

Universidade de Brasília  
PPGI

## Relatório de Resultados de Paletas e Quantização de Imagem Colorida

26 de Outubro de 2023

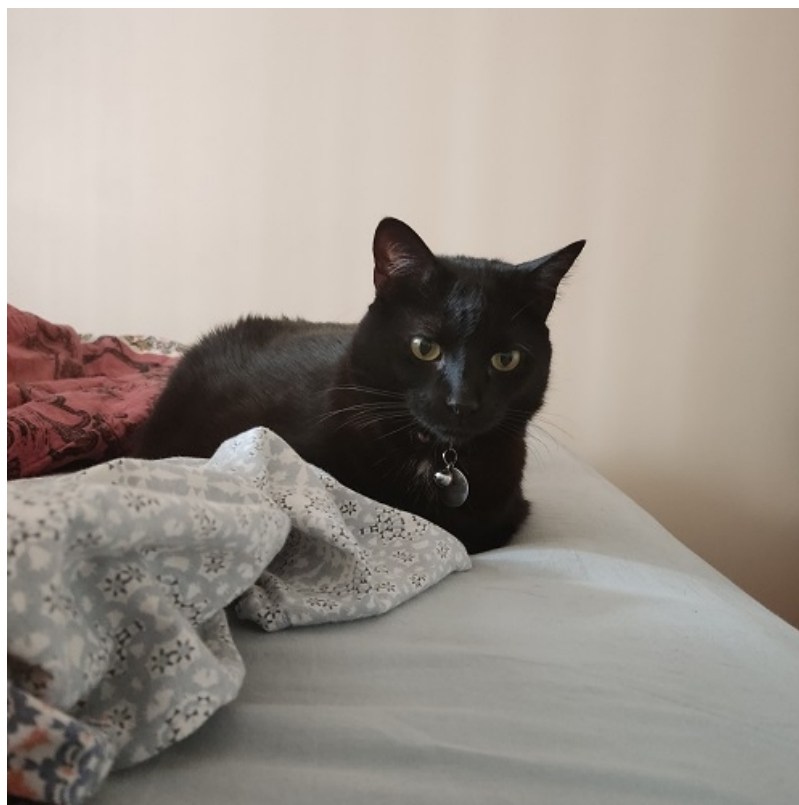
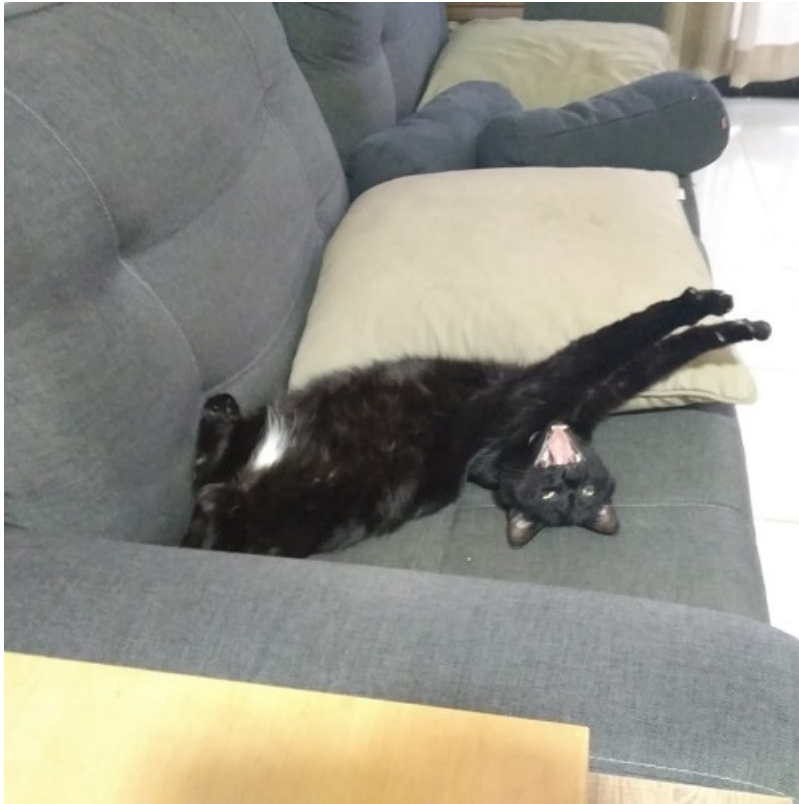
Disciplina: Processamento de Imagens  
Professor: Ricardo Lopes de Queiroz  
Aluno: Henrique Tibério Brandão Vieira Augusto (221101092)

## Metodologia

Escolher imagens para realizar os seguintes experimentos:

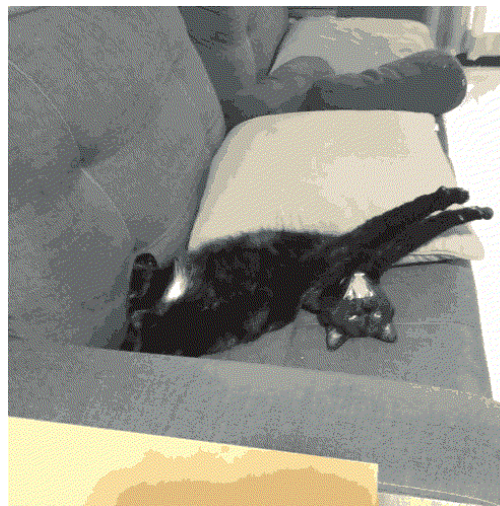
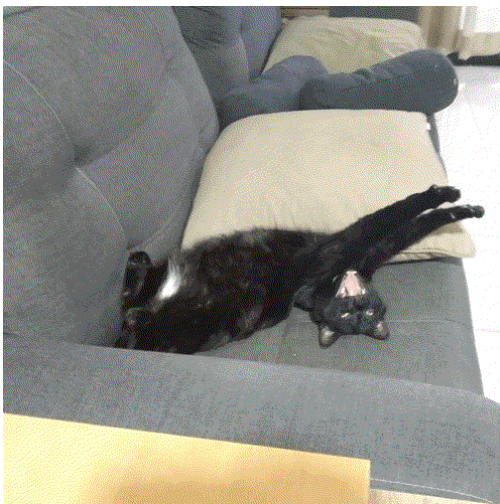
- Escolher um par de imagens coloridas (entre 500 e 700 pixels de largura ou altura);
- Representar cada imagem utilizando uma paleta de 256 cores e outra de 16 cores;
- Quantize as cores utilizando o algoritmo LGB para quantizar as 16M cores da imagem em 256 e 16 cores;
- Repita a representação com difusão de erro.

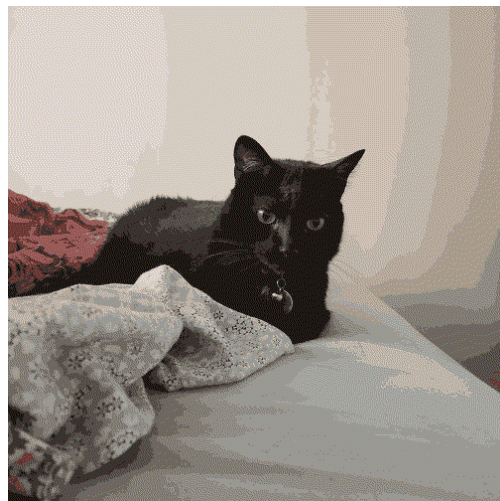
Foram utilizadas as seguintes imagens com altura e largura de 500 pixels:



## Resultados

Foram obtidas as imagens abaixo. Estão dispostas, lado a lado, a imagem quantizada e sua versão com difusão de erro, em ordem decrescente (256 e 16) de paleta de cores:





## **Discussão dos Resultados**

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. Ao efetuar o Dithering nas imagens, o contraste se torna mais suave e os falsos contornos se tornam menos marcantes.

## **Conclusão**

A quantização implementada via K-Means causa o surgimento de *patches* monocromáticos que - em virtude da implementação do algoritmo - causam uma coerência na escolha da cor.

A difusão de erro desempenhou um papel crucial na melhoria da qualidade visual das imagens quantizadas, principalmente por suavizar os falsos contornos e o contraste.

Com base nos resultados deste experimentos, podemos concluir que tanto a quantização quanto a difusão de erro têm um impacto significativo na qualidade visual das imagens. São técnicas são úteis para reduzir a quantidade de bits, mas é importante escolher a abordagem adequada para cada imagem.



## Anexo

Detalhes da implementação e resultados estão disponíveis no repositório <https://github.com/htbrandao/ppgi-proc-img/tree/master/trab02>.

- Código em Python para processamento das imagens:

```
import cv2
import numpy as np

from PIL import Image

def prepara_imagem(fp):
    img = cv2.imread(fp)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    pixels = img.reshape((-1,3))
    return img, np.float32(pixels)

def segmentacao_imagem(fp: str, paleta):
    img, pixels = prepara_imagem(fp)
    params = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 100, 0.85)
    retval, labels, centers = cv2.kmeans(pixels, paleta, None, params, 10,
cv2.KMEANS_RANDOM_CENTERS)
    centers = np.uint8(centers)
    segmented_data = centers[labels.flatten()]
    segmented_image = segmented_data.reshape((img.shape))
    return img, segmented_image

def dithering(pixels):
    return Image.fromarray(pixels).convert("P", dither=Image.Dither.FLOYDSTEINBERG)
```