Assignment 2

COS 470/570: Image Processing and Computer Vision

Objective:

The objective of this assignment is to apply image processing techniques to measure object size and recognize text using Python and OpenCV.

Tasks:

Download "book.jpg", "sign1.jpg", "sign2.jpg", "sign3.jpg", "sign4.jpg", and "text.jpg" from BrightSpace. After downloading, proceed as follows:

Task 1: Object size measurement (50 points).

For this task, work on "book.jpg" to measure the size of a book placed on an A4 paper with dimensions 27.8cm by 21.5cm.

1. **Detection and Measurement:** Utilize appropriate techniques to detect the book in the image and measure its dimensions (height and width in centimeters). Describe the methodology and rationale for your approach in your report.
2. **Annotation:** After detecting the book, annotate the image by drawing a rectangle around the book. Clearly display the measured width and height on the image.
3. **Comparison and Documentation:** The actual size of the book is 8 cm by 10.6 cm. Compare your measurements to these dimensions and document the results in your report. Aim for an error rate of less than 10% with the ideal method.

Ensure that your report is detailed, including descriptions of the methods used, your results (screenshots), and any challenges you encountered.

Task 2: Text recognition (50 points).

In this task, you are asked to explore how to use OpenCV and Python-tesseract, a wrapper for Google's Tesseract-OCR engine, to recognize text. Begin by reviewing the following tutorials:

1. https://pypi.org/project/pytesseract/
2. https://github.com/madmaze/pytesseract

Your tasks:

1. **Text Analysis:** Work on “text.jpg” to recognize the text in the image. Write code to perform the recognition, and include a screenshot of your recognized text in the report.
2. **Sign Recognition**: Apply your methods to “sign1.jpg”, “sign2.jpg”, “sign3.jpg”, and “sign4.jpg” to recognize the text on these traffic signs. Document any additional methods you employ to improve accuracy if direct detection fails. Explain your strategies and summarize the accuracy of your method across these four cases in the report. You can calculate the accuracy as this:

   Acc = (the number of correct recognition)/(the number of total letters)

   For example, the sign is “stop”, while your recognition is “step”. In this case, the accuracy will be 75%. Ensure your report provides enough evidence (screenshots) to prove that your code is functional.

   Hints: Use detection algorithms discussed in class to identify regions of interest (RoI), and apply PyTesseract only to these RoIs. Consider employing image preprocessing methods such as thresholding or noise reduction. Tuning the parameters of specific functions is also encouraged.

Submission:

Submit your Python scripts along with a report detailing your code and approach. Ensure your report provides enough evidence (screenshots) to prove that your code is functional. The deadline of this assignment is October 2 at 11:59 PM.