

Bishop's University
Department of Computer Science
CS462/562
HW3 – Edge detection and evaluation

- **Objective:** Practice various edge detection methods and assess their performance against the ground-truth (reference).
- **Data:** You are provided with four (4) color images and their respective ground-truth (GT) edge maps. The GT maps are stored in ".mat" format. Go to page 4 for more details about this format. A code is provided to read GT maps.
- **Requirements:** Include your code in Jupyter Notebook and feel free to utilize libraries such as Scikit-image, OpenCV, Matplotlib, etc.
- **Task:** Your task is to answer the following questions:
 1. Read the input image (189080.jpg) and display it.



2. Read the ground-truth image of the edge and display it.



3. If the input image is RGB, then convert it to a gray-scale image.



4. Using Roberts and Sobel edge detection methods, detect the edge of the gray-scale image. Display the results as follows:

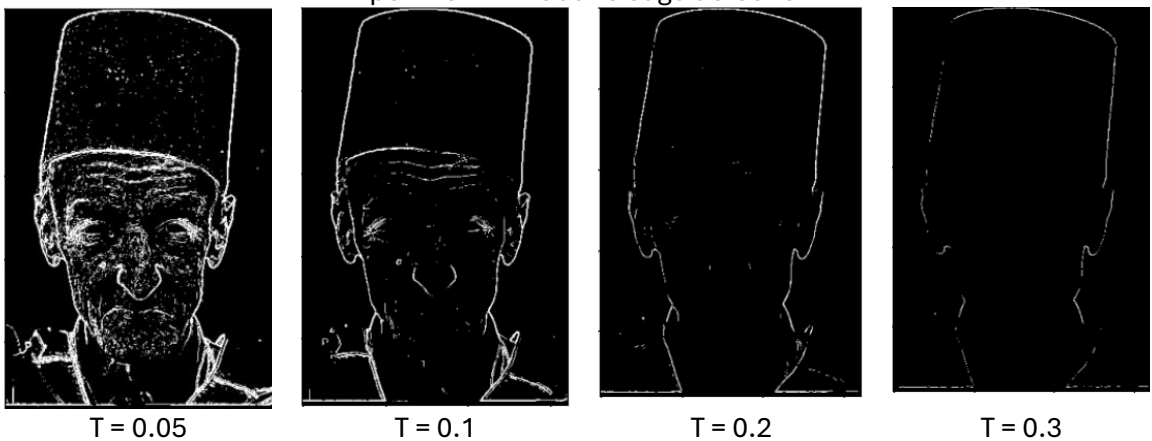


5. The edge pixels' values are not restricted to 0 or 1; they are floating-point numbers. To obtain a binary (0/1) edge output, classify the output of each edge detection method using different threshold values: $T = 0.05, 0.1, 0.2$, and 0.3 . Present the results as follows:

Experiment 1: Sobel edge detection



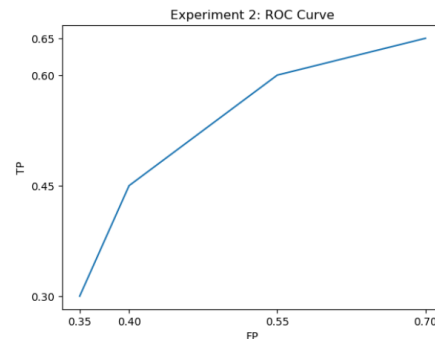
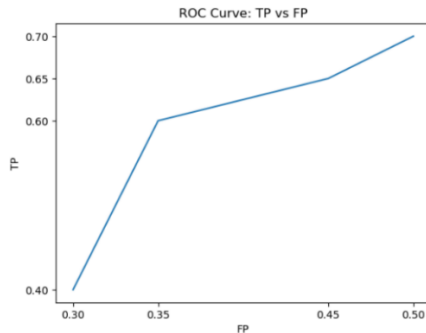
Experiment 2: Roberts edge detection



6. Compute TP, TN, FP, FN, Recall, Precision and F-score for each experiment. Consider the edge as the positive class and the background as the negative class, and show the results in two (2) decimals in this way:

-
- Experiment 1 (Soble):
 - T = 0.05: TP = ..., TN=..., FP=..., FN=..., Recall=..., Precision=..., F-score=...
 - T = 0.10: TP = ..., TN=..., FP=..., FN=..., Recall=..., Precision=..., F-score=...
 - T = 0.20: TP = ..., TN=..., FP=..., FN=..., Recall=..., Precision=..., F-score=...
 - T = 0.30: TP = ..., TN=..., FP=..., FN=..., Recall=..., Precision=..., F-score=...
-
- Experiment 2 (Roberts):
 - T = 0.05: TP = ..., TN=..., FP=..., FN=..., Recall=..., Precision=..., F-score=...
 - T = 0.10: TP = ..., TN=..., FP=..., FN=..., Recall=..., Precision=..., F-score=...
 - T = 0.20: TP = ..., TN=..., FP=..., FN=..., Recall=..., Precision=..., F-score=...
 - T = 0.30: TP = ..., TN=..., FP=..., FN=..., Recall=..., Precision=..., F-score=...
-

7. Plot the ROC curve for the two experiments. Below is an example of what the ROC curve might look like. Please note that these graphs are for illustrative purposes only and may not represent the actual output of the experiments.



8. Display the ROC curves for all provided images. Avoid duplicating the code for each image. Utilize the same code by altering the input image, and finally showcase the ROC curves.
9. Provide a brief description of which edge detection method is better for each image.

Definition of ".mat" format:

The ".mat" format is a file format used to store data in MATLAB. It stands for "MATLAB data" and is commonly used for saving arrays, matrices, variables, and other data structures generated in MATLAB. The format allows for easy interchangeability of data between different MATLAB sessions and provides a convenient way to save and load data for further analysis or processing. ".mat" files can contain various types of data, including numerical arrays, character arrays, structures, and cell arrays, and they can be read and written using MATLAB functions.

Python can read ".mat" files using libraries such as SciPy. The SciPy library provides a module called "io" that includes functions to read and write MATLAB files. Specifically, the `scipy.io.loadmat()` function can be used to load data from a ".mat" file into a Python dictionary, where the keys correspond to variable names and the values contain the data stored in those variables. This allows Python users to access and manipulate MATLAB data within their Python environment.