Install GNU Octave

For windows

- Download the installer:
- Go to the GNU Octave official website.
- Click on the link for the Windows installer (usually a `.exe` file).
- Run the Installer:
- Locate the downloaded `.exe` file and double-click it to run the installer.
- Follow the on-screen instructions. You can choose the default settings for most options.
- Install Dependencies:
- The installer should include all necessary dependencies. If prompted, allow the installer to download and install any additional components.
- Launch Octave:
- After installation, you can find GNU Octave in the Start Menu. Click on it to launch the application.
- Verify Installation:
- Once Octave is open, type `version` in the command window and press Enter. This should display the installed version of Octave.

For macOS

- Download the Installer:
- Visit the GNU Octave download page.
- Look for the macOS installer (usually a `.dmg` file).
- Install Octave:
- Double-click the downloaded `.dmg` file to mount it.
- Drag the Octave application into your Applications folder.
- Install XQuartz:
- Octave requires X11 for graphical displays. Download and install XQuartz.
- Follow the installation instructions provided on the XQuartz website.
- Launch Octave:
- Open the Applications folder and find GNU Octave. Double-click to launch it.
- Verify Installation:
- In the Octave command window, type `version` and press Enter to check the installed version.

For linux

- Using Package Manager:
- Most Linux distributions have GNU Octave available in their package repositories. Open a terminal and use the following commands based on your distribution:
- For Ubuntu/Debian:

```
sudo apt update
sudo apt install octave
```

- For Fedora:

```
sudo dnf install octave
```

- For Arch Linux:

O. ELOUTASSI: o.eloutassi@umi.ac.ma

```
sudo pacman -S octave
```

- Build from Source (Optional):
- If you prefer the latest version and it's not available in your package manager, you can build Octave from source:

```
sudo apt install build-essential
sudo apt install gfortran libblas-dev liblapack-dev libreadline-dev
sudo apt install qt5-default
```

- Download the latest source code from the GNU Octave website.
- Extract the downloaded file and navigate to the directory in the terminal:

```
tar -xvf octave-<version>.tar.gz
cd octave-<version>
```

- Run the following commands:

```
./configure
make
sudo make install
```

- Launch Octave:
- You can start Octave by typing `octave` in the terminal.

Verify Installation:

- Type `version` in the Octave command window to check the installed version.

Additional tips

- Documentation and Help:
- Octave has extensive documentation available online. You can access it by typing `doc` in the Octave command window.
- Installing Packages:
- You can install additional packages using the command `pkg install -forge <package_name>` within Octave.
- Updating Octave:
- Keep your installation up to date by checking for updates through your package manager or the official website.

O. ELOUTASSI: o.eloutassi@umi.ac.ma

Install Python

- Download Python:
  - Go to the official Python website.
  - Click on the "Download Python" button (the latest version will be highlighted).
- Run the Installer:
  - Locate the downloaded installer (usually a .exe file for Windows) and double-click it.
  - Important: Check the box that says "Add Python to PATH" before clicking "Install Now".
  - Follow the on-screen instructions to complete the installation.
- Verify Installation:
  - Open a command prompt (Windows) or terminal (macOS/Linux).
  - Type python --version or python3 --version and press Enter. You should see the installed Python version.

Install PyCharm

- Download PyCharm:
  - Visit the JetBrains PyCharm website.
  - Choose the Community edition (free) or Professional edition (paid) based on your needs.
  - Download the installer appropriate for your operating system.
- Run the Installer:
  - For Windows, double-click the downloaded `.exe` file. For macOS, open the `.dmg` file.
  - Follow the installation instructions. You can keep the default settings.
- Launch PyCharm:
  - After installation, open PyCharm from your applications or Start Menu.

Create a new project in Pycharm

- Start a New Project:
  - When PyCharm opens, click on "New Project".
- Configure the Project:
  - Location: Choose a directory for your project.
- Python Interpreter:
  - PyCharm should automatically detect the Python interpreter you installed. If it doesn't, click on the gear icon next to the interpreter dropdown, select "Add...", and then choose "System Interpreter".
  - Navigate to the Python executable (e.g., `C:\Python39\python.exe` on Windows or `/usr/bin/python3` on macOS/Linux).
- Create the Project:
  - Click "Create" to set up your new project.

Set up a virtual environment (optional but recommended)

- Create a Virtual Environment:

O. ELOUTASSI: o.eloutassi@umi.ac.ma

- In the New Project dialog, you can select "New environment using" and choose "Virtualenv".
- Ensure that the base interpreter is set to the Python version you installed.

➢ Activate the Virtual Environment:

- If you created a virtual environment, you can activate it by running the following command in the terminal:

➢ Windows:

.\venv\Scripts\activate

➢ macOS/Linux:

source venv/bin/activate

Install Packages

➢ Using PyCharm:

- Go to File > Settings (or PyCharm > Preferences on macOS).
- Navigate to Project: [your_project_name] > Python Interpreter.
- Click on the + icon to add packages.
- Search for the package you need (e.g., numpy, pandas) and click Install Package.

➢ Using Terminal:

- Open the terminal in PyCharm (bottom left).
- Use pip to install packages:

```
pip install package_name
```

Write and Run a Python Script

➢ Create a New Python File:

- Right-click on the project folder in the Project pane.
- Select New > Python File and name your file (e.g., main.py).
- Write Your Code:
- Open the newly created file and write your Python code.

➢ Run Your Script:

- Right-click anywhere in the editor and select Run 'main' or click the green play button in the top right corner.

Additional Tips

➢ Documentation**:** PyCharm has built-in documentation. You can access it by pressing Ctrl + Q (Windows/Linux) or F1 (macOS) when your cursor is on a function or class.

➢ Version Control: PyCharm supports Git integration. You can initialize a repository and manage your version control directly from the IDE.

➢ Plugins: Explore PyCharm plugins for additional functionality. Go to File > Settings > Plugins to browse and install plugins.

Install OpenCV Using pip

- **Open Command Prompt:**
  - Press `Win + R`, type `cmd`, and hit Enter to open the command prompt.
- **Upgrade pip (Optional but Recommended):**
  - Before installing OpenCV, it's a good idea to ensure that pip is up to date. Run the following command:

```
python -m pip install --upgrade pip
```

- **Install OpenCV:**
  - Use pip to install the OpenCV package by running the following command:

```
pip install opencv-python
```

  - If you need additional functionalities (like working with video files), you can also install the contrib package:

```
pip install opencv-python-headless
```

Verify OpenCV Installation

- **Open Python:**
  - Type `python` in the command prompt and press Enter to start the Python interpreter.
- **Import OpenCV:**
  - In the Python shell, type the following command:

```
import cv2
```

  - If no error occurs, OpenCV has been successfully installed.
- **Check the Version:**
  - You can check the installed version of OpenCV by running:

```
print(cv2.__version__)
```

Setting Up a Simple Project (Optional)

- **Create a New Directory:**
  - Create a new folder for your OpenCV project (e.g., `OpenCV_Project`).
- **Create a Python File:**
  - Inside your project folder, create a new Python file (e.g., `main.py`).
- **Write a Simple OpenCV Script:**
  - Open the `main.py` file in a text editor or IDE and add the following code:

```
import cv2

# Load an image
image = cv2.imread('path_to_your_image.jpg')

# Display the image in a window
cv2.imshow('Image', image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

- **Run Your Script:**
  - Make sure to replace `'path_to_your_image.jpg'` with the actual path to an image on your computer.
  - Run the script from the command prompt:

O. ELOUTASSI: o.eloutassi@umi.ac.ma

```
python main.py
```

Additional tips

- **Use Virtual Environments:** It's a good practice to use virtual environments for your projects. You can create one using `venv`:

```
python -m venv myenv
myenv\Scripts\activate
```

  - Then install OpenCV within this environment.

- **Explore OpenCV Documentation:** Familiarize yourself with the OpenCV documentation for more information on functions and capabilities.
- **Install Additional Libraries:** Depending on your project, you might want to install other libraries like NumPy:

```
pip install numpy
```