**Image Segmentation using K-Means**

**Introduction**

Image segmentation is a fundamental task in computer vision, involving the partitioning of a digital image into multiple segments or regions. In this project, we focus on color-based segmentation, aiming to reduce image complexity and facilitate various downstream tasks such as object detection.

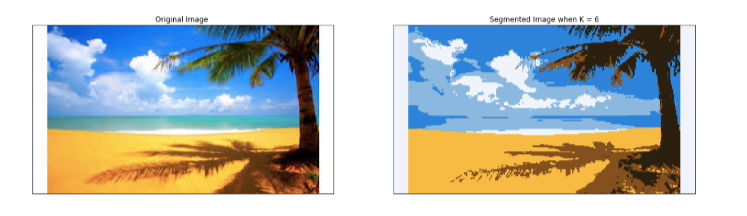
**Approach**

We adopt the K-Means clustering algorithm for image segmentation. Given an input image, we represent each pixel as a point in a three-dimensional space, with the axes corresponding to the red, green, and blue color channels. By iteratively assigning pixels to clusters based on their proximity to cluster centroids and updating centroids accordingly, we effectively group similar pixels together.

**Assignment**

**Color-based Image Segmentation**

Perform color-based image segmentation on a chosen image using a custom implementation of the K-Means algorithm with a predefined value of K (e.g., 3 clusters).



**Task 2: Visualization**

Plot the segmented image using Matplotlib to visually inspect the segmentation results.

**Task 3: Optimization**

Identify an optimal value of K for the chosen image through experimentation and evaluation. Re-run the segmentation algorithm with the new K value and visualize the newly segmented image. Compare and analyze the segmentation quality against the initial result.

**Usage**

To execute the image segmentation process:

1. Prepare your input image.
2. Run the provided Python script, specifying the image file path and desired K value.
3. View the segmented image using the visualization tool of your choice.

**Conclusion**

This project demonstrates the effectiveness of K-Means clustering in image segmentation tasks. By optimizing the choice of K, we can achieve improved segmentation results, enhancing the utility of the segmented images for downstream applications.