

Segmentação de imagens

A segmentação de imagem, uma disciplina-chave no processamento de imagem e visão computacional, desempenha um papel crucial ao subdividir imagens em regiões significativas com base em características visuais como cor e textura. Essa técnica é fundamental em diversas indústrias, como medicina e automação industrial, permitindo a identificação precisa de estruturas em imagens médicas e a inspeção de qualidade de produtos. A capacidade de isolar objetos ou padrões específicos torna a segmentação de imagem uma ferramenta essencial para a análise e interpretação de conteúdo visual em uma variedade de contextos práticos.

Segmentação por k-Means:

A técnica escolhida para a segmentação de imagem neste contexto é o algoritmo K-Means. O K-Means é um método de agrupamento que visa partitionar um conjunto de dados em K clusters, onde cada ponto de dado pertence ao cluster cujo centro é mais próximo. Quando aplicado à segmentação de imagem, o K-Means agrupa pixels semelhantes com base em suas características, como intensidade de cor, em clusters representativos

Por quem foi criada: A técnica de K-Means foi proposta por Stuart Lloyd em 1957, enquanto ele trabalhava na Bell Labs. Entretanto, o algoritmo foi formalizado e popularizado por James MacQueen em 1967, que cunhou o termo "K-Means". O nome "K-Means" refere-se à natureza do algoritmo, onde "K" representa o número de clusters desejado.

Motivação: O K-Means foi originalmente desenvolvido para a quantização vetorial em processamento de sinais e compressão de imagem. A ideia era representar um grande conjunto de pontos de dados por um conjunto menor de pontos, mantendo a estrutura essencial dos dados.

Funcionamento: A técnica de segmentação de imagem usando o algoritmo K-Means opera convertendo a imagem para um espaço de cores adequado e transformando cada pixel em um vetor tridimensional. Inicializa-se centroides para representar clusters, atribuindo pixels ao cluster mais próximo e recalculando os centroides iterativamente. O processo visa minimizar a variação intra-cluster. O K-Means destaca-se por sua simplicidade e eficiência, mas requer a escolha cuidadosa do número de clusters. Estratégias avançadas, como métodos de inicialização robustos, podem ser aplicadas para otimizar o desempenho. Apesar de suas vantagens, o K-Means tem limitações, e a combinação com outras técnicas pode ser explorada para maior flexibilidade e adaptação a diferentes contextos.

Conversão de BGR para RGB

- Definição da função RGB(img_bgr) para trocar os canais BGR pelos canais RGB.
- Aplicação da função à imagem original, resultando na imagem no formato RGB.

```
# FUNÇÃO PARA TRANSFORMAR DE BGR PARA RGB
def RGB(img_bgr):
    # Obtendo as dimensões da imagem
    l, c, p = img_bgr.shape

    # Criando uma matriz vazia para a imagem RGB
    img_rgb = np.zeros(shape=img_bgr.shape, dtype=np.uint8)

    # Iterando sobre cada pixel da imagem BGR
    for i in range(l):
        for j in range(c):
            # Obtendo os valores BGR do pixel
            b = img_bgr[i, j, 0]
            g = img_bgr[i, j, 1]
            r = img_bgr[i, j, 2]

            # Trocando os canais para obter a representação RGB
            img_rgb[i, j, 0] = r
            img_rgb[i, j, 1] = g
            img_rgb[i, j, 2] = b

    return img_rgb

# TRANSFORMANDO A IMAGEM BGR EM RGB
img = RGB(original_image)
```

Vetorização da Imagem

- Utilização da função reshape para converter a matriz da imagem em um vetor de pixels 3D.
- Cada linha representa um vetor RGB de um pixel.

Conversão de Tipo para float32

- Conversão do vetor de pixels para o tipo float32, requisito para o método K-Means.

K-Means Clustering

- Definição dos critérios de parada e número de clusters (K).
- Aplicação do algoritmo K-Means usando cv2.Kmeans.
- Obtenção dos centros dos clusters.

Reconstrução da Imagem Segmentada

- Atribuição dos rótulos obtidos do K-Means para regenerar a imagem segmentada.
- Uso dos centros dos clusters para determinar as cores dominantes.

```
# CONVERTENDO A IMAGEM EM UMA MATRIZ Px3 ONDE P=MxN E CADA LINHA AGORA É UM VETOR NO ESPAÇO 3D
vectorized = img.reshape((-1, 3))

# CONVERTENDO OS VALORES uint8 PARA float (REQUISITO DO MÉTODO K-MEANS)
vectorized = np.float32(vectorized)

# DEFININDO CRITÉRIOS, NÚMERO DE AGRUPAMENTOS(K) E APLICANDO K-MEANS
criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 10, 1.0)

# NÚMERO DE AGRUPAMENTO (NÚMEROS DE CORES QUE QUERO NA IMAGEM SEGMENTADA)
K = 6 # número aproximado de cores da imagem original

# NÚMERO DE VEZES QUE O ALGORITMO É EXECUTADO USANDO DIFERENTES CLASSIFICAÇÕES INICIAIS
# O ALGORITMO RETORNA OS RÓTULOS QUE PRODUZEM MELHOR COMPACTAÇÃO
attempts = 10

# ESPECIFICANDO COMO OS CENTROS INICIAIS SÃO OBTIDOS
ret, label, center = cv2.kmeans(vectorized, K, None, criteria, attempts, cv2.KMEANS_PP_CENTERS)

# CONVERTENDO DE VOLTA PARA uint8
center = np.uint8(center)

# ACESSANDO OS RÓTULOS PARA REGERAR A IMAGEM AGRUPADA
res = center[label.flatten()]
result_image = res.reshape((img.shape))
```

Apresentação Visual

- Uso da biblioteca matplotlib.pyplot para criar uma figura com três subplots.
- Exibição da imagem original, imagem no formato RGB e imagem segmentada.

```
# ACESSANDO OS RÓTULOS PARA REGERAR A IMAGEM AGRUPADA
res = center[label.flatten()]
result_image = res.reshape((img.shape))

# PLOTANDO A IMAGEM ORIGINAL E A IMAGEM RESULTANTE DO AGRUPAMENTO
figure_size = 15
plt.figure(figsize=(figure_size, figure_size))
plt.subplot(1, 3, 1), plt.imshow(original_image)
plt.title('Imagen original'), plt.xticks([]), plt.yticks([])
plt.subplot(1, 3, 2), plt.imshow(img)
plt.title('Imagen RGB'), plt.xticks([]), plt.yticks([])
plt.subplot(1, 3, 3), plt.imshow(result_image)
plt.title('Imagen segmentada com K = %i' % K), plt.xticks([]), plt.yticks([])
plt.show()
```

Pontos fortes da técnica por K-Means

Simplicidade e Facilidade de Implementação:

A simplicidade do algoritmo facilita a implementação e compreensão.

Eficiência Computacional:

O K-Means é computacionalmente eficiente e escalável para grandes conjuntos de dados.

Versatilidade em Diferentes Domínios:

Aplicável em diversas áreas, desde análise de dados até segmentação de imagens.

Interpretabilidade:

Resultados facilmente interpretáveis, já que os clusters são definidos pelos centroides.





Pontos fracos da técnica por K-Means

Sensibilidade à Inicialização:

Os resultados do K-Means podem variar dependendo da escolha inicial dos centroides.

Assunção de Forma Esférica dos Clusters:

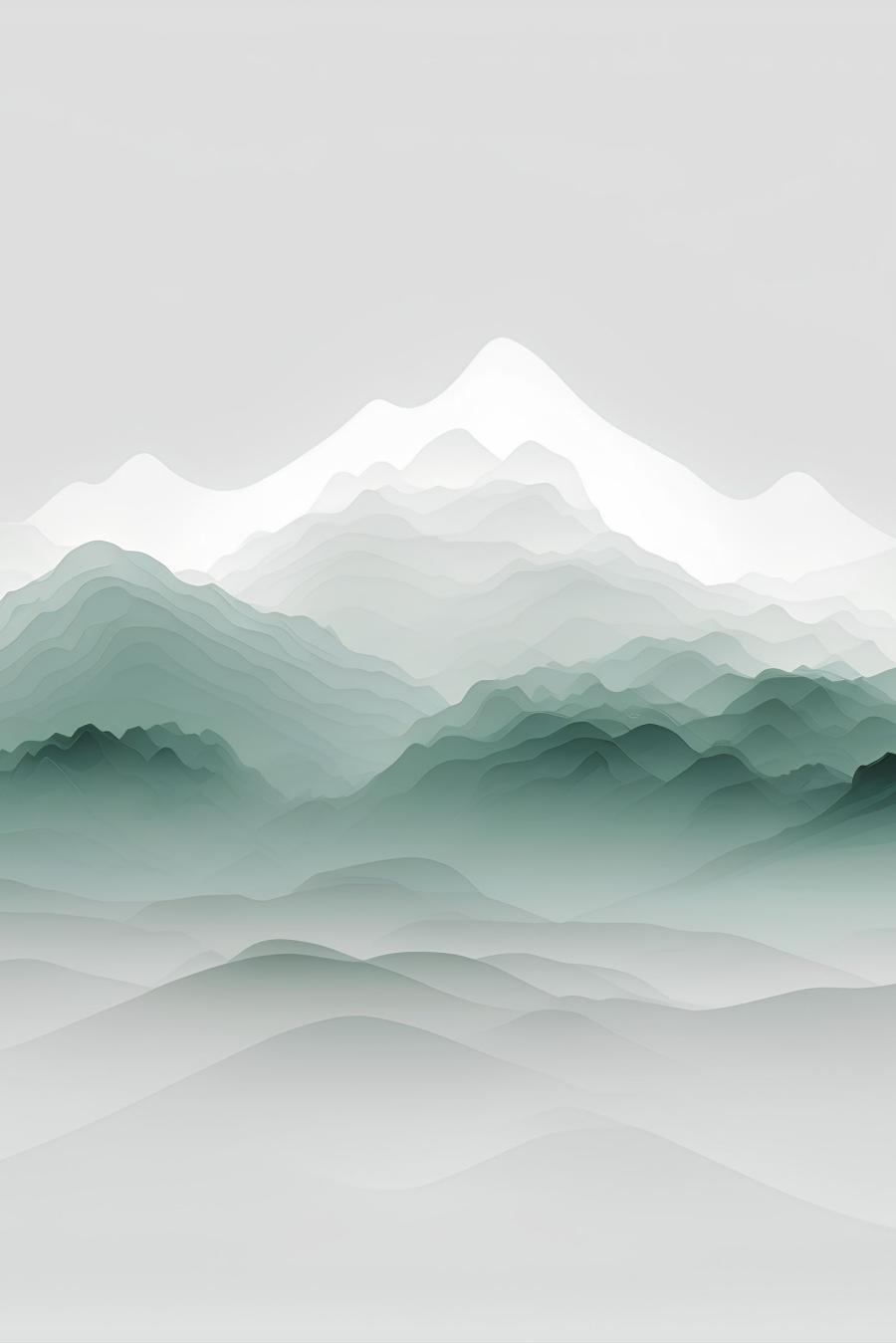
Assume que os clusters têm uma forma esférica e tamanhos aproximadamente iguais.

Sensibilidade a Outliers:

É sensível a outliers, pois podem distorcer a média dos pontos no cluster.

Número de Clusters Pré-definido:

O usuário precisa especificar o número de clusters (K) a priori.



Melhorando a técnica por K-Means

Para aprimorar a eficácia do K-Means, estratégias incluem melhorias na escolha inicial dos centroides (como o método K-Means++), automação da determinação do número de clusters (usando técnicas como o Elbow Method), combinação com métodos de redução de dimensionalidade (como o PCA), exploração de variantes do K-Means (como o Hierárquico) para maior flexibilidade, implementação de estratégias de tratamento de outliers e validação dos resultados através de métricas de avaliação de clusters. Essas abordagens combinadas visam otimizar a segmentação de imagem pelo K-Means em diversos contextos.

Viabilidade de outras técnicas em conjunto com o K-Means

A combinação do K-Means com outras técnicas pode ser viável dependendo da natureza dos dados e dos objetivos da análise. Por exemplo, técnicas de pré-processamento, como normalização de dados ou redução de dimensionalidade, podem melhorar a performance do K-Means. Experimentar diferentes abordagens e ajustar os parâmetros conforme necessário é uma prática comum para otimizar o desempenho da técnica em contextos específicos.



Conclusão

Em resumo, a segmentação de imagem, especialmente usando o K-Means, é fundamental para analisar informações visuais. Embora o K-Means seja eficiente, suas limitações destacam a importância de melhorias. Estratégias como inicialização robusta, automação na determinação de clusters, combinação com técnicas como redução de dimensionalidade, exploração de variantes e tratamento de outliers aprimoram sua eficácia. A validação sistemática dos resultados através de métricas oferece uma avaliação objetiva. Essas estratégias combinadas adaptam o K-Means a diversos contextos, otimizando sua aplicação na interpretação de conteúdo visual em várias áreas práticas. O contínuo aprimoramento dessas abordagens promete soluções mais sólidas na segmentação de imagem.