

CS 560 Advanced Topics in Artificial Intelligence Computer Vision (Winter 2015)

HW1 (2%)

- Assigned date: 01/15/2015, Thursday
- Due date: 01/21/2015
- Electronic submission via CSNS. Include your name and CIN# in your program.
- Submit source codes. Do not submit any images.
- Include a readme.txt file in your submission. Explain how to compile, run and test your program in very detail. Also, disclose any resources/references that you used and explain your methods used to solve given problems.

This homework is designed for you to get familiar with OpenCV programming and outputs produced by different image processing functions/operations for different types of images.

Download [testdata.zip](#). Write an OpenCV program that receives one of the following options from the user and performs the selected task. To implement, refer to “Image Processing” API.

<http://docs.opencv.org/modules/imgproc/doc/imgproc.html>

For each task, display input and output images on the screen.

For each task, allow the user to choose an image from the given test images.

1. Color to Gray conversion
 - Convert a color input image to the gray-scale image.
 - Refer to `cvtColor`.
2. Resize image
 - Accept a resizing factor (1 for the same size, 0.5 for the half size image, etc) from the user, and resize the image.
 - Refer to `resize`.
4. Erode/Dilate/Open/Close
 - Let the user choose which morphological operation to be performed. Perform the selected operation (e.g. erosion, dilation, open, or close.)
 - Refer to `erode/dilate`.
 - The OpenCV morphological functions can process a color image without converting it to a binary image explicitly. Receive an option from the user such that the user can decide whether or not converting the input image into a binary image first.
 - Optional: By default, the operations use a 3x3 rectangular structure element. Define other structural element and perform the operation. Refer to `createMorphologyFilter`.
5. Blur/Smooth
 - Let the user choose an option.
 - Smooth the image. Use `GaussianBlur`.
 - Blur the image using the Box filter provided by the OpenCV.
6. Equalize the histogram
 - Display an output after histogram equalization.

- Refer to `EqualizeHist`.
- 7. Connected component analysis
 - Display connected components of the image.
 - Refer to `findContours` or `floodFill`. An example using the `floodFill` technique can be found at `opencv_source_code/samples/cpp/ffilldemo.cpp` in your computer.
- 8. Implement Otsu's method
 - Implement Otsu's method from scratch and display the result.
 - Display the histogram
 - Do not use any pre-defined Otsu's method. You will be required to show me the code that implements the method based on my lecture note.

Things to remember:

- Do not modify the input image. After each task, the input image should remain untouched.
- When your input is a color image and if OpenCV functions accept a gray-scale image as an input parameter, convert your input to be gray-scaled first.
- Supply your preferred value if any OpenCV functions require a threshold value as a parameter.