

Trabalho 5

MC920 João Vitor Araki Gonçalves (176353)

Introdução

O objetivo desse trabalho é analisar a aplicação de técnicas de clustering de imagens, para realizar quantização do número de cores de uma imagem colorida, e qual o efeito o número de clusters utilizados na imagem final.

O Programa

O programa foi implementado em python 3.7.3 utilizando as seguintes bibliotecas:

- numpy
- opencv2
- matplotlib
- sklearn

Execução

Primeiramente instalar as dependencias do projeto:

```
pip install -r requirements.txt
```

O programa pode ser executado pela linha de comando:

```
python main.py [input_file] [number_of_clusters] [output_folder]
```

O programa irá utilizar o arquivo `input_file1` como entrada e utilizará o `number_of_clusters` como número de grupos para o kmeans.

A imagem resultante será salva na pasta `output_folder`. Também será exibido um gráfico com os centros dos clusters e um histograma das cores selecionadas no final da execução.

Processo

Passo 1: Leitura da imagem e normalização

A imagem é lida pelo método `cv2.imread(filename)` que lê a imagem original rgb, e então normalizamos a imagem no intervalo $[0-1]$



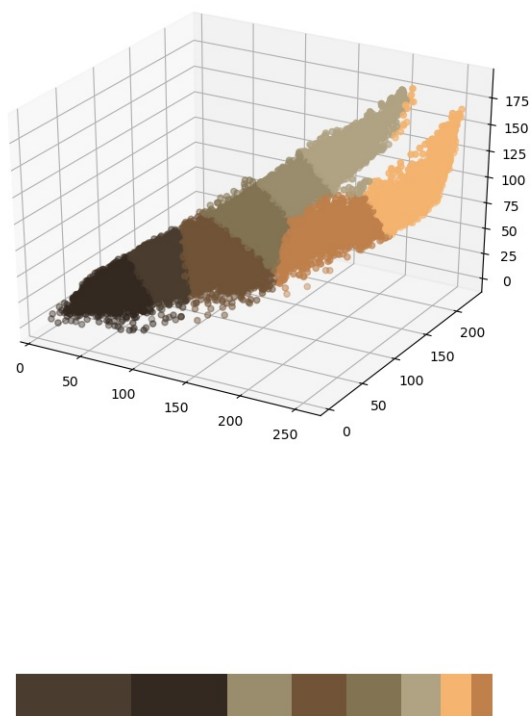
Passo 2: Kmeans

Utilizando o KMeans do sklearn, fazemos o fit do modelo com a imagem lida por meio do método: `KMeans(n_clusters, random_state=0).fit(normalized)`. Então com o dicionário de labels resultante, reconstruímos a imagem a partir desse número de cores reduzido.

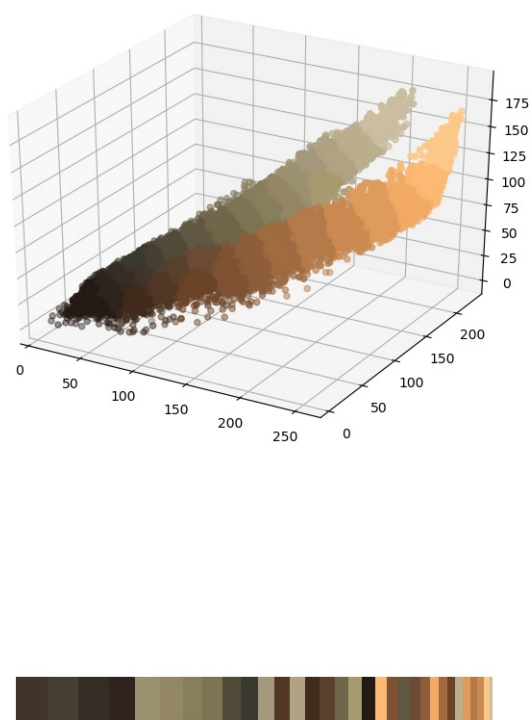
Passo 3: Clusters resultantes

Então com os labels e centros de clusters resultantes, plotamos um gráfico 3D com esses centros de clusters e as cores dos rótulos apropriadas.

Esse é o gráfico dos clusters e histograma de cores resultante da imagem `monalisa.png` com 8 clusters.



Esse é o gráfico dos clusters e histograma de cores resultante da imagem `monalisa.png` com 32 clusters.

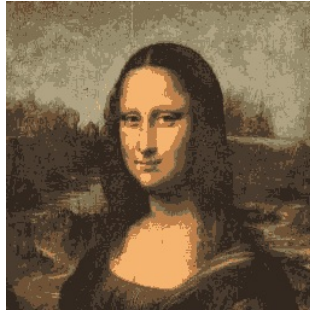


Resultados

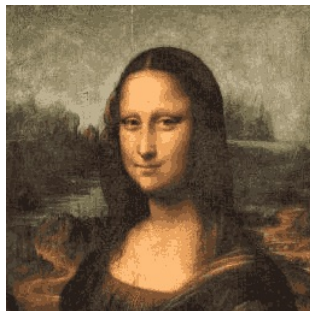
Imagem original



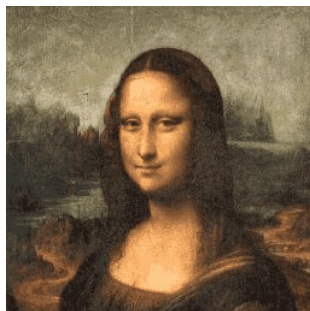
8 clusters (Tempo de execução: 2s)



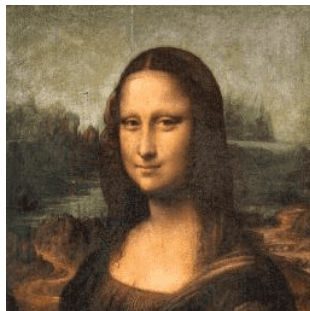
16 clusters (Tempo de execução: 4s)



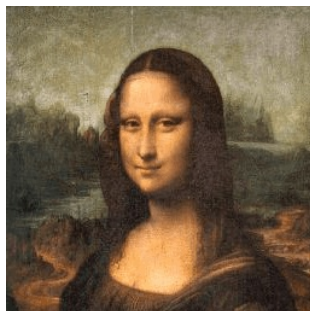
32 clusters (Tempo de execução: 6.6s)



64 clusters (Tempo de execução: 28s)



128 clusters (Tempo de execução: 45s)



Como é possível observar nas imagens, a fidelidade dos resultados de 8 e 16 clusters não é boa, é um número muito reduzido de cores para aproximar as variações da imagem original e é possível observar transições pouco naturais entre as diferentes cores.

Já a partir da imagem de 32 clusters, fica difícil notar as diferenças da imagem com cores reduzidas da original, com apenas pequenos detalhes. E os resultados de 64 e 128 cores há praticamente nenhuma diferença entre os resultados e a imagem original.

Já em relação de tempo de execução, notou-se uma demora cresce para obtenção dos resultados em relação ao número de clusters. A execução para 128 clusters tomou um tempo consideravelmente maior que o de 32 (7.5 vezes) para um resultado apenas marginalmente melhor. E comparando o de 64 e de 128, o tempo de execução praticamente dobrou para um resultado sem diferenças aparentes. Em testes com outras imagens, observou-se um resultado similar, resultados bons com 32 clusters e bem próximos do original com números mais elevados.