# **K - Means Documentation & Log**

* **Algorithm Used:** K-Means
* **Picture Used:** [Tiger Picture](https://www.google.com/imgres?q=tiger%20hd%20wallpaper&imgurl=https%3A%2F%2Fimages7.alphacoders.com%2F710%2F710428.jpg&imgrefurl=https%3A%2F%2Fwall.alphacoders.com%2Fbig.php%3Fi%3D710428&docid=V0qEshNZibVIFM&tbnid=2IOchmbGJq4FbM&vet=12ahUKEwjAr96K8dCLAxUzXUEAHe0CDL4QM3oECH4QAA..i&w=1920&h=1440&hcb=2&ved=2ahUKEwjAr96K8dCLAxUzXUEAHe0CDL4QM3oECH4QAA)
* **Framework:** CRISP-DM
* **Original Notebook**: [Notebook](https://github.com/Adioosin/image-compression-using-k-mean/blob/master/Image%20compression%20using%20K-Mean%20algorithm.ipynb)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Phases** | **Changes Made** | **Reason for the change** | **Duration** | **Difficulty level**  **(1-10)** |
| **Image selection** | Replaced the tiger image with a different image of the tiger. | Instead of high-quality image, I selected a normal quality image to explore the capabilities of K- Means | 20 min | 4 |
| **Algorithm Version** | Changed from K-Means to K-Means++ | To strengthen the initial centroid selection and for better convergence | 20 min | 6 |
| **Further Enhancements** | run\_kmeans | Changed run\_kmeans function to use k-means++ | 5 hours | 9 |
| Elbow Method | To find the best optimal k value I used the elbow method and didn’t take the entire pixels and only took some part of pixels in the image to reduce the internal calculation complexity |
| Downsampling | I haven’t got the desired result, I tried downsampling to reduce the original image resolution a bit. |
| K-Means color quantization | To reduce the colour variability, I used K-Means Color quantization |
| Gaussian Blur | I applied Gaussian Blur to smooth the harsh edges, which is beneficial while compressing the image. |
| **Conclusions** |  | Through this I learned that not only the image, but also the type of the image is crucial to reduce the size of the image. I have explored a lot of image compression techniques, which gave me broader knowledge about images and techniques in k-means to compress the image without compromising the quality.  Finally, the model performed very well compared to the model in the original notebook. | 40 min | 6 |

## **1. Business Understanding**

* **Objective**:  
  Using K-Means clustering, for compressing a tiger image (from 48.77 KB initial file size), reducing its file size while maintaining good visual quality.
* **Reason for Choosing K-Means**:
  + K-Means clustering **decreases color complexity** by providing clusters of similar colors.
  + This method can yield a **reduced range of color** so that it can be easier to compress.

## **2. Data Understanding**

* **Data Source**:
  + The input image is a **JPEG** file (**tiger.jpg**) with dimensions fit for demonstration (not overly large).
  + JPEG is already a **lossy** format, therefore repeatedly saving as JPEG sometimes increase file size unless I am careful in manage parameters (quality, resolution, etc.).
* **Initial Observations**:
  + The original file size is **48.77 KB**.
  + The image has **continuous transitions of colours** and details making it hard for color quantization to compress them properly without seeing visual artifacts.

## **3. Data Preparation**

1. **Read and Normalize Image**
   * I read the image using **skimage.io.imread** and converted pixel values from **[0, 255]** to **[0, 1].**
   * This is done to make sure all operations that happens afterwards (distance calculations in K-Means) goes well.
2. **Reshaping**
   * I converted the 3D image array **(height, width, 3)** into a 2D array **(height\*width, 3)** to treat each pixel as a data point in the K-Means algorithm.

## **4. Modeling**

### **4.1 Initial K-Means Approach**

* **Original Random Initialization**:
  + Implemented **K-Means++** initialization in place of the default random centroid selection to enhance convergence and get better cluster centers.
* **Elbow Method (Sampling)**:
  + Proposed a **sampling approach** for the elbow method (use of only a portion of the pixels) to find a good range of **K** values without running K-Means multiple times on the whole image.
  + This generated **less** **runtime** with still a **good estimation** for the appropriate number of clusters.
* **Choosing K**:
  + Based on the elbow plot, I selected **K=8** as a compromise to preserve color fidelity while allowing for potential compression.

### **4.2 An Enhanced Method for More Compression**

Having confirmed K=8 using the elbow method, I **expanded** upon the compression method with the following:

1. **Downsampling (75%)**
   * I resized the image to 75% of the original dimensions, thus reducing the total number of pixels.
   * This basically resizes a larger resolution down to a smaller one **compressing file size** while not compromising too much detail, particularly if the initial resolution was high.
2. **K-Means Color Quantization**
   * I then applied **K-Means** (with K=8) on the **downsampled** data to **limit the color palette even more**.
3. **Gaussian Blur**
   * A gentle blur will smooth out the **sharp edges** created from K-Means, making the image more flexible to **JPEG compression**.
4. **JPEG Quality (60)**
   * Finally, I saved the image with a **decent JPEG quality** of 60. By this **visual clarity** and **file size can be** balanced.

## **5. Evaluation**

* **Visual Inspection**:
  1. The final compressed image preserves the appearance and color balance of the tiger.
  2. Some **banding** or minor artifacts may be visible upon close inspection, but overall fidelity is good.
* **File Size Comparison**:
  1. **Original**: 48.77 KB
  2. **Compressed**: 19.38 KB
  3. I achieved a **significant reduction** in size (more than 50% smaller) while maintaining **good** image clarity.
* **Analysis of Changes**:
  1. **Downsampling** reduced the resolution, with each pixel cluster represent a larger area.
  2. The color space was simplified by using **K=8** color clusters.
  3. **Gaussian Blur** diminished sharp edges and helped JPEG compression.
  4. **JPEG Quality** at 60 bypassed excessive artifacts while reducing the file size.

## **6. Overview of Changes & Practical Impact**

1. **K-Means++ Initialization**
   * **From**: Random centroid selection.
   * **To**: K-Means++ for improved placement of initial cluster.
   * **Impact**: Faster convergence and usually reduced final distortion in the compressed image.
2. **Elbow Method with Sampling**
   * **Added**: A sampling strategy for fast identification of good range of K.
   * **Impact**: Significant decrease in computation time without losing out much accuracy in finding the best K value.
3. **Downsampling & Blur**
   * **From**: Full-resolution K-Means.
   * **To**: 75% resolution + a light Gaussian blur.
   * **Effect**: Fewer pixel count but smoother transitions thus creating a more compressible image.
4. **JPEG Quality**
   * **Chosen**: Quality=60.
   * **Impact**: Finally, achieved a **size of 19.38 KB** (Lowered from 48.77 KB) while retaining decent level of clarity.