Task #4 Computer Vision

GitHub Repo Link:

Team Information

Name	Sec	BN
Ahmed Khaled	1	4
Alaa Gamal	1	15
Bassam Mostafa	1	22
Mustafa Yehia	2	33

How to Run our Application

For VS code users

1. Open VS Code
2. Download extensions named (CMake, CMake Tools)
3. Open the Project Directory on VS Code
4. Press CTRL + SHIFT + P
5. Choose "CMake: Configure" → If it asks you to choose a kit, Choose GCC 9.3.0 → If it is not in the options then download GCC on your device
6. Press CTRL + SHIFT + P
7. Choose "CMake: Build"
8. Press CTRL + SHIFT + P
9. Choose "CMake: Run Without Debugging"
10. Enter the full "absolute" path of the image you want to detect the edges on
11. Go check out the results in the outputs folder

For UNIX terminal users

1. Create "build" directory if it does not exist
2. Run a terminal in the build folder
3. Run the command ```cmake ..```
4. Run the command ```make```
5. Run the project ```./CVProject```
6. Enter the full "absolute" path of the image you want to detect the edges on
7. Go check out the results in the outputs folder

Demo Link

- Demo Link

GUI Design and Implementation

Framework

- The used framework is GTK and is implemented in C++ using gtkmm wrapper which supports all OOP features of C++ that would be impossible to use without it.

Reasons for this framework

- GTK is chosen over other C++ frameworks such as Qt and wxWidgets for multiple reasons:
    - Seamless integration with GNOME desktop environment (the GNOME environment itself is built using GTK)
    - Is free to use for proprietary software (as opposed to Qt)
    - Uses C++ std classes (as opposed to Qt which uses its own classes)

Design

- The application is designed using the “Glade” app, which produces an XML file that contains the properties of the widgets (components) of the app.

Thresholding

Assumptions for Thresholding

1. Intensity values are different in different regions.
2. Within each region, which represents the corresponding object in a scene, intensity values are similar.

1. Optimal Thresholding

Main Concept

Idea: Histogram of image is sum of two overlapping distributions.

Main Concept: Overlapping point of these distributions (corresponds to the minimum probability between the maxima of 2 distributions).

Problem: Distributions are unknown.

Algorithm

1. Select initial estimate for T_a
2. Segment image using T.
   This produces 2 groups: $G_1\ pixels\ with\ value\ >\ T$ and $G_2\ with\ value\ <\ T$
3. Compute μ_1 and μ_2, average pixel values of G_1 and G_2
4. **New threshold:** $T = \frac{1}{2}(\mu_1 + \mu_2)$
5. Repeat steps 2 to 4 until T stabilizes.
   Optimal thresholding methods select threshold based on minimization of criterion function.

Multi-threading ** (CPU: 4.5 GHz) un-affected
Clock Error

2. Otsu Thresholding

Algorithm

Algorithm

The criterion for Otsu is the minimization of the within-group variance of the two groups of pixels separated by the threshold

3. Spectral Thresholding

Result

Image Segmentation

Motivation

Definition: Process of partitioning image into multiple segments.

Goal: Change representation of image into something that is more meaningful and easier to analyze. It is usually used for **locating** objects and **creating** boundaries.

Concept: Image is basically a set of given pixels. In image segmentation, pixels which have similar attributes are grouped together. Image segmentation creates a pixel-wise mask for objects in an image which gives us a more comprehensive and granular understanding of the object.

K-means++ Algorithm

Main Concept

It is a **clustering algorithm**. Clustering algorithms are **unsupervised algorithms** which means that there is no labelled data available. It is used to identify different classes or clusters in the given data based on how similar the data is. Data points in same group are more like other data points in that same group than those in other groups (k: Number of Clusters).

Algorithm

1. Choose number of clusters you want (k).
2. Randomly assign data points to any of k-clusters.
3. Calculate center of clusters.
4. Calculate distance of data points from centers of each of clusters.
5. Re-assign data points to the nearest clusters depending on the distance.
6. Again, calculate the new cluster center.

Repeat steps 4, 5 and 6 till convergence is almost the same.

Region Growing Algorithm

Main Concept

Assumption that **neighboring pixels** within one region have similar values.

The common procedure is to compare one pixel with its neighbors.

If a similarity criterion is satisfied, the pixel can be set to belong to the cluster as one or more of its neighbors. The selection of the similarity criterion is significant, and the results are influenced by noise in all instances.

This method takes a set of seeds as input along with the image. The seeds mark each of the objects to be segmented. The regions are iteratively grown by comparison of all unallocated neighboring pixels to the regions. The difference between a pixel's intensity value and the region's mean, is used as a measure of similarity. The pixel with the smallest difference measured in this way is assigned to the respective region. This process continues until all pixels are assigned to a region. Because seeded region growing requires seeds as additional input, the segmentation results are dependent on the choice of seeds, and noise in the image can cause the seeds to be poorly placed.

Agglomerative (Hierarchical) Algorithm

Main Concept

Algorithm

Mean Shift Algorithm

Main Concept

Algorithm

Results

K-means++ Algorithm

Region Growing Algorithm

Agglomerative Algorithm

Mean Shift Algorithm