **Realce de imagem:** o realce de imagem envolve a melhoria da qualidade de uma imagem por meio de técnicas como aumento de contraste, realce de bordas e redução de ruído, a fim de tornar as características mais visíveis.
- **Segmentação:** a segmentação é o processo de dividir uma imagem em regiões ou objetos distintos com base em características como cor, intensidade, textura ou forma. É frequentemente usada em reconhecimento de objetos e análise de imagem.
- **Filtro:** um filtro de imagem é uma máscara ou matriz de convolução que é aplicada a uma imagem com o objetivo de realizar operações como suavização (borramento), realce de borda e detecção de características.
- **Transformada de Fourier:** é uma técnica matemática que converte uma imagem do domínio espacial para o domínio de frequência. É usada para análise de textura, detecção de padrões e compressão de imagem.

- **Morfologia matemática:** uma técnica que lida com a forma e estrutura dos objetos em uma imagem. Ela é usada em tarefas como remoção de ruído, segmentação e extração de características.
- **Processamento em tempo real:** refere-se ao processamento de imagens à medida que são adquiridas, geralmente com o objetivo de tomada de decisão em tempo real, como em aplicações de visão computacional.
- **Aprendizado de máquina em processamento de imagem:** o uso de algoritmos de aprendizado de máquina, como redes neurais convolucionais (CNNs), para tarefas de análise de imagem, como reconhecimento de objetos e detecção de padrões.

Esses são alguns dos termos e conceitos básicos no campo do processamento de imagem. Compreender esses fundamentos é crucial para trabalhar com imagens digitais e aplicar técnicas avançadas para análise, melhoramento e interpretação de imagens.

## 1.2 Percepção visual (sistema biológico versus computacional)

A percepção visual é uma capacidade complexa que tanto os sistemas biológicos quanto os sistemas computacionais têm em comum. Embora os sistemas biológicos, como o sistema visual humano, tenham sido a inspiração para muitas abordagens em processamento de imagem computacional, eles também têm diferenças significativas. Vamos explorar as semelhanças e diferenças entre esses dois sistemas.

### Sistema biológico (percepção visual humana)

A percepção visual humana refere-se ao processo pelo qual os seres humanos interpretam e compreendem as informações visuais que recebem por meio de seus olhos.

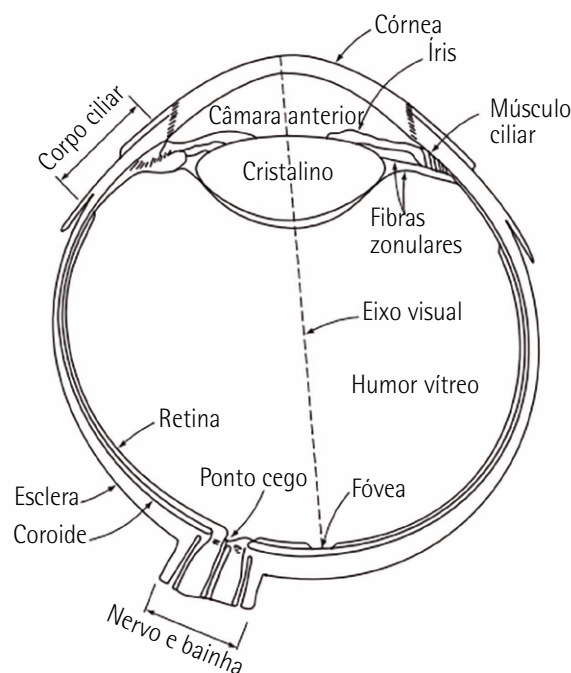


Figura 1 – Diagrama simplificado de um corte transversal do olho humano

Fonte: Gonzalez e Woods (2010, p. 23).

No livro *Processamento digital de imagens*, Gonzalez e Woods (2010) fizeram uma comparação entre a câmera fotográfica e o olho humano na formação da imagem:

Em uma câmera fotográfica comum, a lente tem uma distância focal fixa, e a focalização para diferentes distâncias é obtida variando a distância entre a lente e o plano-imagem, onde o filme (ou o chip de captura de imagem, no caso de uma câmera digital) se localiza. No olho humano, ocorre o oposto: a distância entre a lente e o plano-imagem (a retina) é fixa, e a distância focal necessária para atingir uma focalização adequada é obtida variando o formato do cristalino (que equivale a uma lente flexível).

A percepção visual humana é um campo de estudo complexo que envolve a análise e a interpretação de estímulos visuais, incluindo a percepção de formas, cores, profundidade, movimento e objetos no ambiente. Podemos listar as diversas etapas e processos:

- **Captura da luz:** a primeira etapa envolve a captação da luz pelos olhos, que atuam como dispositivos ópticos, focalizando a luz que passa pela córnea e pelo cristalino para formar uma imagem na retina.
- **Transdução:** as células da retina, chamadas de fotorreceptores (cones e bastonetes), transformam os estímulos luminosos em sinais elétricos que podem ser processados pelo cérebro.
- **Processamento visual:** os sinais elétricos gerados pelos fotorreceptores são transmitidos ao longo do nervo óptico e entram no cérebro, onde passam por uma série de estágios de processamento em áreas como o córtex visual. Durante esse processamento, as informações visuais são organizadas e interpretadas.
- **Organização visual:** a percepção visual envolve a organização de elementos visuais, como linhas, formas, cores e texturas, para criar uma representação coerente do mundo ao nosso redor. Isso inclui a identificação de objetos, rostos e cenas.
- **Percepção de profundidade:** o cérebro é capaz de perceber a profundidade por meio de pistas visuais, como o paralaxe (do grego *paralaxis*, mudança), sombreamento, perspectiva e tamanho relativo. Isso nos permite perceber objetos como estando mais próximos ou mais distantes.
- **Reconhecimento de cores:** os humanos percebem cores com base na sensibilidade dos cones da retina a diferentes comprimentos de onda de luz. Essa percepção de cores é essencial para distinguir objetos e entender o ambiente.
- **Percepção de movimento:** o cérebro é capaz de perceber o movimento por meio da detecção de mudanças na posição dos objetos ao longo do tempo. Isso é fundamental para perceber objetos em movimento e acompanhar eventos visuais dinâmicos.

O sistema visual humano é altamente paralelo, o que significa que processa informações de várias partes da cena simultaneamente. Isso permite que os humanos percebam rapidamente objetos, cores, movimentos e profundidade em seu campo de visão. O cérebro humano é especializado na detecção de padrões e na interpretação de informações visuais. Reconhecimento de faces, objetos e leitura de texto são habilidades complexas que os humanos executam naturalmente.

O sistema visual humano é robusto em relação às variações de iluminação. As pessoas conseguem reconhecer objetos em diferentes condições de luz, adaptando-se automaticamente às mudanças de iluminação. A percepção visual humana é capaz de ser influenciada pelo contexto. As pessoas podem usar o contexto para entender melhor as cenas e tomar decisões com base nisso. Os olhos humanos têm alta resolução e são sensíveis a uma ampla gama de cores. Isso permite uma percepção rica de detalhes em imagens.

A percepção visual humana é fundamental na compreensão do mundo ao nosso redor e é essencial para atividades cotidianas, como navegar no ambiente, reconhecer rostos, ler e interpretar informações visuais. Ela também é um campo de estudo importante em disciplinas como psicologia, neurociência, design gráfico e visão computacional.

## **Sistema computacional (processamento de imagem computacional)**

O processamento de imagem em sistemas computacionais é geralmente determinístico e segue algoritmos predefinidos. O processamento é controlado por programação, e os resultados são consistentes. As máquinas de processamento de imagem podem executar tarefas com alta precisão e repetibilidade. Elas podem medir com precisão intensidades de pixel, identificar padrões e executar cálculos complexos.

Sistemas de processamento de imagem podem processar imagens em tempo real, o que é essencial em aplicações como visão computacional, inspeção de qualidade e veículos autônomos. Ao contrário dos sistemas biológicos, os sistemas de processamento de imagem computacional não têm uma compreensão natural do contexto. Eles dependem de regras e algoritmos para interpretar as imagens.

Sistemas computacionais são sensíveis às condições de iluminação e podem exigir iluminação controlada para obter resultados precisos.

## **Integração de ambos os sistemas**

Há uma tendência crescente de integrar aspectos da percepção visual humana em sistemas de processamento de imagem computacional. Isso é conhecido como "visão computacional inspirada na biologia" e envolve o uso de algoritmos que simulam algumas das habilidades perceptivas humanas, como o reconhecimento de objetos em cenas complexas.

A percepção visual é uma área de sobreposição entre sistemas biológicos e computacionais. Embora os sistemas biológicos sejam altamente adaptativos e flexíveis, os sistemas computacionais oferecem precisão, repetibilidade e processamento em tempo real. A combinação dessas abordagens tem o potencial de criar sistemas de visão artificial mais poderosos e versáteis em áreas como medicina, automação industrial e tecnologia autônoma.



### 1.3 Fundamentos de imagens digitais e suas características

As imagens digitais são representações numéricas de cenas ou objetos visuais e formam a base do processamento de imagem. Compreender os fundamentos das imagens digitais e suas características é essencial para realizar tarefas de processamento de imagem de maneira eficaz. Aqui estão os principais fundamentos e características das imagens digitais:

- **Pixels:** as imagens digitais são compostas de pixels, que são os elementos de menor resolução em uma imagem. Cada pixel é representado por um valor numérico que denota sua cor ou intensidade. A resolução da imagem determina o número de pixels em largura e altura.

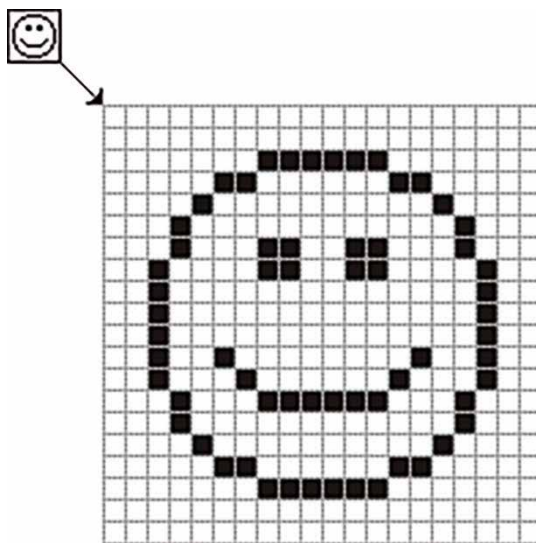


Figura 2 – Exemplo mostra imagem de rosto sorridente 16x16 pixel

Disponível em: <https://tinyurl.com/ycx6nfk8>. Acesso em: 24 out. 2023.

- **Cores:** em imagens coloridas, cada pixel é composto de três canais de cores: vermelho (R), verde (G) e azul (B). A combinação desses canais em diferentes intensidades cria uma ampla gama de cores. Em imagens em escala de cinza, há apenas um canal representando a intensidade luminosa.

Vamos exemplificar os canais de cores criando um programa em Python que utiliza a biblioteca Matplotlib para exibir os três canais de cores RGB de uma imagem.



#### Observação

Neste exemplo e nos próximos, carregamos as imagens necessárias para o entendimento do conceito estudado. Ao executar os programas você deve substituir "sua\_imagem.jpg" pelo caminho da imagem deseja utilizar. Certifique-se sempre que a imagem está no local que você definiu e o caminho está indicado corretamente no programa.



```
1 import matplotlib.pyplot as plt
2 import cv2

3 # Carregue a imagem
4 imagem = cv2.imread('sua_imagem.jpg')

5 # Separe os canais RGB
6 canal_r = imagem[:, :, 2]
7 canal_g = imagem[:, :, 1]
8 canal_b = imagem[:, :, 0]

9 # Mostra as imagens
10 plt.rcParams.update({'font.size': 20})
11 plt.figure(figsize=(15, 10))

12 plt.subplot(221) # Subplot para imagem original
13 plt.imshow(cv2.cvtColor(imagem, cv2.COLOR_BGR2RGB))
14 plt.title('Imagem original')
15 plt.axis('off')

16 plt.subplot(222) # Subplot para o canal vermelho
17 plt.imshow(canal_r, cmap='Reds')
18 plt.title('Canal Vermelho')
19 plt.axis('off')

20 plt.subplot(223) # Subplot para o canal verde
21 plt.imshow(canal_g, cmap='Greens')
22 plt.title('Canal Verde')
23 plt.axis('off')

24 plt.subplot(224) # Subplot para o canal azul
25 plt.imshow(canal_b, cmap='Blues')
26 plt.title('Canal Azul')
27 plt.axis('off')

28 plt.tight_layout() # Para garantir que os subplots não se sobreponham
29 plt.show()
```

Saída:

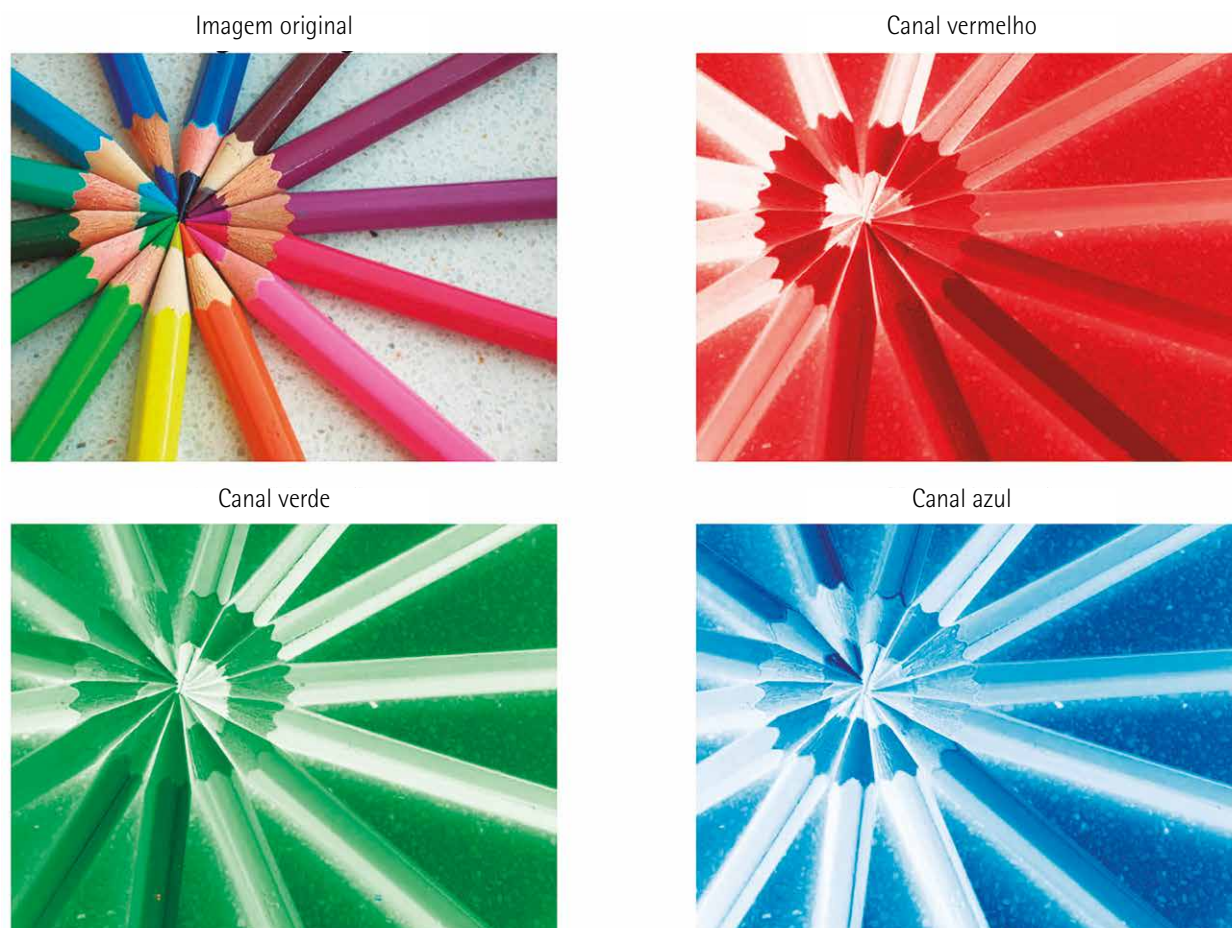


Figura 3 – Lápis coloridos

Adaptada de: <https://tinyurl.com/4tyfr36s>. Acesso em: 24 out. 2023.

**Resolução:** a resolução de uma imagem digital refere-se à quantidade de detalhes que pode ser representada. Imagens de alta resolução têm mais pixels por unidade de área, o que resulta em maior detalhe, enquanto imagens de baixa resolução parecem mais granuladas.



Figura 4 – Exemplo de variação da resolução da maior para a menor (esquerda para direita)

Disponível em: <https://tinyurl.com/2xnsbyy3>. Acesso em: 24 out. 2023.

- **Profundidade de bits:** a profundidade de bits determina a quantidade de informação que cada pixel pode conter. Por exemplo, em uma imagem de 8 bits por canal de cor, cada canal pode representar 256 cores diferentes ( $2^8$ ), enquanto em uma imagem de 16 bits por canal, podem ser representadas 65.536 cores ( $2^{16}$ ).
- **Formatos de imagem:** existem vários formatos de imagem, como JPEG, PNG, GIF e TIFF. Cada formato possui suas próprias características e níveis de compressão, afetando a qualidade da imagem e o espaço de armazenamento necessário.
- **Tamanho do arquivo:** o tamanho de um arquivo de imagem digital depende da resolução, profundidade de bits e formato. Imagens de alta resolução com mais bits por pixel tendem a ter arquivos maiores.
- **Espaço de cor:** o espaço de cor descreve a gama de cores que uma imagem pode representar. Exemplos incluem o espaço de cor RGB (vermelho, verde, azul) e o espaço de cor CMYK (ciano, magenta, amarelo, preto) usados na impressão.
- **Relação de aspecto:** a relação de aspecto é a proporção entre a largura e a altura de uma imagem. É importante para a exibição adequada da imagem em diferentes dispositivos e formatos de exibição.
- **Ruído:** o ruído é uma interferência indesejada que pode aparecer em imagens digitais devido a condições de aquisição, compressão ou outros fatores. O processamento de imagem muitas vezes envolve a redução de ruído para melhorar a qualidade da imagem. Exemplos de ruídos em imagens:
  - **Ruído sal e pimenta:** o ruído sal e pimenta, às vezes chamado de ruído impulsivo, é um tipo de distorção de imagem que ocorre quando pixels individuais são alterados aleatoriamente para valores extremos em uma imagem. Esses valores extremos geralmente são 0 (preto) e 255 (branco) em imagens em tons de cinza. Essa distorção cria pequenas manchas escuras (sal) e pontos claros (pimenta) na imagem, como se sal e pimenta tivessem sido jogados na imagem, daí o nome.

O ruído sal e pimenta é comum em imagens adquiridas por meio de sensores ou câmeras com problemas, como sensores de câmeras defeituosos, interferências elétricas, transmissões de dados com perda ou problemas no armazenamento de imagens. Esse tipo de ruído pode prejudicar a qualidade das imagens e dificultar a análise ou o processamento posterior.

A remoção do ruído sal e pimenta geralmente envolve técnicas de filtragem de imagem, como filtros de mediana, que substituem o valor do pixel afetado pela mediana dos valores dos pixels vizinhos. Isso ajuda a eliminar os pontos de ruído sal e pimenta, restaurando a qualidade da imagem. A escolha da técnica de filtragem depende do grau e da natureza do ruído na imagem.

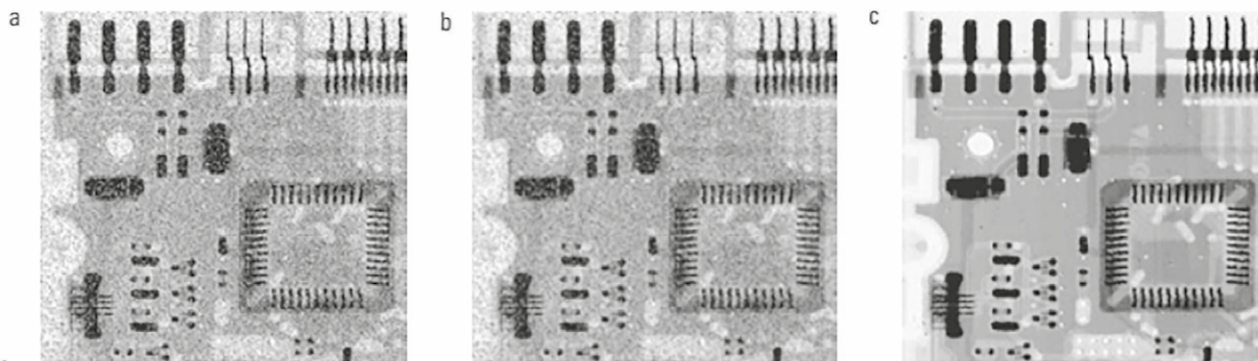


Figura 5 – (a) Imagem de raios X de uma placa de circuito corrompida pelo ruído sal e pimenta; (b) Redução de ruído com um filtro de média  $3 \times 3$ ; (c) Redução de ruído com um filtro de mediana  $3 \times 3$

Fonte: Gonzalez e Woods (2010, p. 103).

- **Ruído gaussiano:** o ruído gaussiano ou ruído gaussiano aditivo é um tipo comum de ruído presente em imagens e sinais. Ele é chamado de "aditivo" porque é causado pela adição de valores aleatórios, que seguem uma distribuição gaussiana (normal), a cada pixel ou amostra do sinal. Esse tipo de ruído é uma simplificação útil para modelar distorções que ocorrem em sistemas de aquisição de imagem, transmissões de dados ou processamento de sinal.

Os valores de ruído seguem uma distribuição normal (gaussiana), o que significa que a maioria dos valores está centrada em torno de um valor médio, com desvio padrão controlando a dispersão dos valores. Cada valor de pixel é afetado de forma independente, e os valores de ruído são independentes e aleatórios em cada ponto da imagem ou sinal. A distribuição do ruído é simétrica em relação à média, o que significa que a probabilidade de valores positivos e negativos é igual.

O ruído gaussiano aditivo pode resultar em variações suaves e contínuas nos valores dos pixels, tornando-o menos perceptível do que outros tipos de ruído, como o ruído sal e pimenta.

Para remover ou reduzir o ruído gaussiano aditivo em uma imagem ou sinal, técnicas de filtragem de imagem são frequentemente aplicadas. Um dos filtros mais comuns para reduzir o ruído gaussiano é o filtro de média, que substitui o valor de cada pixel pela média ponderada dos valores dos pixels vizinhos. Outras técnicas, como filtragem por mediana e filtragem por desvio padrão, também podem ser aplicadas, dependendo das características da imagem e do nível de ruído.

A remoção eficaz do ruído gaussiano aditivo é importante em áreas como processamento de imagens médicas, visão computacional, processamento de sinais e comunicações, em que a qualidade dos dados é fundamental para a precisão das análises e interpretações.



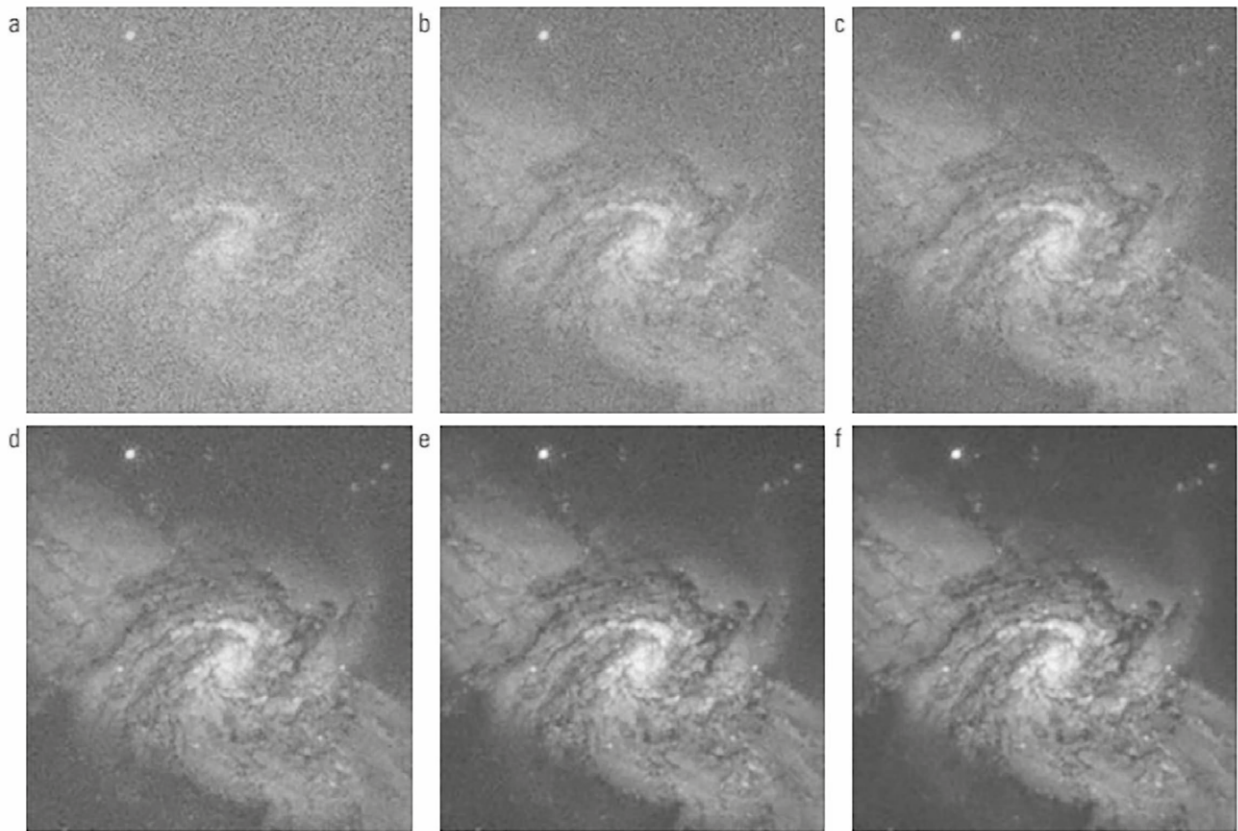


Figura 6 – (a) Imagem do par de galáxias NGC 3314 corrompida pelo ruído gaussiano aditivo; (b) a (f) Resultados do cálculo da média de 5, 10, 20, 50 e 100 imagens ruidosas, respectivamente

Fonte: Gonzalez e Woods (2010, p. 49).

- **Escala de cinza e cores:** as imagens podem ser representadas em escala de cinza (tons de preto e branco) ou em cores. A escolha depende da aplicação e da informação que se deseja extrair da imagem.
- **Compressão de imagem:** a compressão de imagem é o processo de reduzir o tamanho do arquivo da imagem, minimizando a perda de qualidade. Existem técnicas de compressão com perdas e sem perdas.

Compreender esses fundamentos e características das imagens digitais é crucial para realizar tarefas de processamento de imagem, como realce, segmentação, reconhecimento de padrões e análise de imagem. O conhecimento desses conceitos ajuda os profissionais a escolher as técnicas apropriadas para atender às necessidades específicas de cada aplicação.

### 1.4 Fontes de dados

As fontes de dados em processamento de imagem referem-se às diferentes origens de imagens digitais que podem ser usadas para análise, processamento e tomada de decisões. Ter acesso a fontes de dados relevantes e de alta qualidade é fundamental para o sucesso de muitas aplicações de processamento de imagem. Aqui estão algumas das principais fontes de dados em processamento de imagem:

- **Câmeras digitais:** câmeras fotográficas digitais, câmeras de vídeo e até mesmo câmeras de smartphones são fontes comuns de imagens digitais. Elas são usadas em diversas aplicações, como fotografia, videovigilância, reconhecimento facial e automação industrial.



Figura 7 – Foto de nuvens (1,600 × 1,200 pixels). Modelo da câmera: GT-S5360B

Disponível em: <https://tinyurl.com/4rtkxnf>. Acesso em: 24 out. 2023.

- **Sensores remotos:** satélites, aeronaves e drones equipados com sensores remotos capturam imagens de superfícies terrestres, oceânicas e atmosféricas. Essas imagens são usadas em sensoriamento remoto para monitorar o clima, a agricultura, o meio ambiente e muito mais.

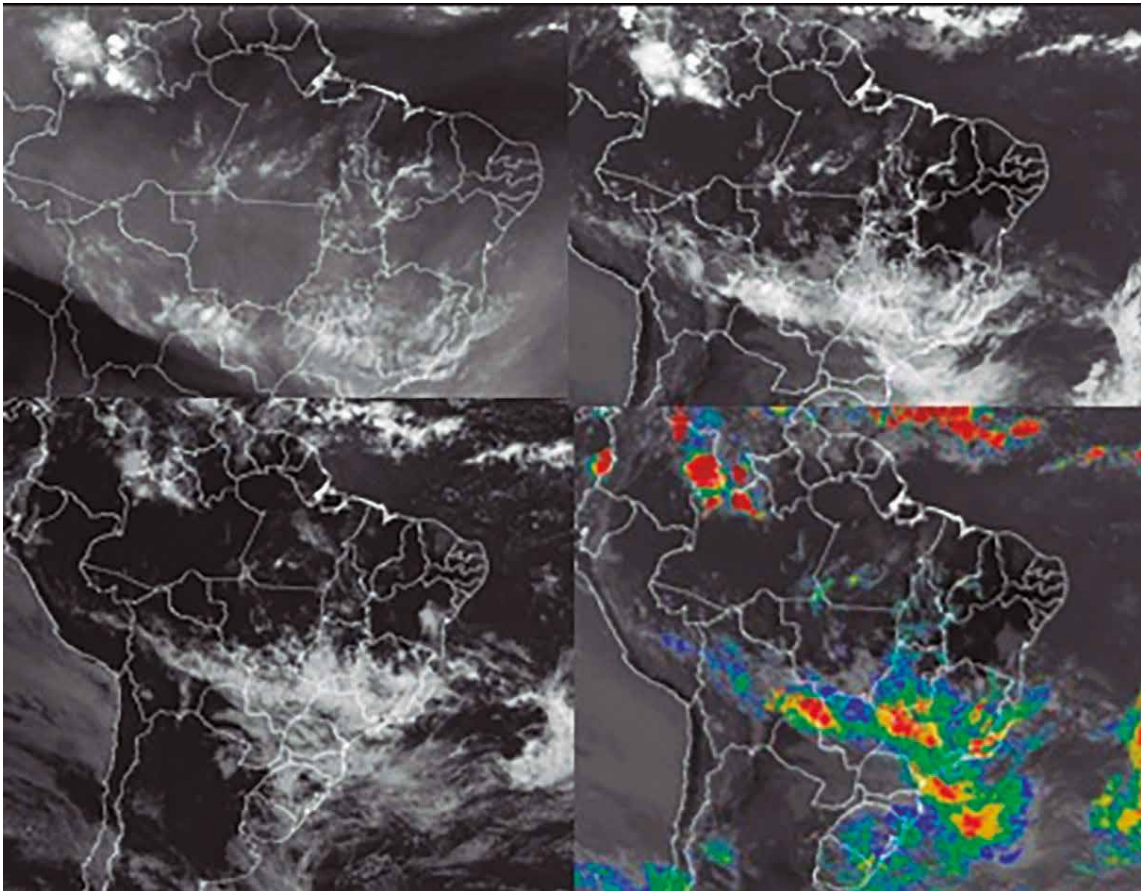


Figura 8 – Exemplo de imagem gerada pelo infravermelho termal do satélite Goes

Disponível em: <https://tinyurl.com/4n7w366f>. Acesso em: 24 out. 2023.



### Saiba mais

O Geostationary Operational Environmental Satellite (Goes) é uma série de satélites meteorológicos operados pela Administração Nacional Oceânica e Atmosférica dos Estados Unidos (NOAA) e pela Agência Espacial Federal dos Estados Unidos (NASA). Esses satélites são projetados para monitorar e fornecer informações em tempo real sobre as condições meteorológicas e ambientais na região da América do Norte e ao redor do globo. Para saber mais sobre esse satélite, acesse:

NOAA. *GOES Image Viewer*. [s.d.].

Disponível em: <https://tinyurl.com/3bhswuun>. Acesso em: 24 out. 2023.



**Imagens médicas:** equipamentos médicos, como ressonância magnética (MRI), tomografia computadorizada (CT) e ultrassonografia, geram imagens que são cruciais para diagnóstico médico, planejamento cirúrgico e pesquisa biomédica.

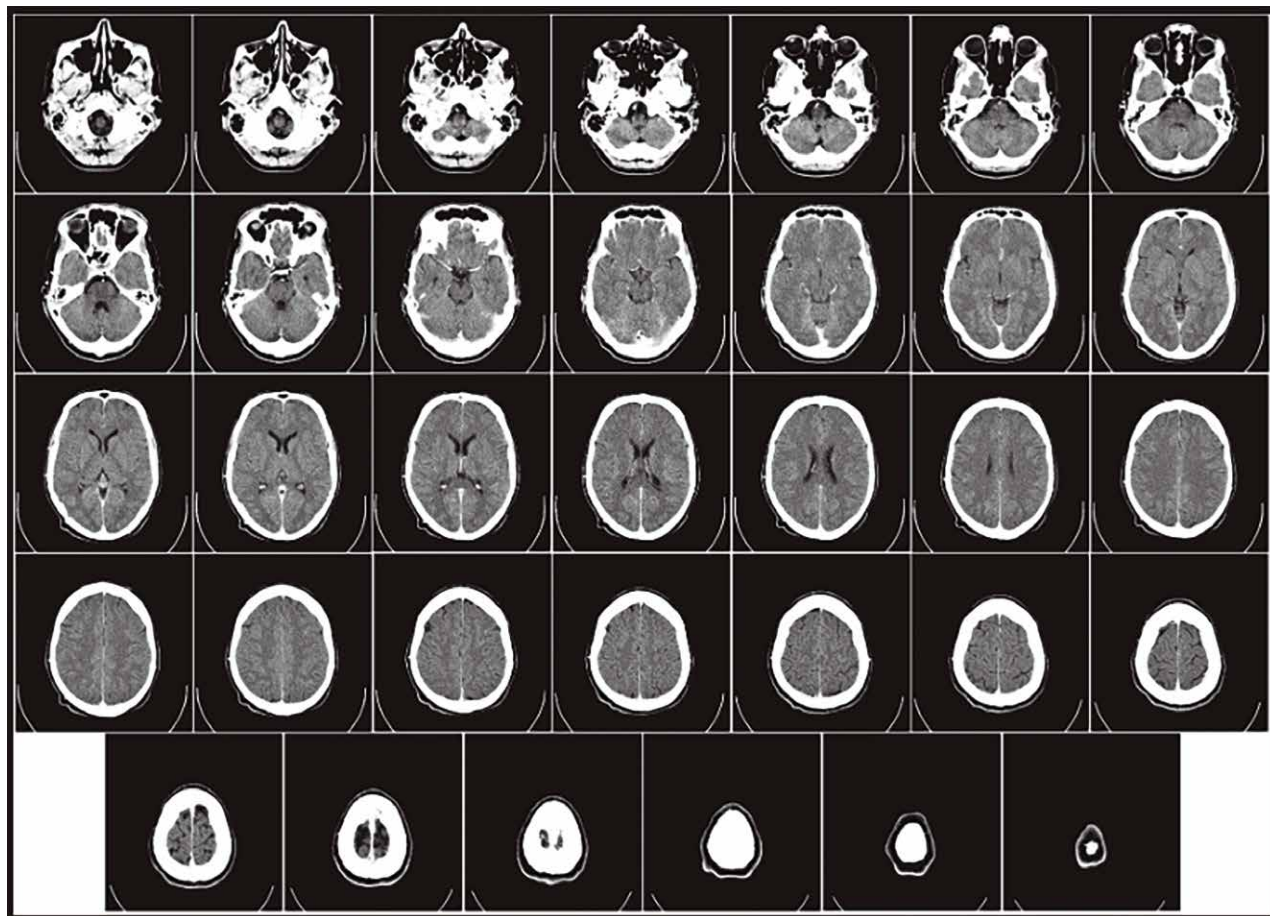


Figura 9 – Tomografia computadorizada do cérebro humano

Disponível em: <https://tinyurl.com/3b8xtyuu>. Acesso em: 24 out. 2023.

- **Bancos de dados públicos:** muitos bancos de dados públicos disponibilizam imagens digitais para fins de pesquisa. Podemos mencionar como exemplos o banco de dados ImageNet (<https://www.image-net.org/index.php>) para reconhecimento de objetos e o banco de dados de faces da Universidade de Massachusetts (LFW) (<https://vis-www.cs.umass.edu/lfw/index.html>).
- **Imagens de microscopia:** em ciências biológicas e materiais, microscópios geram imagens de alta resolução de células, tecidos e materiais em escalas micro e nanométricas. Essas imagens são usadas em pesquisa e diagnóstico.



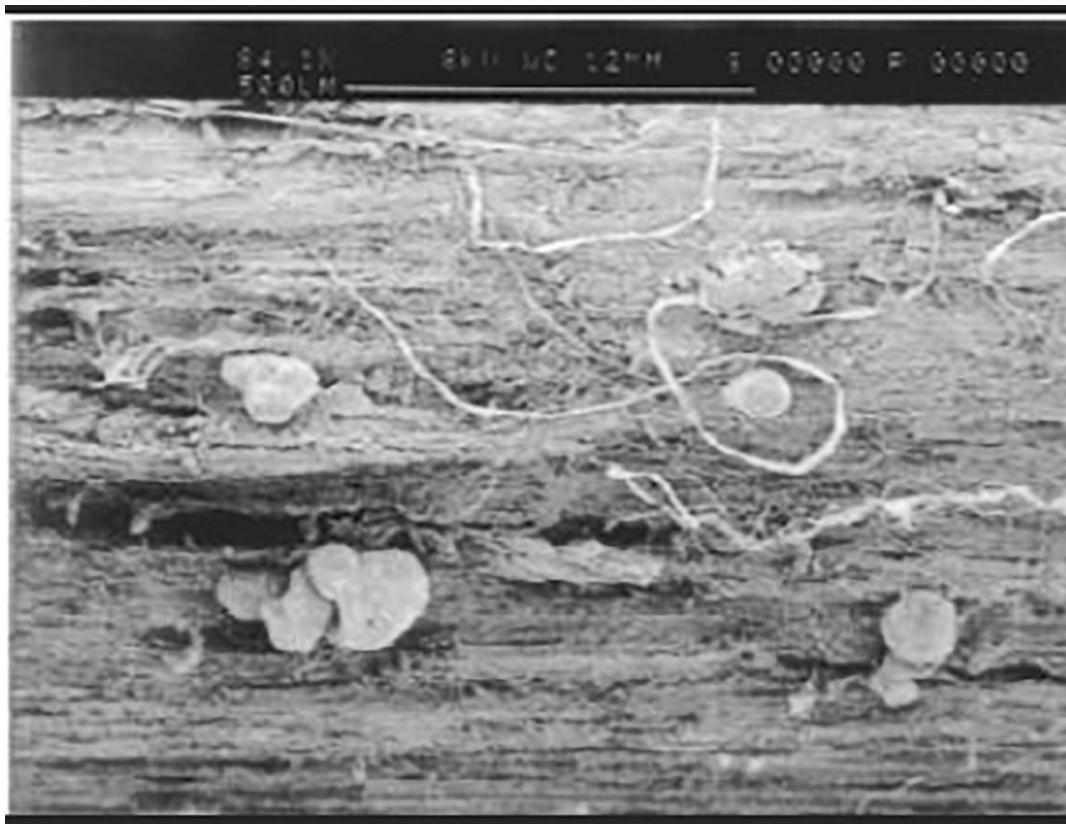


Figura 10 – Os escleródios (esporos) de *Macrophomina phaseolina* vistos por um microscópio eletrônico de varredura (MEV)

Disponível em: <https://tinyurl.com/yubfkwzr>. Acesso em: 24 out. 2023.

- **Imagens de segurança e vigilância:** câmeras de vigilância em locais públicos, edifícios e empresas capturam imagens para fins de segurança. O processamento de imagem é usado para detecção de intrusos, análise de comportamento e reconhecimento de placas de veículos.
- **Imagens de satélite:** satélites de observação da Terra fornecem imagens de alta resolução que são utilizadas em cartografia, monitoramento ambiental, planejamento urbano e agricultura de precisão. Por exemplo, ao entrar no site do Instituto Nacional de Pesquisas Espaciais (INPE – <http://www.dgi.inpe.br/catalogo/explore>) você pode acessar cenas de uma série de satélites (AQUA, TERRA, S-NPP, UK-DMC-2, LANDSAT-1, LANDSAT-2, LANDSAT-3, LANDSAT-5, LANDSAT-7, LANDSAT-8, CBERS-2, CBERS-2B, CBERS-4 – Satélite Sino-Brasileiro de Recursos Terrestres, RESOURCESAT-1 e RESOURCESAT-2).

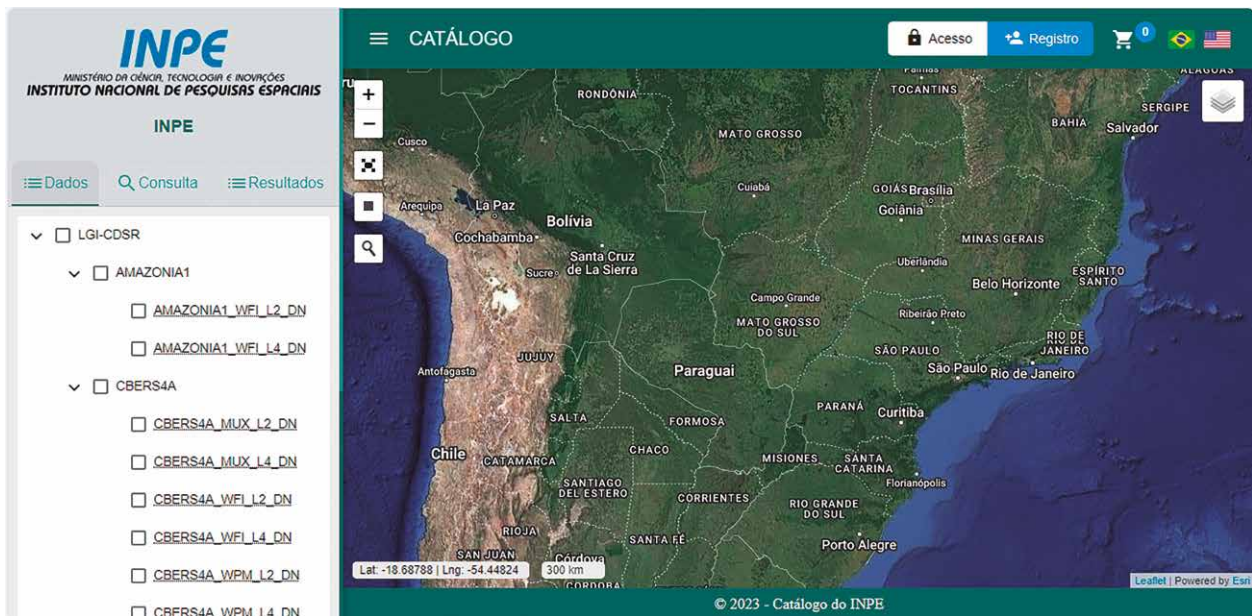


Figura 11 – Site do Instituto Nacional de Pesquisas Espaciais (INPE)

Disponível em: <https://tinyurl.com/bpjsyb2>. Acesso em: 24 out. 2023.

- **Imagens de redes sociais:** plataformas de redes sociais como o Instagram, o Facebook e o Twitter hospedam uma enorme quantidade de imagens compartilhadas pelos usuários. Essas imagens podem ser usadas para análise de tendências e reconhecimento de objetos e sentimentos.
- **Câmeras de veículos autônomos:** carros autônomos e veículos equipados com sistemas de assistência ao motorista capturam continuamente imagens para navegação e segurança. Essas imagens são processadas em tempo real para tomar decisões de direção.

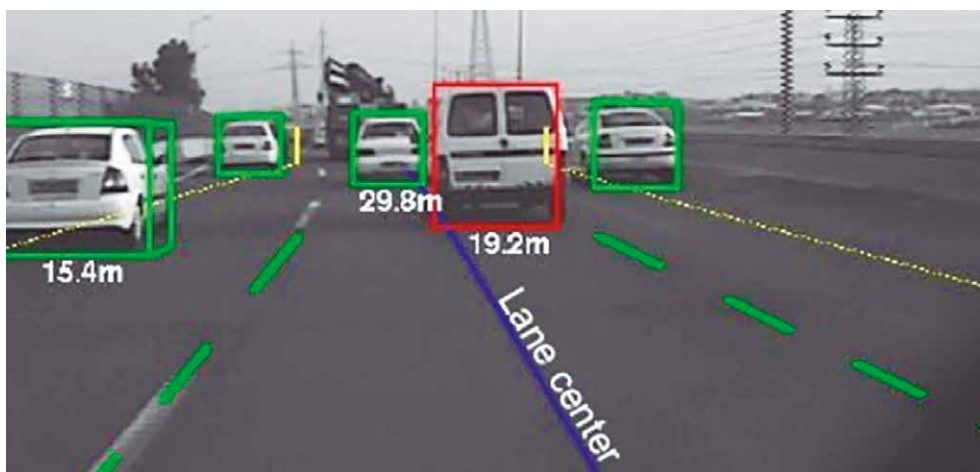


Figura 12 – Uma amostra da câmera com visão artificial em tempo real com detecção de objetos para condução autônoma

Fonte: Abualhoul, Munoz e Nashashibi (2018, p. 4).

- **Imagens históricas e arquivadas:** bibliotecas, museus e arquivos históricos mantêm coleções de imagens antigas e documentos digitalizados. Essas imagens podem ser usadas para estudos históricos, conservação e pesquisa.
- **Imagens geradas por computador (CGI):** em aplicações de entretenimento, como animações e jogos, as imagens são frequentemente criadas por computador. O processamento de imagem pode ser usado para melhorar a qualidade visual dessas imagens.



Figura 13 – Imagem gerada por computador do jogo Minecraft

Disponível em: <https://tinyurl.com/32h4kh9s>. Acesso em: 24 out. 2023.

- **Imagens de documentos digitalizados:** documentos, manuscritos e fotos antigas são digitalizados para preservação e compartilhamento. O processamento de imagem ajuda a melhorar a qualidade e a legibilidade dessas imagens digitalizadas.

Selecionar a fonte de dados correta é crucial para atender às necessidades específicas de uma aplicação de processamento de imagem. Além disso, é importante considerar a qualidade da imagem, a resolução, a precisão e a ética em relação à coleta e uso das imagens, especialmente em contextos sensíveis, como reconhecimento facial e segurança de dados.

## 1.5 Canais de imagem (espaço RGB, CMY, CMYK e HSV)

Os canais de imagem são componentes individuais que compõem uma imagem colorida ou em tons de cinza. Dois espaços de cores comuns usados para representar esses canais são o espaço RGB (Red, Green, Blue) e o espaço HSV (Hue, Saturation, Value). Cada espaço de cores oferece uma maneira única de representar e manipular cores em imagens.

### Espaço RGB (Red, Green, Blue)

Seus componentes são:

- **Canal vermelho (R)**: representa a quantidade de vermelho em cada pixel da imagem. Quanto mais intenso o valor no canal vermelho, mais forte é a contribuição da cor vermelha naquele pixel.
- **Canal verde (G)**: representa a quantidade de verde em cada pixel. Semelhante ao canal vermelho, valores mais altos indicam uma presença maior da cor verde.
- **Canal azul (B)**: representa a quantidade de azul em cada pixel. Valores maiores no canal azul indicam uma presença maior da cor azul.

Combinando os três canais RGB em diferentes intensidades, podemos criar uma ampla gama de cores e tons. A mistura de vermelho, verde e azul em intensidades diferentes permite a representação de qualquer cor visível. Por exemplo, uma intensidade máxima nos canais R, G e B resulta em branco, enquanto nenhuma intensidade nos três canais resulta em preto.

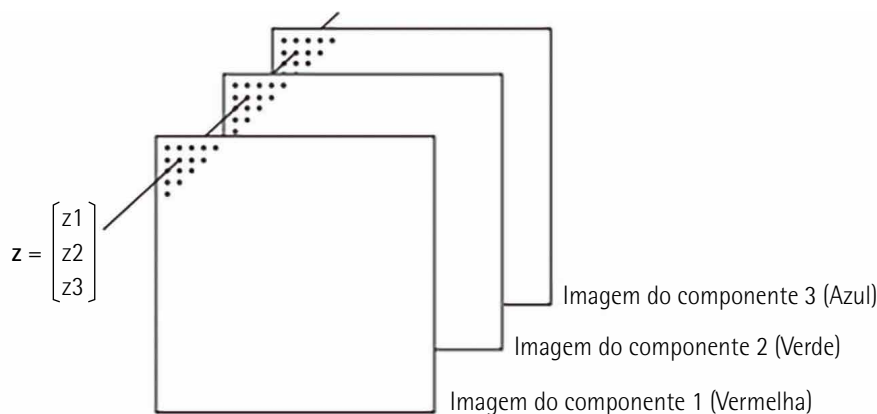


Figura 14 – Formação de um vetor a partir de valores de pixels correspondentes em três imagens de componentes RGB

Fonte: Gonzalez e Woods (2010, p. 60).

### Espaço CMY (Ciano, Magenta e Amarelo)

O espaço de cores CMY é um modelo de cores subtrativas. Isso significa que as cores são criadas subtraindo-se cores de uma fonte de luz branca. Os componentes principais desse espaço de cores são:

- **Ciano (C)**: representa a cor azul-esverdeada.
- **Magenta (M)**: representa a cor magenta, um rosa profundo.
- **Amarelo (Y)**: representa a cor amarela.



Ao misturar essas três cores primárias, é possível criar uma vasta gama de cores. No entanto, o CMY não é perfeito para reproduzir cores muito escuras, pois a sobreposição completa de ciano, magenta e amarelo não produz preto puro, mas uma tonalidade escura.

### CMYK (Ciano, Magenta, Amarelo e Preto)

O espaço de cores CMYK é uma extensão do CMY que inclui um quarto componente, o preto (K). A adição do preto permite uma reprodução mais precisa das cores escuras e é particularmente útil na impressão gráfica. Os componentes do CMYK são:

- ciano (C);
- magenta (M);
- amarelo (Y);
- preto (K).

Ao usar o CMYK, os designers podem criar uma ampla variedade de cores, incluindo tons mais escuros e profundos, que seriam difíceis de obter apenas com o CMY.

A principal diferença entre os espaços de cores CMY e CMYK é que o primeiro é adequado para exibição em telas de computador e projetores, enquanto o CMYK é o padrão para impressão. Quando você cria um design digital para ser impresso, geralmente converte suas cores do espaço de cores RGB para o espaço de cores CMYK para garantir uma correspondência mais precisa entre a tela do computador e a impressão final.

O espaço de cores CMY e o CMYK são sistemas fundamentais para o gerenciamento de cores na indústria gráfica e no design, permitindo a representação precisa de cores em várias mídias e garantindo que os projetos gráficos sejam reproduzidos fielmente durante o processo de impressão.

### Espaço HSV (Hue, Saturation, Value)

- **Canal matiz (Hue – H):** o canal de matiz representa a cor pura ou o tom da imagem. Ele é medido em graus de 0 a 360, onde 0 (ou 360) representa o vermelho, 120 representa o verde e 240 representa o azul. Os outros valores correspondem a cores intermediárias.
- **Canal saturação (Saturation – S):** a saturação mede a pureza ou intensidade da cor. Valores baixos de saturação indicam tons de cinza ou cores desbotadas, enquanto valores altos indicam cores vibrantes.
- **Canal valor (Value – V):** o canal de valor representa a intensidade da cor ou o brilho. Ele determina o quão claro ou escuro um pixel é. Valores mais altos correspondem a pixels mais brilhantes.

Na figura a seguir, apresentamos em (a) a imagem composta das cores RGB primárias e secundárias. De (b) até (d), apresentamos as imagens monocromáticas dos componentes H (matiz), S (saturação) e I (intensidade) dessa imagem.

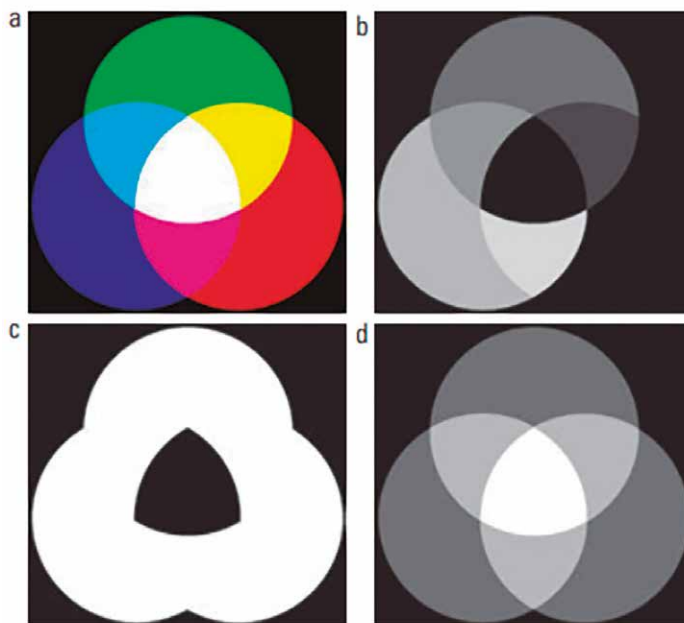


Figura 15 – (a) Imagem RGB e os componentes de sua imagem HSI correspondente: (b) matiz, (c) saturação e (d) intensidade

Fonte: Gonzalez e Woods (2010, p. 273).

O espaço de cores HSV é útil para muitas tarefas de processamento de imagem, como segmentação de objetos com base em cor, ajuste de brilho e contraste e detecção de cores em uma imagem. Ele separa a informação de cor, intensidade e matiz, tornando mais fácil trabalhar com essas características de forma individual.

Os canais de imagem nos espaços de cores RGB e HSV são componentes essenciais para representar e manipular informações de cor em processamento de imagem. Eles desempenham um papel importante em uma variedade de aplicações, desde o ajuste de cores em fotos até a detecção de objetos com base em cores em visão computacional. A escolha do espaço de cores e a manipulação dos canais dependem das necessidades específicas da tarefa de processamento de imagem.



### Saiba mais

Para saber mais sobre os fundamentos das cores e os modelos de cores, recomendamos a leitura do capítulo 6 do livro a seguir:

GONZALEZ, R. G.; WOODS, R. E. *Processamento digital de imagens*. São Paulo: Pearson, 2010.

## 1.6 Variação de canais

A variação de canais refere-se à manipulação dos canais de cores em uma imagem para obter diferentes efeitos visuais ou para extrair informações específicas. Isso é feito principalmente em imagens coloridas, em que cada pixel é composto de informações de três canais de cores principais: vermelho (R), verde (G) e azul (B) no espaço de cores RGB (Red, Green, Blue). A variação de canais é uma técnica poderosa que permite uma série de manipulações interessantes em imagens. Podemos utilizar a variação de canais em processamento de imagem para:

- **Realce de canais individuais:** aumentar ou diminuir a intensidade de um canal específico pode realçar ou atenuar uma cor na imagem. Por exemplo, aumentar o canal vermelho pode realçar áreas vermelhas em uma foto.
- **Separação de canais:** às vezes, é útil isolar um canal específico para análise. Por exemplo, no diagnóstico médico, pode-se isolar o canal verde em uma imagem médica para destacar os detalhes da pele.
- **Combinação de canais:** a combinação de diferentes canais pode criar efeitos visuais interessantes. Por exemplo, combinar o canal vermelho e azul, enquanto exclui o canal verde, cria uma imagem com tons de magenta.
- **Deteção de bordas:** a detecção de bordas pode ser realizada pela subtração dos canais. Subtrair o canal verde do canal vermelho, por exemplo, pode destacar as bordas em que a cor vermelha muda para a cor verde.
- **Ajuste de cor e matiz:** a variação de canais pode ser usada para ajustar a cor global de uma imagem. Aumentar ou diminuir a intensidade de cada canal de forma independente pode alterar a aparência geral da imagem.
- **Remoção de cores indesejadas:** seletivamente, zerar um canal específico pode remover uma cor da imagem. Por exemplo, zerar o canal azul pode eliminar o componente azul, tornando a imagem com tons de sépia.
- **Filtros de canal:** aplicar filtros específicos a um único canal pode criar efeitos artísticos. Por exemplo, aplicar um filtro de suavização apenas ao canal vermelho pode criar um efeito de sonho.
- **Análise de características:** em aplicações de visão computacional, a análise de canais pode ajudar a destacar características específicas em uma imagem, como objetos vermelhos em um ambiente.
- **Correção de balanço de branco:** ajustar os canais de cores de forma independente é uma técnica comum para correção de balanço de branco, garantindo que as cores na imagem sejam representadas de forma precisa.

A variação de canais em processamento de imagem é uma técnica versátil que oferece controle granular sobre a cor e a intensidade em uma imagem. Ela é muito utilizada em fotografia, design gráfico, visão computacional e em muitas outras aplicações nas quais a manipulação de cores desempenha um papel fundamental na obtenção dos resultados desejados.

### 1.7 Resolução, cores e formatos

No processamento de imagem, a resolução, as cores e os formatos desempenham papéis cruciais, pois afetam diretamente a qualidade, a quantidade de informações e a compatibilidade das imagens digitais. Vamos explorar esses conceitos:

#### Resolução

A resolução de uma imagem digital se refere à quantidade de detalhes que ela pode representar e é expressa em pixels. Ela é determinada pela largura e altura da imagem em pixels. Por exemplo, uma imagem de 1920 × 1080 pixels possui uma resolução de 1920 pixels na largura e 1080 pixels na altura.

- **Resolução alta:** imagens com alta resolução têm mais pixels por unidade de área e, portanto, exibem mais detalhes. São usadas em aplicações que exigem alta qualidade, como impressão, fotografia profissional e visualização em telas de alta definição.
- **Resolução baixa:** imagens com baixa resolução têm menos pixels e, portanto, exibem menos detalhes. São usadas quando a economia de espaço de armazenamento ou de largura de banda é mais importante do que a qualidade visual, como em imagens para web, ícones e miniaturas.

A resolução é um fator crítico em aplicações como reconhecimento de objetos, em que a capacidade de identificar detalhes sutis pode ser crucial.

#### Cores

As cores em imagens digitais podem ser representadas de diferentes maneiras, dependendo dos espaços de cores utilizados. Os espaços de cores mais comuns incluem:

- **RGB (Red, Green, Blue):** o espaço de cores RGB representa cores combinando intensidades de vermelho, verde e azul em cada pixel. A combinação dessas intensidades cria uma ampla gama de cores.
- **CMYK (Cyan, Magenta, Yellow, Black):** muito utilizado na impressão, o espaço de cores CMYK é semelhante ao RGB, mas usa ciano, magenta, amarelo e preto para criar cores. É mais adequado para representar cores de tintas de impressão.
- **HSV (Hue, Saturation, Value):** o espaço de cores HSV separa a informação de cor, saturação e brilho, o que pode ser útil em tarefas como detecção de cor e ajuste de cores.
- **Escala de cinza:** em imagens em escala de cinza, cada pixel é representado por um único valor de intensidade luminosa, variando de preto a branco.



## Formatos

Os formatos de imagem determinam como os dados da imagem são armazenados e comprimidos. Alguns dos formatos mais comuns incluem:

- **Joint Photographic Experts Group (JPEG):** é muito usado para imagens fotográficas. Ele usa compressão com perdas, o que significa que a qualidade pode diminuir com alta compressão, mas os arquivos tendem a ser menores.
- **Portable Network Graphics (PNG):** é usado para imagens com transparência, como logotipos ou elementos gráficos. Ele usa compressão sem perdas, o que significa que a qualidade não é afetada pela compressão.
- **Graphics Interchange Format (GIF):** é usado para imagens animadas e imagens com poucas cores. Ele suporta compressão sem perdas, mas a paleta de cores é limitada.
- **Tagged Image File Format (TIFF):** é usado para imagens de alta qualidade e imagens médicas. Pode suportar tanto compressão com perdas quanto sem perdas, dependendo das configurações.

A escolha do formato depende das necessidades específicas da aplicação. Por exemplo, JPEG é adequado para fotos na web, enquanto TIFF é preferido em aplicações médicas de alta qualidade. Escolher a combinação certa de resolução, espaço de cores e formato é fundamental para atender às necessidades de diferentes aplicações, desde fotografia e design gráfico até visão computacional e medicina.

## 1.8 Manipulação matricial e representação em matriz

A manipulação matricial e a representação em matriz são conceitos fundamentais no processamento de imagem. Elas envolvem a representação de uma imagem como uma matriz numérica, em que cada elemento da matriz corresponde a um pixel na imagem. Isso permite a aplicação de uma variedade de operações matemáticas para processar e analisar imagens de forma eficiente. Vamos explorar esses conceitos:

### Representação em matriz

Uma imagem digital é composta de pixels, e cada pixel é representado como um valor numérico. Para imagens em tons de cinza, geralmente usa-se uma matriz bidimensional em que cada elemento representa a intensidade do pixel naquela posição. Quanto maior o valor, mais claro o pixel será. Por exemplo, uma matriz de  $8 \times 8$  pode representar uma pequena imagem em tons de cinza da seguinte forma:

[128, 127, 130, 132, 135, 140, 142, 144],

[129, 131, 134, 136, 138, 141, 143, 145],

[130, 132, 135, 137, 139, 142, 144, 146],

[132, 134, 136, 138, 141, 143, 145, 148],

[135, 137, 139, 142, 144, 147, 149, 151],

[139, 141, 143, 146, 148, 151, 154, 156],

[142, 144, 147, 149, 152, 155, 158, 160],

[145, 147, 150, 153, 156, 159, 162, 165]]

Para imagens coloridas, cada pixel é representado como uma combinação de três valores numéricos correspondentes aos canais de cores vermelho (R), verde (G) e azul (B), resultando em uma matriz tridimensional.

### Manipulação matricial

A manipulação matricial envolve a aplicação de operações matemáticas a essas matrizes de pixels para realizar tarefas de processamento de imagem. Algumas operações comuns incluem:

- **Realce e suavização:** aplicação de filtros (máscaras de convolução) para realçar ou suavizar áreas da imagem, melhorando a qualidade visual ou reduzindo o ruído.
- **Segmentação:** divisão da imagem em regiões com base em critérios de intensidade de pixel, cor ou outros atributos.
- **Transformações geométricas:** rotação, redimensionamento, corte e projeção de imagens.
- **Filtragem de frequência:** aplicação da transformada de Fourier para realizar filtragem de frequência em imagens, útil para detecção de bordas e realce de características.
- **Operações lógicas:** aplicação de operadores lógicos (AND, OR, NOT) para combinar ou modificar imagens.
- **Histograma:** análise do histograma de uma imagem para entender a distribuição de intensidades e ajustar o contraste.
- **Aprendizado de máquina:** utilização de algoritmos de aprendizado de máquina para classificação, detecção de objetos, reconhecimento de padrões e outras tarefas avançadas.

A manipulação matricial é a base para muitas técnicas de processamento de imagem e visão computacional. O uso eficaz dessas operações permite melhorar a qualidade das imagens, extrair informações úteis e automatizar tarefas complexas.

A representação de uma imagem como uma matriz numérica permite uma série de operações matemáticas que podem ser aplicadas para realizar tarefas diversas, desde realce de imagem até análise de padrões. Esses conceitos são essenciais para o desenvolvimento de soluções de processamento de imagem eficazes.

## 1.9 Sistema de coordenadas e manipulação de pixels

### Sistema de coordenadas

Um sistema de coordenadas é uma estrutura que define como as posições de pontos ou elementos são descritas em relação a um plano ou espaço. Em processamento de imagem, frequentemente utilizamos sistemas de coordenadas para representar a posição de pixels em uma imagem.

Existem dois sistemas de coordenadas comuns em imagens digitais:

- **Sistema de coordenadas cartesiano:** é o sistema de coordenadas bidimensional mais utilizado. Nele, uma imagem é representada como uma grade retangular de pixels. As coordenadas  $(x, y)$  indicam a posição de um pixel na horizontal ( $x$ ) e na vertical ( $y$ ) a partir de um ponto de origem no canto superior esquerdo da imagem.
- **Sistema de coordenadas polar:** este sistema é menos comum em processamento de imagem, mas é utilizado em algumas aplicações, especialmente em reconhecimento de formas ou transformações de imagens em domínio de frequência, como na transformada de Fourier.

### Manipulação de pixels

A manipulação de pixels envolve a leitura e a alteração dos valores dos pixels em uma imagem para realizar várias operações de processamento. Aqui estão algumas das operações de manipulação de pixels comuns:

- **Acesso a pixels:** para acessar um pixel em uma imagem, você especifica suas coordenadas  $(x, y)$  e lê ou altera seu valor. Isso permite a extração de informações de pixel específicas para análise.
- **Realce de imagem:** para melhorar a qualidade de uma imagem, você pode ajustar os valores dos pixels para aumentar o contraste, a nitidez ou outras características visuais.
- **Filtragem:** a aplicação de filtros envolve a convolução de uma máscara ou kernel com a matriz de pixels da imagem para realizar tarefas como suavização (borramento), detecção de bordas e realce de características.
- **Transformações geométricas:** isso inclui rotação, redimensionamento, corte e projeção de imagens. As transformações geométricas envolvem a modificação das coordenadas dos pixels.
- **Segmentação de imagem:** para detectar objetos ou regiões de interesse em uma imagem, você pode definir critérios de segmentação e atribuir valores de pixel correspondentes a essas regiões.
- **Operações lógicas:** as operações lógicas, como AND, OR e NOT, são usadas para combinar ou modificar imagens com base em condições específicas.
- **Histograma:** a análise do histograma de uma imagem envolve a contagem da frequência de valores de intensidade de pixel e pode ser usada para ajustar o contraste e o brilho.

A manipulação de pixels é a base para muitas técnicas de processamento de imagem. Ela permite a modificação das características visuais de uma imagem, a extração de informações importantes e a realização de tarefas avançadas, como detecção de objetos, reconhecimento de padrões e análise de textura.

O sistema de coordenadas e a manipulação de pixels permitem que os profissionais de imagem acessem, modifiquem e analisem imagens digitais de maneira eficaz, abrindo possibilidades para uma ampla gama de aplicações, desde edição de fotos até visão computacional e diagnóstico médico.

### 1.10 Região de interesse (ROI, do inglês region of interest)

Em processamento de imagem, uma região de interesse (ROI), traduzida do inglês como region of interest, refere-se a uma parte específica de uma imagem que é selecionada ou recortada para análise ou processamento separadamente do restante da imagem. As ROIs são ferramentas valiosas em várias aplicações de processamento de imagem, pois permitem focalizar a atenção e os recursos computacionais em áreas de interesse específicas. Alguns aspectos importantes relacionados às regiões de interesse em processamento de imagem:

- **Seleção de áreas relevantes:** as ROIs são usadas para destacar áreas de interesse em uma imagem que contenham informações relevantes para uma tarefa específica. Por exemplo, a figura a seguir apresenta um exame médico de mamografia. À esquerda vemos a imagem original e após o processamento dela temos a imagem do lado direito com a ROI (parte branca) que está destacando uma parte específica do corpo para uma análise mais detalhada.

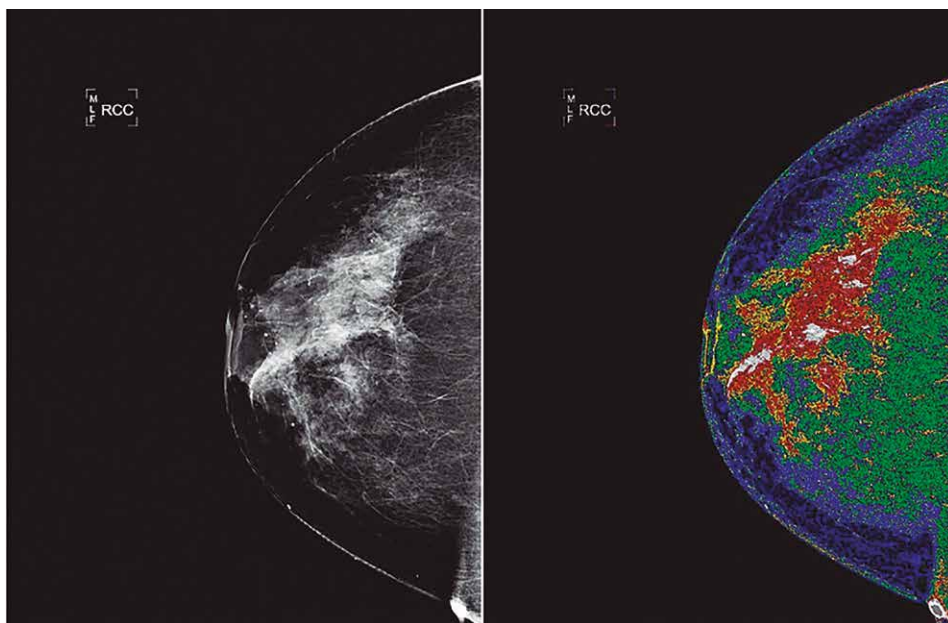


Figura 16 – A imagem à esquerda mostra uma mamografia original antes do processamento. A imagem à direita, com a região de interesse (branca) marcada, mostra uma mamografia após processamento

- **Redução de ruído e economia de recursos:** focar em uma ROI pode ajudar a reduzir o impacto de ruídos ou informações irrelevantes presentes em outras partes da imagem. Isso economiza tempo e recursos computacionais ao aplicar algoritmos de processamento apenas na área relevante.
- **Melhorias de detecção e reconhecimento:** em tarefas de detecção de objetos, reconhecimento de padrões ou análise de texto, a segmentação de uma ROI pode melhorar significativamente o desempenho, pois a concentração de recursos em uma área menor pode aumentar a precisão.
- **Análise de texturas e características locais:** em aplicações de visão computacional, como reconhecimento facial ou análise de texturas, as ROIs permitem a extração de características locais específicas, como pontos de interesse, bordas ou texturas, sem considerar informações contextuais irrelevantes.
- **Extração de recursos:** as ROIs podem ser usadas para extrair características específicas de uma imagem, como a intensidade média de uma região, histogramas de cores ou descritores de texturas.
- **Interatividade e edição de imagens:** em software de edição de imagens, as ROIs podem ser selecionadas manualmente pelo usuário para aplicar filtros, ajustes de cor ou outras transformações apenas à parte escolhida da imagem.
- **Delimitação de regiões em detecção de objetos:** em algoritmos de detecção de objetos, as ROIs frequentemente atuam como candidatos iniciais em que objetos podem estar presentes, reduzindo o escopo da busca e acelerando o processo de detecção.

As regiões de interesse são uma ferramenta versátil que desempenha um papel essencial em muitos aspectos do processamento de imagem. Sua capacidade de isolar áreas específicas de uma imagem para análise e processamento se traduz em melhorias significativas na eficácia e eficiência de muitas aplicações de visão computacional, análise de imagem e diagnóstico.

## 2 TÉCNICAS DE PRÉ-PROCESSAMENTO

As técnicas de pré-processamento em processamento de imagem referem-se a uma série de etapas realizadas antes da análise ou processamento principal de uma imagem. Essas etapas são projetadas para melhorar a qualidade da imagem, destacar características relevantes, reduzir o ruído e facilitar tarefas subsequentes, como detecção de objetos, segmentação ou reconhecimento de padrões. Vamos apresentar algumas das técnicas de pré-processamento mais comuns em processamento de imagem:

- **Redimensionamento e normalização:** o redimensionamento envolve a alteração das dimensões da imagem para um tamanho específico. Isso pode ser útil para garantir que todas as imagens tenham a mesma resolução. A normalização de intensidade de pixel ajusta os valores de pixel para uma escala específica, como 0 a 255, para garantir que todas as imagens tenham o mesmo intervalo de intensidade.

- **Conversão de espaço de cores:** converter a imagem para diferentes espaços de cores, como RGB para escala de cinza ou HSV, pode ajudar a destacar características específicas. Por exemplo, a conversão para escala de cinza é útil quando a cor não é relevante para a tarefa.
- **Correção de balanço de branco:** o balanço de branco corrige problemas de iluminação na imagem, garantindo que as cores sejam representadas de forma precisa. Isso é especialmente importante em aplicações de fotografia e visão computacional.
- **Filtros de suavização e realce:** filtros de suavização, como a média e a mediana, podem ser aplicados para reduzir o ruído na imagem. Filtros de realce, como o filtro laplaciano e o filtro de Sobel, destacam bordas e características importantes.
- **Equalização de histograma:** a equalização de histograma é usada para ajustar o contraste da imagem, redistribuindo as intensidades de pixel para ocupar toda a faixa dinâmica. Isso pode realçar detalhes e melhorar a qualidade visual.
- **Filtragem de frequência:** a transformada de Fourier e filtros de frequência podem ser aplicados para realçar ou suavizar padrões em uma imagem. Isso é útil em tarefas de processamento de imagens médicas e análise de textura.
- **Segmentação de imagem:** a segmentação é a subdivisão da imagem em regiões ou objetos. Técnicas como limiarização (thresholding) e crescimento de região são usadas para identificar áreas de interesse.
- **Remoção de fundo:** em aplicações de detecção de objetos, remover o fundo da imagem pode facilitar a detecção dos objetos de interesse.
- **Eliminação de artefatos:** artefatos de imagem, como manchas, podem ser removidos usando técnicas de restauração de imagem.
- **Deteção de borda e contorno:** a detecção de borda destaca as transições entre objetos na imagem, tornando-as mais perceptíveis. Isso é usado em detecção de objetos e análise de forma.
- **Remoção de ruído:** técnicas de remoção de ruído, como filtragem gaussiana ou filtragem bilateral, são aplicadas para reduzir a interferência de ruído na imagem.
- **Normalização de iluminação:** em imagens com variações significativas de iluminação, a normalização pode ser usada para tornar a iluminação mais uniforme.
- **Rotação e alinhamento:** quando necessário, imagens podem ser rotacionadas ou alinhadas para uma orientação específica.

Essas técnicas de pré-processamento são aplicadas de forma seletiva, dependendo da natureza da imagem e das tarefas que serão realizadas posteriormente. O objetivo principal é melhorar a qualidade

da imagem, destacar características importantes e tornar a imagem adequada para análises subsequentes, tornando o processamento de imagem mais eficaz e preciso.

## 2.1 Rotação

A rotação é uma operação fundamental que envolve a mudança da orientação de uma imagem, geralmente em um ângulo específico, mantendo seus atributos visuais. A rotação pode ser aplicada em uma grande variedade de cenários, desde a correção de orientação de imagens até a transformação de objetos em visão computacional. Os principais aspectos da rotação em processamento de imagem são:

- **Rotação de imagens:** a rotação de uma imagem envolve girá-la em torno de seu centro ou de outro ponto de referência em um ângulo específico. Isso é útil quando uma imagem foi capturada em uma orientação indesejada e precisa ser corrigida. Por exemplo, você pode rotacionar uma imagem de retrato para a orientação paisagem.
- **Interpolação:** ao realizar uma rotação, os pixels originais da imagem nem sempre se encaixarão perfeitamente em novas posições. Portanto, é necessário usar técnicas de interpolação para calcular os valores dos pixels na nova imagem. Interpolação bilinear e interpolação por vizinho mais próximo são abordagens comuns.
- **Ângulo de rotação:** o ângulo de rotação pode ser especificado em graus ou radianos, dependendo da implementação. A rotação é realizada no sentido anti-horário em relação ao ponto de referência.
- **Rotação afim versus rotação arbitrária:** a rotação afim preserva linhas retas, enquanto a rotação arbitrária pode distorcer a geometria da imagem. A rotação afim é mais comum em aplicações de processamento de imagem.
- **Rotação de objetos:** além de girar imagens, a rotação é usada para transformar objetos em visão computacional, como detecção e reconhecimento de objetos. Ela permite alinhar objetos com uma orientação padrão para análise.
- **Rotação em 90 graus:** a rotação de 90 graus é uma operação comum que pode ser usada para alterar a orientação de uma imagem em incrementos de 90 graus, geralmente em sentido horário ou anti-horário.
- **Rotação de imagens médicas:** em aplicações médicas, como tomografia computadorizada (TC) ou ressonância magnética (RM), a rotação de fatias de imagem pode ser necessária para análise e reconstrução tridimensional.
- **Rotação de texto e documentos:** a rotação é frequentemente usada para endireitar texto ou documentos digitalizados para facilitar a leitura ou o processamento automático.
- **Rotação de imagens panorâmicas:** em imagens panorâmicas ou esféricas, a rotação pode ser usada para criar diferentes vistas panorâmicas a partir da mesma imagem de base.



A rotação é uma operação essencial no processamento de imagem, pois permite a adaptação de imagens e objetos a diferentes necessidades de análise, visualização ou reconhecimento. A precisão da rotação e as técnicas de interpolação desempenham um papel fundamental na preservação da qualidade da imagem durante o processo de rotação. A figura a seguir ilustra a rotação de uma imagem utilizando uma transformação.

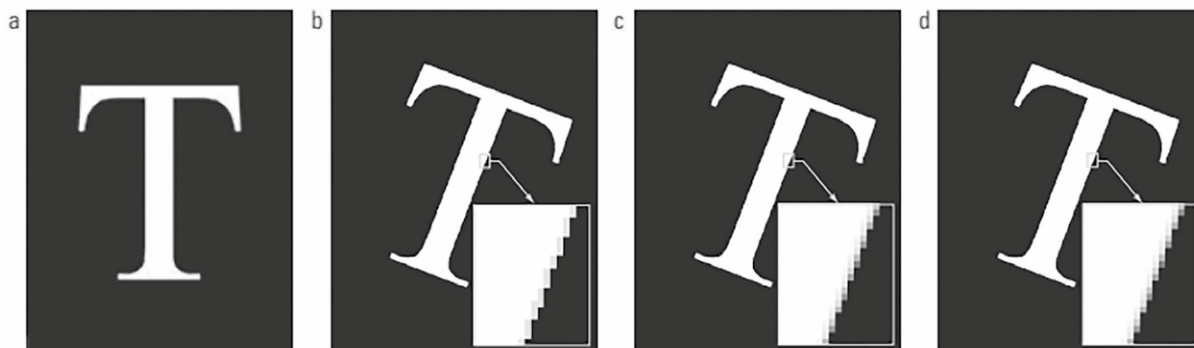


Figura 17 – (a) Uma imagem de 300 dpi da letra T. (b) Imagem rotacionada a 21° utilizando a interpolação do vizinho mais próximo para atribuir valores de intensidade aos pixels transformados espacialmente. (c) Imagem rotacionada a 21° utilizando a interpolação bilinear. (d) Imagem rotacionada a 21° utilizando a interpolação bicúbica. As seções ampliadas mostram detalhes de borda para os três métodos de interpolação

Fonte: Gonzalez e Woods (2010, p. 58).

Para realizar a rotação de uma imagem em Python, podemos utilizar o seguinte código como exemplo:

```
1 import cv2
2 import numpy as np
3 import matplotlib.pyplot as plt

4 # Carrega a imagem
5 imagem = cv2.imread('sua_imagem.jpg')

6 # Define o ângulo de rotação (em graus)
7 angulo_rotacao = 15

8 # Obtém o centro da imagem
9 altura, largura = imagem.shape[:2]
10 centro = (largura // 2, altura // 2)

11 # Define a matriz de rotação
12 matriz_rotacao = cv2.getRotationMatrix2D(centro, angulo_rotacao, 1.0)

13 # Aplica a rotação à imagem
14 imagem_rotacionada = cv2.warpAffine(imagem, matriz_rotacao, (largura, altura))
```



```
15 # Mostra a imagem original
16 plt.subplot(121)
17 plt.title('Imagem Original')
18 plt.imshow(cv2.cvtColor(imagem, cv2.COLOR_BGR2RGB))
19 plt.axis('off')

20 # Mostra a imagem rotacionada
21 plt.subplot(122)
22 plt.title('Imagem Rotacionada')
23 plt.imshow(cv2.cvtColor(imagem_rotacionada, cv2.COLOR_BGR2RGB))
24 plt.axis('off')

25 # Exibe as imagens
26 plt.show()
```

Saída:

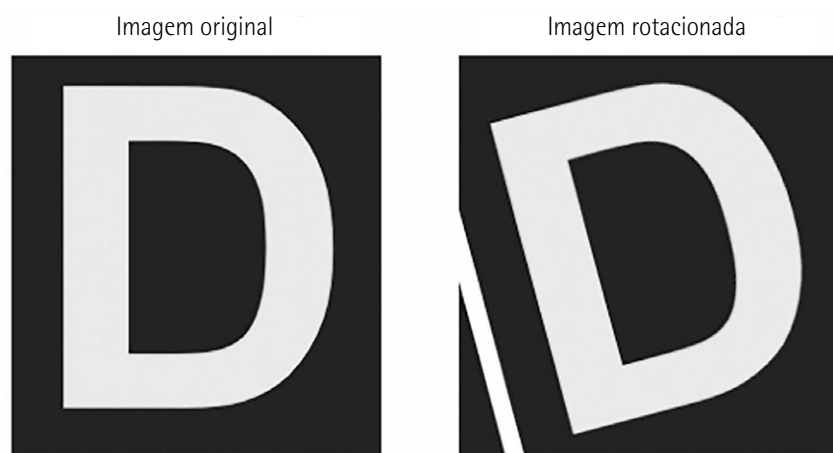


Figura 18

## 2.2 Histograma de cores

O histograma de cores em processamento de imagem é uma representação gráfica da distribuição das cores presentes em uma imagem. Ele é uma ferramenta fundamental para a análise e o ajuste de imagens, permitindo uma compreensão visual das características de cores da imagem. O histograma de cores é frequentemente usado em uma variedade de aplicações, como ajuste de contraste, correção de cores, detecção de características e segmentação de imagem. Os principais aspectos do histograma de cores em processamento de imagem:

- **Composição do histograma:** um histograma de cores é geralmente representado como um gráfico de barras, no qual o eixo horizontal representa as cores e o eixo vertical representa a frequência de ocorrência dessas cores na imagem. Cada barra no histograma corresponde a uma faixa de tons de cor, e a altura da barra indica quantos pixels da imagem têm essa cor.
- **Distribuição de cores:** o histograma revela a distribuição de cores na imagem. Por exemplo, se houver uma barra alta no extremo direito do histograma, isso indica que a imagem contém muitos pixels claros ou brancos.
- **Ajuste de contraste:** o histograma de cores é frequentemente usado para ajustar o contraste de uma imagem. Isso é feito redistribuindo os valores de intensidade de pixel para ocupar uma faixa dinâmica mais ampla, resultando em uma imagem com mais detalhes visíveis.
- **Correção de balanço de branco:** o histograma pode ser usado para corrigir problemas de balanço de branco, ajustando as intensidades de cor para que os tons de branco, preto e cinza sejam representados de maneira apropriada.
- **Deteção de características:** em tarefas de visão computacional, o histograma de cores pode ser usado para detectar características específicas em uma imagem. Por exemplo, um pico no histograma pode indicar a presença de um objeto de cor específica.
- **Segmentação de imagem:** a análise do histograma de cores pode ajudar na segmentação de imagens, em que diferentes faixas de tons de cor podem ser usadas para identificar objetos ou regiões de interesse.
- **Avaliação de qualidade de imagem:** o histograma de cores também pode ser usado para avaliar a qualidade da imagem. Por exemplo, a presença de um histograma equilibrado pode indicar uma imagem bem exposta e de alta qualidade.
- **Equalização de histograma:** a equalização de histograma é uma técnica que ajusta a distribuição de intensidade de pixel para uma distribuição uniforme, melhorando o contraste da imagem.
- **Redução de ruído:** a análise do histograma pode revelar picos que representam ruído na imagem. Isso pode ajudar na identificação e remoção de ruído.

O histograma de cores é uma ferramenta poderosa em processamento de imagem, pois fornece informações valiosas sobre a distribuição de cores em uma imagem. O entendimento adequado do histograma permite que os profissionais de imagem apliquem técnicas de processamento adequadas para melhorar a qualidade da imagem, destacar características relevantes e realizar análises mais precisas. A seguir apresentaremos um código em Python que exibe os histogramas dos canais RGB de uma imagem. Utilizamos a imagem a seguir para gerar os histogramas:



Figura 19 – Botão de rosa vermelha

Disponível em: <https://tinyurl.com/26u5b84z>. Acesso em: 25 out. 2023.

```
1 import cv2
2 import matplotlib.pyplot as plt
3 import numpy as np

4 # Carregue a imagem RGB
5 imagem = cv2.imread('sua_imagem.jpg')

6 # Separe os canais R, G e B
7 canal_r, canal_g, canal_b = cv2.split(imagem)

8 # Calcule os histogramas para cada canal
9 hist_r = cv2.calcHist([canal_r], [0], None, [256], [0, 256])
10 hist_g = cv2.calcHist([canal_g], [0], None, [256], [0, 256])
11 hist_b = cv2.calcHist([canal_b], [0], None, [256], [0, 256])

12 # Plote os histogramas
13 plt.rcParams.update({'font.size': 12})
14 plt.figure(figsize=(5, 10))

15 plt.subplot(311)
16 plt.title('Histograma do Canal Vermelho (R)')
17 plt.xlabel('Valor do Pixel')
18 plt.ylabel('Frequência')
19 plt.plot(hist_r, color='red')
```

```
20 plt.subplot(312)
21 plt.title('Histograma do Canal Verde (G)')
22 plt.xlabel('Valor do Pixel')
23 plt.ylabel('Frequência')
24 plt.plot(hist_g, color='green')

25 plt.subplot(313)
26 plt.title('Histograma do Canal Azul (B)')
27 plt.xlabel('Valor do Pixel')
28 plt.ylabel('Frequência')
29 plt.plot(hist_b, color='blue')

30 plt.tight_layout()
31 plt.show()
```

Saída:

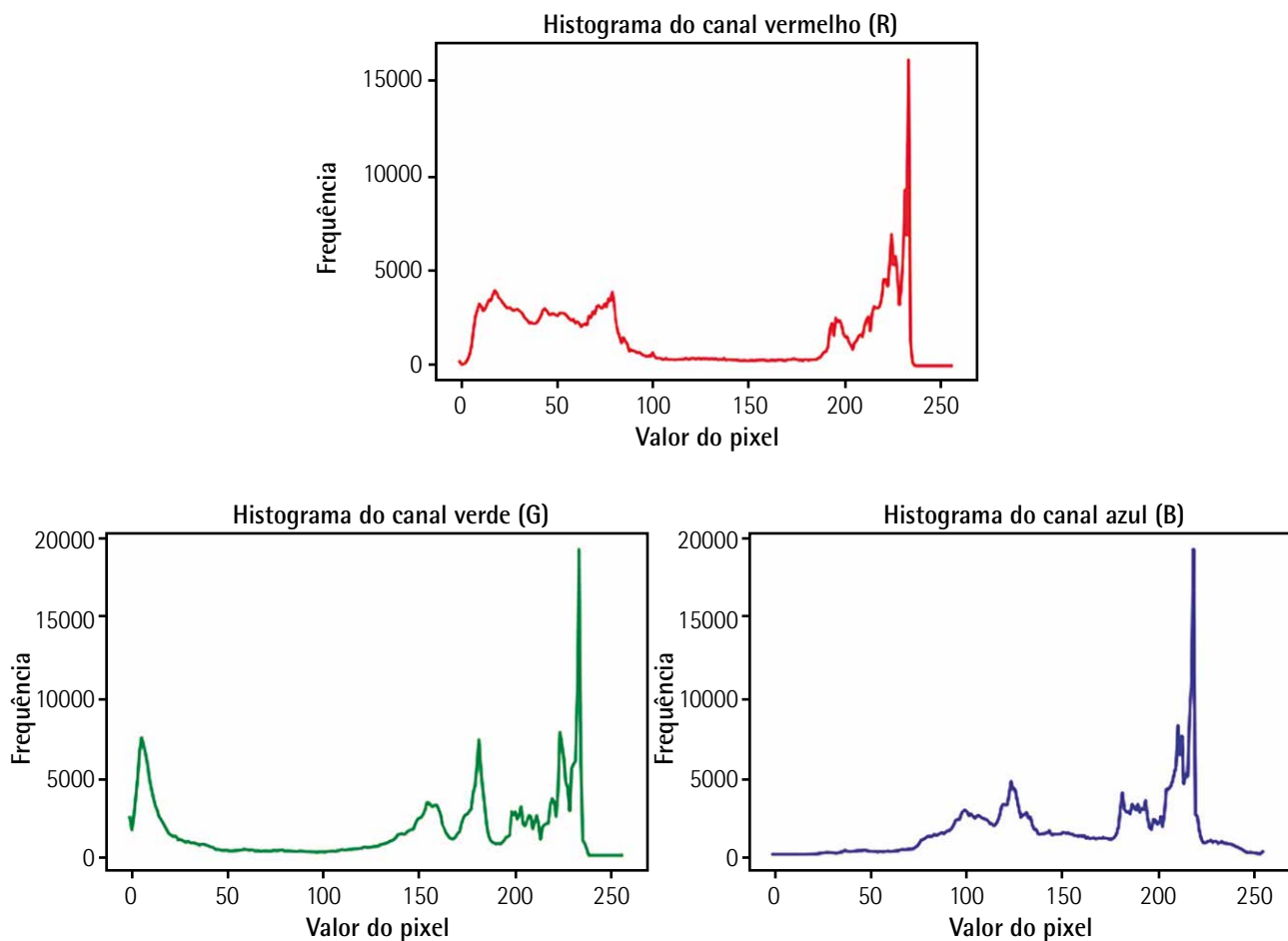


Figura 20

A interpretação de um histograma de cores depende do espaço de cores da imagem (RGB ou CMYK), mas uma explicação geral de como ele funciona seria considerar os eixos:

- **Eixo horizontal (X):** cada canal de cor é representado por um valor que varia de 0 a 255 em uma imagem em tons de cinza (8 bits por canal), onde 0 representa a ausência da cor e 255 representa a máxima intensidade daquela cor. No caso do espaço de cores RGB, cada canal (R, G, B) representa uma das três cores primárias. Portanto, o eixo horizontal em um histograma de cor representa a intensidade ou tom de uma cor específica.
- **Eixo vertical (Y):** este eixo representa a frequência ou a contagem de pixels com uma determinada intensidade de cor.

Ao plotar um histograma de uma imagem RGB, você obterá três curvas separadas, uma para cada canal de cor (vermelho, verde e azul). Essas curvas mostram como as intensidades de cada canal estão distribuídas na imagem. Um pico no histograma indica que uma determinada intensidade de cor é predominante na imagem.

Por exemplo, se houver um pico alto na parte direita do histograma vermelho, isso significa que muitos pixels da imagem têm intensidades de vermelho mais altas. Isso pode indicar áreas de realces vermelhos na imagem.

## 2.3 Operações aritméticas

As operações aritméticas são técnicas de pré-processamento de imagem muito utilizadas, permitindo combinar ou modificar imagens de diferentes maneiras para atingir objetivos específicos. Essas operações envolvem cálculos matemáticos simples que podem ser aplicados a pares de pixels ou a uma imagem inteira. Algumas das operações aritméticas comuns usadas no pré-processamento de imagens:

- **Adição e subtração:** a adição e a subtração de imagens envolvem a combinação de dois ou mais conjuntos de pixels. Por exemplo, adicionar uma imagem com uma imagem deslocada pode corrigir problemas de alinhamento ou sobreposição.
- **Multiplicação e divisão:** a multiplicação e a divisão de imagens são úteis para ajustar a intensidade ou o contraste de uma imagem. Multiplicar uma imagem por uma máscara de ponderação pode realçar ou atenuar regiões específicas.
- **Combinação ponderada:** a combinação ponderada envolve a multiplicação de cada pixel de uma imagem por um fator de escala específico e, em seguida, a soma dos resultados. Isso é comumente usado para criar efeitos de fusão de imagens, como realce de contraste ou redução de ruído.
- **Máximo e mínimo:** a operação de máximo seleciona o valor máximo entre dois pixels, enquanto a operação de mínimo seleciona o valor mínimo. Essas operações são usadas em várias técnicas de filtragem e na fusão de imagens.
- **Complemento:** o complemento de uma imagem é obtido invertendo a intensidade de pixel. Pixels mais claros se tornam mais escuros e vice-versa. Isso pode ser útil em tarefas de realce ou detecção de bordas.

- **Soma e média:** a soma e a média de um conjunto de imagens são usadas para criar uma imagem média ou somar os valores de intensidade de pixel. Isso pode ajudar a reduzir o ruído em imagens.
- **Substituição de fundo:** a substituição de fundo envolve a combinação de uma imagem de objeto com um novo fundo, ajustando os valores de intensidade para criar uma transição suave.
- **Limiarização (thresholding):** a limiarização é uma técnica de pré-processamento que usa operações de comparação, como maior ou igual e menor que, para converter uma imagem em uma imagem binária com base em um valor de limiar. Pixels que atendem ao critério se tornam brancos, enquanto os outros se tornam pretos.
- **Realce de detalhes:** as operações aritméticas também são usadas para realçar detalhes em uma imagem, destacando diferenças de intensidade entre pixels próximos.
- **Filtragem com máscara:** a aplicação de máscaras de convolução que contêm valores de peso para pixels vizinhos é uma forma de operação aritmética usada para realizar filtragem em imagens, como suavização ou detecção de bordas.

As operações aritméticas são flexíveis e versáteis, permitindo a criação de efeitos visuais desejados, a correção de problemas e a adaptação da imagem para tarefas subsequentes de processamento. A escolha das operações e dos valores de parâmetros depende das necessidades específicas de cada aplicação de processamento de imagem.

## 2.4 Transformações geométricas

Transformações geométricas referem-se à aplicação de operações que alteram a geometria da imagem, como sua posição, orientação, escala ou perspectiva. Essas transformações são valiosas para corrigir distorções, alinhar objetos, melhorar a visualização e preparar imagens para análises posteriores. Algumas das transformações geométricas comuns em pré-processamento de imagem:

- **Rotação:** a rotação envolve a mudança da orientação da imagem em torno de um ponto de referência, geralmente o centro da imagem. Isso é útil para corrigir a orientação incorreta, alinhar objetos ou criar diferentes visualizações da mesma cena.
- **Redimensionamento:** o redimensionamento modifica as dimensões da imagem, tornando-a maior ou menor em escala. Isso é útil para ajustar o tamanho da imagem, criar miniaturas ou ampliar detalhes.
- **Corte:** o corte envolve a remoção de partes indesejadas da imagem, resultando em uma imagem menor com uma área de interesse específica. Isso é usado para destacar uma região de interesse e remover informações irrelevantes.
- **Translação:** a translação desloca a imagem em relação a um eixo, mudando sua posição. Isso é útil para ajustar a posição de objetos em uma cena.

- **Espelhamento (reflexão):** o espelhamento cria uma imagem espelhada em relação a um eixo. Isso é usado para criar simetrias ou corrigir imagens espelhadas incorretamente.
- **Projeção perspectiva:** a projeção perspectiva é usada para corrigir distorções de perspectiva em imagens que resultam de ângulos de visualização não ideais. Isso é comum em aplicações de visão computacional, como reconhecimento de placas de carro.
- **Correção de distorção de lente:** em fotografia, a correção de distorção de lente é uma transformação geométrica usada para corrigir distorções, como barril ou almofada, que podem ocorrer em imagens devido à ótica da câmera.
- **Alinhamento de imagens:** a transformação geométrica pode ser usada para alinhar imagens, tornando-as consistentes em termos de orientação e posição. Isso é útil em mosaicos de imagens ou em análises que exigem registro preciso.
- **Transformações não lineares:** além das transformações lineares, transformações não lineares, como a deformação livre (warping), podem ser usadas para ajustar a geometria da imagem de maneira mais complexa, permitindo a correção de distorções de forma livre.
- **Correção de perspectiva:** a correção de perspectiva é usada para transformar uma imagem para simular uma visualização a partir de um ponto de vista diferente. Isso é útil em aplicações de realidade aumentada e simulação.

As transformações geométricas são essências no pré-processamento de imagem, pois permitem que os profissionais de imagem corrijam imperfeições, ajustem a geometria e melhorem a visualização das imagens antes de realizar análises posteriores. A escolha da transformação depende das necessidades específicas de cada aplicação.

## 2.5 Ruído em imagens

O ruído em imagens é uma ocorrência indesejada que introduz variações aleatórias nos valores dos pixels, tornando a imagem menos nítida e prejudicando a qualidade visual. Essas variações podem resultar de várias fontes, como aquisição de imagem, transmissão, compressão ou processamento inadequado. O pré-processamento de imagem frequentemente envolve técnicas para reduzir ou remover o ruído, melhorando assim a qualidade da imagem e facilitando análises subsequentes. Aqui estão algumas considerações importantes sobre o ruído em imagens e as técnicas para mitigá-lo.

Principais tipos de ruído em imagens:

- **Ruído de sensor (granular):** resulta da variação nos valores de pixel devido à eletrônica do sensor de imagem. Pode se assemelhar a pontos brancos ou coloridos na imagem.
- **Ruído aleatório (sensível à luz):** pode ocorrer devido à variação na quantidade de luz que atinge o sensor da câmera, criando pequenas variações nos valores de pixel.

- **Ruído de Poisson:** é comum em imagens com baixa intensidade de luz, em que os fótons atingem o sensor de forma aleatória, criando um padrão de ruído de Poisson.
- **Ruído de sal e pimenta:** causa a ocorrência de pixels extremamente brilhantes (sal) ou escuros (pimenta) na imagem, resultando em pontos isolados de intensidade extrema.

Técnicas de pré-processamento para redução de ruído:

- **Suavização (filtros de média):** filtros de suavização, como a média, a mediana e o filtro gaussiano, são usados para reduzir o ruído de forma geral, substituindo os valores de pixel por médias ponderadas de pixels vizinhos.
- **Filtragem não linear (filtros de mediana):** filtros de mediana são particularmente eficazes na redução de ruído de sal e pimenta, pois substituem cada pixel pela mediana dos valores em uma vizinhança.
- **Filtragem por difusão (denoising):** algoritmos de filtragem por difusão, como o filtro de difusão anisotrópica, aplicam processos iterativos para remover o ruído enquanto preservam as bordas e os detalhes.
- **Filtros adaptativos:** esses filtros ajustam a intensidade do filtro de acordo com as características locais da imagem, proporcionando uma redução eficaz de ruído.
- **Transformada de Wavelet:** a transformada de Wavelet é usada para decompor a imagem em diferentes frequências e escalas, permitindo a separação do sinal de ruído do sinal de interesse.
- **Filtragem por Fourier:** a filtragem no domínio da frequência, usando a transformada de Fourier, pode ser usada para eliminar componentes de alta frequência associadas ao ruído.
- **Filtragem de cor:** em imagens coloridas, os filtros podem ser aplicados em canais de cores individuais (R, G, B) ou em espaços de cores alternativos, como o espaço de cores  $L^*a^*b^*$ , para preservar melhor os detalhes de cor.
- **Métodos de reconstrução:** métodos de reconstrução de imagem, como a filtragem Wiener, usam o conhecimento do ruído e da imagem original para melhorar a qualidade da imagem.

A escolha da técnica de redução de ruído depende do tipo de ruído presente na imagem, bem como dos requisitos da aplicação. O objetivo é encontrar equilíbrio entre a redução eficaz do ruído e a preservação das características importantes da imagem. A remoção do ruído em imagens é uma etapa crucial em muitas aplicações, como processamento de imagem médica, visão computacional e análise de imagem.

O espaço de cores  $L^*a^*b^*$  (também conhecido como CIELAB ou Lab) é um espaço de cores tridimensional utilizado para representar e descrever cores de uma forma que seja independente do dispositivo ou da mídia de exibição. Ele foi desenvolvido pela Comissão Internacional de Iluminação (CIE) e é utilizado na indústria gráfica, design, pesquisa de cores, entre outros.



O espaço de cores Lab é projetado para ser perceptualmente uniforme, o que significa que as diferenças entre cores no espaço Lab correspondem de forma mais precisa às diferenças percebidas pelo olho humano. Ele consiste em três componentes principais:

- **L (luminosidade):** o canal "L" representa a intensidade luminosa ou a luminosidade da cor. Ele varia de 0 (preto) a 100 (branco). À medida que o valor L aumenta, a cor se torna mais clara.
- **a (tons de vermelho/verde):** o canal "a" descreve a posição da cor ao longo do eixo vermelho/verde. Valores positivos de "a" representam tons de vermelho, enquanto valores negativos representam tons de verde. O valor zero representa um tom de cinza.
- **b (tons de amarelo/azul):** o canal "b" descreve a posição da cor ao longo do eixo amarelo/azul. Valores positivos de "b" representam tons de amarelo, enquanto valores negativos representam tons de azul. O valor zero representa um tom de cinza.

O espaço de cores Lab é útil para várias aplicações, incluindo correspondência de cores, calibração de dispositivos de exibição, análise de cores em imagens e medição de diferenças de cores perceptuais. Uma das principais vantagens desse espaço de cores é a capacidade de descrever uma variedade de cores visíveis e suas relações de forma precisa.

O espaço de cores Lab\* é um sistema de representação de cores que se concentra na perceptibilidade humana das cores e fornece uma maneira precisa e padronizada de descrever e comparar cores, independentemente do dispositivo de exibição ou da mídia.

## 2.6 Escala de cinza

A escala de cinza é uma representação de imagem na qual os pixels são apresentados em tons de cinza, variando de preto (intensidade zero) a branco (intensidade máxima). Em uma imagem em escala de cinza, a informação de cor é descartada, resultando em uma imagem em preto e branco que representa apenas a variação de intensidade de pixel. A transformação de uma imagem colorida em uma imagem em escala de cinza é uma técnica comum no pré-processamento de imagem e é útil em várias aplicações. Algumas considerações importantes sobre a escala de cinza e seu uso em técnicas de pré-processamento de imagem:

Vantagens da escala de cinza:

- **Simplicidade:** imagens em escala de cinza são mais simples de manipular e analisar, pois não possuem informações de cor, o que pode ser benéfico em muitos cenários.
- **Economia de espaço:** as imagens em escala de cinza geralmente requerem menos espaço de armazenamento do que suas contrapartes coloridas, tornando-as eficientes em termos de armazenamento de dados.

- **Redução de complexidade:** a remoção da informação de cor pode simplificar tarefas de análise de imagem, como detecção de bordas e segmentação.
- **Ênfase em intensidade:** em muitos casos, a intensidade é o atributo mais relevante da imagem, e a escala de cinza destaca essa informação.

A conversão de uma imagem colorida para escala de cinza envolve a atribuição de um único valor de intensidade (ou luminância) a cada pixel da imagem. Existem várias técnicas para fazer essa conversão:

- **Média dos canais de cor:** a intensidade de um pixel pode ser calculada como a média dos valores dos canais de cor (vermelho, verde e azul) na imagem colorida.
- **Luminância pesada:** a conversão pode levar em consideração as diferenças de sensibilidade do olho humano às cores. Nesse caso, os valores dos canais de cor são ponderados antes de calcular a intensidade.
- **Decomposição em espaços de cores alternativos:** outra abordagem é converter a imagem colorida em um espaço de cores alternativo, como o espaço Lab, onde o canal L representa a luminância e é usado como a imagem em escala de cinza.

Aplicações da escala de cinza:

- **Melhoria de contraste:** a representação em escala de cinza pode ser usada para ajustar o contraste de uma imagem, realçando detalhes e tornando-a mais perceptível.
- **Filtragem de ruído:** em muitos casos, a filtragem de ruído é mais eficaz em imagens em escala de cinza, pois simplifica o processo de análise.
- **Segmentação de imagem:** a segmentação de imagens em escala de cinza é frequentemente usada para identificar regiões de interesse com base na intensidade.
- **Extração de características:** em aplicações de visão computacional, características como bordas e contornos são frequentemente extraídas de imagens em escala de cinza.
- **Processamento de imagem médica:** a escala de cinza é utilizada em imagens médicas, como radiografias e imagens de ressonância magnética, para realçar estruturas e características relevantes.

A representação em escala de cinza é uma técnica valiosa no pré-processamento de imagem que simplifica a análise e pode ser usada em uma variedade de aplicações, desde melhoria de contraste até segmentação e extração de características. A escolha de converter uma imagem para escala de cinza depende das necessidades específicas da aplicação e dos objetivos de análise de imagem.

A seguir apresentaremos um código em Python que transforma uma imagem colorida em escala de cinza e exibe seu histograma.

```
1 import cv2
2 import numpy as np
3 import matplotlib.pyplot as plt

4 # Carrega a imagem colorida
5 imagem_colorida = cv2.imread('sua_imagem.jpg')

6 # Converte a imagem colorida em tons de cinza
7 imagem_em_tons_de_cinza = cv2.cvtColor(imagem_colorida, cv2.COLOR_BGR2GRAY)

8 # Calcula o histograma da imagem em tons de cinza
9 histograma = cv2.calcHist([imagem_em_tons_de_cinza], [0], None, [256], [0, 256])

10 # Ajuste do tamanho da saída
11 plt.rcParams.update({'font.size': 20})
12 plt.figure(figsize=(10, 4))

13 # Exibe a imagem original
14 plt.subplot(221)
15 plt.title('Imagem Original')
16 plt.imshow(cv2.cvtColor(imagem_colorida, cv2.COLOR_BGR2RGB))
17 plt.axis('off')

18 # Exibe a imagem em tons de cinza
19 plt.subplot(222)
20 plt.title('Imagem em Tons de Cinza')
21 plt.imshow(imagem_em_tons_de_cinza, cmap='gray')
22 plt.axis('off')

23 # Exibe o histograma
24 plt.subplot(223)
25 plt.hist(imagem_em_tons_de_cinza.ravel(), 256, [0, 256])
26 plt.title('Histograma de Tons de Cinza')
27 plt.xlabel('Valor do Pixel')
28 plt.ylabel('Frequência')

29 # Ajusta o layout para melhor visualização
30 plt.tight_layout()

31 # Exibe as três imagens e o histograma
32 plt.show()
```

Saída:

Imagem original



Imagem em tons de cinza



Histograma de tons de cinza

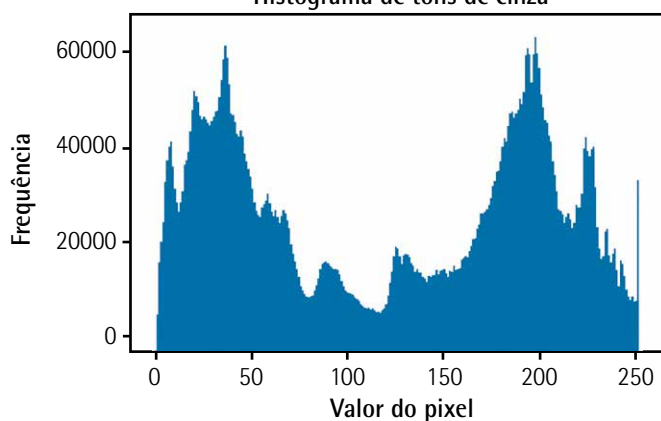


Figura 21 – Imagem original do Museu de Arte de São Paulo (Masp)

Adaptada de: <https://tinyurl.com/334sxvbs>. Acesso em: 25 out. 2023.

## 2.7 Métodos de filtragem

Os métodos de filtragem envolvem a aplicação de filtros ou máscaras a uma imagem para melhorar sua qualidade, realçar características específicas ou reduzir o ruído. Essas técnicas são usadas em processamento de imagem para preparar imagens para análises subsequentes. Aqui estão alguns dos métodos de filtragem mais comuns:

- **Filtros de suavização (filtros de média):** os filtros de suavização são usados para reduzir o ruído em uma imagem, tornando-a mais suave. O filtro de média calcula a média dos valores de pixel em uma vizinhança ao redor de cada pixel e substitui o valor original pelo resultado. Isso ajuda a reduzir o ruído, mas também pode reduzir os detalhes.
- **Filtro de mediana:** o filtro de mediana é eficaz na remoção de ruído do tipo sal e pimenta, no qual alguns pixels são extremamente escuros ou claros. Ele substitui cada pixel pelo valor da mediana em sua vizinhança, preservando melhor os detalhes da imagem.

- **Filtros de suavização gaussiana:** esses filtros aplicam uma ponderação maior aos pixels mais próximos ao centro da vizinhança e uma ponderação menor aos pixels mais distantes, criando uma suavização suave, que ainda mantém detalhes na imagem.
- **Filtros de difusão (denoising):** algoritmos de filtragem por difusão, como o filtro de difusão anisotrópica, são usados para remover o ruído enquanto preservam as bordas e os detalhes da imagem.
- **Filtros de detecção de bordas (filtros laplacianos, filtros de Sobel):** esses filtros realçam bordas e transições de intensidade na imagem, tornando-as mais visíveis. São usados em detecção de características.
- **Filtros de realce (filtros de nitidez):** esses filtros realçam detalhes finos e estruturas na imagem, tornando-a mais nítida. O filtro de nitidez é um exemplo comum.
- **Filtros de frequência (filtros de Fourier):** a transformada de Fourier é usada para decompor a imagem em frequências. Os filtros de frequência podem ser aplicados para realçar ou atenuar determinadas frequências, o que é útil em tarefas de remoção de ruído ou realce de padrões.
- **Filtragem morfológica:** a filtragem morfológica envolve operações como erosão e dilatação, que são usadas para realçar ou reduzir características específicas na imagem, como bordas ou regiões de interesse.
- **Filtragem de cor:** em imagens coloridas, os filtros podem ser aplicados em canais de cores individuais (R, G, B) ou em espaços de cores alternativos, como o espaço Lab, para realçar ou ajustar características de cor.
- **Filtros adaptativos:** esses filtros ajustam a intensidade do filtro com base nas características locais da imagem, proporcionando uma filtragem mais adaptável.
- **Filtros de anisotropia perceptual:** usados em aplicações de realidade virtual e jogos, esses filtros alteram a imagem de acordo com a percepção visual humana, melhorando a qualidade visual.

A escolha do método de filtragem depende das características da imagem e dos objetivos de pré-processamento. Em muitos casos, uma combinação de vários filtros é usada para alcançar o resultado desejado. A filtragem é uma etapa crítica no processamento de imagem, pois pode melhorar sua qualidade, destacar informações importantes e prepará-la para análises subsequentes.

### 3 SUAVIZAÇÃO DE IMAGENS

A suavização de imagens, ou filtragem de imagens, é uma técnica de processamento de imagem que tem como objetivo reduzir o ruído e as variações abruptas de intensidade, tornando a imagem mais suave e uniforme. Essa técnica é utilizada em diversas aplicações, desde aprimoramento de imagens até análises de computação visual. Os principais conceitos e métodos relacionados à suavização de imagens são:

- **Ruído em imagens:** o ruído em imagens refere-se a variações indesejadas de intensidade de pixel que podem ser causadas por vários fatores, como sensores de imagem, iluminação irregular ou transmissão de dados. O objetivo da suavização é reduzir ou eliminar esse ruído.
- **Filtros de suavização:** os filtros de suavização são os principais instrumentos utilizados para realizar a suavização de imagens. Eles funcionam substituindo o valor de um pixel pela média ponderada dos valores dos pixels em sua vizinhança. Os filtros de suavização mais comuns incluem o filtro de média, o filtro de mediana e o filtro gaussiano.
- **Filtro de média:** o filtro de média calcula a média dos valores dos pixels em uma vizinhança e substitui o valor do pixel central por esse valor médio. Isso reduz o ruído aleatório, mas pode causar uma perda de detalhes.
- **Filtro de mediana:** o filtro de mediana substitui o valor do pixel central pela mediana dos valores dos pixels em uma vizinhança. É especialmente eficaz na remoção de ruído do tipo sal e pimenta e preserva melhor os detalhes da imagem.
- **Filtro gaussiano:** o filtro gaussiano utiliza uma função gaussiana para ponderar os valores dos pixels em uma vizinhança. Ele produz uma suavização suave e é eficaz na remoção de ruído.
- **Parâmetros de filtro:** a eficácia da suavização depende dos parâmetros do filtro, como o tamanho da vizinhança e os pesos aplicados a cada pixel. Ajustar esses parâmetros permite controlar o nível de suavização e a capacidade de preservação de detalhes.

A suavização de imagens é usada em várias aplicações, incluindo:

- Remoção de ruído de imagens.
- Melhoria da qualidade visual de imagens.
- Redução de detalhes irrelevantes para simplificar análises.
- Preparação de imagens utilizada para processamentos subsequentes, como em detecção de bordas ou segmentação.



É importante destacar que a suavização de imagens geralmente envolve um trade-off entre a redução de ruído e a preservação de detalhes. Filtros de suavização mais fortes podem remover mais ruído, mas também podem prejudicar a nitidez dos detalhes.

A suavização de imagens é fundamental no processamento de imagem, ajudando a melhorar a qualidade das imagens e torná-las mais adequadas para análises posteriores. A escolha do método de suavização e seus parâmetros depende das características da imagem e dos objetivos da aplicação específica.

## 3.1 Suavização por cálculo da média

A suavização de imagens utilizando o método de suavização por cálculo da média é uma técnica simples e eficaz para reduzir o ruído em imagens, tornando-as mais suaves e uniformes. Esse método é usado em processamento de imagem e é conhecido como filtro de média. A ideia básica é substituir o valor de cada pixel na imagem pela média dos valores dos pixels em sua vizinhança. Os principais conceitos e passos envolvidos na suavização por cálculo da média:

- **Vizinhança do pixel:** para cada pixel na imagem, uma vizinhança é definida ao redor dele. A vizinhança é geralmente uma janela retangular ou quadrada de tamanho predefinido. O tamanho da vizinhança determina a extensão da suavização. Na figura a seguir, temos um exemplo de uma janela de cálculo  $3 \times 3$  pixels:

30	100	130
130	?	160
50	100	210

Figura 22

### Observação

Alguns autores chamam esta caixa de janela de cálculo ou kernel (do inglês núcleo).

Sugerimos utilizar os números ímpares para as linhas e colunas da janela de cálculo para que exista sempre o pixel central que será alvo do cálculo.

- **Cálculo da média:** para suavizar um pixel específico, a média dos valores dos pixels dentro da vizinhança é calculada. Isso é feito somando os valores dos pixels na vizinhança e dividindo o resultado pelo número total de pixels na vizinhança.
- **Substituição do valor do pixel:** o valor do pixel original é substituído pela média calculada. Isso suaviza a intensidade do pixel e ajuda a reduzir o impacto do ruído.
- **Repetição para todos os pixels:** o processo é repetido para todos os pixels na imagem, garantindo que cada um seja suavizado com base na média de seus vizinhos.
- **Ajuste de parâmetros:** para otimizar a suavização, os parâmetros do filtro de média podem ser ajustados, como o tamanho da vizinhança. O tamanho maior da vizinhança resulta em uma suavização mais forte, mas pode prejudicar a preservação de detalhes.
- **Efeitos na imagem:** a suavização por cálculo da média é eficaz na remoção de ruído aleatório, tornando a imagem mais uniforme. No entanto, ela pode causar a perda de detalhes finos e transições de intensidade abrupta.

Aplicações:

- Imagens médicas para redução de ruído.
- Processamento de imagem em câmeras digitais.
- Melhoria da qualidade visual de imagens.
- Preparação de imagens para análises subsequentes, como detecção de bordas.

Veja a seguir um exemplo de pseudocódigo em Python. Trata-se de um exemplo simples de como implementar a suavização (kernel de 5, 9 e 15) por cálculo da média em Python usando a biblioteca OpenCV:

```
1 import cv2
2 import matplotlib.pyplot as plt

3 # Carregue a imagem
4 imagem = cv2.imread('sua_imagem.jpg')

5 # Aplique o filtro de média com uma vizinhança de 5x5 pixels
6 imagem_suavizada_5 = cv2.blur(imagem, (5, 5))

7 # Aplique o filtro de média com uma vizinhança de 9x9 pixels
8 imagem_suavizada_9 = cv2.blur(imagem, (9, 9))
```

```
9 # Aplique o filtro de média com uma vizinhança de 15x15 pixels
10 imagem_suavizada_15 = cv2.blur(imagem, (15, 15))

11 # Exiba a imagem original e a imagem suavizada
12 plt.figure(figsize=(12, 6))

13 # Exibir a imagem original
14 plt.subplot(2, 2, 1)

15 plt.imshow(cv2.cvtColor(imagem, cv2.COLOR_BGR2RGB)) # Converter de BGR para RGB para
exibição correta
16 plt.title('Imagem Original')

17 # Exibir a imagem suavizada
18 plt.subplot(2, 2, 2)
19 plt.imshow(cv2.cvtColor(imagem_suavizada_5, cv2.COLOR_BGR2RGB))
20 plt.title('Imagem Suavizada 5x5')

21 # Exibir a imagem suavizada
22 plt.subplot(2, 2, 3)
23 plt.imshow(cv2.cvtColor(imagem_suavizada_9, cv2.COLOR_BGR2RGB))
24 plt.title('Imagem Suavizada 9x9')

25 # Exibir a imagem suavizada
26 plt.subplot(2, 2, 4)
27 plt.imshow(cv2.cvtColor(imagem_suavizada_15, cv2.COLOR_BGR2RGB))
28 plt.title('Imagem Suavizada 15x15')

29 # Configurar layout
30 plt.tight_layout()

31 # Exibir as imagens
32 plt.show()
```

Saída:

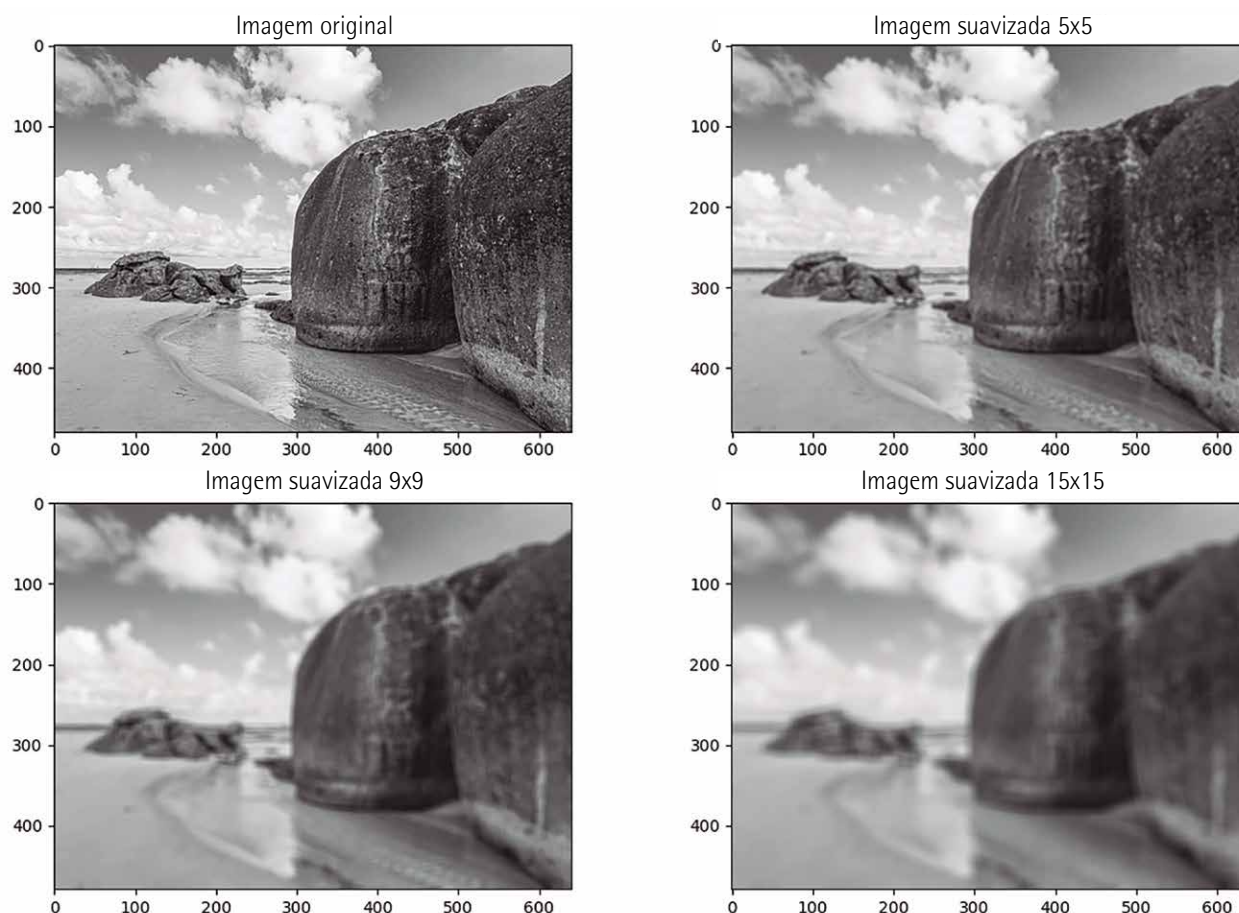


Figura 23 – Squeaky Beach em Wilsons Promontory National Park, Victoria, Austrália, 2019 (640 × 479 pixels)

Disponível em: <https://tinyurl.com/sze7hysm>. Acesso em: 25 out. 2023.

A suavização por cálculo da média é uma técnica simples e eficaz para reduzir o ruído em imagens, tornando-as mais adequadas para análises posteriores ou melhorando sua qualidade visual. A escolha do tamanho da vizinhança e dos parâmetros depende das características específicas da imagem e dos objetivos da aplicação.

## 3.2 Suavização pela gaussiana

A suavização de imagens utilizando o método de suavização pela gaussiana é uma técnica empregada no processamento de imagens para reduzir o ruído e suavizar a imagem. Esse método é baseado na aplicação de um filtro gaussiano, que utiliza uma distribuição gaussiana (ou normal) para calcular a média ponderada dos valores dos pixels em uma vizinhança. A suavização pela gaussiana é uma das técnicas mais comuns e eficazes de filtragem para esse fim. Os principais conceitos e passos envolvidos na suavização pela gaussiana:

- **Vizinhança do pixel:** para aplicar a suavização pela gaussiana, define-se uma vizinhança em torno de cada pixel na imagem. A vizinhança é geralmente uma janela quadrada ou retangular de tamanho predefinido.
- **Filtro gaussiano:** o filtro gaussiano é uma máscara (kernel) que tem uma distribuição gaussiana de pesos. Essa distribuição de pesos é usada para calcular a média ponderada dos valores dos pixels na vizinhança de acordo com a distribuição gaussiana.
- **Cálculo da média ponderada:** para suavizar um pixel específico, os valores dos pixels na vizinhança são multiplicados pelos pesos do filtro gaussiano e somados para calcular a média ponderada. O resultado desse cálculo é o novo valor do pixel suavizado.
- **Substituição do valor do pixel:** o valor do pixel original é substituído pelo valor calculado usando o filtro gaussiano. Isso reduz a variação de intensidade e ajuda a reduzir o ruído na imagem.
- **Repetição para todos os pixels:** o processo de suavização pela gaussiana é repetido para todos os pixels na imagem, garantindo que cada um seja suavizado com base na média ponderada de seus vizinhos.
- **Ajuste de parâmetros:** é possível ajustar os parâmetros do filtro gaussiano, como o tamanho da vizinhança e o desvio padrão (sigma), para controlar o grau de suavização. Um desvio padrão maior resulta em uma suavização mais forte.
- **Efeitos na imagem:** a suavização pela gaussiana é eficaz na redução de ruído, mantendo melhor a nitidez dos detalhes em comparação com filtros de média. No entanto, ela ainda pode causar uma perda suave de detalhes finos.

Aplicações:

- Redução de ruído em imagens.
- Melhoria da qualidade visual de imagens.
- Preparação de imagens para análises subsequentes, como detecção de bordas.
- Processamento de imagens médicas para melhorar a qualidade das imagens de diagnóstico.

Veja a seguir um exemplo de pseudocódigo em Python. Trata-se de um exemplo simples de como implementar a suavização pela gaussiana (kernel de 5, 9 e 15) em Python usando a biblioteca OpenCV:

```
1 import cv2
2 import matplotlib.pyplot as plt

3 # Carregue a imagem
4 imagem = cv2.imread('sua_imagem.jpg')
5 # Aplique o filtro Gaussiano com um tamanho de kernel (vizinhança) de 5x5 e desvio padrão de 1
6 imagem_suavizada_5 = cv2.GaussianBlur(imagem, (5, 5), 1)

7 # Aplique o filtro Gaussiano com um tamanho de kernel (vizinhança) de 9x9 e desvio padrão de 1
8 imagem_suavizada_9 = cv2.GaussianBlur(imagem, (9, 9), 1)
9 # Aplique o filtro Gaussiano com um tamanho de kernel (vizinhança) de 15x15 e desvio padrão de 1
10 imagem_suavizada_15 = cv2.GaussianBlur(imagem, (15, 15), 1)

11 # Exiba a imagem original e a imagem suavizada
12 plt.figure(figsize=(12, 8))

13 # Exibir a imagem original
14 plt.subplot(2, 2, 1)
15 plt.imshow(cv2.cvtColor(imagem, cv2.COLOR_BGR2RGB)) # Converter de BGR para RGB para
exibição correta
16 plt.title('Imagem Original')

17 # Exibir a imagem suavizada
18 plt.subplot(2, 2, 2)
19 plt.imshow(cv2.cvtColor(imagem_suavizada_5, cv2.COLOR_BGR2RGB))
20 plt.title('Imagem Suavizada 5x5')

21 # Exibir a imagem suavizada
22 plt.subplot(2, 2, 3)
23 plt.imshow(cv2.cvtColor(imagem_suavizada_9, cv2.COLOR_BGR2RGB))
24 plt.title('Imagem Suavizada 9x9')

25 # Exibir a imagem suavizada
26 plt.subplot(2, 2, 4)
27 plt.imshow(cv2.cvtColor(imagem_suavizada_15, cv2.COLOR_BGR2RGB))
28 plt.title('Imagem Suavizada 15x15')

29 # Configurar layout
30 plt.tight_layout()

31 # Exibir as imagens
32 plt.show()
```

Saída:



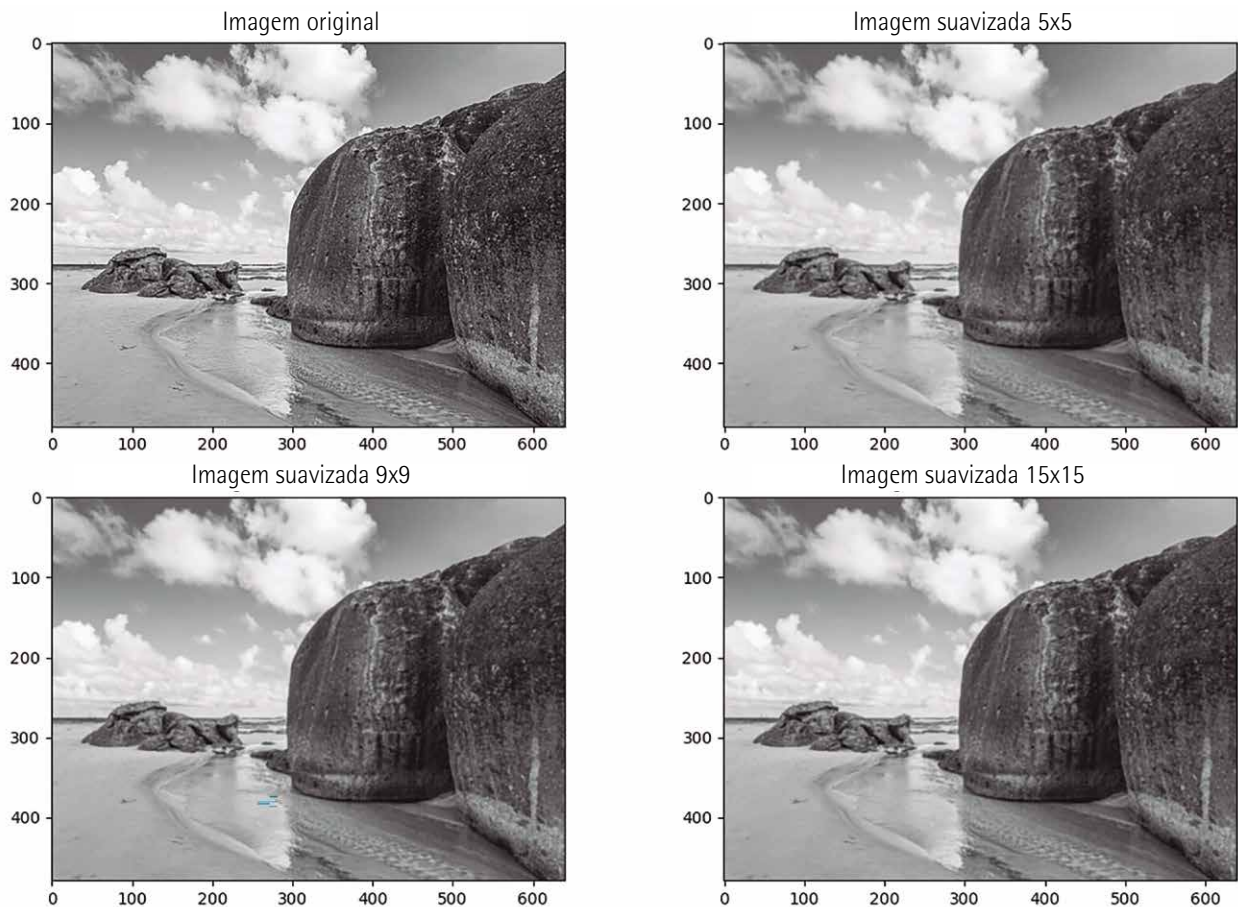


Figura 24 – Squeaky Beach em Wilsons Promontory National Park, Victoria, Austrália, 2019 (640 × 479 pixels)

Disponível em: <https://tinyurl.com/sze7hysm>. Acesso em: 25 out. 2023.

Podemos observar nas imagens como o filtro de kernel gaussiano gera menos borrão na imagem, mas também gera um efeito mais natural e reduz o ruído na imagem. A suavização pela gaussiana é uma técnica eficaz para melhorar a qualidade de imagens e reduzir o ruído. A escolha dos parâmetros, como o tamanho do kernel e o desvio padrão, depende das características específicas da imagem e dos objetivos da aplicação.

### 3.3 Suavização pela mediana

A suavização de imagens utilizando o método de suavização pela mediana é uma técnica eficaz para reduzir o ruído em imagens, especialmente o tipo de ruído conhecido como ruído de sal e pimenta. Esse método envolve a substituição do valor de cada pixel na imagem pela mediana dos valores dos pixels em sua vizinhança. A suavização pela mediana é muito usada em processamento de imagem, pois preserva melhor os detalhes da imagem em comparação com filtros de suavização que utilizam médias ponderadas. Os principais conceitos e passos envolvidos na suavização pela mediana:

- **Vizinhança do pixel:** para aplicar a suavização pela mediana, define-se uma vizinhança ao redor de cada pixel na imagem. A vizinhança é geralmente uma janela quadrada ou retangular de tamanho predefinido.
- **Cálculo da mediana:** para suavizar um pixel específico, os valores dos pixels na vizinhança são coletados e organizados em ordem crescente. A mediana, que é o valor do pixel que se encontra no meio desse conjunto ordenado, é calculada.
- **Substituição do valor do pixel:** o valor do pixel original é substituído pela mediana calculada. Isso reduz o impacto do ruído, pois a mediana é menos sensível a valores extremos em comparação com a média aritmética. A mediana é semelhante à média, mas ela despreza os valores muito altos ou muito baixos que podem distorcer o resultado. A mediana é o número que fica exatamente no meio do intervalo.
- **Repetição para todos os pixels:** o processo de suavização pela mediana é repetido para todos os pixels na imagem, garantindo que cada um seja suavizado com base na mediana de seus vizinhos.
- **Ajuste de parâmetros:** é possível ajustar o tamanho da vizinhança para controlar o grau de suavização. Um tamanho maior da vizinhança resulta em uma suavização mais forte.
- **Efeitos na imagem:** a suavização pela mediana é eficaz na remoção de ruído de tipo sal e pimenta, em que alguns pixels são extremamente escuros (sal) ou claros (pimenta). Ela preserva melhor os detalhes da imagem em comparação com filtros de suavização que usam médias ponderadas.

Aplicações:

- Remoção de ruído em imagens de câmeras digitais.
- Melhoria da qualidade visual de imagens.
- Processamento de imagens médicas, como radiografias.
- Pré-processamento de imagens antes da detecção de bordas ou segmentação.

Veja a seguir um exemplo de pseudocódigo em Python. Trata-se de um exemplo simples de como implementar a suavização pela mediana (kernel de 3, 9 e 15) em Python usando a biblioteca OpenCV:

```
1 import cv2
2 import matplotlib.pyplot as plt

3 # Carregue a imagem
4 imagem = cv2.imread('sua_imagem.jpg')

5 # Aplique a suavização pela mediana com uma vizinhança de 5x5 pixels
6 imagem_suavizada_5 = cv2.medianBlur(imagem, 5)

7 # Aplique a suavização pela mediana com uma vizinhança de 9x9 pixels
8 imagem_suavizada_9 = cv2.medianBlur(imagem, 9)

9 # Aplique a suavização pela mediana com uma vizinhança de 15x15 pixels
10 imagem_suavizada_15 = cv2.medianBlur(imagem, 15)

11 # Exiba a imagem original e a imagem suavizada
12 plt.figure(figsize=(12, 8))

13 # Exibir a imagem original
14 plt.subplot(2, 2, 1)
15 plt.imshow(cv2.cvtColor(imagem, cv2.COLOR_BGR2RGB)) # Converter de BGR para RGB para
exibição correta
16 plt.title('Imagem Original')

17 # Exibir a imagem suavizada
18 plt.subplot(2, 2, 2)
19 plt.imshow(cv2.cvtColor(imagem_suavizada_5, cv2.COLOR_BGR2RGB))
20 plt.title('Imagem Suavizada 5x5')

21 # Exibir a imagem suavizada
22 plt.subplot(2, 2, 3)
23 plt.imshow(cv2.cvtColor(imagem_suavizada_9, cv2.COLOR_BGR2RGB))
24 plt.title('Imagem Suavizada 9x9')

25 # Exibir a imagem suavizada
26 plt.subplot(2, 2, 4)
27 plt.imshow(cv2.cvtColor(imagem_suavizada_15, cv2.COLOR_BGR2RGB))
28 plt.title('Imagem Suavizada 15x15')

29 # Configurar layout
30 plt.tight_layout()

31 # Exibir as imagens
32 plt.show()
```

Saída:

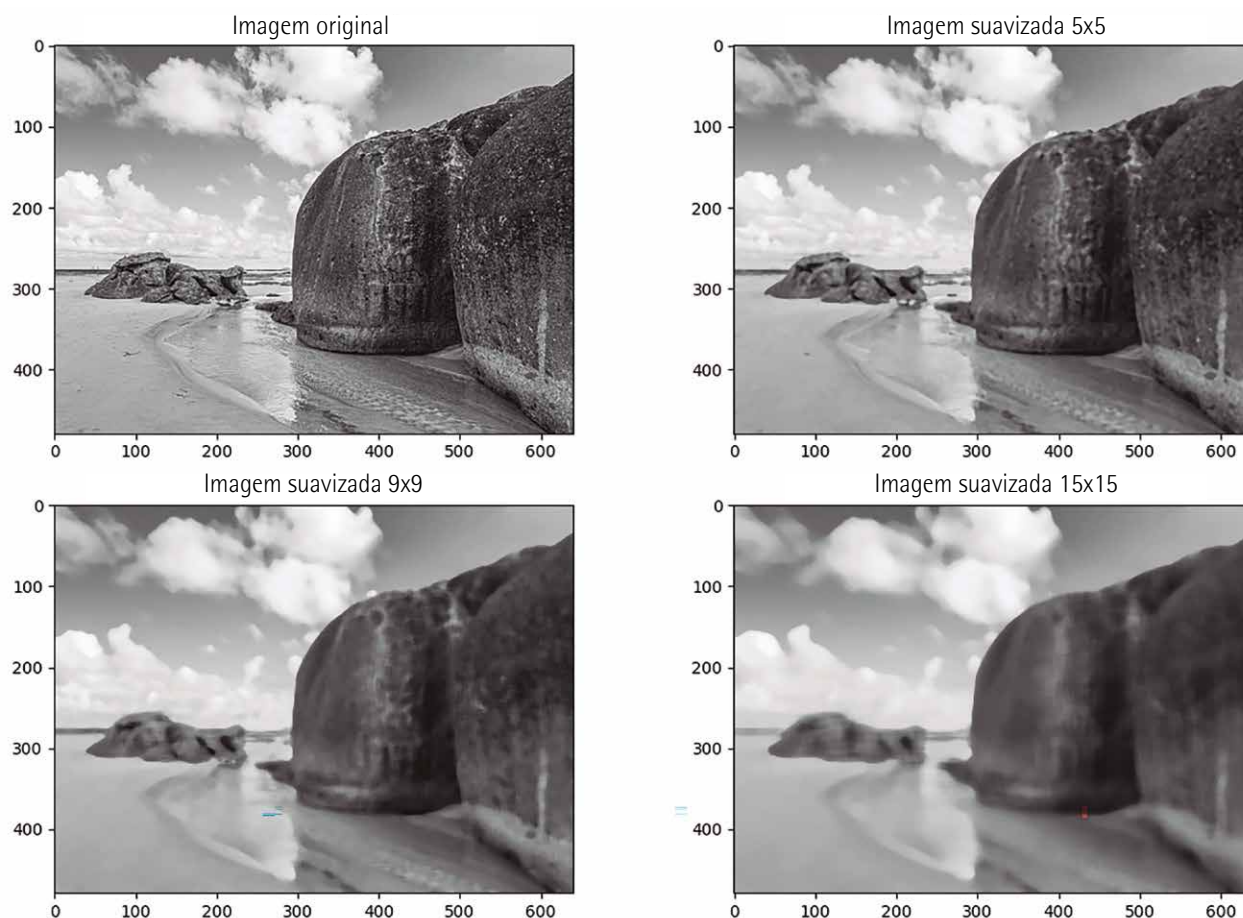


Figura 25 – Squeaky Beach em Wilsons Promontory National Park, Victoria, Austrália, 2019 (640 × 479 pixels)

Disponível em: <https://tinyurl.com/sze7hysm>. Acesso em: 25 out. 2023.

A suavização pela mediana é uma técnica eficaz para reduzir o ruído em imagens, especialmente ruído de sal e pimenta. Ela é muito usada em processamento de imagem devido à sua capacidade de preservar detalhes importantes enquanto reduz o impacto do ruído. A escolha do tamanho da vizinhança depende das características da imagem e dos objetivos da aplicação.

### 3.4 Suavização com filtro bilateral

A suavização de imagens utilizando o método de suavização com filtro bilateral é uma técnica avançada e eficaz para reduzir o ruído em imagens enquanto preserva bordas e detalhes importantes. Esse método é particularmente útil quando se deseja manter a clareza das bordas e estruturas na imagem, ao mesmo tempo em que se reduz o impacto do ruído. O filtro bilateral é uma alternativa aos filtros tradicionais de suavização, como a suavização gaussiana ou de média, que não consideram a estrutura da imagem. Os principais conceitos e passos envolvidos na suavização com filtro bilateral são:

- **Vizinhança do pixel:** como em outras técnicas de suavização, uma vizinhança é definida ao redor de cada pixel na imagem. A vizinhança é geralmente uma janela quadrada ou retangular de tamanho predefinido.
- **Cálculo do peso bilateral:** o filtro bilateral leva em consideração dois tipos de pesos, um peso espacial e um peso de intensidade. O peso espacial é baseado na distância entre os pixels na vizinhança e o pixel central, enquanto o peso de intensidade é calculado com base na diferença de intensidade entre os pixels.
- **Suavização ponderada:** para suavizar um pixel específico, o filtro bilateral calcula uma média ponderada dos valores dos pixels na vizinhança. A ponderação leva em consideração tanto o peso espacial quanto o peso de intensidade. Pixels próximos e com intensidades semelhantes têm mais peso na média, enquanto pixels distantes ou com intensidades significativamente diferentes têm menos peso.
- **Substituição do valor do pixel:** o valor do pixel original é substituído pelo valor da média ponderada calculada. Isso suaviza a imagem, reduzindo o ruído, enquanto preserva as bordas e detalhes.
- **Repetição para todos os pixels:** o processo de suavização com filtro bilateral é repetido para todos os pixels na imagem.
- **Ajuste de parâmetros:** a eficácia da suavização com filtro bilateral pode ser controlada ajustando os parâmetros, como a largura da janela espacial e a largura da janela de intensidade. Valores maiores resultam em uma suavização mais forte, enquanto valores menores mantêm mais detalhes.

Aplicações:

- Remoção de ruído em imagens fotográficas.
- Melhoria da qualidade visual de imagens.
- Preservação de detalhes importantes em imagens de visão computacional e processamento de imagem médica.

Veja a seguir um exemplo de pseudocódigo em Python. Trata-se de um exemplo simples de como implementar a suavização com filtro bilateral (kernel de 5, 9 e 15 e variando a intensidade de 40 a 140) em Python usando a biblioteca OpenCV:

```
1 import cv2
2 import matplotlib.pyplot as plt

3 # Carregue a imagem
4 imagem = cv2.imread('sua_imagem.jpg')

5 # Aplique o filtro bilateral com janelas espaciais e de intensidade de tamanho 5
6 imagem_suavizada_5 = cv2.bilateralFilter(imagem, 5, 40, 40)

7 # Aplique o filtro bilateral com janelas espaciais e de intensidade de tamanho 9
8 imagem_suavizada_9 = cv2.bilateralFilter(imagem, 9, 80, 80)

9 # Aplique o filtro bilateral com janelas espaciais e de intensidade de tamanho 15
10 imagem_suavizada_15 = cv2.bilateralFilter(imagem, 15, 140, 140)

11 # Exiba a imagem original e a imagem suavizada
12 plt.figure(figsize=(12, 8))

13 # Exibir a imagem original
14 plt.subplot(2, 2, 1)
15 plt.imshow(cv2.cvtColor(imagem, cv2.COLOR_BGR2RGB)) # Converter de BGR para RGB para
exibição correta
16 plt.title('Imagem Original')

17 # Exibir a imagem suavizada
18 plt.subplot(2, 2, 2)
19 plt.imshow(cv2.cvtColor(imagem_suavizada_5, cv2.COLOR_BGR2RGB))
20 plt.title('Imagem Suavizada 5x5 - 40')

21 # Exibir a imagem suavizada
22 plt.subplot(2, 2, 3)
23 plt.imshow(cv2.cvtColor(imagem_suavizada_9, cv2.COLOR_BGR2RGB))
24 plt.title('Imagem Suavizada 9x9 - 80')

25 # Exibir a imagem suavizada
26 plt.subplot(2, 2, 4)
27 plt.imshow(cv2.cvtColor(imagem_suavizada_15, cv2.COLOR_BGR2RGB))
28 plt.title('Imagem Suavizada 15x15 - 140')

29 # Configurar layout
30 plt.tight_layout()

31 # Exibir as imagens
32 plt.show()
```

Saída:



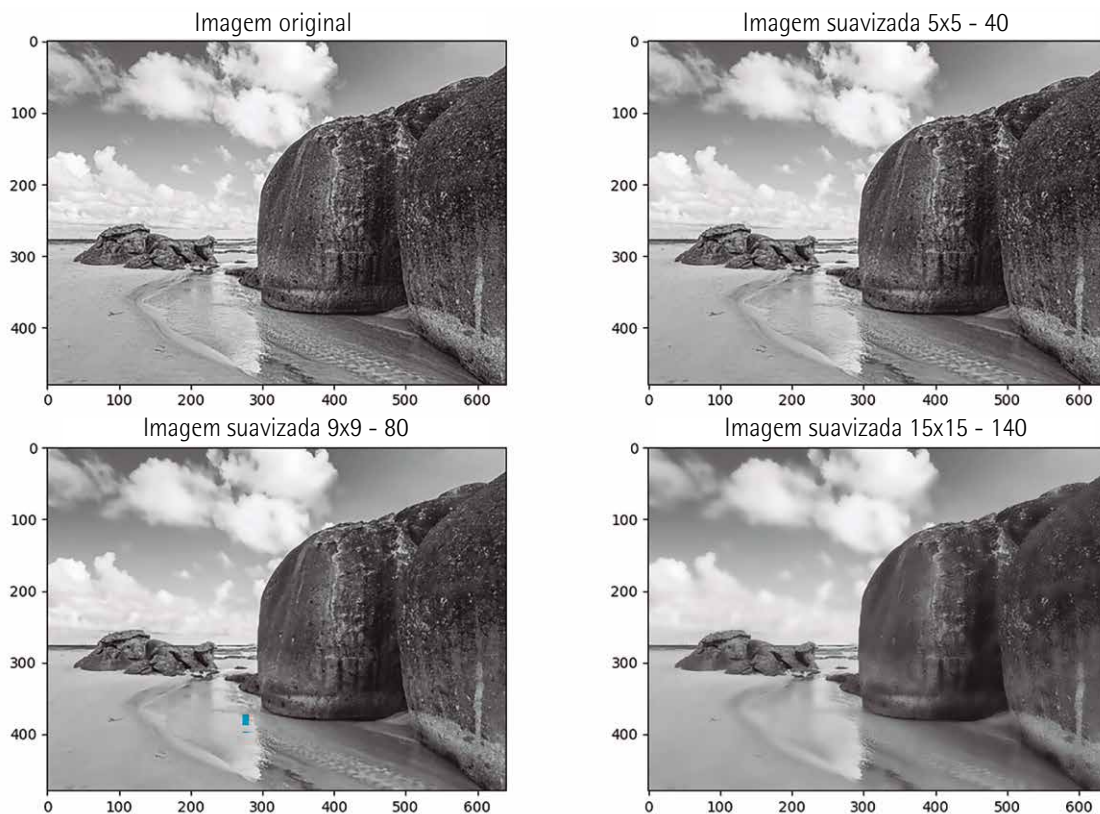


Figura 26 – Squeaky Beach em Wilsons Promontory National Park, Victoria, Austrália, 2019 (640 × 479 pixels)

Disponível em: <https://tinyurl.com/sze7hysm>. Acesso em: 25 out. 2023.

Observe como mesmo com a grande interferência na imagem no caso da imagem mais à baixo e à direita as bordas são preservadas. O filtro bilateral é uma técnica poderosa para suavização de imagens, especialmente quando se deseja reduzir o ruído sem comprometer as bordas e os detalhes importantes. Os parâmetros do filtro devem ser ajustados de acordo com as características da imagem e os objetivos da aplicação.

## 4 SEGMENTAÇÃO DE IMAGENS

A segmentação de imagens é um dos tópicos fundamentais no processamento de imagens e visão computacional. Ela envolve a subdivisão de uma imagem em regiões ou objetos significativos, com o objetivo de identificar e isolar áreas de interesse. A segmentação é uma etapa crítica em muitas aplicações de processamento de imagem, pois permite a extração de informações relevantes da imagem, facilitando análises e tomada de decisões. Os principais conceitos e técnicas relacionados à segmentação de imagens são:

- **Objetivo da segmentação:** o principal objetivo da segmentação é dividir uma imagem em regiões que contenham objetos ou características semelhantes. Isso pode incluir a detecção de bordas, a identificação de objetos, a separação de fundo e primeiro plano, entre outros.

Tipos de segmentação:

- **Segmentação por detecção de bordas:** identifica as fronteiras entre diferentes regiões da imagem.
- **Segmentação por região:** agrupa pixels semelhantes em regiões distintas.
- **Segmentação semântica:** atribui significado semântico aos objetos na imagem.
- **Segmentação de instância:** identifica instâncias individuais de objetos, permitindo diferenciar objetos da mesma classe.
  - **Métodos de segmentação:** existem várias técnicas e algoritmos para realizar a segmentação de imagens, incluindo:
    - **Métodos baseados em intensidade:** esses métodos se baseiam nas intensidades de pixel para separar regiões, como o algoritmo de limiarização.
    - **Métodos baseados em bordas:** identificam bordas e contornos na imagem usando gradientes de intensidade.
    - **Métodos baseados em regiões:** agrupam pixels semelhantes em regiões com base em critérios como cor, textura ou intensidade.
- **Métodos de aprendizado de máquina:** utilizam algoritmos de aprendizado de máquina para aprender a segmentação com base em dados de treinamento.
- **Desafios na segmentação:** a segmentação de imagens pode ser uma tarefa desafiadora devido a diversos fatores, incluindo a presença de ruído, variações de iluminação, sobreposição de objetos e a escolha adequada de parâmetros.
- **Aplicações da segmentação de imagens:** a segmentação de imagens é utilizada em diversas aplicações, tais como:
  - **Medicina:** segmentação de órgãos em imagens médicas, como ressonâncias magnéticas.
  - **Visão computacional:** identificação de objetos em cenas, rastreamento de objetos em vídeos e reconhecimento de gestos.
  - **Processamento de imagens geoespaciais:** extração de informações de imagens de satélite.
  - **Controle de qualidade industrial:** inspeção e controle de qualidade de produtos usando visão computacional.
  - **Condução autônoma:** identificação de objetos e obstáculos em veículos autônomos.

- **Avaliação da segmentação:** a qualidade de uma segmentação pode ser avaliada usando métricas como precisão, revocação, índice Jaccard (IoU), entropia cruzada, entre outras, que medem a sobreposição entre a segmentação resultante e uma segmentação de referência.
  - Precisão (accuracy):
    - Precisão é uma métrica que mede a proporção de previsões corretas em relação ao número total de previsões.
    - É calculada pela fórmula:  $(\text{Verdadeiros Positivos} + \text{Verdadeiros Negativos}) / (\text{Verdadeiros Positivos} + \text{Falsos Positivos} + \text{Verdadeiros Negativos} + \text{Falsos Negativos})$ .
    - É uma medida geral da exatidão do modelo.
  - Revocação (recall):
    - Revocação é uma métrica que calcula a capacidade do modelo de identificar todos os exemplos positivos corretamente.
    - Também é chamada de sensibilidade ou taxa de verdadeiros positivos.
    - É calculada pela fórmula:  $\text{Verdadeiros Positivos} / (\text{Verdadeiros Positivos} + \text{Falsos Negativos})$ .
  - Índice Jaccard (IoU – intersection over union):
    - O índice Jaccard mede a sobreposição entre dois conjuntos, frequentemente usado para avaliar a precisão da segmentação de objetos em imagens.
    - É calculado pela fórmula:  $\text{IoU} = (\text{Área da Interseção}) / (\text{Área da União})$ .
    - Quanto maior o valor do IoU, melhor a sobreposição entre a segmentação prevista e a verdadeira.
  - Entropia cruzada (cross-entropy):
    - A entropia cruzada é uma métrica comum para avaliar a qualidade das previsões de um modelo de classificação.
    - Em tarefas de classificação binária, é calculada pela fórmula:  $-\sum y \log p - \sum (1-y) \log (1-p)$ , onde  $y$  é o rótulo verdadeiro (0 ou 1) e  $p$  é a probabilidade prevista.
    - Quanto menor a entropia cruzada, melhor o desempenho do modelo, pois ela mede a divergência entre as previsões e os rótulos reais.

Essas métricas são essenciais para avaliar o desempenho de modelos de aprendizado de máquina em tarefas de classificação e segmentação de imagem. Elas fornecem insights sobre a qualidade das previsões e a capacidade do modelo de identificar corretamente os exemplos positivos.

### Desenvolvimentos recentes

Com o avanço da inteligência artificial e do aprendizado profundo (deep learning), redes neurais convolucionais têm sido aplicadas com sucesso à segmentação de imagens, permitindo resultados mais precisos em muitas aplicações.

A segmentação de imagens é importante em muitas áreas e é uma ferramenta poderosa para extrair informações valiosas de dados visuais. A escolha do método de segmentação depende das características da imagem e dos objetivos da aplicação específica.

O aprendizado profundo é um subcampo da inteligência artificial que se concentra no treinamento de modelos de aprendizado de máquina profundos, compostos de várias camadas de neurônios artificiais. Essas redes profundas são capazes de aprender automaticamente a representação de dados complexos e extrair características hierárquicas.

O aprendizado profundo é chamado "profundo" devido à profundidade das redes neurais envolvidas, com muitas camadas intermediárias, o que permite que esses modelos capturem padrões e informações de alto nível em dados brutos.

Redes neurais convolucionais, ou CNNs, são um tipo específico de arquitetura de rede neural projetada para processar dados com estrutura de grade, como imagens.

As CNNs são adequadas para tarefas de visão computacional, pois usam camadas de convolução para detectar características locais em diferentes partes da imagem, como bordas, texturas e padrões. Essas redes são caracterizadas por sua capacidade de compartilhar pesos, o que as torna eficientes em termos de aprendizado e essenciais para o treinamento bem-sucedido de modelos em grandes volumes de dados.



### Lembrete

O aprendizado profundo é um subcampo da inteligência artificial que se concentra no treinamento de modelos de aprendizado de máquina profundos, compostos de várias camadas de neurônios artificiais. Essas redes profundas são capazes de aprender automaticamente a representação de dados complexos e extrair características hierárquicas.

## 4.1 Detecção de bordas

A segmentação de imagens utilizando detecção de bordas é uma técnica que visa identificar as transições abruptas de intensidade de pixel na imagem, o que muitas vezes corresponde a bordas ou limites entre objetos. A detecção de bordas é um passo importante para a segmentação, pois fornece informações sobre onde os objetos começam e terminam na imagem.

A detecção de bordas frequentemente começa com a aplicação de operadores de gradiente, como o operador de Sobel, o operador de Prewitt ou o operador de Roberts, para calcular as derivadas da intensidade dos pixels na imagem em relação às coordenadas  $x$  e  $y$ . Essas derivadas indicam as taxas de mudança de intensidade em direções horizontal e vertical e, portanto, ajudam a identificar regiões onde ocorrem transições abruptas.

A magnitude do gradiente em cada ponto da imagem é calculada a partir das derivadas obtidas na etapa anterior. A magnitude é frequentemente calculada como a raiz quadrada da soma dos quadrados das derivadas (usando o teorema de Pitágoras). A magnitude do gradiente fornece informações sobre o quão rápido a intensidade muda em um ponto da imagem. Pontos com magnitudes de gradiente maiores são mais prováveis de estarem localizados em uma borda.

Além da magnitude, a orientação do gradiente é calculada em cada ponto da imagem. Isso ajuda a determinar a direção da mudança de intensidade. A orientação do gradiente é usada para quantizar a direção das bordas, dividindo a faixa de ângulos em um número limitado de direções (por exemplo, vertical, horizontal e diagonais).

O próximo passo é aplicar um limiar à magnitude do gradiente. A limiarização envolve a definição de um valor de corte: abaixo desse valor, os pixels são considerados não pertencentes a bordas; acima dele, os pixels são considerados parte de uma borda. A escolha adequada do valor de limiar é uma etapa crítica e pode variar dependendo da aplicação.

Em muitos algoritmos de detecção de bordas, é realizada uma etapa de supressão de não máximos para reduzir o número de pixels considerados como bordas. Isso envolve verificar se os pixels vizinhos têm uma magnitude de gradiente maior e, se não, o pixel atual é suprimido.

Para obter segmentos de borda contínuos, é comum realizar uma etapa de conexão de borda que une pixels adjacentes com intensidades acima do limiar em segmentos de borda.

## 4.2 Sobel

A segmentação de imagens utilizando o operador Sobel é uma técnica comum para detectar bordas e transições de intensidade em uma imagem. O operador Sobel é usado para calcular gradientes de intensidade em direções horizontal e vertical, o que pode ajudar a identificar regiões de mudança significativa na imagem, muitas vezes correspondentes a bordas de objetos. Veja um exemplo de como realizar a segmentação de imagens usando o operador Sobel em Python com a biblioteca OpenCV:

```
1 import cv2
2 import numpy as np
3 import matplotlib.pyplot as plt

4 # Carregue a imagem em tons de cinza
5 imagem = cv2.imread('sua_imagem.jpg', cv2.IMREAD_GRAYSCALE)

6 # Aplique o operador de Sobel para detecção de bordas na direção x (horizontal)
7 bordas_x = cv2.Sobel(imagem, cv2.CV_64F, 1, 0, ksize=3)

8 # Aplique o operador de Sobel para detecção de bordas na direção y (vertical)
9 bordas_y = cv2.Sobel(imagem, cv2.CV_64F, 0, 1, ksize=3)

10 # Calcule a magnitude das bordas (combinação das bordas horizontais e verticais)
11 magnitude_bordas = np.sqrt(bordas_x**2 + bordas_y**2)

12 # Aplique um limiar para segmentar as bordas
13 limiar = 100 # Você pode ajustar esse valor conforme necessário
14 imagem_segmentada = cv2.threshold(magnitude_bordas, limiar, 255, cv2.THRESH_BINARY)[1]

15 # Exiba a imagem original e a imagem suavizada
16 plt.figure(figsize=(9, 10))

17 plt.subplot(3, 2, 1)
18 plt.imshow(imagem, cmap='gray')
19 plt.title('Imagem Original')
20 plt.axis('off')

21 plt.subplot(3, 2, 2)
22 bordas_x = np.uint8(np.absolute(bordas_x))
23 plt.imshow(bordas_x, cmap='gray')
24 plt.title('Sobel Horizontal (SobelX)')
25 plt.axis('off')

26 plt.subplot(3, 2, 3)
27 bordas_y = np.uint8(np.absolute(bordas_y))
28 plt.imshow(bordas_y, cmap='gray')
29 plt.title('Sobel Vertical (SobelY)')
30 plt.axis('off')

31 plt.subplot(3, 2, 4)
32 plt.imshow(magnitude_bordas, cmap='gray')
33 plt.title('Magnitude das Bordas')
34 plt.axis('off')
35 plt.subplot(3, 2, 5)
36 plt.imshow(imagem_segmentada, cmap='gray')
37 plt.title('Imagem Segmentada')
38 plt.axis('off')
```



39 plt.show()

Saída:

Imagem original



Sobel horizontal (SobelX)



Sobel vertical (SobelY)



Magnitude das bordas



Imagem segmentada



Figura 27



Neste exemplo, carregamos uma imagem em tons de cinza. Aplicamos o operador Sobel na direção x (horizontal) e na direção y (vertical) para detectar bordas nas duas direções. Calculamos a magnitude das bordas combinando as bordas horizontais e verticais usando a fórmula da hipotenusa. Aplicamos um limiar para segmentar as bordas da imagem. Pixels com intensidade maior que o limiar são definidos como objetos, enquanto pixels com intensidade menor que o limiar são definidos como fundo.

A imagem resultante, "imagem\_segmentada", mostra as regiões da imagem em que as bordas foram detectadas. Você pode ajustar o valor do limiar de acordo com a sua imagem e requisitos de segmentação. Este é um exemplo básico de como realizar a segmentação de imagens usando o operador Sobel em Python.

### 4.3 Operador laplaciano

A segmentação de imagens utilizando o operador laplaciano é uma técnica que visa realçar áreas em que ocorrem mudanças abruptas de intensidade, o que muitas vezes corresponde a bordas e detalhes significativos na imagem. O operador laplaciano é usado para calcular a segunda derivada da intensidade da imagem e pode ser aplicado para identificar áreas de interesse na imagem. Veja um exemplo de como realizar a segmentação de imagens usando o operador laplaciano em Python com a biblioteca OpenCV:

```
1 import cv2
2 import numpy as np
3 import matplotlib.pyplot as plt

4 # Carregue a imagem em tons de cinza
5 imagem = cv2.imread('sua_imagem.jpg', cv2.IMREAD_GRAYSCALE)

6 # Aplique o operador laplaciano para detecção de bordas
7 laplaciano = cv2.Laplacian(imagem, cv2.CV_64F)

8 # Converta a imagem resultante para valores não negativos
9 laplaciano_absoluto = np.abs(laplaciano)

10 # Aplique um limiar para segmentar as bordas
11 limiar = 100 # Você pode ajustar esse valor conforme necessário
12 imagem_segmentada = cv2.threshold(laplaciano_absoluto, limiar, 255, cv2.THRESH_BINARY)[1]

13 # Exiba a imagem original, o operador laplaciano e a imagem segmentada
14 plt.figure(figsize=(10, 8))

15 plt.subplot(2, 2, 1)
16 plt.imshow(imagem, cmap='gray')
17 plt.title('Imagem Original')
18 plt.axis('off')
```

```
19 plt.subplot(2, 2, 2)
20 plt.imshow(laplaciano_absoluto, cmap='gray')
21 plt.title('Operador Laplaciano')
22 plt.axis('off')

23 plt.subplot(2, 2, 3)
24 plt.imshow(imagem_segmentada, cmap='gray')
25 plt.title('Imagem Segmentada')
26 plt.axis('off')

27 plt.show()
```

Saída:

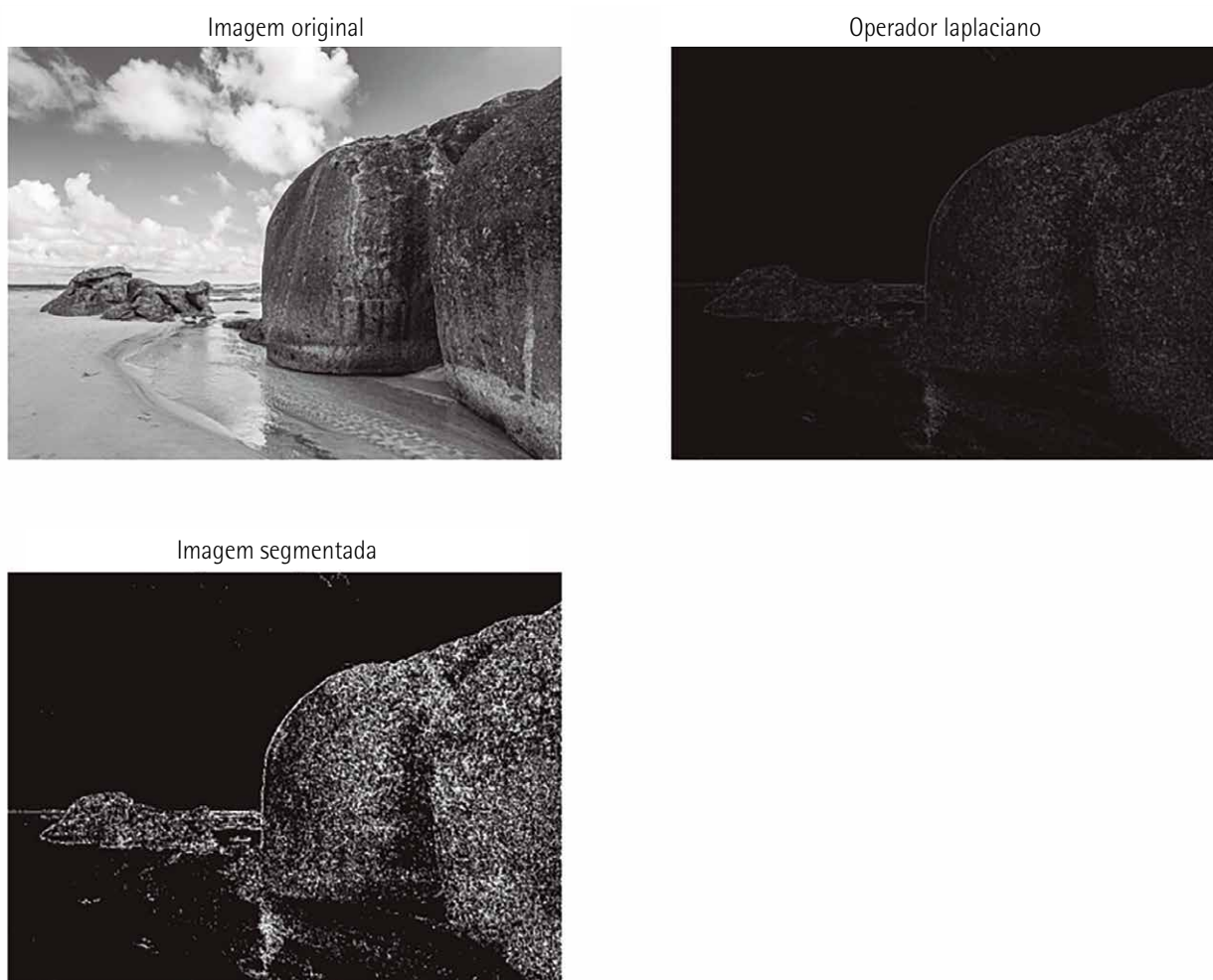


Figura 28

Neste exemplo, carregamos uma imagem em tons de cinza. Aplicamos o operador laplaciano para calcular a segunda derivada da intensidade da imagem, que realça as mudanças abruptas de intensidade. Para garantir que a imagem resultante seja não negativa, calculamos o valor absoluto do resultado do operador laplaciano. Aplicamos um limiar para segmentar as áreas da imagem em que as bordas foram detectadas. Pixels com intensidade maior que o limiar são definidos como objetos, enquanto pixels com intensidade menor que o limiar são definidos como fundo.

A imagem resultante, "imagem\_segmentada", destaca as áreas em que ocorrem mudanças abruptas de intensidade, que geralmente correspondem a bordas e detalhes na imagem. Você pode ajustar o valor do limiar de acordo com a sua imagem e requisitos de segmentação. Este é um exemplo simples de como realizar a segmentação de imagens usando o operador laplaciano em Python.

### 4.4 Filtro máscara de desaguçamento

A segmentação de imagens utilizando um filtro de máscara de desfoque (ou desagucinação) é uma técnica que visa reduzir o ruído e as pequenas variações de intensidade na imagem, tornando-a mais homogênea. Isso pode ajudar a destacar as regiões de interesse na imagem, tornando a segmentação posterior mais eficaz. Um filtro comum usado para desfoque é o filtro da média, que substitui o valor de um pixel pela média dos valores dos pixels em sua vizinhança. Veja um exemplo de como realizar a segmentação de imagens usando um filtro de máscara de desfoque em Python com a biblioteca OpenCV:

```
1 import cv2
2 import numpy as np
3 import matplotlib.pyplot as plt

4 # Carregue a imagem em tons de cinza
5 imagem = cv2.imread('sua_imagem.jpg', cv2.IMREAD_GRAYSCALE)

6 # Aplique um filtro de máscara de desfoque (filtro da média) para reduzir o ruído
7 tamanho_kernel = 5 # Tamanho da janela do filtro (ímpar)
8 imagem_desfocada = cv2.blur(imagem, (tamanho_kernel, tamanho_kernel))

9 # Aplique um limiar para segmentar a imagem desfocada
10 limiar = 100 # Você pode ajustar esse valor conforme necessário
11 imagem_segmentada = cv2.threshold(imagem_desfocada, limiar, 255, cv2.THRESH_BINARY)[1]

12 # Exiba a imagem original, a imagem desfocada e a imagem segmentada
13 plt.figure(figsize=(10, 8))

14 plt.subplot(2, 2, 1)
15 plt.imshow(imagem, cmap='gray')
16 plt.title('Imagem Original')
17 plt.axis('off')
```

```
18 plt.subplot(2, 2, 2)
19 plt.imshow(imagem_desfocada, cmap='gray')
20 plt.title('Imagem Desfocada')
21 plt.axis('off')

22 plt.subplot(2, 2, 3)
23 plt.imshow(imagem_segmentada, cmap='gray')
24 plt.title('Imagem Segmentada')
25 plt.axis('off')

26 plt.show()
```

Saída:

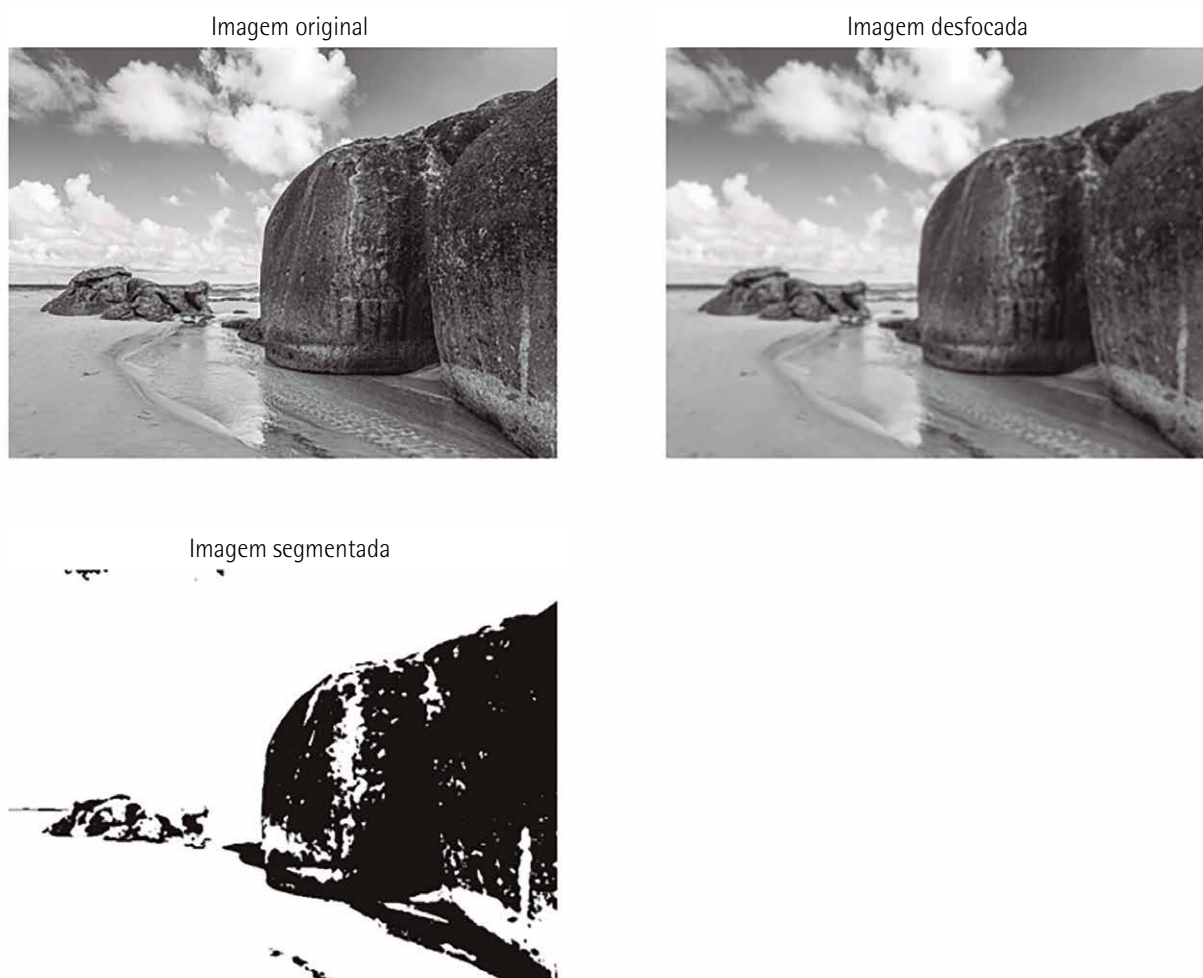


Figura 29

Neste exemplo, carregamos uma imagem em tons de cinza. Aplicamos um filtro de máscara de desfoque (filtro da média) para suavizar a imagem e reduzir o ruído. O tamanho do kernel define o tamanho da janela usada para calcular a média. Aplicamos um limiar na imagem desfocada para segmentar as áreas da imagem que desejamos destacar. Pixels com intensidade maior que o limiar são definidos como objetos, enquanto pixels com intensidade menor que o limiar são definidos como fundo.

A imagem resultante, "imagem\_segmentada", destaca as regiões da imagem que são mais homogêneas após a aplicação do filtro de desfoque. Você pode ajustar o tamanho do kernel do filtro e o valor do limiar de acordo com a sua imagem e requisitos de segmentação.

### 4.5 Detector de bordas de Canny

A segmentação de imagens usando o detector de bordas de Canny é uma técnica muito utilizada para identificar as bordas e limites entre objetos em uma imagem. O detector de bordas de Canny é conhecido por ser eficaz na detecção de bordas finas e bem definidas, enquanto suprime o ruído na imagem. Veja um exemplo de como realizar a segmentação de imagens usando o detector de bordas de Canny em Python com a biblioteca OpenCV:

```
1 import cv2
2 import matplotlib.pyplot as plt

3 # Carregue a imagem em tons de cinza
4 imagem = cv2.imread('sua_imagem.jpg', cv2.IMREAD_GRAYSCALE)

5 # Aplique o detector de bordas de Canny
6 bordas_canny = cv2.Canny(imagem, threshold1=210, threshold2=240) # Você pode ajustar
    esses valores conforme necessário

7 # Exiba a imagem original e as bordas detectadas
8 plt.figure(figsize=(10, 5))

9 plt.subplot(1, 2, 1)
10 plt.imshow(imagem, cmap='gray')
11 plt.title('Imagem Original')
12 plt.axis('off')

13 plt.subplot(1, 2, 2)
14 plt.imshow(bordas_canny, cmap='gray')
15 plt.title('Bordas Detectadas (Canny)')
16 plt.axis('off')

17 plt.show()
```

Saída:



Figura 30

Neste exemplo, carregamos uma imagem em tons de cinza. Aplicamos o detector de bordas de Canny usando a função `cv2.Canny()`. Os parâmetros `threshold1` e `threshold2` controlam os limites inferior e superior de intensidade de gradiente para a detecção de bordas. Você pode ajustar esses valores conforme necessário para obter as bordas desejadas. O resultado é a imagem de bordas detectadas, em que as regiões brancas correspondem às bordas na imagem original.

O detector de bordas de Canny é útil quando você deseja identificar bordas nítidas e bem definidas em uma imagem. As bordas detectadas podem ser usadas posteriormente para realizar segmentação de objetos ou para análise de características na imagem. Ajustar os limiares é importante para obter resultados adequados, pois os valores ideais podem variar dependendo da imagem e das condições específicas.



### Resumo

Vimos nesta unidade que o processamento de imagem é uma área da visão computacional que se concentra em manipular imagens digitais para melhorar a qualidade, extrair informações significativas e facilitar análises posteriores. O processamento de imagem é usado em uma variedade de aplicações, como visão de máquina, diagnóstico médico, visão de computador e muito mais.

Apresentamos algumas técnicas de pré-processamento, suavização de imagem e segmentação de imagens.

Como técnica de pré-processamento, a rotação é usada para corrigir a orientação da imagem, alinhando-a corretamente. Isso pode ser útil em casos em que a imagem está inclinada. O histograma de cores é uma representação gráfica da distribuição das cores na imagem. Analisar o histograma pode ajudar a ajustar o equilíbrio de cores e corrigir problemas de saturação. As operações aritméticas permitem combinar duas ou mais imagens para obter um novo resultado. Por exemplo, a soma ou subtração de imagens pode ser usada para realçar características.

As transformações geométricas, como escala, translação e cisalhamento, são usadas para modificar a geometria da imagem. Elas são úteis para ajustar o tamanho e a posição da imagem. A adição de ruído artificial à imagem pode ser usada para aumentar a robustez do modelo de processamento de imagem, permitindo que ele se adapte a diferentes níveis de ruído nos dados. Converter uma imagem colorida em escala de cinza é útil quando a cor não é relevante e se deseja simplificar o processamento.

A suavização de imagem é uma técnica de processamento de imagem que tem como objetivo reduzir o ruído e as pequenas variações na intensidade dos pixels. Isso é feito aplicando um filtro que faz uma média das intensidades dos pixels em uma vizinhança específica.

A suavização pela média, também conhecida como filtragem média, calcula a intensidade do pixel de saída como a média das intensidades dos pixels em uma vizinhança. Esse filtro é eficaz na redução de ruído de alta frequência, mas pode causar alguma perda de detalhes. A suavização gaussiana é um método que aplica um filtro ponderado pela distribuição gaussiana. A intensidade do pixel de saída é calculada como uma média ponderada das intensidades dos pixels na vizinhança,



em que os pesos seguem uma função gaussiana. A suavização gaussiana é eficaz na redução de ruído e, ao mesmo tempo, preserva detalhes.

A suavização pela mediana é uma técnica que substitui o valor do pixel de saída pela mediana das intensidades dos pixels na vizinhança. Essa técnica é excelente para remover ruídos de sal e pimenta, pois não é afetada por valores extremos.

O filtro bilateral é uma técnica mais sofisticada que realiza uma suavização ponderada, levando em consideração não apenas a proximidade espacial, mas também a diferença de intensidade entre os pixels. Isso ajuda a preservar bordas e detalhes enquanto suaviza áreas de baixo contraste.

A detecção de bordas envolve a identificação de transições abruptas de intensidade na imagem. Técnicas como Sobel, laplaciano, filtro de máscara de desaguçamento e detector de bordas de Canny são comumente usadas.

O operador de Sobel é usado para calcular gradientes de intensidade na direção  $x$  e  $y$ . A magnitude do gradiente indica onde as bordas estão presentes. O operador laplaciano enfatiza mudanças de segunda ordem na intensidade da imagem, identificando regiões de transição abrupta.

Filtro de máscara de desaguçamento realça as bordas, enfatizando as transições de intensidade e suavizando o restante da imagem. O detector de bordas de Canny é um método mais sofisticado que combina detecção de bordas, supressão de não máximos e histerese para identificar bordas nítidas.

O pré-processamento, suavização e segmentação de imagens são etapas fundamentais no processamento de imagens e desempenham um papel importante na extração de informações úteis para análises subsequentes, como reconhecimento de objetos, segmentação de regiões de interesse e muito mais. Cada técnica tem suas próprias vantagens e desvantagens e é escolhida com base na natureza dos dados e na aplicação específica.



### Exercícios

**Questão 1.** (FGV/2014, adaptada) Na área de processamento de imagens, o histograma de cores é uma ferramenta fundamental para a análise e o ajuste de uma imagem, o que permite a compreensão visual das suas características de cores. Sobre o histograma e seu uso durante o tratamento de imagens, avalie as afirmativas:

I – O histograma ilustra como os pixels são distribuídos em uma imagem na forma de um gráfico do número de pixels em cada nível de intensidade de cor.

II – Um histograma pode ajudar a determinar se uma imagem tem detalhes suficientes para fazer uma boa correção.

III – O histograma pode descrever apenas a imagem inteira, sem informações individuais dos seus canais de cores.

É correto o que se afirma em:

A) I, apenas.

B) III, apenas.

C) I e II, apenas.

D) II e III, apenas.

E) I, II e III.

Resposta correta: alternativa C.

#### Análise das afirmativas

I – Afirmativa correta.

Justificativa: em processamento de imagens, o histograma é uma representação gráfica que mostra a distribuição de intensidades de cor ou de brilho em uma imagem. No eixo horizontal, temos os diferentes níveis de intensidade de cor, e no eixo vertical, temos as frequências de pixels com cada intensidade. Isso permite visualizar como os tons de cores estão distribuídos na imagem, o que é útil para compreender as características de cores da imagem.

II – Afirmativa correta.

Justificativa: ao observar o histograma de uma imagem, é possível identificar se a imagem tem bom alcance de tons e detalhes. Se o histograma se estender por toda a faixa de intensidades (ou seja, se ele não estiver muito concentrado em um único ponto), é mais provável que haja detalhes suficientes na imagem.

III – Afirmativa incorreta.

Justificativa: é possível criar histogramas separados para os canais de cores individuais. Por exemplo, para uma imagem em espaço de cores RGB, é possível criar três histogramas: um para o canal vermelho, um para o canal verde e um para o canal azul.

**Questão 2.** A segmentação de imagens é um dos tópicos fundamentais no processamento de imagens e na visão computacional. Ela envolve a subdivisão de uma imagem em regiões, com o objetivo de identificar e isolar áreas de interesse. A respeito desse tema, avalie as afirmativas:

I – Em uma imagem segmentada, cada região deve apresentar grande diversidade de características.

II – Entre os tipos de segmentação, estão a segmentação por detecção de bordas e a segmentação semântica.

III – A segmentação de imagens pode ser aplicada em áreas, como a medicina e o controle de qualidade industrial.

É correto o que se afirma em:

A) I, apenas.

B) III, apenas.

C) I e II, apenas.

D) II e III, apenas.

E) I, II e III.

Resposta correta: alternativa D.

### Análise das afirmativas

I – Afirmativa incorreta.

Justificativa: o principal objetivo da segmentação é dividir uma imagem em regiões que contenham objetos ou características semelhantes, e não diversas. Cada região, portanto, deve representar uma porção relativamente homogênea da imagem original.

II – Afirmativa correta.

Justificativa: ao longo do conteúdo do livro-texto, foram abordados quatro tipos de segmentação, elencados a seguir:

- **Segmentação por detecção de bordas:** há a identificação de fronteiras entre diferentes regiões da imagem.
- **Segmentação por região:** há o agrupamento de pixels semelhantes.
- **Segmentação semântica:** há a atribuição de significado semântico aos objetos da imagem.
- **Segmentação de instância:** há a identificação das instâncias individuais de objetos, o que permite diferenciar objetos de mesma classe.

III – Afirmativa correta.

Justificativa: a segmentação de imagens é utilizada em diversas aplicações, como medicina, visão computacional, processamento de imagens geoespaciais, controle de qualidade industrial e condução autônoma.

[illegible]