# Department of Software Engineering

**CS 474: Computer Vision**

**Class: BESE-7**

**Lab 1: Image filtering**

**Date: 21-01-2020**

**Time: 10:00 am-1:00 pm**

**Instructor: Dr. Muhammad Moazam Fraz**

**Lab Engineer : Ms Anum Asif**

**Course Learning Outcomes (CLOs)**

|  |  |  |  |
| --- | --- | --- | --- |
| Upon completion of the course, students should demonstrate the ability to: | | **PLO Mapping\*\*** | **BT Level\*** |
| CLO 1 | Understand computer vision algorithms and tools and techniques. | PLO 1 | C2 |
| CLO 2 | Develop solutions for image/video understanding and recognition. | PLO 3 | C3 |
| CLO 3 | Use modern tools to solve practical problems. | PLO 5 | C5 |

\* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

* *Knowledge(C-1), Comprehension(C-2), Application(C-3), Analysis(C-4), Synthesis(C-5), Evaluation(C-6)*

*\*\* PLOs are published on department website*

# Lab 1: Convolution and Image Filtering

**Learning Outcome**

CLO 1: Understand computer vision algorithms and tools and techniques.

**Tools/Software Requirement**

Python , Jupyter, Numpy, PIL, OpenCV, Scikit Image

**Objectives**

This lab gives introduction on how to use images in python using various libraries. It focuses on understanding and implementing the basic concepts such as image manipulation operation using python libraries.

**Introduction**

This lab is based on basic image processing operations which includes

* Reading, displaying and writing images, applying 1st order and 2nd order image filters to compute gradients and display.
* Capture Video Stream from Webcam and display in a window, apply Image Gradient operation on live video and display in the video (optional/extra credit)

A comprehensive list of Python libraries and packages used for computer vision can be seen [here](https://opensource.com/article/19/3/python-image-manipulation-tools).

**Lab Tasks**

**Task 1**

* Read and display an image into the workspace. [you may use MatplotLib for image display ]
* To see the distribution of intensities in the image, create a histogram by calling the imhist function.
* Improve the contrast in an image, you may like to use histogram equalization. If you compare the two histograms (equalized and original), you can see that the histogram of equalized image is more spread out over the entire range of intensities.

**Hints:**

* <https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_histograms/py_histogram_equalization/py_histogram_equalization.html>
* <https://scikit-image.org/docs/dev/auto_examples/color_exposure/plot_equalize.html>

**Task 2**

Use functions available in various python packages for convolution, which includes

* scipy method scipy.signal.convolve2d(img, kernel, 'valid')
* OpenCV method cv2.filter2D(... , .. , ...).
* Apply box filter using convolution, and display the resultant image
* Apply Gaussian filter to the image, with varying sigma values.
* Add Gausian Noise and Salt and Pepper Noise to them.
* Apply Gaussian Filter and Median Filters.
* Apply Sobel operator, computer gradient magnitude and display the results (original image, gradient images and gradient magnitude image)
* Apply Canny Edge Deteciton Operator and display the results.

**Task 3 [optional / extra credit]**

* Read the video from webcam and apply any of the filtering operation (blurring, gradient magnitude, edge detection e.t.c on the video stream in the real time and display the resultant video stream, showing the effect of image filtering operation in real time.
* **Hint: You can use ipywebrtc library**
* <https://ipywebrtc.readthedocs.io/en/latest/VideoStream.html>
* <https://github.com/maartenbreddels/ipywebrtc>

**Deliverables**

* Jypyter Notebook
* PDF of Jupyter Notebook. . In order to export notebook as a PDF, you will need to install [additional libraries](https://nbconvert.readthedocs.io/en/latest/install.html#installing-nbconvert).

## How to start working on Labs and setting up Jupyter Notebooks?

Before working on each homework, you need to setup a few things:

1: **Installing Python 3.6+:** To use python3, make sure to install version 3.6 or 3.7 on your local machine. If you are on Mac OS X, you can do this using Homebrew with [brew install](https://raw.githubusercontent.com/Homebrew/homebrew-core/f2a764ef944b1080be64bd88dca9a1d80130c558/Formula/python.rb). You can find instructions for Ubuntu [here](https://www.digitalocean.com/community/tutorials/how-to-install-python-3-and-set-up-a-local-programming-environment-on-ubuntu-16-04).

2: **Setting up a virtual environment:** we strongly recommend working using a [virtual environment](http://docs.python-guide.org/en/latest/dev/virtualenvs/) for each Assignment. This will allow you to have a working environment with all the package dependencies within the repository of your homework, without messing up your work environment in other repositories.

To set up a virtual environment with name .env, run the following inside your Labs Home directory (ex: inside cs474/Labs/):

sudo pip install virtualenv # You will need to do this only once

virtualenv -p python3 .env # Creates a virtual environment with python3

source .env/bin/activate # Activate the virtual environment

pip install -r requirements.txt # Install all the dependencies

# Work on the assignement for a while...

deactivate # Exit the virtual environment when you're done

Note that every time you want to work on the assignment, you should run source .env/bin/activate (from within your assignment folder) to re-activate the virtual environment, and deactivate again whenever you are done.

The above mentioned instructions for virtual environment setup are for UBUNTU. It is a little bit different for Windows.

3 :**Working with Jupyter notebooks:** In your assignment repository, start the *notebook* with the *jupyter notebook* command. You might have issues if you are in a virtual environment, as the notebook might not recognize your virtual environment and might not find the kernel located in .env to execute code. In this case, refer to this page and do the following within your virtual environment:

python -m ipykernel install --user --name=my-virtualenv-name

If you are unfamiliar with Jupyter notebooks, you can also refer this [IPython tutorial](http://cs231n.github.io/ipython-tutorial).

When working with a Jupyter notebook, you can edit the \***.py** files either in the Jupyter interface (in your browser) or with your favorite editor (vim, Atom...). Whenever you save a \***.py** file, the notebook will reload their content directly.