

# CP 6475 Assignment 2: Image I/O and Python Libraries

The purpose of this assignment is for you to gain experience using the numpy and openCV libraries in python. The main task will be required to complete several short functions in the assignment2.py file, and then write a report documenting your work. Some of these functions will be useful in later assignments. You will need to use some of your own digital pictures (either old or new) in the template report.

## Instructions

### 1) Clone the Git Repository

Clone the assignment repository into a vagrant shared folder (by default, the folder containing your Vagrantfile from the assignment 1 repository is shared with the VM). You can do this from your host OS if you have git installed, otherwise you can do it from within the VM by running `vagrant up`, then connecting by running `vagrant ssh`.

```
git clone --recursive --branch fall2016 https://github.gatech.edu/omscs6475/assignment2.git
```

**Note:** If you receive a server certification failure error, add `-c http.sslVerify=false` to your git commands. (e.g., `git -c http.sslVerify=false clone ..`)

**Note:** If you want to use the git repository locally to track your changes, you should checkout a new branch immediately after cloning: `git checkout --b <new_branch_name>`

### 2) Implement the Functions in assignment2.py

The python file assignment2.py is supplied in the git repo. Specific instructions and requirements are provided in the docstrings in this file. Write your code only in the specified areas, and do not add any libraries. You may ask general questions about these functions on Piazza, but do not share code on class-wide posts.

- **numberOfPixels:** This function takes in a grayscale image, and you must output an integer which represents the number of pixels in this grayscale image.
- **averagePixel:** This function takes in a grayscale image and outputs the average pixel value of that image as an integer.

**Note:** You will need to run the next three functions on your own images (images that you have taken, now or in the past) after it passes the autograder for the template report.

- **convertToBlackAndWhite:** This function converts a grayscale image to a 1-bit black and white image. Details are provided in the function description.
- **averageTwoImages:** This function requires you to average two input images into the output. Further details are provided in the function description.
- **flipHorizontal:** This function flips an image along the horizontal axis. Further details are provided in the function description.

**Note:** You should write code to try your functions on test\_image.jpg and test\_image\_2.jpg (a horizontally flipped version of test\_image.jpg), but *you will only submit the five*

*functions listed above in assignment2.py*. You can use the [cv2.imread](#) and [cv2.imwrite](#) functions to read and write images.

**Note:** Upon submission, your code will be tested by the autograder. You will receive test results immediately, and you can resubmit an unlimited number of times before the due date -- *but we will only grade your last submission before the late submission deadline*.

**Note:** The autograder test cases will not be released to students, but you are encouraged to use the skeleton provided in the file *assignment2\_test.py* to write your own unit tests. You may discuss tests with other students, but *you are not allowed to share any source code (applies both to assignments & test cases)*.

### 3) Complete the Report

Make a copy of the [assignment 2 report template](#) and complete the report by answering all of the questions. Save your report as “assignment2.pdf” (the name is important because the submission script searches the current working directory to find this file).

**Note:** Your report must be 6MB or less. If you need to compress your PDF before submitting, you can use `cv2.resize` or [smallpdf](#). You must answer *all* questions in the template to receive full credit on the assignment.

### 4) Submit the Assignment Code & Report

Use the course VM to upload your report (as a PDF) and your code to Bonnie using the `submit.py` script from the assignment2 repository:

- Launch the VM & initiate a remote connection

```
vagrant up  
vagrant ssh
```

- Change to the project directory (the assignment2 git repository) and submit your work. You must submit the code (`assignment2.py`) to the “assignment2” quiz, and the report (`assignment2.pdf`) to the “writeup2” quiz.

```
cd /vagrant/<a2_git_repository_folder>  
python submit.py [assignment2|writeup2]
```

You will be asked to acknowledge the course late policy and the Georgia Tech honor code pledge, then the script will upload your files and display a confirmation message. You can verify your submission by visiting <https://bonnie.udacity.com>

**Note:** If you use two-factor authentication, you will need to manually download the jwt. [https://bonnie.udacity.com/auth\\_tokens/two\\_factor](https://bonnie.udacity.com/auth_tokens/two_factor)

## Evaluation Criteria

- Submission of working code. (scored by the autograder)
- Thoughtful and appropriate answers to all questions in the template. (Use complete sentences.)
- Including all images required for the report.

# Setting up a Native Development Environment (For Reference Only)

The course VM has all of the required software and libraries required for this course pre-installed. However, in the interest of completeness, we use the following libraries in this course:

1. Python  $\geq 2.7.6$ ; stay in the 2.7 releases
2. OpenCV  $= 2.4.13$ ; stay in the 2.4 releases
3. Numpy  $\geq 1.8.2$
4. Scipy  $\geq 0.13.3$

The following guides for configuring your native environment were prepared for past semesters, however, ***remember that the TAs will not provide official support for any problems you encounter if you are not using the VM. Different versions of these four have resulted in discrepancies in the past.***

- [Mac OSX](#)
- [Windows](#)