**Week 1: Foundations of Computer Vision & Image Processing**

**Goal**: Understand how to read, manipulate, and analyze images using OpenCV.

**1. Set Up Your Environment**

**Tasks**:

* Install **Miniforge** (ARM-native Conda):

[Download Link](https://github.com/conda-forge/miniforge)

* Create a new Python environment:

conda create -n cv101 python=3.10

conda activate cv101

* Install OpenCV and Jupyter:

pip install opencv-python jupyterlab matplotlib

**2. Watch Introductory Videos**

**Tasks**:

* Watch this short intro:

[What is Computer Vision? (10 min)](https://www.youtube.com/watch?v=2ePf9rue1Ao)

* Start this longer hands-on tutorial:

[OpenCV Crash Course (FreeCodeCamp – 4 hours)](https://www.youtube.com/watch?v=oXlwWbU8l2o)

Aim to finish the first 60–90 minutes this week

**3. Hands-On Practice in Jupyter**

**Create a cv101-week1.ipynb notebook** and complete the following:

**Tasks**:

* Load and display an image:

import cv2

import matplotlib.pyplot as plt

img = cv2.imread('your\_image.jpg')

img\_rgb = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

plt.imshow(img\_rgb)

plt.axis('off')

plt.show()

* Convert the image to grayscale
* Resize and crop the image
* Apply a Gaussian blur
* Detect edges using Canny edge detection
* Draw shapes or text on an image using OpenCV

**4. Mini Project: Image Filter Explorer**

**Task**:

* Create a Python script or notebook to apply a series of filters (blur, grayscale, edge detection) on any image you upload
* Display all versions side-by-side using matplotlib.pyplot.subplot

**5. Document What You Learned**

**Tasks**:

* Write short notes in your notebook or a Markdown file:
  + What is an image in computer vision?
  + What is the difference between BGR and RGB?
  + What is edge detection and why is it useful?

**6. Stretch Goals (Optional)**

**Tasks**:

* Try real-time webcam capture with OpenCV:

*(might need permissions or may not work on Mac depending on camera access)*

cap = cv2.VideoCapture(0)

while True:

ret, frame = cap.read()

cv2.imshow("Live", frame)

if cv2.waitKey(1) & 0xFF == ord('q'):

break

cap.release()

cv2.destroyAllWindows()