

# Quantização de cores

Nathália Harumi Kuromiya  
Engenharia da Computação - Graduação  
E-mail: n175188@dac.unicamp.br  
RA 175188

24 de junho de 2019

## 1 Introdução

Esse trabalho tem o objetivo de mostrar como a quantização de cores reflete no resultado da imagem. Para isso, vamos comparar diferentes quantidades de cores na imagem, mas sempre utilizando o mesmo método de quantização. Para a elaboração desse trabalho, foi necessário um estudo maior sobre registros, incluindo detectores e descritores. Para esse fim, utilizou-se o livro[1] e os slides de técnicas de agrupamento [2], disponibilizados pelo professor *Hélio Pedrini*. Além dessas fontes, foi utilizado tutoriais do OpenCV ([3] e [4]) para a aplicação de métodos utilizando essa biblioteca.

## 2 Trabalho Proposto [5]

O trabalho proposto se resume em: uma imagem de entrada no formato PNG e a quantidade de cores para qual a imagem deve ser quantizada, o algoritmo faz o agrupamento das partes da imagem e retorna a mesma imagem, porém com as cores quantizadas para a quantidade escolhida na entrada. Essa imagem também está no formato PNG.

### 3 Programas e bibliotecas utilizadas

A versão de Python[6] utilizada para implementação foi 3.5.4 e as bibliotecas de auxílio foram NumPy[7] 1.11.2 e OpenCV[8] 4.1.0-dev.

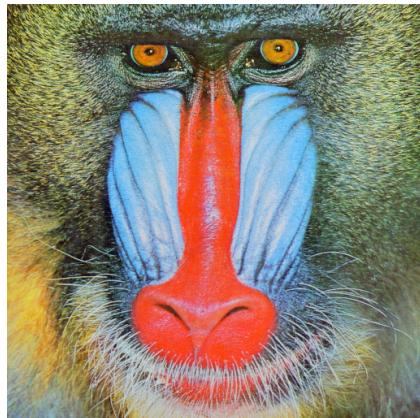
### 4 Entradas e Saídas

A imagem de entrada e as de saída possuem o formato PNG (*Portable Network Graphics*).

### 5 Implementação e Resultados

#### 5.1 Imagem-referência

Para comparação das técnicas utilizadas, usaremos a Figura 1 como base para todos os experimentos.



(a) Baboon



(b) Watch

Figura 1: Figuras originais que servirão de base para comparações.

#### 5.2 Metodologia

Para fazer o agrupamento dos pontos da imagem de acordo com a semelhança, foi utilizada a função `cv2.kmeans()`. Isso quer dizer que todo o agrupamento da imagem foi feito através da utilização do método conhecido como

“*K-Means*”. Esse método consiste em inicializar  $k$  centróides na imagem aleatoriamente, em que  $k$  é o número de grupos em que a imagem será separada. Nesse caso de quantificação, o número de grupos representa quantas cores haverão na imagem resultante. Então, para cada ponto da imagem, é escolhido a que grupo esse ponto pertence e atualizado as características do grupo. Ao final, os centróides são movidos para o centro do grupo relacionado a tal centróide. Esse processo é feito até atingir um  $\epsilon < 1.0$  ou atingir 10 iterações, de acordo com a configuração adotada nesse problema. Com esse agrupamento, a imagem então é reconstruída de acordo com os grupos encontrados, limitando a cor de cada grupo.

## 6 Resultados

Para as imagens-referências 1, foram aplicados  $k$  de valores iguais a [2, 4, 8, 16, 32, 64, 128]. Os resultados podem ser vistos a seguir nas figuras 2, 3, 4, 5, 6, 7 e 8.



(a) Baboon

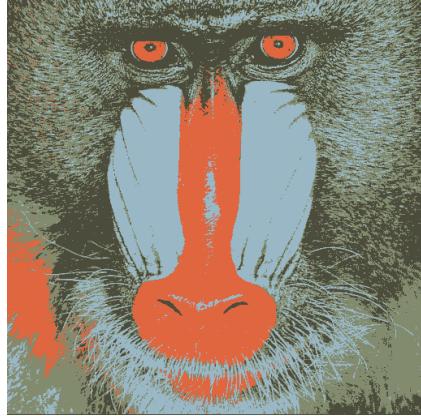


(b) Watch

Figura 2: Figuras com 2 cores.

## 7 Conclusão

A implementação para criar quantizar as cores das imagens se mostrou eficiente, dado que conseguimos resultados claros em todos os testes.

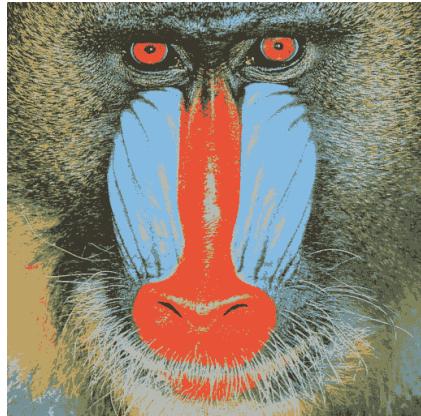


(a) Baboon



(b) Watch

Figura 3: Figuras com 4 cores.



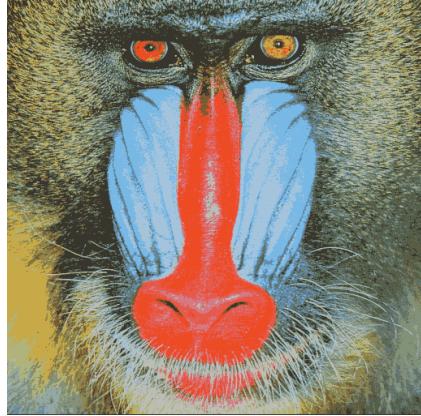
(a) Baboon



(b) Watch

Figura 4: Figuras com 8 cores.

Já em relação a quantização, percebemos a importância das cores para diferenciar certas informações da imagem. A redução da quantidade de cores traz a simplicidade para a apresentação da imagem, mas mostra uma clara perda de detalhes. Como exemplo, temos a imagem 2b, em que não é possível distinguir com clareza todas as setas da imagem, bem como a régua e detalhes do livro. Já na imagem 3, com o aumento de 2 cores apenas, já conseguimos um resultado significativamente melhor. A ideia é sempre balancear a quantidade de detalhes que é admissível a perda. Em caso de

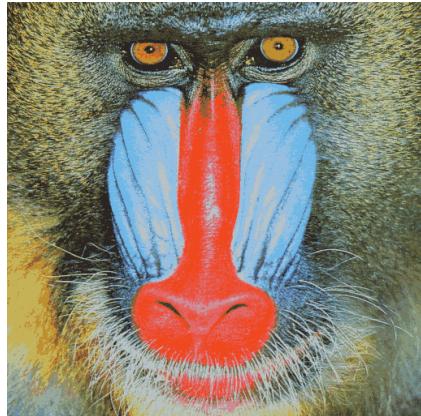


(a) Baboon



(b) Watch

Figura 5: Figuras com 16 cores.



(a) Baboon



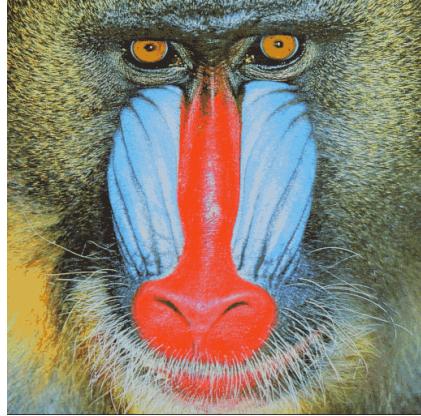
(b) Watch

Figura 6: Figuras com 32 cores.

transmissões televisivas, por exemplo, há uma certa tolerância a perda de detalhes, onde caberia um método de quantização para otimizar a manipulação das imagens. Já no meio médico, não é possível admitir perda de detalhes, então uma quantização não é viável.

Com toda essa análise, conseguimos dizer que o trabalho conseguiu mostrar tanto a utilização da quantização quanto seus resultados esperados com as diferentes imagens e suas configurações (em relação a quantidade de cores).

+ — - +

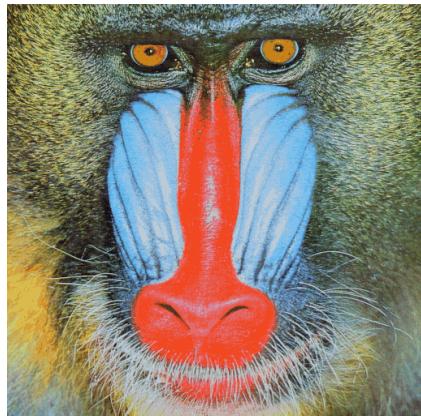


(a) Baboon



(b) Watch

Figura 7: Figuras com 64 cores.



(a) Baboon



(b) Watch

Figura 8: Figuras com 128 cores.

## Referências

- [1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed. Pearson, 2017. 1
- [2] H. Pedrini, “Aula: Técnicas de agrupamento.” [Online]. Available: [http://www.ic.unicamp.br/~helio/disciplinas/MC920/aula\\_agrupamentos.pdf](http://www.ic.unicamp.br/~helio/disciplinas/MC920/aula_agrupamentos.pdf) 1

- [3] “Understanding k-means clustering.” [Online]. Available: [https://docs.opencv.org/4.1.0/de/d4d/tutorial\\_py\\_kmeans\\_understanding.html](https://docs.opencv.org/4.1.0/de/d4d/tutorial_py_kmeans_understanding.html)
- [4] “K-means clustering in opencv.” [Online]. Available: [https://docs.opencv.org/4.1.0/d1/d5c/tutorial\\_py\\_kmeans\\_opencv.html](https://docs.opencv.org/4.1.0/d1/d5c/tutorial_py_kmeans_opencv.html)
- [5] H. Pedrini, *trabalho 5*. [Online]. Available: <http://www.ic.unicamp.br/~helio/disciplinas/MC920/trabalho5.pdf>
- [6] “Python 3.” [Online]. Available: <https://www.python.org/>
- [7] “Numpy.” [Online]. Available: <https://www.numpy.org/>
- [8] “Opencv.” [Online]. Available: <https://opencv.org/>