**Pseudo code**

Here is pseudo code of the general k-means clustering. (partially presented in LaTeX)

|  |
| --- |
| **Algorithm** The general algorithm of k-means clustering |
| 1. **Import:** packages |
| 1. **Input:** image 2. Acquire the numbers of three dimensions 3. **imgRGB<-**list of all pixels with columns of x, y, R, G, B data 4. Plot of original image 5. **K<-**the number of k clustering 6. Generate **k** means of **imgRGB** and classify them in k indexes 7. Give them colors based on indexes. 8. Plot of compressed image 9. **Output:** Draw original plot and compressed plot in a window |
| **End of the algorithm** |

**Experiments**

**I try to organize the experiments with ideas. Please feel free to adapt or partially adopt in your parts.**

I would try to analyze this k-means clustering compression tool from two aspects, efficiency of compressing and fidelity.

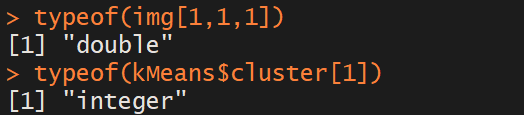
**Efficiency:** The proportion of compressed image takes compared with the original image. A double takes place 64 bits in R while an integer takes place 32 bits. Assume that the image is imported with resolution of $n\times m$. During compressing, it is imported with $n\times m\times RGB$doubles while will be compressed by k-means clustering as $n\times m$ integers of classified indexes and k means of color data with $k\times RGB$ doubles. Then the efficiency of this compression tool is given by:

$\begin{align}

& e=\frac{n\times m\text{ integers indexes(32 bits)}}{n\times m\times RGB\text{ doubles(64 bits)}}+\frac{k\times RGB\text{ doubles(64 bits)}}{n\times m\times RGB\text{ doubles(64 bits)}} \\

& \text{ }=16.667%+\frac{k}{n\times m}\times 100% \\

\end{align}$.

(complement: Although pdf says the images stored with integers RGB, I did an experiment that it is stored in doubles in R. )

**Fidelity:** How much the compressed image looks like the original one. We have a educated guess that the fidelity is affected by diversity of colors in the original image.

**Experiments of efficiency**

|  |
| --- |
| **Experiment 1:** the basic experiment with k=16 and $e=16.673%$.  **Experiment result:** (you can make some conclusion and adjust the picture with the corresponding figure in the directory. Here is just showing some ideas of mine.) |

|  |
| --- |
| **Experiment 2:** smaller k=8 and $e=16.670%$.  **Experiment result:** The efficiency does not improve(decrease) significantly while it looks more abstract and the fidelity get worse compared to experiment 1. (you can make some conclusion and adjust the picture with the corresponding figure in the directory. Here is just showing some ideas of mine.) |

|  |
| --- |
| **Experiment 3:** bigger k=88 and $e=16.700%$.  **Experiment result:** The efficiency does not increase significantly. However, we already cannot tell the difference between the original and the compressed. |

(We may have that adopt a bigger k is better.)

**Experiments of fidelity**

|  |
| --- |
| **Experiment 4:** Using a simpler picture with k=16.  **Experiment result:** The fidelity loss seems to be improved compared with experiment 1 with the same k=16. We think may be because the image is dominated by several main colors, with less color diversity, thus has higher fidelity after compressed. However, some colors are twisted, i.e.: red become green. |

|  |
| --- |
| **Experiment 5:** Reduce the resolution of original image to $\left( {n}/{2}\; \right)\times \left( {m}/{2}\; \right)$.  **Experiment result:** It seems that fidelity difference is not affected by original resolution of a same picture. |

|  |
| --- |
| **Experiment 6:** Cut the right.  **Experiment result:** We can notice some fidelity improvement when focus on the chair. But it is not significant as cutting the right side does not result in reduce the color diversity significantly. |