Problem 6

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from skimage import io
from sklearn.metrics import silhouette_score

image = io.imread('/content/drive/MyDrive/sem 7/ID5055/Assignment
3/Problem 6/frog.jpg')
pixels = np.array(image)
print(pixels.shape)
```

```
(392, 562, 3)
```

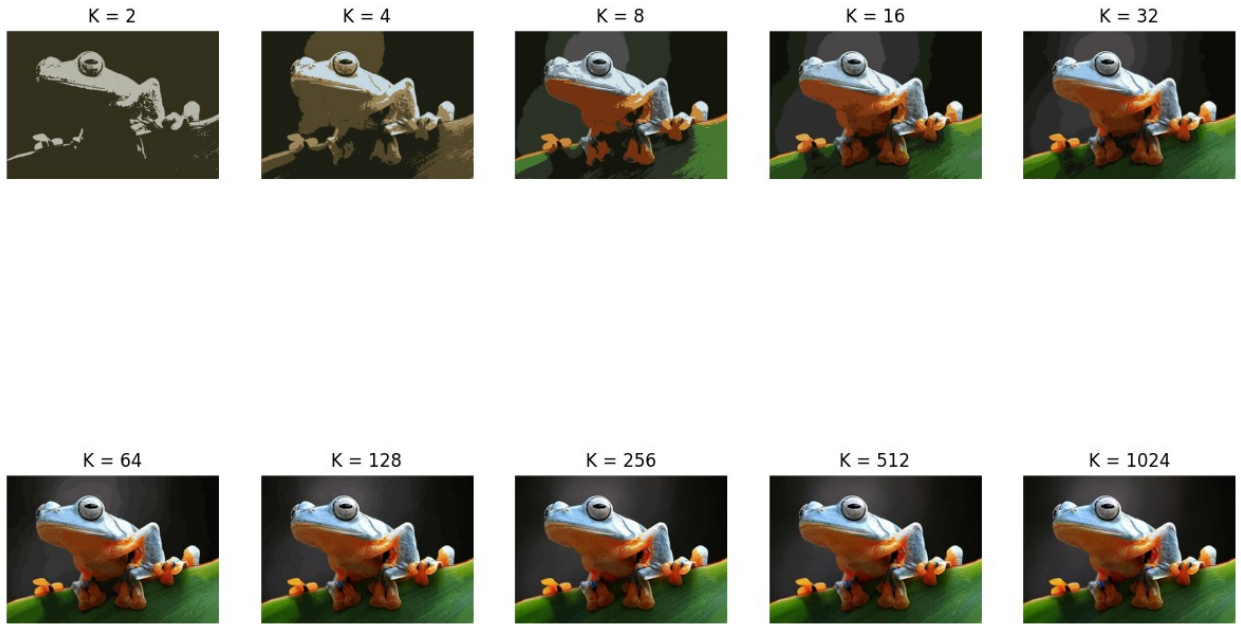
Converting the pixels super matrix into 2D matrix that represent all pixels.

```
pixels = pixels.reshape(-1, 3)
```

Visualizing the image after compression

```
k_values = [2, 4, 8, 16, 32, 64, 128, 256, 512, 1024]
ss_dist_elbow_check = []
# ss_dist_silhouette_score = []

plt.figure(figsize=(15, 10))

for i, k in enumerate(k_values):

    kmeans = KMeans(n_clusters = k, random_state = 0, init = 'k-means+
+', n_init = 1, max_iter = 30)
    clustered_pixels = kmeans.fit_predict(pixels)
    ss_dist_elbow_check.append(kmeans.inertia_)

    # score = silhouette_score(pixels, clustered_pixels,
metric='euclidean')
    # ss_dist_silhouette_score.append(score)

    compressed_pixels =
kmeans.cluster_centers_[clustered_pixels].astype(int)

    compressed_image = compressed_pixels.reshape(image.shape)

    plt.subplot(2, 5, i + 1)
    plt.title(f'K = {k}')
    plt.imshow(compressed_image)
    plt.axis('off')
```

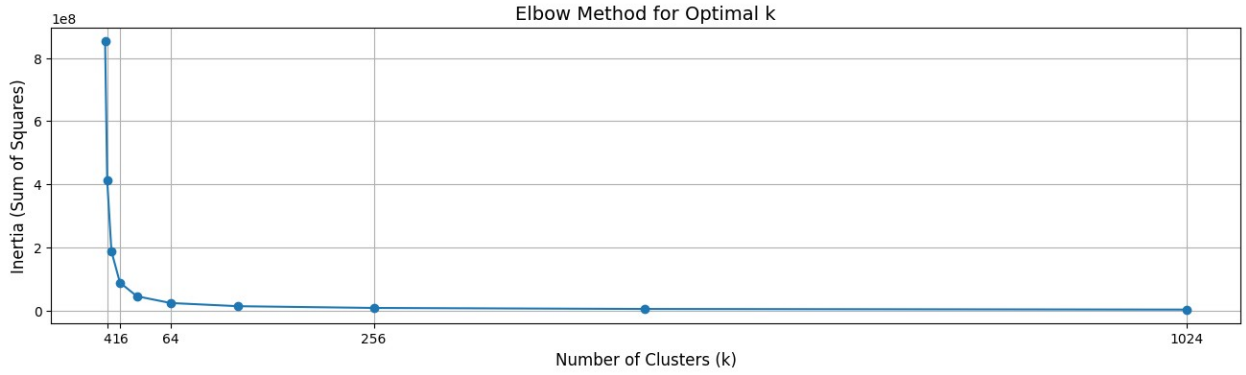
```
plt.figure(figsize=(10, 8))
plt.show()
```




```
<Figure size 1000x800 with 0 Axes>
```

Getting the optimal k value based on elbow test and silhouette score.

```
# Elbow curve
plt.figure(figsize = (16, 4))
plt.plot(k_values, ss_dist_elbow_check, marker='o')
plt.xlabel('Number of Clusters (k)', fontsize = 12)
plt.ylabel('Inertia (Sum of Squares)', fontsize = 12)
plt.title('Elbow Method for Optimal k', fontsize = 14)
plt.xticks(k_values[1::2])
plt.grid(True)
```



1. From the above curve it is clear that k = 64 is the optimal value as per the elbow method for k value selection.
2. From the images shown above it is also clear that for k = 64 the image has almost all the features of the original image.

NOTE : I tried getting the Silhouette score but it is taking too much computational power, which my laptop can't process.

Is the compression obtained lossy or lossless? What is the effect of varying the value of K in terms of overfitting or underfitting the data?

1. The k mean compression is a lossy compression because we are approximating each pixel colour using nearest centroid. We are loosing information in the process and can't revet back, hence it is also known as irreversible compression.
2. If we increase the k values then it will result in overfitting of the data, and if we use low value of k it will result in underfitting. This is evident from the elbow curve, where as we increase the k values the sum of square of error decreases.