

Image Processing and Computer Vision

Introduction to Image Processing and Image Acquisition

Sanath Thilakarathna

BSc Eng (Hons), MSc (Reading)

CINEC Campus

January 29, 2026



Why Python for Image Processing?

Definition

Python is a high-level programming language widely used for scientific computing and computer vision.

- Simple and readable syntax
- Strong numerical and scientific libraries
- Widely used in academia and industry

Why OpenCV?

Definition

OpenCV is an open-source library for image processing and computer vision.

- Real-time capable
- Industry-standard library
- Supports Python and C++

Why Use Anaconda?

Definition

Anaconda is a Python distribution designed for scientific and engineering applications.

- Simplified package management
- Environment isolation
- Avoids dependency conflicts

Mandatory Instruction

Important

All students must use Anaconda and create a separate environment for this module.

Warning

Do NOT install OpenCV or libraries in the base environment.

What Is a Python Environment?

Definition

A Python environment is an isolated workspace with its own Python version and libraries.

- Prevents version conflicts
- Ensures reproducibility
- Required for professional workflows

Important

Each student must create one dedicated environment for this module.

- Environment name: ee4216
- Used for all labs and projects

Step 1: Install Anaconda

Example

- Download Anaconda (Python 3.x)
- Install with default settings
- Available for Windows, Linux, macOS

Step 2: Create the EE4216 Environment

Example

```
conda create --name ee4216 python=3.10
```

Note

Press y when prompted.

Step 3: Activate the Environment

Example

```
conda activate ee4216
```

Note

The environment name should appear in the command prompt.

Important

The following libraries must be installed inside the ee4216 environment.

- **NumPy** – numerical arrays and matrix operations
- **OpenCV** – image processing and computer vision
- **Matplotlib** – image display and plotting
- **SciPy** – scientific and signal processing utilities

Install Required Libraries

Example

```
conda install numpy matplotlib scipy
```

Example

```
pip install opencv-python
```

Warning

Always activate the environment before installing packages.

Why Jupyter Notebook?

Definition

Jupyter Notebook provides an interactive environment for writing, executing, and visualizing Python code.

- Immediate visual feedback
- Ideal for image processing experiments
- Widely used in research and industry

Install Jupyter and Kernel Support

Important

Jupyter Notebook and kernel support must be installed in the EE4216 environment.

Example

```
conda install jupyter ipykernel
```

Register the EE4216 Kernel

Example

```
python -m ipykernel install -user -name ee4216 -display-name "EE4216  
(Python) "
```

Note

This makes the EE4216 environment available inside Jupyter Notebook.

Example

- Launch Jupyter Notebook
- Create a new notebook
- Select kernel: EE4216 (Python)

Important

All lab work must use this kernel.

Mandatory Rules

Warning

- Do not use system Python
- Do not use the base environment
- Do not mix environments