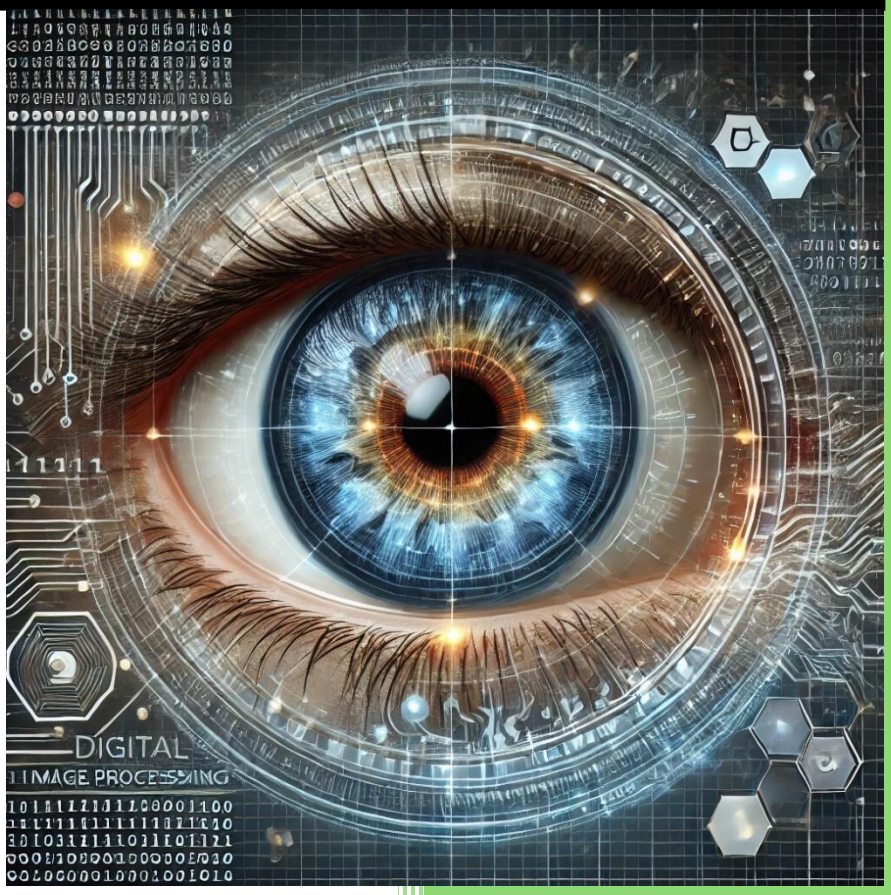
LAB 1 : DIGITAL IMAGE PROCESSING



Maryam Arshad

PAF-IAST (B22F0031AI085)

Lab Report 1:

Digital Image Processing

Lab Overview:

This lab provided a practical introduction to managing software environments and dependencies using Conda. These skills are foundational for effectively implementing Digital Image Processing (DIP) projects, where managing multiple libraries and dependencies is crucial for smooth execution of algorithms. The lab covered three main objectives:

- ✓ Mastering basic Conda commands.
- ✓ Managing virtual environments using the Conda CLI.
- ✓ Creating and managing environments through the Anaconda Navigator GUI.

Tasks and Outputs

Task 1:

Mastering Basic Conda Commands

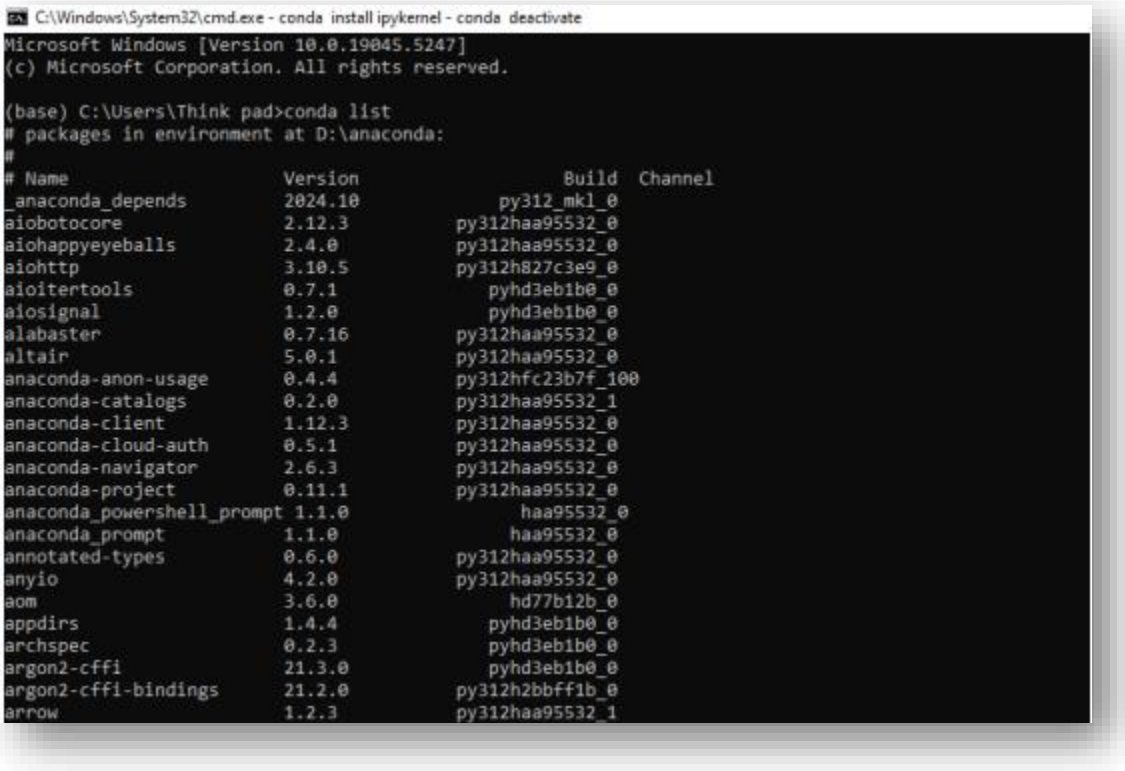
Objective:

Gain hands-on experience with fundamental Conda commands for managing packages in digital image processing projects. These skills ensure that necessary libraries, like OpenCV or scikit-image, can be installed and updated efficiently.

List Installed Packages:

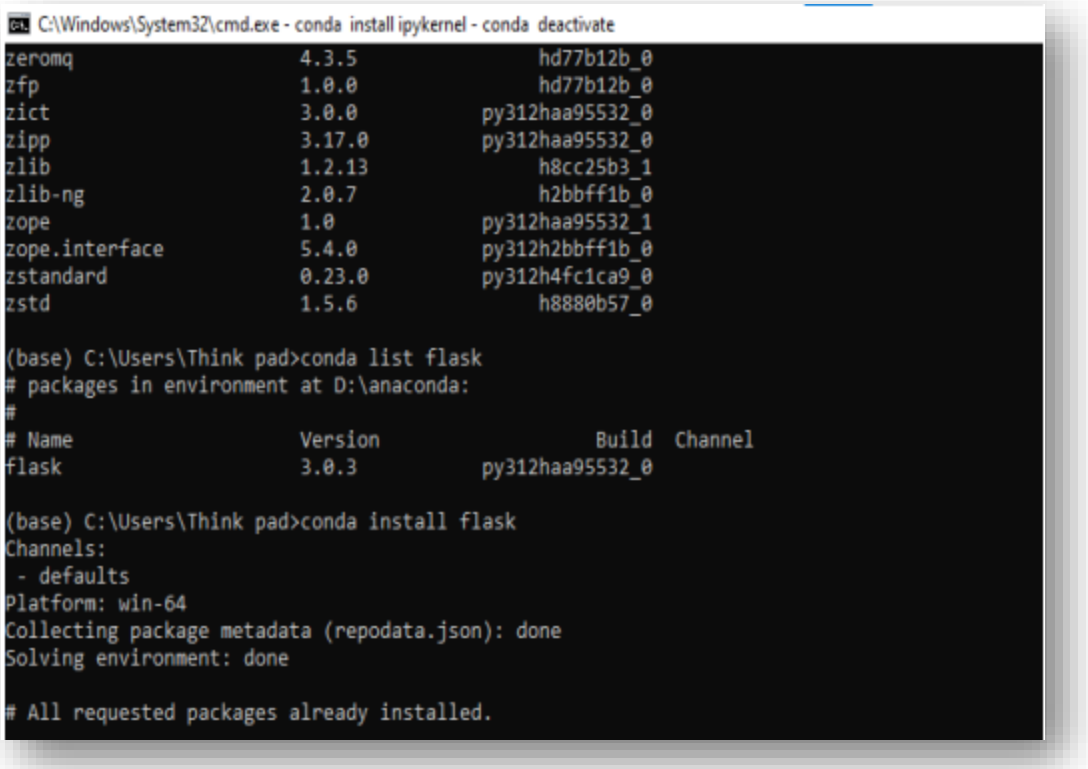
- **Command:** *conda list*

- **Purpose:** Display all currently installed packages in the active environment. This is particularly useful in DIP to verify if image processing libraries like OpenCV or Matplotlib are installed.



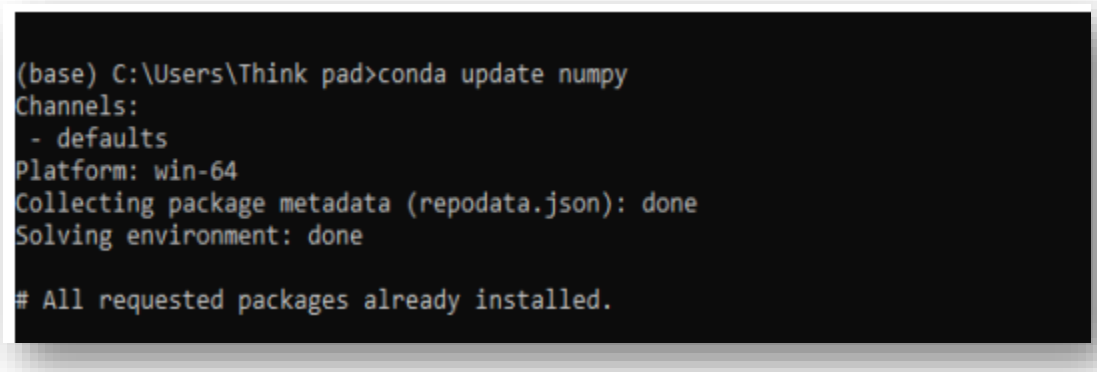
Install Flask Package:

- **Command:** *conda install flask*
- **Purpose:** Practice installing packages. Flask, while a web framework, was used to demonstrate how external dependencies can be added seamlessly, a skill transferable to installing DIP-specific libraries.



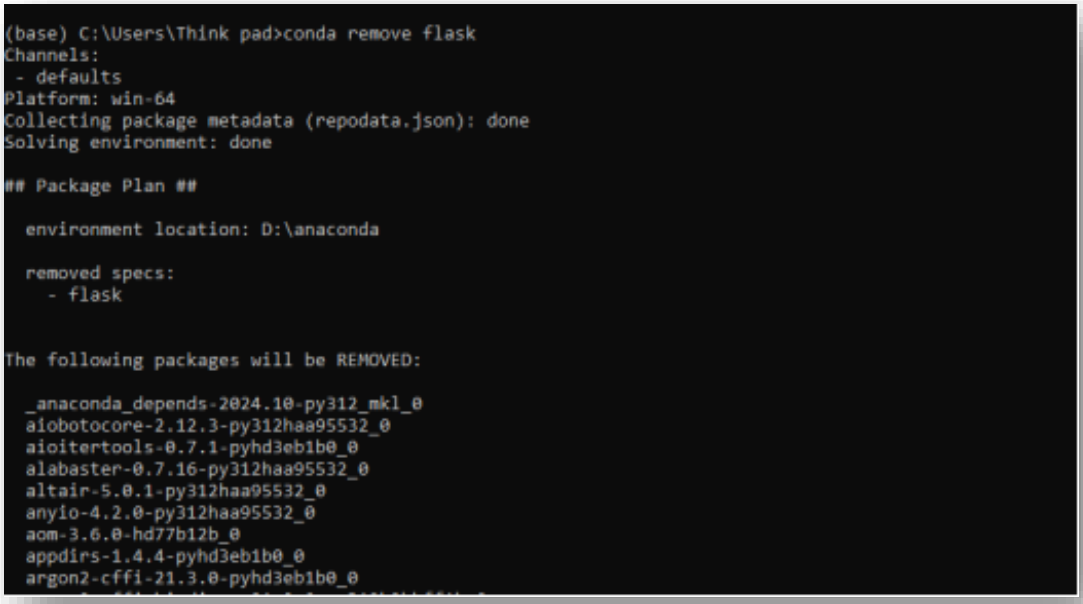
Update a Package:

- **Command:** *conda update <package-name>*
- **Purpose:** Ensure packages stay updated. For DIP, updated libraries often include improved algorithms or compatibility fixes.



Remove a Package:

- **Command:** *conda remove flask*
- **Purpose:**Practice package removal. This helps maintain a clean environment by removing unnecessary or conflicting libraries.



Task 2:

Managing Virtual Environments Using Conda

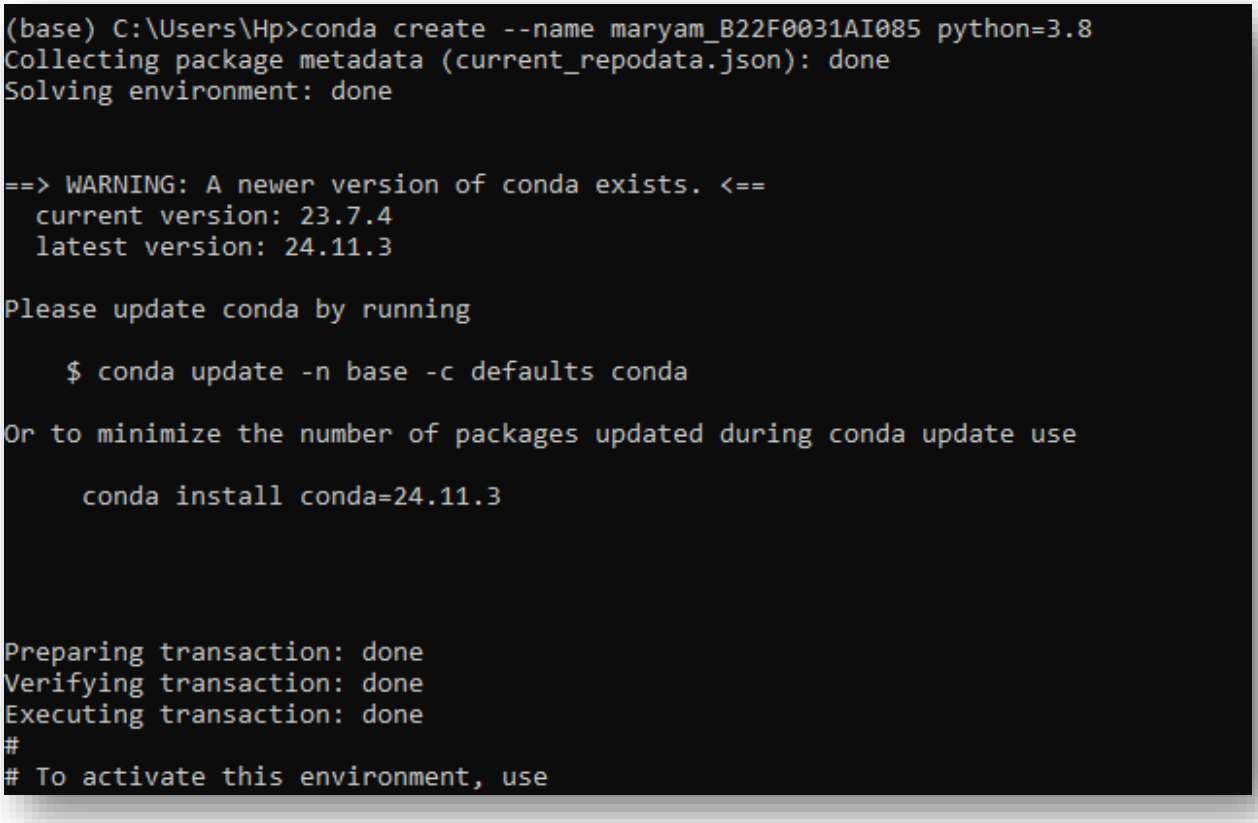
Objective:

Learn to create and manage isolated environments, a critical practice in DIP where projects often require different library versions.

Environment Creation:

- **Command:** *conda create --name Maryam_B22F0031AI085 python=3.8*

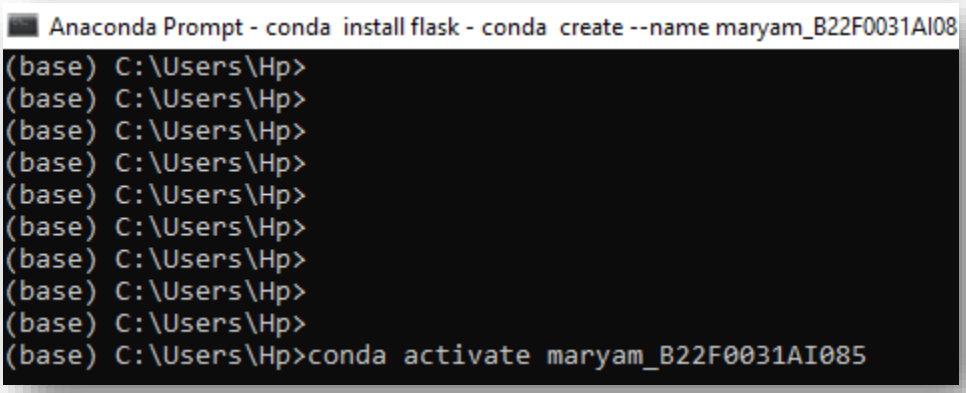
- **Purpose:** Create a virtual environment with Python 3.8, ensuring



compatibility with DIP libraries like TensorFlow or PyTorch.

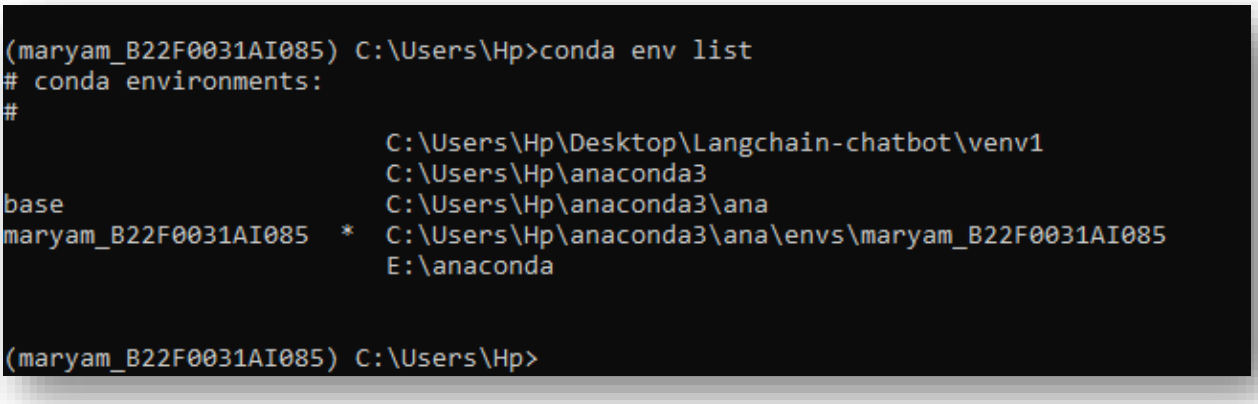
Activating the Environment:

- **Command:** *conda activate Maryam_B22F0031AI085*
- **Purpose:** Switch to the created environment to work on specific DIP tasks.



Listing All Environments:

- **Command:** *conda env list*
- **Purpose:** View all available environments, making it easy to manage multiple DIP projects.



Installing Jupyter:

- **Command:** *conda install jupyter*
- **Purpose:** Add Jupyter Notebook, a popular tool for testing DIP algorithms interactively.

Anaconda Prompt - conda install flask - conda create --name maryam_B22F0031AI085 python-3.8 - conda install jupyter

```
                         C:\Users\Hp\Desktop\Langchain-chatbot\venv1
                         C:\Users\Hp\anaconda3
base                     C:\Users\Hp\anaconda3\ana
maryam_B22F0031AI085  *  C:\Users\Hp\anaconda3\ana\envs\maryam_B22F0031AI085
                         E:\anaconda


(maryam_B22F0031AI085) C:\Users\Hp>
(maryam_B22F0031AI085) C:\Users\Hp>conda install jupyter
Collecting package metadata (current_repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
  current version: 23.7.4
  latest version: 24.11.3

Please update conda by running

    $ conda update -n base -c defaults conda

Or to minimize the number of packages updated during conda update use

     conda install conda=24.11.3



## Package Plan ##

  environment location: C:\Users\Hp\anaconda3\ana\envs\maryam_B22F0031AI085

  added / updated specs:
    - jupyter


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    anyio-4.2.0                |   py38haa95532_0         186 KB
    argon2-cffi-bindings-21.2.0|   py38h2bbff1b_0          36 KB
    async-lru-2.0.4            |   py38haa95532_0          18 KB
    attrs-23.1.0               |   py38haa95532_0         143 KB
    babel-2.11.0               |   py38haa95532_0         6.8 MB
```

```
Proceed ([y]/n)?


Downloading and Extracting Packages
libclang13-14.0.6    | 22.6 MB   | #####################################################################################1                                      |  69%
beautifulsoup4-4.12. | 214 KB    | ############################################################################################################################ | 100%
ipython-8.12.2       | 1.1 MB    | ############################################################################################################################ | 100%
exceptiongroup-1.2.0 | 30 KB     | ############################################################################################################################ | 100%
pygments-2.15.1      | 1.7 MB    | ############################################################################################################################ | 100%
httpx-0.27.0         | 194 KB    | ############################################################################################################################ | 100%
typing_extensions-4. | 61 KB     | ############################################################################################################################ | 100%
pywin32-305          | 10.7 MB   | ############################################################################################################################ | 100%
async-lru-2.0.4      | 18 KB     | ############################################################################################################################ | 100%

Preparing transaction: done
Verifying transaction: done
Executing transaction: |
Warning: using menuinst v1 shortcuts
Please update menuinst in the base environment and reinstall notebook.

done
```

Adding to Jupyter Kernel:

- **Command**: *python -m ipykernel install --user --name=Maryam_B22F0031AI085*
- **Purpose:** Integrate the environment into Jupyter, simplifying workflows for DIP experimentation.

```
(maryam_B22F0031AI085) C:\Users\Hp>
(maryam_B22F0031AI085) C:\Users\Hp>python -m ipykernel install --user --name=maryam_B22F0031AI085
Installed kernelspec maryam_B22F0031AI085 in C:\Users\Hp\AppData\Roaming\jupyter\kernels\maryam_b22f0031ai085
```

Deactivating the Environment:

- **Command:** *conda deactivate*
- **Purpose:** Exit the environment, ensuring no unintentional modifications.

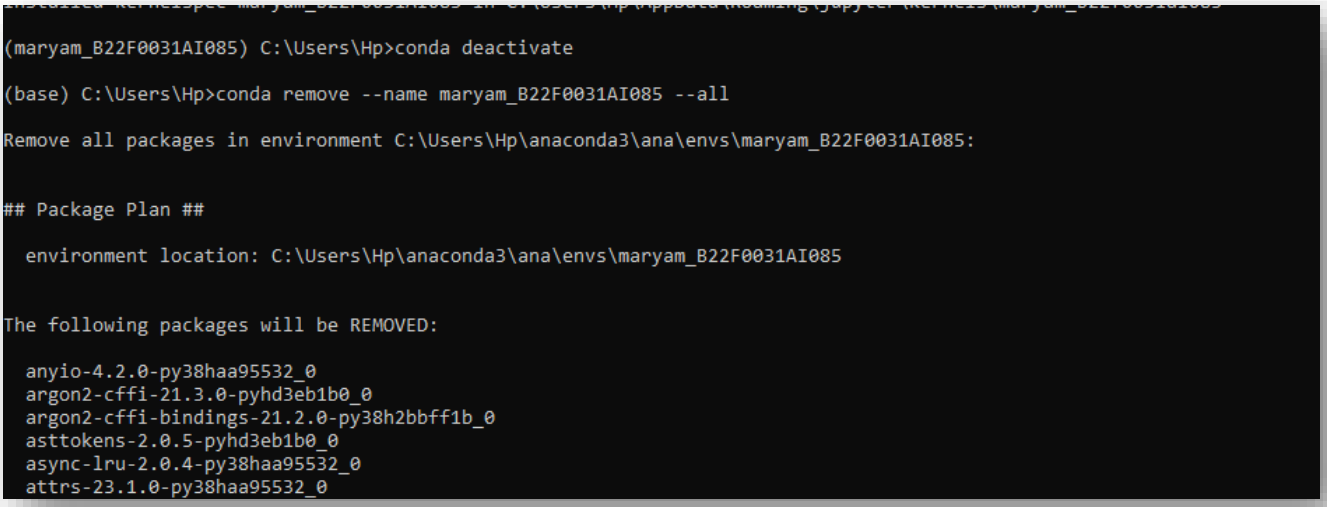
```
(maryam_B22F0031AI085) C:\Users\Hp>
(maryam_B22F0031AI085) C:\Users\Hp>
(maryam_B22F0031AI085) C:\Users\Hp>
(maryam_B22F0031AI085) C:\Users\Hp>
(maryam_B22F0031AI085) C:\Users\Hp>python -m ipykernel install --user --name=maryam_B22F0031AI085
Installed kernelspec maryam_B22F0031AI085 in C:\Users\Hp\AppData\Roaming\jupyter\kernels\maryam_b22f0031ai085

(maryam_B22F0031AI085) C:\Users\Hp>conda deactivate

(base) C:\Users\Hp>
```

Removing the Environment:

- **Command:** *conda remove --name Maryam_B22F0031AI085 --all*
- **Purpose:** Delete the environment when it is no longer needed, freeing up system resources.



Task 3:

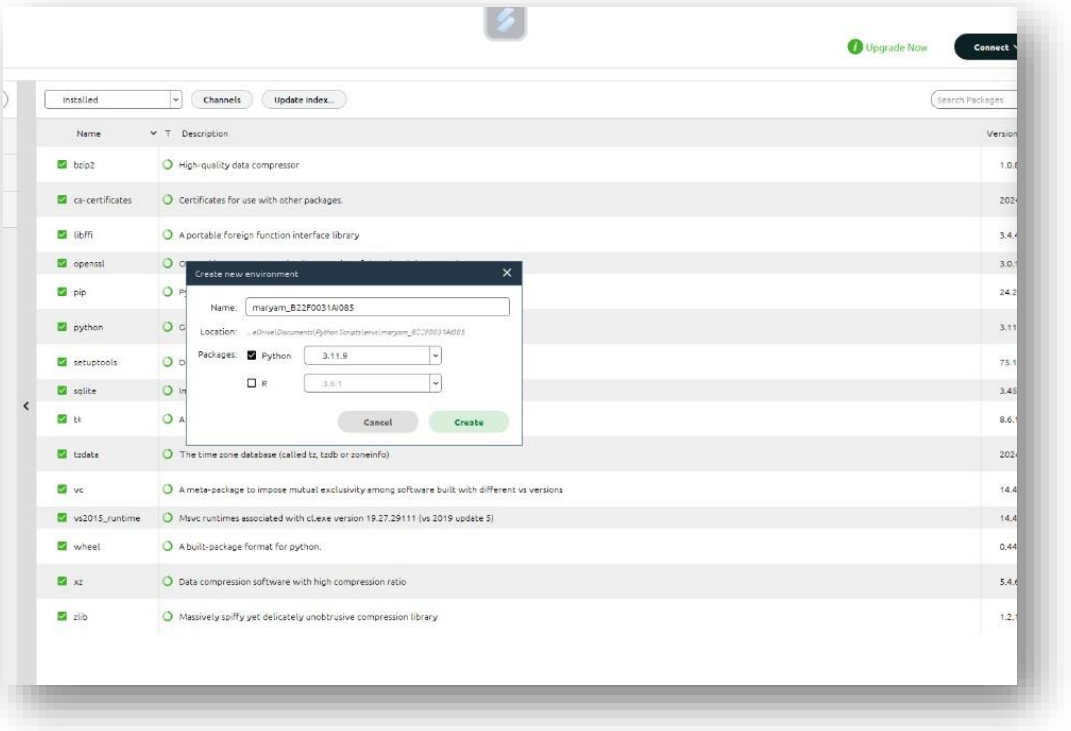
Creating Virtual Environments via Anaconda Navigator

Objective:

Explore the graphical interface for managing environments, offering a user-friendly alternative to the command line.

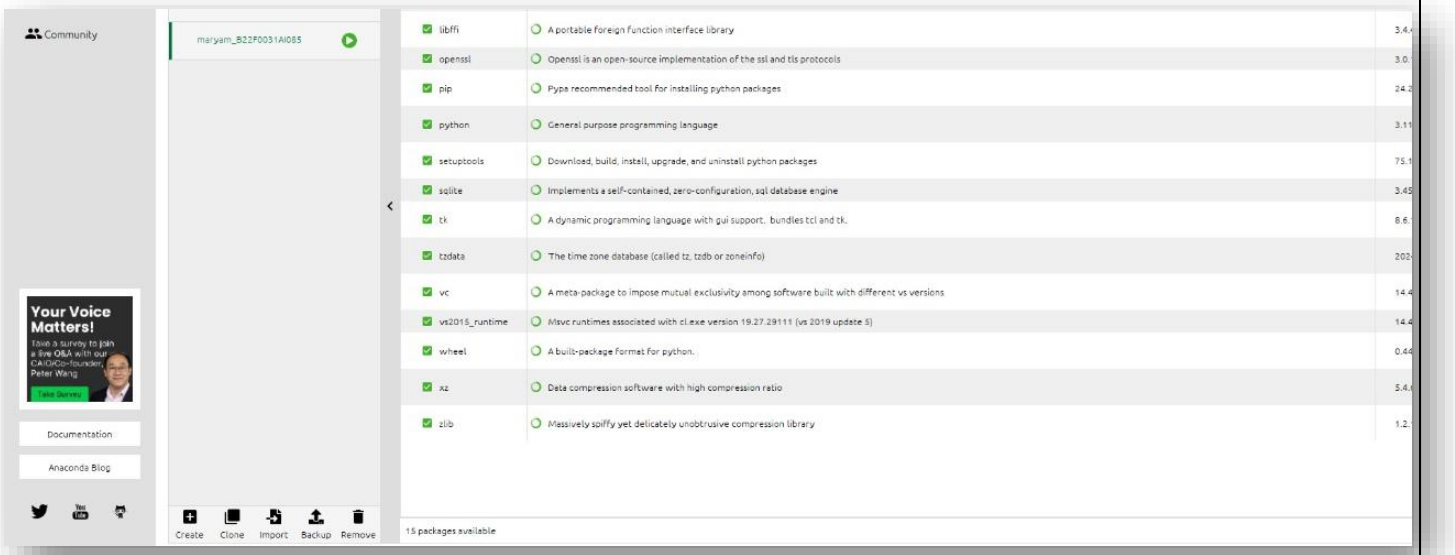
Opening Anaconda Navigator:

- **Purpose:** Access a visual tool for managing environments and packages.



Creating a New Environment:

- **Name:** *maryam_B22F0031AI085*
- **Python Version:** *3.8*
- **Purpose**: Create an environment similar to Task 2 but through a GUI, demonstrating multiple approaches to managing DIP workflows.



Summary:

This lab session was an introduction to Conda's capabilities for managing packages and environments. The tasks reinforced dependency management, especially important in digital image processing projects where diverse libraries like **OpenCV, PIL**, and **TensorFlow** may be required.

Conclusion:

This lab emphasized the significance of tools like **Conda** and **Anaconda Navigator**, bridging the gap between theoretical knowledge and practical application.