

## Segmentação de Imagens

## **1. Introdução**

O processamento de imagens desempenha um papel fundamental em diversas áreas, por exemplo: Reconhecimento de Padrões e Visão Computacional. Uma técnica importante nesse contexto é a Segmentação de Imagem. Este relatório explora essa técnica e suas aplicações. Este relatório trata de uma atividade consistente em sua aplicação.

## 2. Metodologia

A tarefa a ser realizada nesta atividade consiste em escolher duas imagens já utilizadas ao longo da disciplina realizar a seguinte tarefa:

1. Segmente as imagens em pelo menos 5 regiões homogêneas sob algum critério que você defina.

Para tal, foram escolhidas as imagens da Figura 1.



*Figure 1: Imagens utilizadas na atividade*

### 3. Processamento de Imagem

A segmentação de imagens refere-se ao processo de dividir uma imagem em regiões ou objetos significativos com base em certas características, como cor, intensidade, textura ou borda. O objetivo é simplificar a representação da imagem e facilitar a análise de regiões específicas.

Para realizar a segmentação, foi escolhida a técnica de K-Means, com 5 clusters conforme a metodologia. O algoritmo K-Means é frequentemente utilizado para realizar tarefas análogas, aproveitando seus princípios de agrupamento para identificar grupos de *pixels* semelhantes. Este documento explora a aplicação do algoritmo K-Means na segmentação de imagens.

Para realizar esta tarefa, foi utilizada a linguagem Python em conjunto com as seguintes bibliotecas: Numpy, PIL, OpenCV e Matplotlib.

## 4. Resultados

De posse das imagens da Figura 1, aplicamos a segmentação de imagem. Conforme a instrução metodológica, deveríamos escolher as regiões com base em um critério. A Figura 2 apresenta os representantes de cada região para ambas as imagens, respectivamente.

Na primeira imagem, os pontos representam os seguintes elementos: Mesa de madeira; mancha branca no gato; pelagem do gato; língua do gato; e pedaço da cortina transparecendo iluminação externa.

Na segunda imagem, os pontos representam os seguintes elementos: Porção do lençol com pouca iluminação; porção do lençol bem iluminado; pelagem do gato; fronteira entre a cama e a parede ao fundo; e parede ao fundo da imagem.

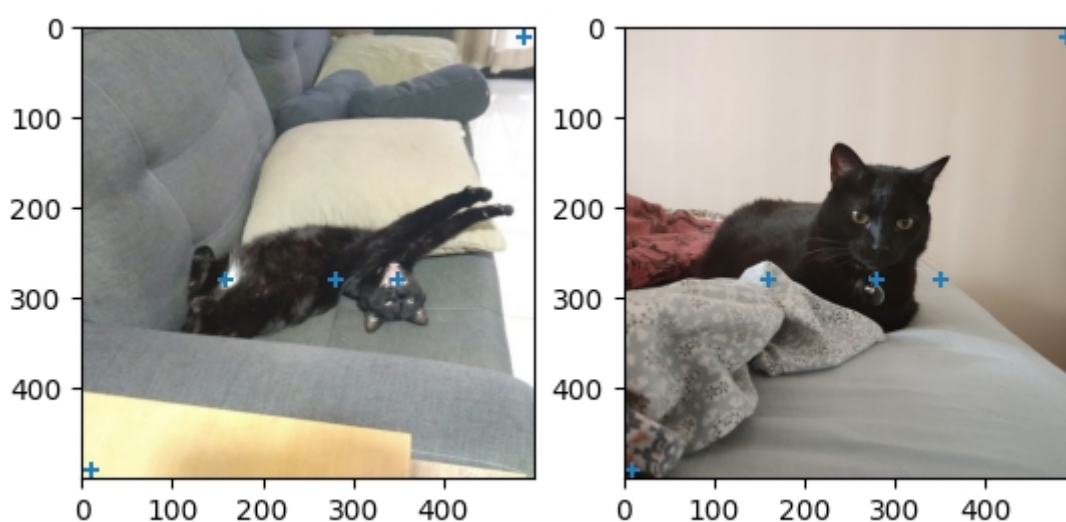


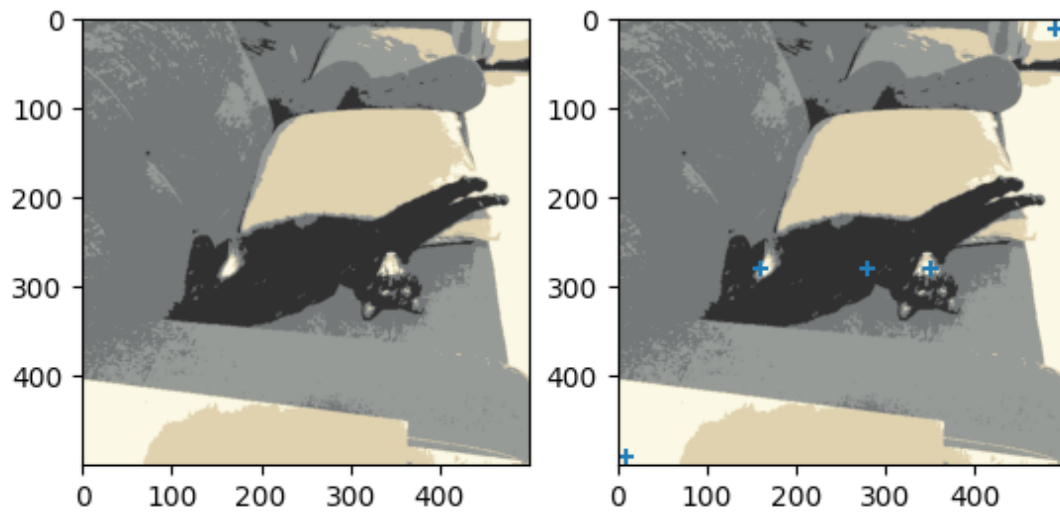
Figure 2: Pontos escolhidos como base para a segmentação marcados com “+”

Aplicando a segmentação através do algoritmo K-Means (com  $k = 5$ ), obtivemos os resultados expressos nas Figuras 3 e 4, para cada imagem.

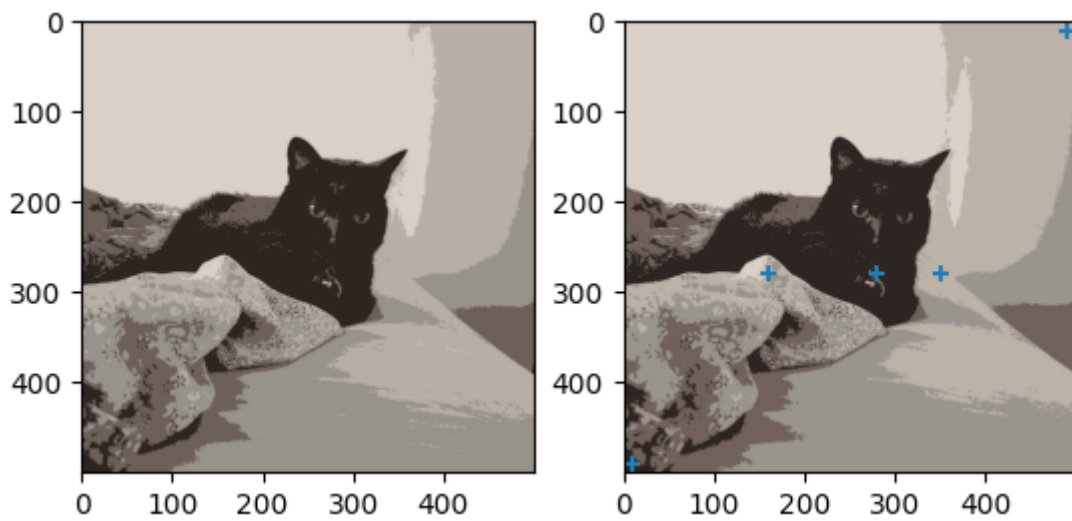
As Figura 3 e 4 nos mostram os resultados da segmentação quando selecionamos os centróides de maneira aleatória e de maneira determinada pelos pontos mencionados anteriormente, de maneira respectiva.

Em ambos os casos, para as duas imagens, não percebemos diferenças significativas frente aos dois casos de estudo: Quando escolhemos os representantes das regiões contra quando deixamos que o algoritmo escolha de maneira pseudo-aleatória.

Quanto a imagem final, a medida que reduzimos a paleta de cores imagem, percebemos um aumento do contraste bem como a perda significativa de detalhes. Percebemos também o surgimento de falsos contornos, que podem ser observados, por exemplo, na almofada na primeira imagem ou na parede ao fundo da segunda imagem.



*Figure 3: Resultados da Segmentação para a primeira imagem*



*Figure 4: Resultados da Segmentação para a segunda imagem*

## **5. Conclusões**

A segmentação de imagens é uma ferramenta essencial no processamento de imagens, permitindo a extração eficiente de informações relevantes em diversas aplicações.

### **Anexo**

Detalhes da implementação utilizando código em Python para processamento das imagens:

```
import cv2

import numpy as np
import seaborn as sns

from math import sqrt
from random import choice

from PIL import Image
from matplotlib import pyplot as plt

%matplotlib inline

def segmentar(img, n=5):
    shape_old = img.shape
    pixels = img.reshape((-1, 3))
    kmeans = KMeans(n_clusters=n, random_state=42)
    kmeans.fit(pixels)
    labels = kmeans.labels_
    centros = kmeans.cluster_centers_
    segmentada = centros[labels].reshape(shape_old)
    return segmentada
```