

Trabalho MC920 5: Redução de Cores

Esdras R. Carmo - RA 170656

15 de junho de 2019

1 Introdução

Exploraremos neste trabalho a técnica de agrupamento através de KMeans para reduzir a quantidade de cores de uma fotografia.

Para isso, primeiramente iremos rodar o algoritmo de aprendizado de máquina não supervisionado para encontrar as n principais cores que representam a imagem. Feito isso, será reduzido as demais cores para a cor central de cada cluster.

2 Especificação do Problema

O algoritmo receberá uma imagem colorida de entrada, assumindo colorspace RGB. Logo após será linearizado os pixels da imagem, criando uma tabela de features que será utilizado na clusterização, sendo que essa tabela terá $w * h$ linhas e 3 colunas (features), cada coluna representando um dos canais da imagem (R, G e B).

2.1 Clusterização

O algoritmo de clusterização utilizado foi o KMeans. Ele é responsável por dividir o conjunto de pixels coloridos da imagem em k grupos de modo a minimizar a variância de cada grupo. Formalmente, o objetivo é encontrar:

$$\arg \min_S \sum_{i=1}^k |S_i| \text{Var } S_i$$

Sendo $S = \{S_1, S_2, \dots, S_k\}$ o conjunto de k clusters.

Desta forma, para cada cluster S_i temos um conjunto de pixels pertencentes ao grupo e o centróide C_i que representa a cor que melhor descreve o cluster.

2.2 Reconstrução

Após a etapa de clusterização é realizada a reconstrução da imagem com base nos centróides calculados na etapa anterior.

Para isso, é criado uma nova imagem com as mesmas dimensões da imagem de entrada, sendo que cada pixel da nova imagem receberá os valores RGB do centróide do cluster ao qual o respectivo pixel da imagem de entrada pertence.

Seja $h : \mathbb{N}^2 \rightarrow \mathbb{N}$ a função que mapeia cada pixel da imagem de entrada f para o índice i do cluster em que está contido. Seja $\phi : \mathbb{N} \rightarrow \mathbb{R}^3$ a função que retorna o centróide do cluster de índice i em RGB. Portanto, a imagem de saída $g : \mathbb{N}^2 \rightarrow \mathbb{R}^3$ é definida para todo par (x, y) pertencente ao domínio de f :

$$g(x, y) := \phi \circ h(x, y)$$

2.3 Otimização

O processo de clusterização por KMeans pode ficar muito caro dependendo das dimensões da imagem de entrada e da quantidade de cores / clusters desejadas.

A fim de encontrar uma forma mais eficiente de clusterizar utilizamos Mini Batches KMeans, que consiste em dividir o conjunto de pixels em batches menores e realizar a clusterização por batch.

Utilizando desse algoritmo, em uma imagem de dimensões 1024×768 e redução para 32 cores conseguimos uma redução no tempo de execução da clusterização de 60 s para 0.65 s, dividindo em batches de tamanho 2^{12} . Isso representa um speedup de aproximadamente 92 vezes.

3 Entrada de Dados

O script foi escrito utilizando *Python* 3.7, com as bibliotecas *scikit-image*, *scikit-learn* e *NumPy*.

Para a execução do programa basta utilizar o seguinte comando:

```
python3 reduce_colors.py input_path output_path n_colors
```

Sendo os parâmetros:

- *reduce_colors.py*: Nome do script;
- *input_path*: Caminho da imagem de entrada;
- *output_path*: Caminho da imagem de saída;
- *n_colors*: Número de cores para o qual o script deverá reduzir a imagem de entrada

4 Resultado e Discussão

Utilizando a técnica de agrupamento, percebemos que imagens que possuem cores bem definidas como na Figura 1 podem ser reduzidas a uma quantidade bem pequena de cores sem perder o aspecto geral. Apenas algumas perdas na região dos olhos foi notada nesse caso.

Já para imagens que possuem gradiente e uma gama maior de cores diferenciando o background do foreground, como é o caso da Figura 2, a redução de cores acaba prejudicando o aspecto da imagem. Embora os objetos da imagem ainda sejam bem reconhecidos com 8 cores, o gradiente de transição entre foreground e background é quantizado de modo a apresentar bastante perda.

Outra imagem que foi prejudicada na quantização com poucas cores foi o retrato de Monalisa da Figura 3. Notamos que para 8 cores os gradientes apresentaram o aspecto quantizado, enquanto que para 64 cores ou mais esse efeito já desapareceu.

5 Conclusão

Com os resultados apresentados, concluímos que conseguimos uma forma eficiente de reduzir o número de cores de uma imagem aplicando aprendizado de máquina não supervisionado.

Essa técnica pode ser utilizada em métodos de compressão, sendo que na maioria dos casos conseguimos manter a qualidade geral da imagem representando-a em apenas 256 cores ou até menos. Portanto, pode-se criar um formato de imagem que armazene um dicionário com as k cores escolhidas pela clusterização e os pixels da imagem com o índice referenciando esse dicionário de cores.

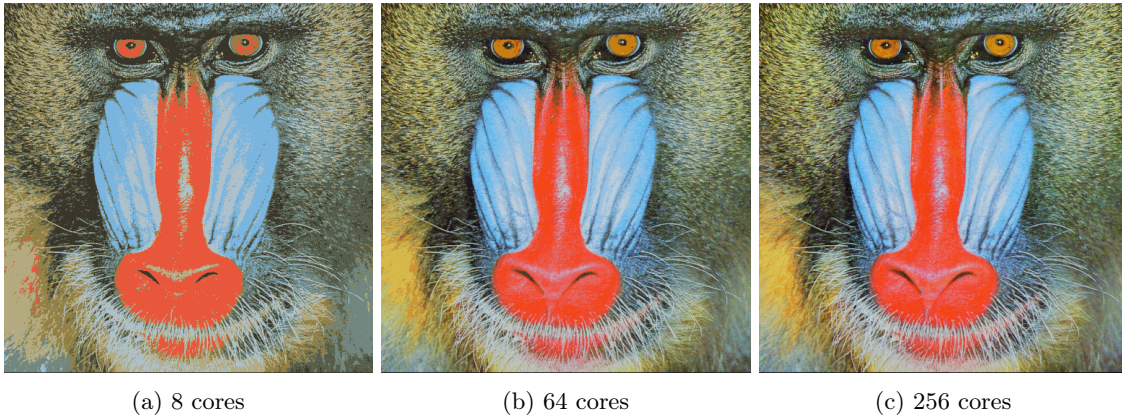


Figura 1: Imagem bem representada em poucas cores

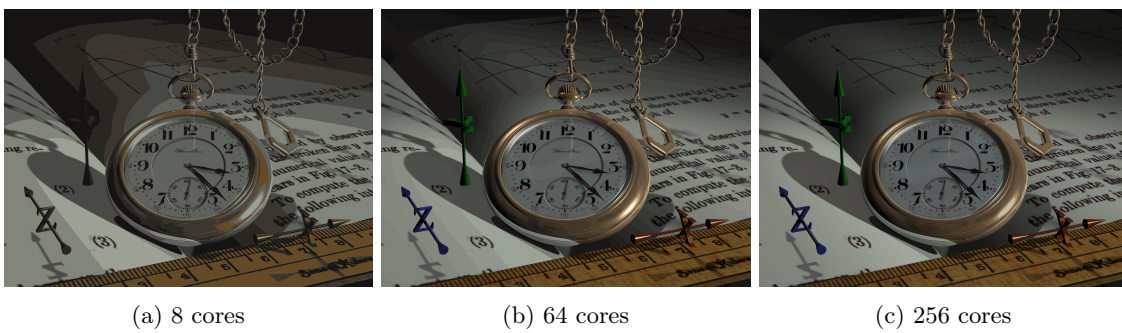


Figura 2: Imagem com perda nas regiões de gradiente

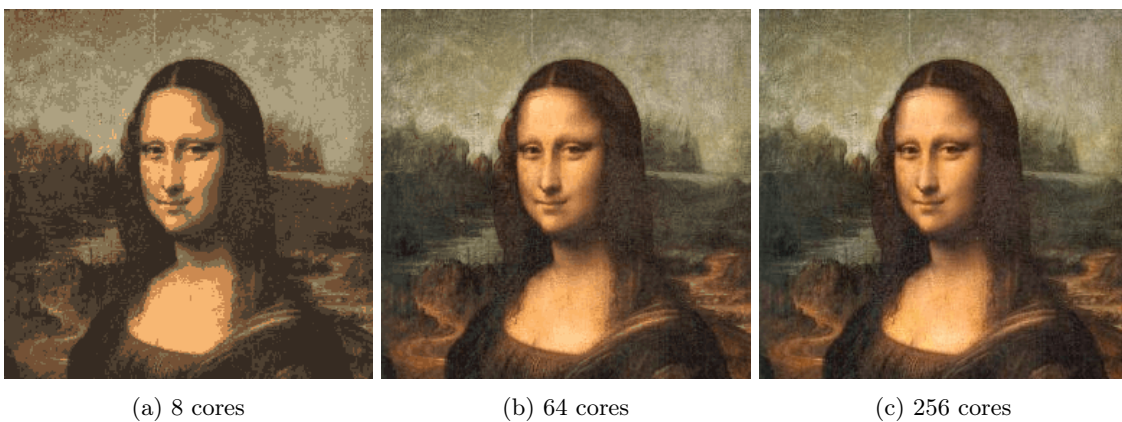


Figura 3: Imagem de retrato com perda em poucas cores