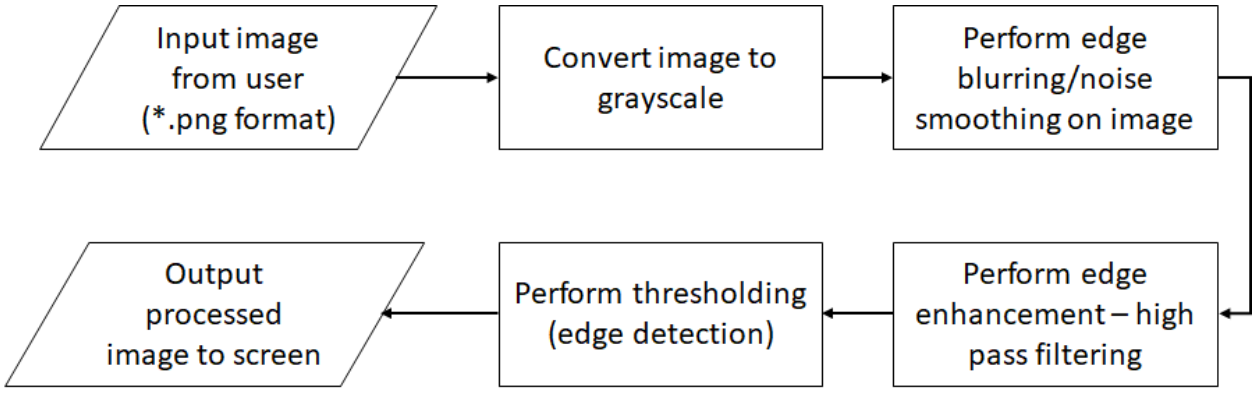
ENGR 13300
Python Group Project Fall 2020

Project Description:

Nearly all of the engineering disciplines use image and video analysis to implement solutions to engineering problems. For example, computer vision is used to identify objects such as potholes in roadways, structural damage in materials and infrastructures, and product defects in manufacturing. Image analysis and computer vision are used to guide autonomous vehicles and drones, assist in medical imaging evaluation, and in agricultural and retail and banking applications. Facial recognition employing big data, deep learning, and computer vision is used in many applications. The applications continue to be expanded due to the development of extensive libraries such as scikit-image and OpenCV which contain powerful functions that can complete these complex operations in just a few lines of code. As a first step in exploring these broader applications, your team will develop a Python program to perform the steps of basic edge detection on a PNG (.png) color image as defined below and shown in Figure 1. Y***our team will develop your program using the core functions of Python and the basic scientific libraries of numpy, matplotlib and scipy what come installed with the anaconda package. The goal is to help you better understand these general libraries. Therefore, do not use the advanced libraries with built-in image processing libraries such as scikit-image or OpenCV***.

Figure 1. Systems Diagram of Edge Detection:

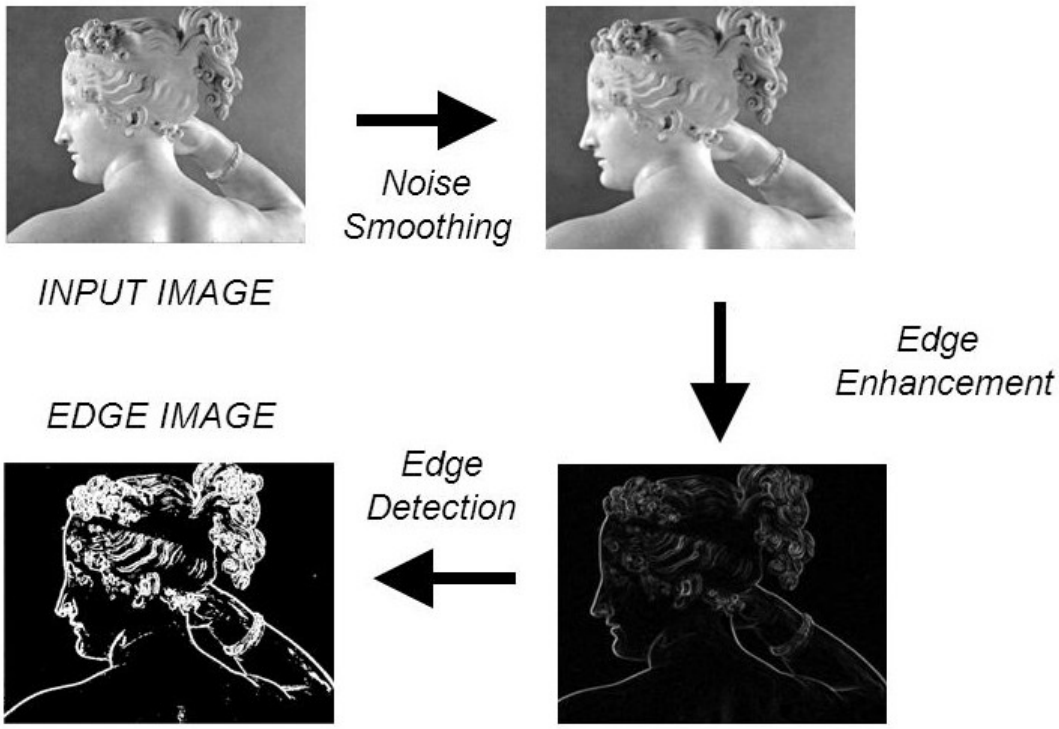


Your program should:

- **Prompt the user** for the name of a color PNG image file, complete the processing, then save the various processed images to `.png` output files. Use matplotlib to read the image file, as well as export it back as `.png`. Implement your image processing algorithms using only the `numpy` library.
- **Conduct error checking** on inputs, inform the user if they enter an incorrect filename, then ask the user to re-enter proper inputs or exit appropriately.

- **Convert color** PNG image to grayscale using an appropriate standard. Your program should ***only*** use `numpy` functions and save the converted image as a `.png` image.
- Perform edge detection using the steps below and output the processed image as a `.png` image after each step (see Figure 2) using only the `numpy` library:
    - Edge blurring/noise smoothing
    - Edge enhancement – high pass filtering
    - Thresholding (edge detection)
- Output image of the image with edge detection highlighted. This file should be a png format viewable in web browser window.
- **Utilize function(s) in this project** to allow for modularity and reusability. This will also allow your project team to assign each group member with a specific task, and the member will individually work on their task and build their user-defined function, that accepts inputs they need and return the desired output.

Figure 2: Edge Detection Example (Source: Image Processing and Analysis Tutorial – Part II by Edward J. Delp. 8/29/2020.)



Usability Design

In addition to developing a reliable algorithm to detect edges, the team should develop a useful process for interacting with the algorithm. Future users of a team's image processing library will want a robust and easy method to interact with it to perform basic operations on an image. For example, performing

fine tuning to identify the values for parameters that optimize edge detection for various images. These needs could include, but are not limited to –

- Easy input of desired files to process
- Graceful exit from program errors (no system errors and useful message to users of what the problem is)
- easy adjustment of parameters to improve edge detection
- Useful output of edge detection performance

The algorithm already anticipates the first two items. The other two will make a user's experience better when performing multiple trials on an image to find the best settings.

Deliverables

Each team will submit a project report outlining their research into the image processing application of their choice, the image processing methods they used, and the code the team developed. The project report will follow the common structure of a technical report. The Python files along with a report will be uploaded to Gradescope as the final submission. Make sure that the files will run as they are uploaded (e.g. have them all in the same folder). You may zip all your files together. The report will consist of:

1. **Project Motivation** describing how image processing techniques are used in the engineering discipline of the choice citing at least three credible sources using either APA or MLA format. This section will outline the major problems trying to be solved and how image processing is a better alternative to other methods for solving the problem. The description should include the unique challenges that application has for using image processing techniques.
2. **Project Overview and Methods** section describes how the team's smoothing, edge enhancement, and edge detection/thresholding algorithm implementations work and brief summary of background research and evidence to support on methods chosen for each of the processing steps.
3. **Discussion of Algorithm Design**: Flow charts of the overall program and each of the user-defined functions. This section should describe the design rationale for modularizing the code in a way to meet the needs of future uses of the image processing methods. In addition, this section should discuss ways in which your design addressed issues related to Usability Design.
4. **References:** List of resources used to research the application of image processing in disciplines and any additional resources used for methods for conducting image processing.
5. **Appendices: Include as appendices to your main report**
    a. **User manual** with sample inputs and outputs (including pictures).
    b. **Project management Plan**: Summary of the contributions of each member of the team to the project. Describe your team's methods for collaboration that facilitated active participation of and learning by all team members. Include a discussion of opportunities for improvement for future team projects.
    c. **Discussion of Design Process**: Description of how your team followed the design process, which is defined as "a process of devising a system, component, or process, to meet desired needs within constraints. It is an iterative, creative, decision-making process that involves

developing requirements, performing analysis and synthesis, generating multiple solutions, evaluating solutions against requirements, considering risks, and making trade- offs."

d. **Code:** The team should upload the Python files, but they should also include the code as an appendix in the report. Code should be well commented to increase readability for future users of the algorithm.

Deadlines:

In class **demo: October 15th**

Final Reports and Programs **uploaded to Gradescope, Thursday, October 15th at the start of class**.

Project Grading Summary (a detailed grading rubric will be provided):

Points	Code
70	Code runs with no errors and image processing portions are implemented using only matplotlib and NumPy modules
6	Errors handled and structured appropriately
10	Comments in code
	Report
5	Project motivation and application to major or career interest(s)
10	Project overview and methods discussion
10	Algorithm Design rationale and Flow charts
5	User manual for code
3	Project Management Plan (Summary of contributions)
3	Discussion of Design Process
3	Appendix includes the code
125	**Total**