Computer Vision

CVI620

Session 1

Ellie Azizi

AI and tech enthusiast

Machine Learning Engineer at Sinn Studio

Graduated from Ontario Tech University in 2023

Thought programming

Bachelor in Computer Engineering

Started as a Computer Vision Engineer

Spend my leisure time studying and develop skills

Sometimes I write poems

Play video games

Habit of working out

Initial Words

2 sections (image processing, ML)

Roadmap

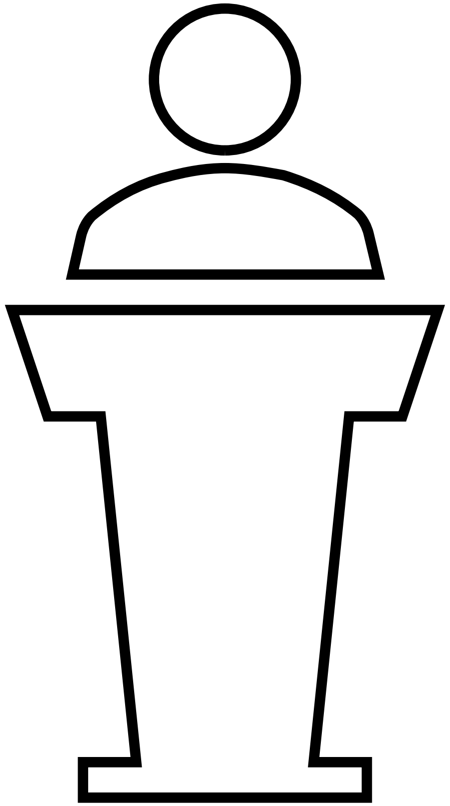
Skills and tools

Hands-on and theoretical

Python - OpenCV

Mode of Instruction

- Mode: Online lecture, quizzes, and hands-on activity
- Hours per week: 3.5
- Tuesdays and Thursdays, 3:20 PM – 5:05 PM
- Room configurations: online sessions
- Typical scheduling pattern: Summer term



Syllabus and Course Description

As cameras are becoming a popular feature of standard PCs and mobile devices, it is important to be able to extract useful information from this data. This course covers simple computer vision concepts, such as capturing digital images and video, applying simple and arithmetic changes to images(e.g. crop, rotate, enhance color), and extracting information from images (e.g. finding or counting objects). Also, concepts such as tracking objects in a video sequence are introduced. Students will work on a project targeting a computer vision application such as face recognition, security systems, medical imaging, robotics, and others.

Modes of Evaluation

Assignments	15%
Labs	20%
Quizzes	15%
Midterm Exam	15%
Final Exam	20%
Final Project	15%

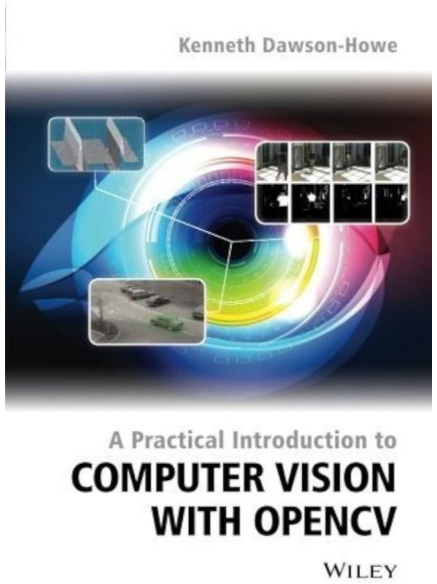
Tentative and might change according to the semester progress with pre-announcement

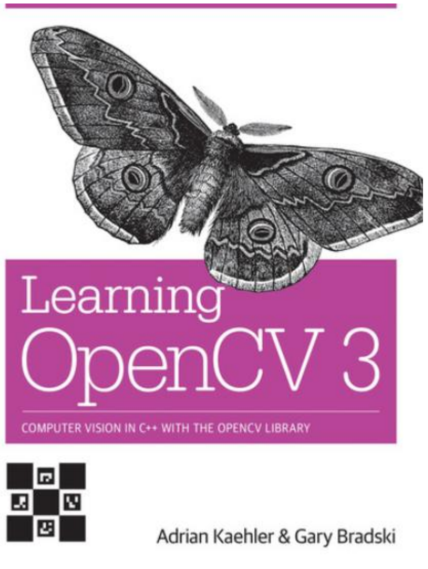
- All exams and some of the assignments in this course will be followed by a brief oral Q&A session.

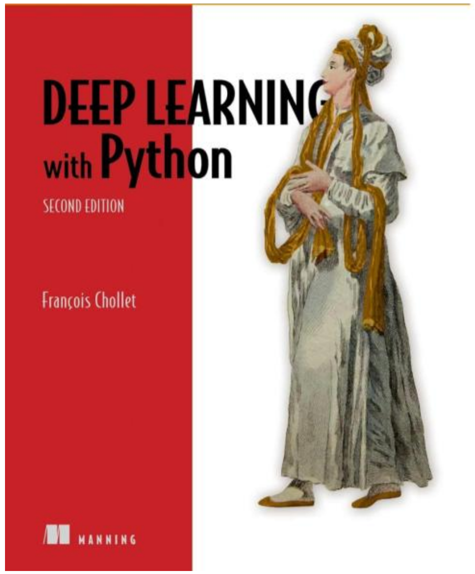
To obtain a credit in this subject, a student must:

- Achieve 50% or better of the overall course.
- Achieve 50% or better of the quizzes and exams.
- Successfully complete the final project

Reference Material




Learning
OpenCV 3
COMPUTER VISION IN C++ WITH THE OPENCV LIBRARY
Adrian Kaehler & Gary Bradski



What field are you planning to pursue in the future?
Which semester are you currently in?

https://forms.office.com/r/AAxvdcmBxG?origin=lprLink

Image Processing

- Signal processing
- Image processing
- Classical Machine Learning - Before 2012
- Deep Learning - After 2012

History

- 1960s: The Birth of Digital Image Processing
- Early Development:
  - Image processing emerged as a subfield of computer science and electrical engineering.
  - First applications were related to space exploration.
  - NASA used digital image processing to enhance images of the Moon during the Apollo missions.

History

- 1970s: Advancements in Medical and Remote Sensing
- 1980s: Rise of Personal Computing
- 1990s: Multimedia Revolution
- 2000s: Integration with Technology
- 2010s - Present: Image Processing Meets AI

Definition

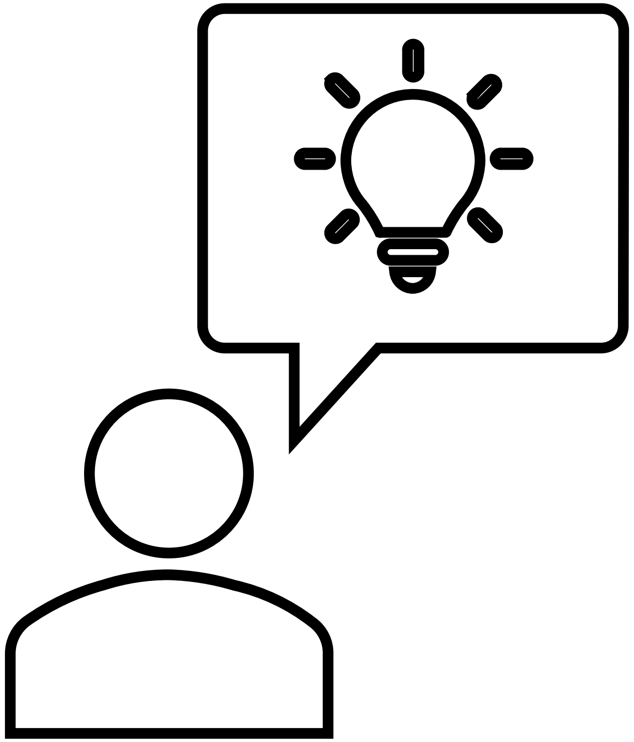
Techniques for manipulating and analyzing images to extract information or enhance quality

Edge detection, object segmentation

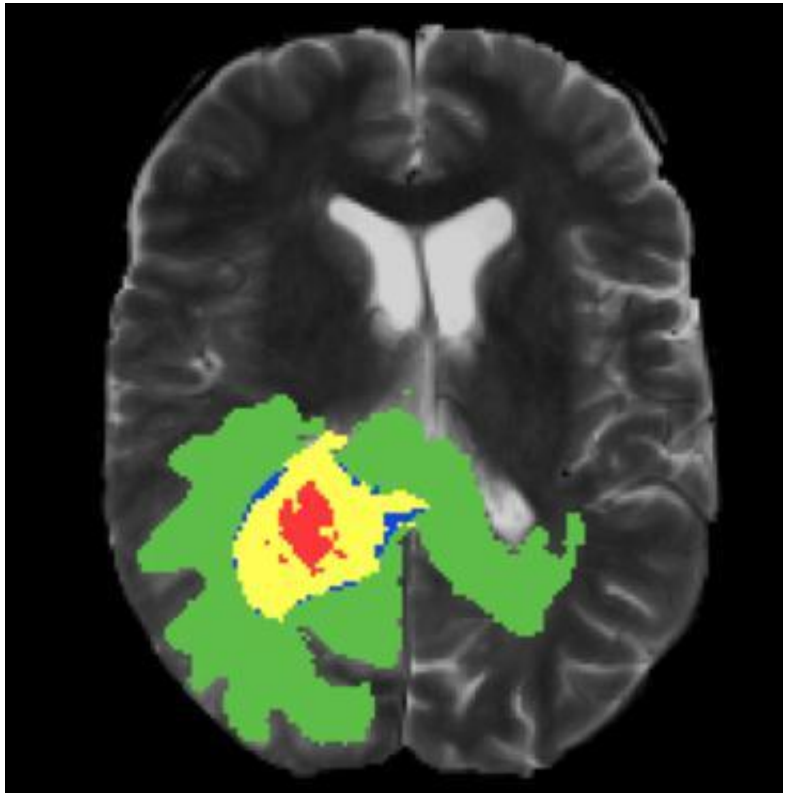
Why Image Processing?

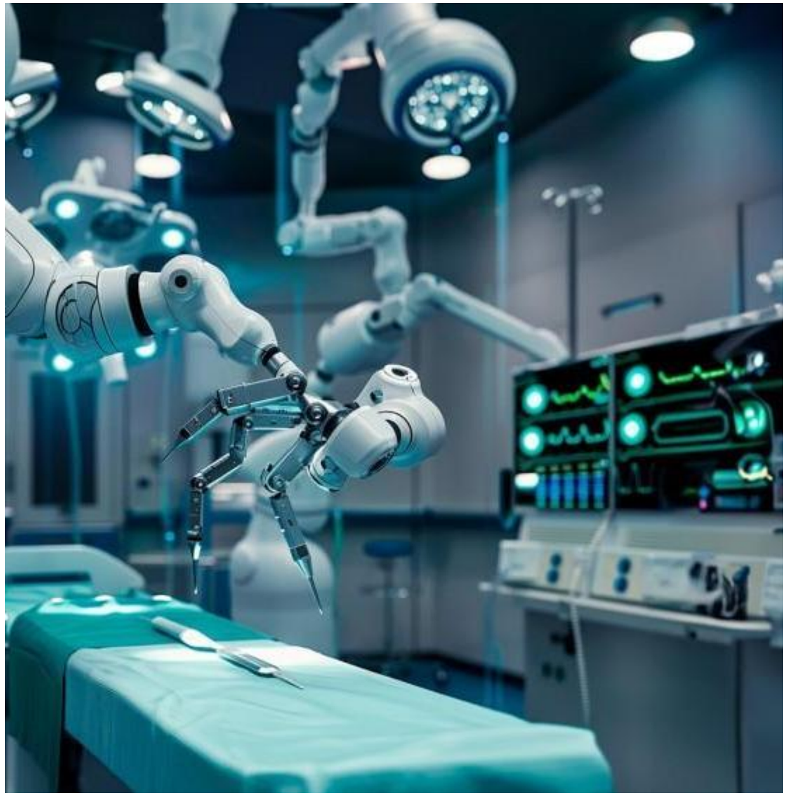
Automates and enhances human-like vision tasks.

Drives innovation across industries.



Medical Imaging



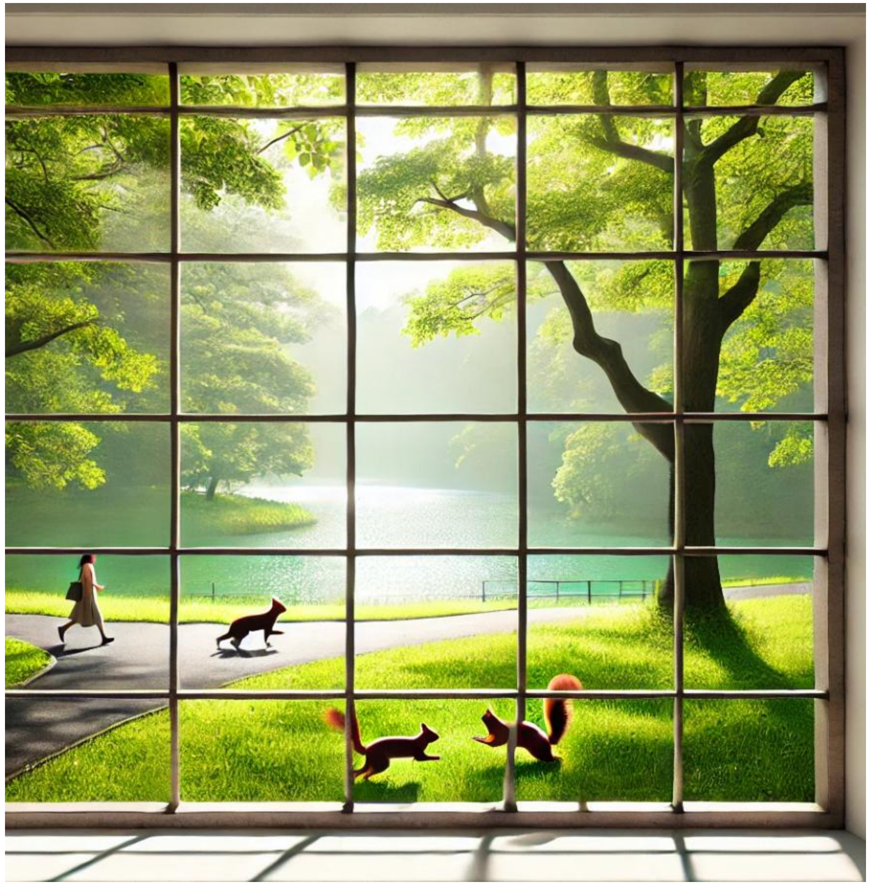


Photography

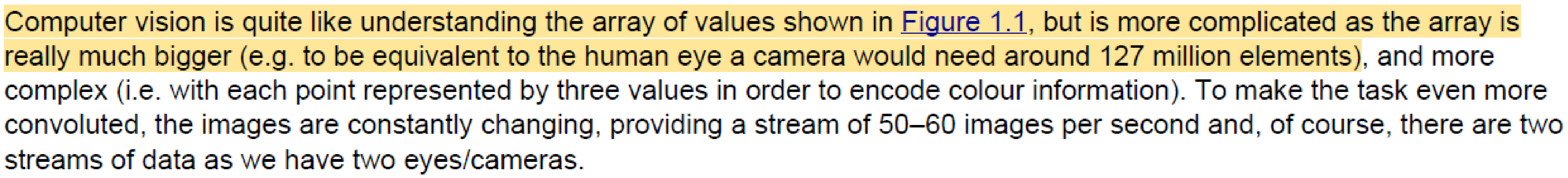
Security and Surveillance

Education

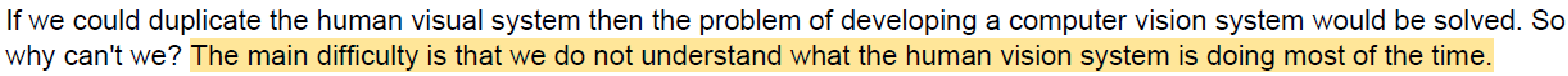
- As I look out of my window, I see grass and trees, gently swaying in the wind, with a lake beyond … An asphalt path leads down through the trees to the lake and two squirrels are chasing each other to and fro across it, ignoring the woman coming up the path …
- This is the scene I experience, a world of objects with background, acted upon and sometimes acting and interacting in events. I have no problem seeing and hearing and smelling and feeling all these things because they affect my senses directly and they make up the real world.
- Or do they? I can look again and notice things I missed before or see the scene in new ways. There is a white wall framing the window I am looking through and the window in fact fills less of my field of view that the wall, but I did not even notice the wall at first, and my impression was that the scene through the window was a panorama right across in front of me. There are metal bars dividing the window into squares and the glass is obscured with dust and spots but for me the view seems complete and un-obscured. The ‘grass’ is patches of colour ranging from nearly white in the bright sun to nearly black in the shade but I ‘saw’ green grass in light and shade. Other changing greenish shapes were for me permanent leafy branches moved by a wind I neither saw nor felt, and two constantly varying grey shapes were squirrels moving with a purpose. Another shape increasing in size and changing in position was an approaching woman. (*Wilding, 1983*)



Human Vision VS Digital Image

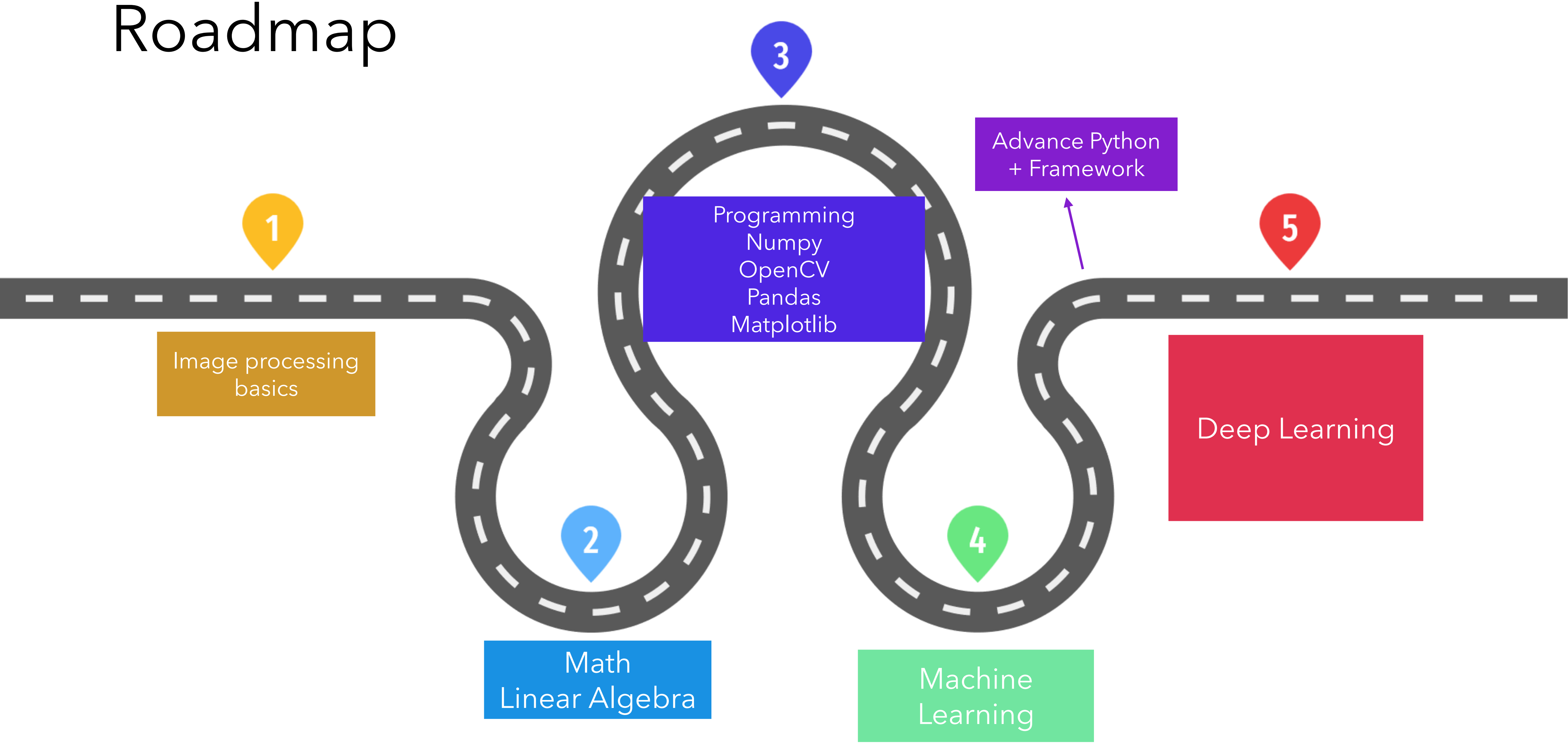


Computer vision is quite like understanding the array of values shown in Figure 1.1, but is more complicated as the array is really much bigger (e.g. to be equivalent to the human eye a camera would need around 127 million elements), and more complex (i.e. with each point represented by three values in order to encode colour information). To make the task even more convoluted, the images are constantly changing, providing a stream of 50–60 images per second and, of course, there are two streams of data as we have two eyes/cameras.



If we could duplicate the human visual system then the problem of developing a computer vision system would be solved. So why can't we? The main difficulty is that we do not understand what the human vision system is doing most of the time.

Roadmap



1

Image processing basics

2

Math
Linear Algebra

3

Programming
Numpy
OpenCV
Pandas
Matplotlib

Advance Python + Framework

4

Machine Learning

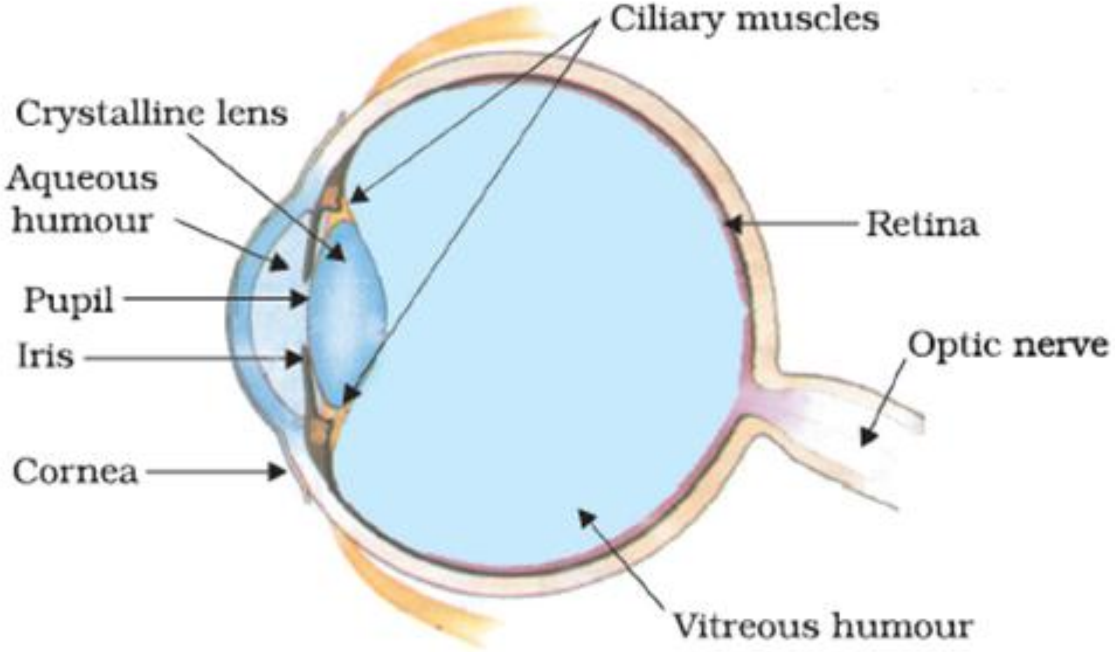
5

Deep Learning

Terminology

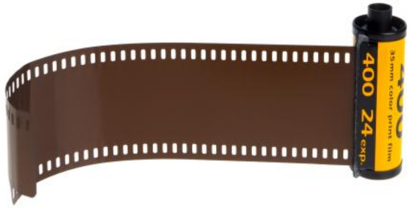
- Image Processing: enhancing and manipulating images to improve quality or extract features.
  - Examples: Noise reduction, contrast enhancement, resizing.
  - Tools: Traditional algorithms like filtering, edge detection, histogram equalization.
- Computer Vision: goes beyond image processing to interpret and understand visual data.
  - Examples: Object detection, facial recognition, image segmentation.
  - Tools: AI/ML techniques, deep learning (e.g., CNNs), frameworks like OpenCV, TensorFlow.

Image

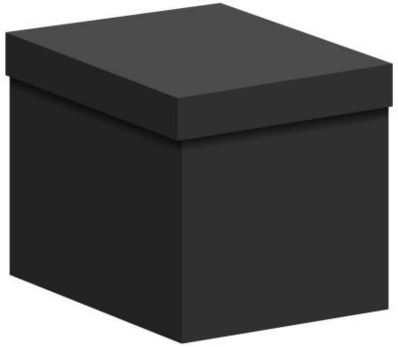


Ciliary muscles
Crystalline lens
Aqueous humour
Pupil
Iris
Cornea
Retina
Optic nerve
Vitreous humour

Capturing Image



film, atomic properties

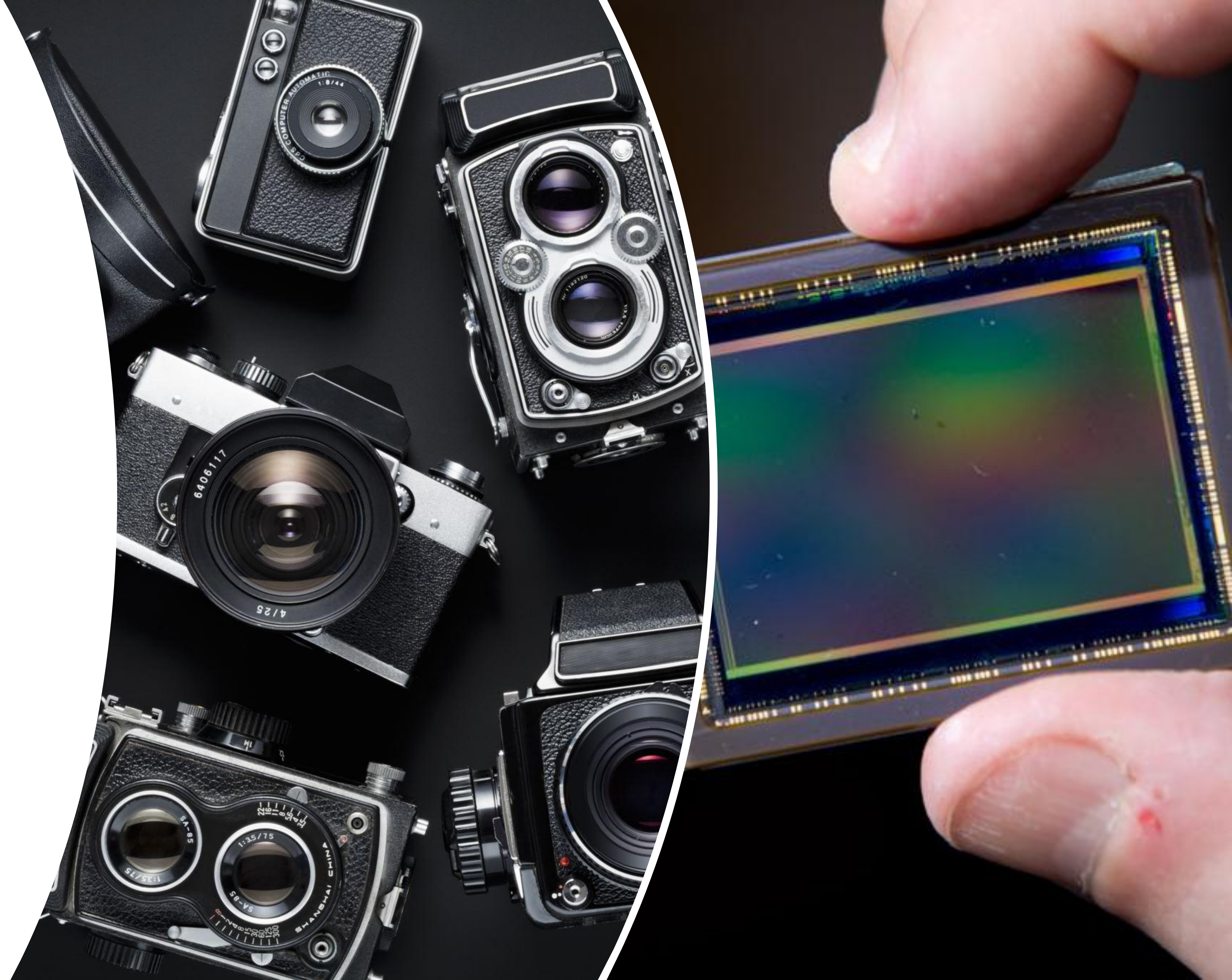


black box

cameras with options

Chemical reactions

Digital Cameras



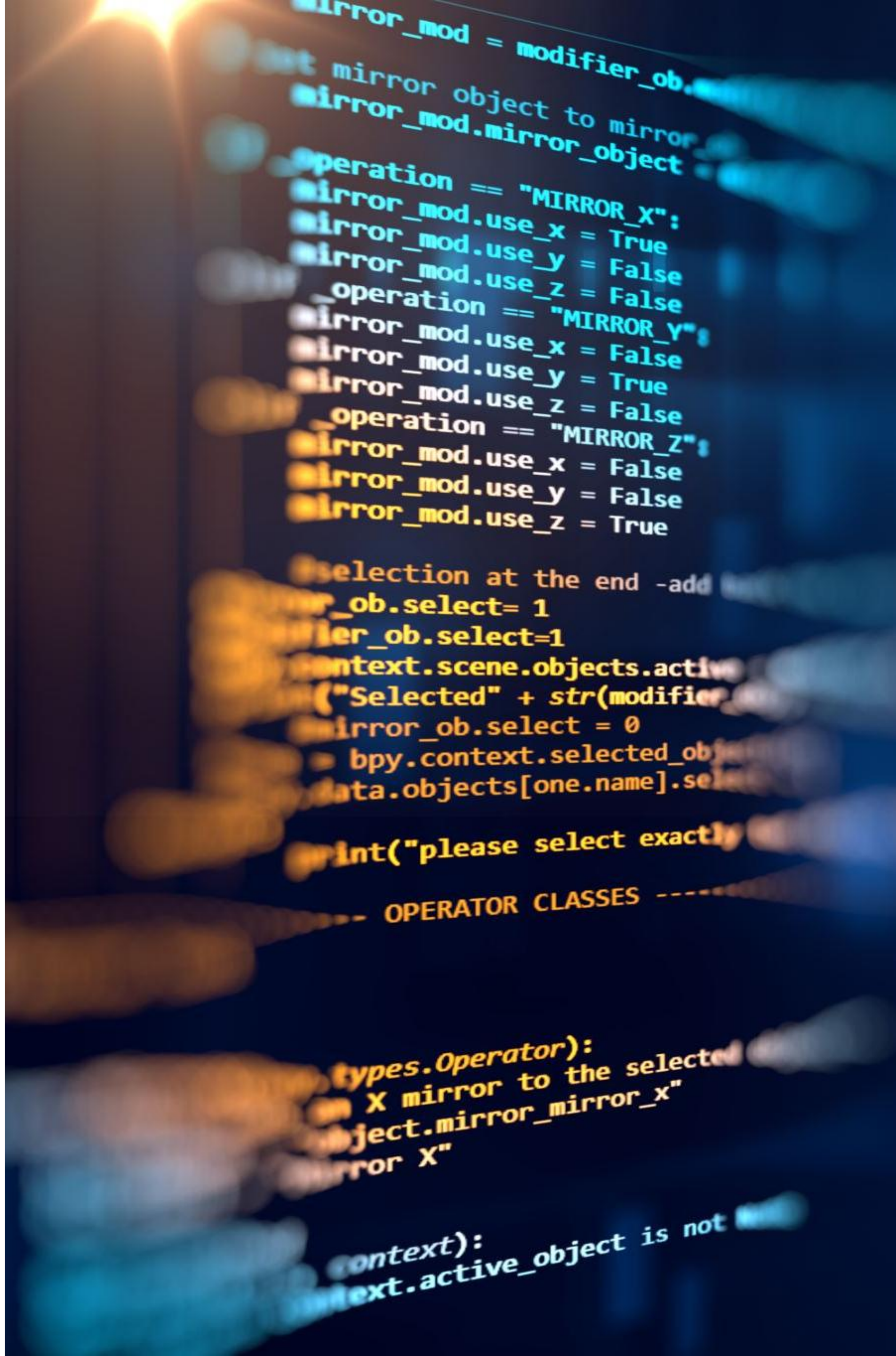
Install Prerequisite

1. Install VSCode
2. Install Python = download + add to path
3. Add Python extension in VSCode
4. Install virtualenv
5. Create virtual environment
6. Install numpy
7. Install OpenCV
8. Install matplotlib
9. Install ipykernel
10. Add jupyter notebook extension
11. Add ipykernel to kernels
12. python -m ipykernel install --user --name=<venv_name> --display-name "Python <venv_name>"
13. Setup Git

VSCode

- A lightweight, open-source code editor developed by Microsoft.
- Supports multiple programming languages and platforms.
- Offers a rich ecosystem of extensions and integrations.

1. For installation go to: https://code.visualstudio.com/
2. Download based on your system spec



```
irror_mod = modifier_ob.
mirror object to mirror
irror_mod.mirror_object
peration == "MIRROR_X":
irror_mod.use_x = True
irror_mod.use_y = False
irror_mod.use_z = False
_operation == "MIRROR_Y":
irror_mod.use_x = False
irror_mod.use_y = True
irror_mod.use_z = False
_operation == "MIRROR_Z":
irror_mod.use_x = False
irror_mod.use_y = False
irror_mod.use_z = True

selection at the end -add
_ob.select= 1
er_ob.select=1
ntext.scene.objects.active
("Selected" + str(modifier
irror_ob.select = 0
 bpy.context.selected_ob
ata.objects[one.name].se

int("please select exactly

- OPERATOR CLASSES ----

types.Operator):
 X mirror to the selected
ject.mirror_mirror_x"
rror X"

ontext):
ext.active_object is not
```

Python

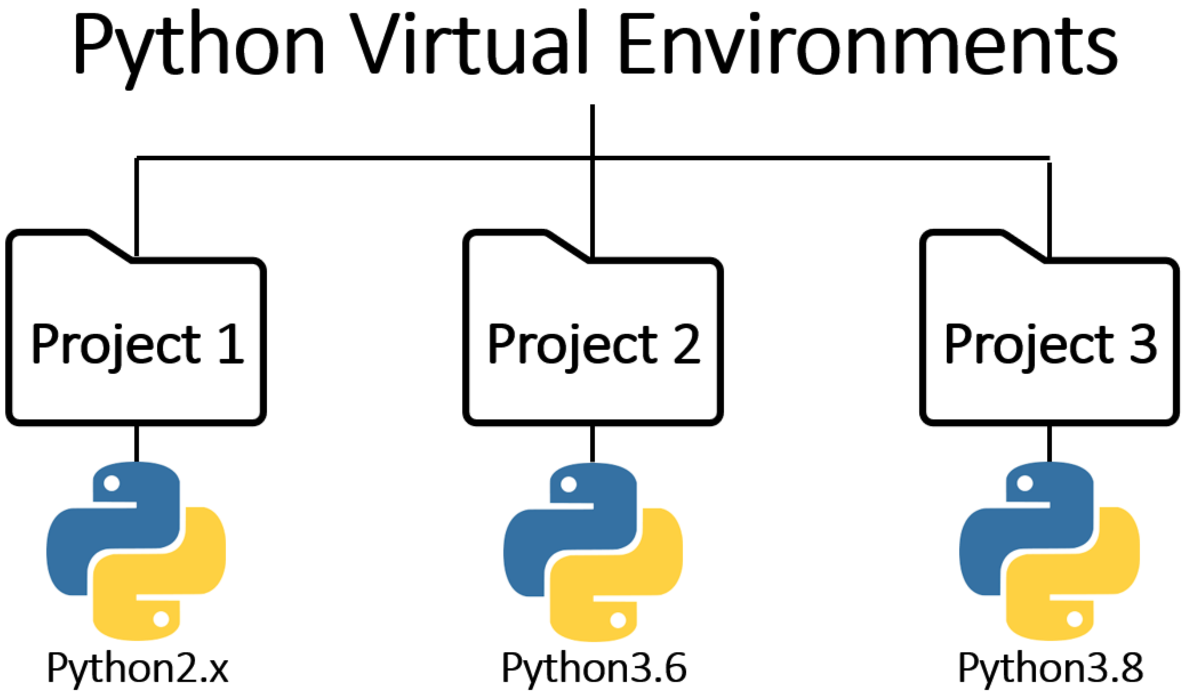
- A versatile, high-level, and interpreted programming language.
- Intuitive syntax resembling natural language.
- Rich standard library and thousands of third-party libraries.

1. For installation go to: https://www.python.org/downloads/
2. Download the desired version
3. Follow the instructions in .exe file
4. Add python to your environment variables
5. Add python extension in VSCode

Virualenv

- A tool to create isolated Python environments.
- Prevents dependency conflicts between projects.

- Avoid global package installation issues.
- Manage different versions of libraries for multiple projects.
- Enable easier collaboration and deployment.



Virtualenv

- Installation

1. Open terminal in VScode
2. Write: pip install virtualenv
3. Check the installation: pip list   or   virtualenv --version

(pip is the standard package manager for Python, used to install and manage libraries and dependencies)

- Activate and use

1. *<first go to the address of your project or wherever you want the virtual env to be created> virtualenv -p <path to python version we want to use> <name>*
2. activate and use : <path to project> <env_name> \Scripts\activate
3. Or cntrl+shift+p in VSCode and choose your virtualenv as python interpreter

Numpy

- A powerful library for numerical computations in Python.
- Provides support for multi-dimensional arrays and matrices.
- Offers a wide range of mathematical functions to operate on arrays.

- For installation: `pip install numpy`
- Make sure you are installing at the right virtualenv

Matplotlib

- A popular Python library for creating static, interactive, and animated visualizations.

- Offers a wide range of plotting capabilities, including line plots, bar charts, histograms, and more.

- For installation: `pip install matplotlib`

Jupyter Notebook

- An open-source web-based tool for interactive computing

- pip install ipykernel
- Add jupyternotebook extension in VSCode

OpenCV

- An open-source library for computer vision and image processing tasks.
- Developed firs for testing CPUs. Images had high computational costs.
- Written in C++ with bindings for Python, Java, and other languages
- C++ -> very efficient

- pip install opencv-python

Gary Bradski