**ENEE408I: Lab Tutorial 2**

Introduction to NumPy, Matplotlib, SciPy, and OpenCV



# Introduction

The purpose of this laboratory tutorial is to introduce you to the fundamental concepts and functionalities of NumPy, Matplotlib, SciPy, and OpenCV, and to equip you with the skills to utilize these libraries in Python programming.

# Pre-requisites

* Python 3.x installed
* Code editor (like Visual Studio Code, PyCharm, or Jupyter Notebook)
* Internet connection for downloading necessary libraries

# Procedure

## Setting Up a Virtual Environment

A virtual environment is a self-contained directory tree that contains a Python installation for a particular version of Python, plus a number of additional packages. It is a tool that helps to keep dependencies required by different projects separate by creating isolated Python environments for them.

### Creating a Virtual Environment

To create a virtual environment, open your terminal or command prompt and run the following command:



This command will create a new directory called ***lab\_env*** which contains a fresh Python installation.

**(\*)** - If this command does not work, it is most probably because you don't have Python installed or Python has not been set in the environment variables.

### Activating the Virtual Environment

Once the virtual environment is created, you need to activate it. The activation process differs slightly between Windows and Linux/Mac:

On Windows:



On Linux/Mac:



### Deactivating the Virtual Environment

To deactivate the virtual environment and return to your global Python environment, simply run:



### Installing Libraries within your Virtual Environment

Activate the virtual environment you created and install the necessary libraries in the active virtual environment using the following command:



**Create a python file for each of the sections after this. You will need to upload these files to github and submit before the deadline for this tutorial.**

## NumPy

Documentation: <https://numpy.org/devdocs/user/quickstart.html>

1. Why is numpy faster than using for loops in Python for operations such as matrix multiplications?

Numpy is implemented with C, and since C is a compiled language it runs faster than Python. Numpy also has better memory management when it comes to storing arrays and matrices.

1. What is the data type used in numpy which makes such operations feasible? Also name a few differences between this data type and its similar counterpart in Python

The NumPy’s array class, ndarray, allows handling of arrays of more than 1 dimension. Some other differences are:

* NumPy arrays have a fixed size at creation
* Elements in a NumPy array are all required to be of the same data type.

1. Go through the documentation and create a numpy array with the elements [1,2,3,4]
2. Use **np.ones**, **np.zeros** to create an array of 1’s with dimension 3x4 and an array of 0’s with dimension 4x3.
3. Create a 2x3 matrix A and a 3x4 matrix B and perform a matrix multiplication using numpy.
4. Find the eigenvalues and eigenvectors of the matrix given below?

## Matplotlib ⎡⎢⎣ 31 12⎤⎥⎦

Documentation: <https://matplotlib.org/stable/tutorials/introductory/quick_start.html>

1. Create a line plot of the sine function over the interval [0, 2π] using Matplotlib.
2. Add labels to the axes in a Matplotlib plot.
3. Plot the 3d graph of the function given below using Matplotlib.

## SciPy 𝑧 = 𝑠𝑖𝑛( 𝑥2 + 𝑦2)

Documentation: <https://docs.scipy.org/doc/scipy/tutorial/index.html#user-guide>(1) Solve the linear system of equations given below using SciPy. 3𝑥 𝑥 ++2 𝑦𝑦 == 98

(2) Find the minimum of the function given below using SciPy's optimization module.

### 𝑦 = 𝑥2 + 2𝑥

(3) Perform the Fourier transformation of the function given below using SciPy. Plot the frequency response using matplotlib. **3.5. OpenCV** 𝑓(𝑥) = 𝑠𝑖𝑛(100π𝑥) + 12 𝑠𝑖𝑛(160π𝑥)

Documentation: <https://docs.opencv.org/4.x/>

1. Read an image and convert it to grayscale using OpenCV.
2. Perform edge detection on the image using OpenCV.
3. Use a Haar cascade classifier to implement face detection using OpenCV on an image which contains faces.

**Homework**

**Explain how Haar cascades work to detect faces.**

Haar cascades use a series of haar features. These features are rectangular patterns of black and white regions that resemble things like edges and lines. These features work by computing the sum of the pixel intensities in the white and black regions. The haar cascade is trained using a large database of positive (faces) and negative (non-face) images. The reason it’s called a cascade is because it goes through a cascade of stages. The first stage may discard regions that obviously don’t contain a face, but as the stages progress, the classifications get more specific, eventually looking for things like eyes and noses. Eventually, a full face can be found.