```python
import cv2
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

image_path = '/content/download.jpeg'
image = cv2.imread(image_path)
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

def show_image(img, title="Image"):
    plt.imshow(img)
    plt.title(title)
    plt.axis('off')
    plt.show()

show_image(image, "Original Image")
```


```python
def apply_kmeans(img, K):
    img_flat = img.reshape((-1, 3))
    kmeans = KMeans(n_clusters=K, random_state=42)
    kmeans.fit(img_flat)
    clustered = kmeans.cluster_centers_[kmeans.labels_]
    clustered_img = clustered.reshape(img.shape).astype(np.uint8)
    return clustered_img

kmeans3_img = apply_kmeans(image, 3)
show_image(kmeans3_img, "K-means with K=3")
```

K-means with K=3

```
K_values = [2, 5, 7, 10, 20]
fig, axes = plt.subplots(3, 3, figsize=(12, 12))
axes = axes.flatten()

axes[0].imshow(image)
axes[0].set_title("Original Image")
axes[0].axis('off')

for i, K in enumerate(K_values):
    clustered_img = apply_kmeans(image, K)
    axes[i+1].imshow(clustered_img)
    axes[i+1].set_title(f"K={K}")
    axes[i+1].axis('off')

for j in range(len(K_values) + 1, 9):
    axes[j].axis('off')

plt.tight_layout()
plt.show()
```


```
a, b, c = 10, 10, 10
modified_img = image.copy()
modified_img = np.clip(modified_img + [a, b, c], 0,
255).astype(np.uint8)
show_image(modified_img, "Color Changed Image")
```

Color Changed Image

```
K_values = [3, 5, 7, 10]
fig, axes = plt.subplots(3, 3, figsize=(12, 12))
axes = axes.flatten()


axes[0].imshow(image)
axes[0].set_title("Original Image")
axes[0].axis('off')

axes[1].imshow(modified_img)
axes[1].set_title("Color Changed Image")
axes[1].axis('off')

for i, K in enumerate(K_values):
    clustered_img_mod = apply_kmeans(modified_img, K)
    axes[i+2].imshow(clustered_img_mod)
    axes[i+2].set_title(f"K={K}")
    axes[i+2].axis('off')

for j in range(len(K_values) + 2, 9):
    axes[j].axis('off')

plt.tight_layout()
plt.show()
```

After applying K-means clustering with different numbers of clusters, it is observed that as the value of K increases, the image segmentation becomes more detailed and better captures variations in color and texture, making the distinct regions more visually defined. When the image colors are altered by adding fixed values to the RGB channels, the overall appearance becomes brighter and slightly shifted in hue, which influences the clustering results by changing how similar pixel colors are grouped together. Applying K-means again on the modified image shows that the clusters adapt to the new color distribution, resulting in segmentation patterns that differ from the original image, demonstrating how color changes can impact the way regions are grouped and perceived.