Image Quantization Project

**Team No. 102**

Noran Mohamed Ali Hussein

Norhan Gamal Abdelsanea Faid

Hanaa Ashraf Rezk Mohamed

Project Documentation

* **Introduction**:

The idea of image quantization is to reduce the number of colors in a full resolution digital color image (24 bits per pixel) to a smaller set of representative colors called color palette. Reduction should be performed so that the quantized image differs as little as possible from the original image. Algorithmic optimization task is to find such a color palette that the overall distortion is minimized.

* **Some Usages**

**Targetdifferentdevices*:*** color quantization is critical for displaying images with many colors on devices that can only display a limited number of colors, usually due to memory limitations.

**Image compression*:*** by reducing number of bits per pixels without affecting the image view. It’s used as a step in the compression pipeline of most common formats like JPEG and MPEG.

**Image segmentation*:*** is the process extracting useful objects from an image. It usually done by assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.

* **Main Functions:**

1. To begin working on the project, identify the unique colors and then search for smaller representative colors.

About function: Distinct () function is in charge of gathering all distinct colors and storing them in a list. This function iterates over the 2d array picture matrix, looking for all possible distinct colors while using a unique id to enable for faster access. This function runs in O(N^2) where N represents image width / height.

1. After that, we began generating the MST with the lowest cost in a single pass, rather than constructing a fully connected graph, which uses a lot of memory and wastes resources.

About function: MST () function discover the minimum spanning tree, which requires us to determine the lowest costs between the colors in order to group them. We created the tree using the list of different colors and prim's algorithm.

At first calculate distance between every two vertices and save it as cost of edge array, it runs in order (1)

Then Get index of node with the smallest cost of edge this runs in O(D)

construct graph with MST by looping on the Childs of the smallest node and check if any of them is discovered or not, and if not calculate distance for this node and its parent then storing these indices in arrays, this function runs in O(D^2) where D is the number of distinct colors.

Finally, we loop on all distinct colors and sum the smallest cost of edges that considered the shortest path in MST this function runs in O(V)

3) Then we go through the clustering part, where we break our MST into K-1 maximum sections to determine the colors that are the closest.

About function: looping on K clusters – 1 to extract K clusters from MST with maximal spacing between them and set them by negative 1 to distinguish them, it runs in O(K\*D)

In this step, we should construct a graph from our final MST, so we chose to implement the graph via adjacency list implemented as an array of list.

Graphical user interface, text, application

Description automatically generated

Then looping on adjacent list to group clusters and this when the distinct color has not adjacent list put it into dictionary and if has adjacent list loop on this adjacent list by making DFS graph on it and put them into dictionary

Graphical user interface, text, application, email

Description automatically generated

4)Find Representative color Function bycalculating average for each cluster to generate the new palette by using clusters dictionary that contains all the needed information for each cluster, it runs in order (D)

Text

Description automatically generated

5)Mapping colors in image matrix by average colors in palette byreplacing each distinct color in the input image by the average of its cluster by using cluster dictionary then loop on all distinct color for specific cluster and set average of this cluster in each distinct color, it runs in order (N^2).

* **Total complexity of all the project is O(D^2)**