Assignment 2:
Python Programming for Machine Learning and Computer Vision

Objective

The objective of this assignment is to assess your ability to implement Machine Learning (ML) techniques in Python, specifically for Computer Vision tasks. You will work with image datasets, apply fundamental image processing techniques, and implement deep learning models for classification and object detection.
Note: Look at Submission Guidelines carefully

Section A: Image Processing Basics

1. Load an image using OpenCV and perform the following operations:
   - Convert it to grayscale
   - Apply Gaussian blur
   - Detect edges using Canny edge detection
2. Perform image transformations such as rotation, scaling, and flipping using OpenCV. Display the original and transformed images.

Section B: Image Classification using CNN

Task: Implement a CNN for Handwritten Digit Recognition

Instructions:

- Use TensorFlow/Keras and train a Convolutional Neural Network (CNN) on the MNIST dataset(Fetch from ML course code or online).
- Split the dataset into training and testing sets.
- Design a CNN with at least one convolutional layer, one pooling layer, and a dense output layer.
- Train the model and evaluate its performance using accuracy and loss metrics.
- Visualize training loss and accuracy using Matplotlib.
- Submit your Python code along with a brief explanation of the steps involved.

Section C: Object Detection using Pre-Trained Models

1. Use a pre-trained model such as YOLO, SSD, or Faster R-CNN to detect objects in an image of your choice.
2. Load the model using OpenCV or TensorFlow/Keras.
3. Apply object detection to at least three images (maybe a clumsy room or a forest) and visualize the results with bounding boxes.
4. Provide a brief explanation of how the model works and its applications, as a text block.

Section D: Detect Triangle inside a Circle

1. Use OpenCV API to detect shapes within an image, here triangles and circles.
2. Implement logic to detect whether a triangle lies within another circle, if so, mark(with box or circle) both the circle and traingle. A triangle may lie within multiple circles.
3. Submit atleast 3 images(testcases) validating your implementation.

Submission Guidelines

- Submit your code in a single Jupyter Notebook.
- Include Text corresponding to each code block describing your unique way of implementation.
- Once again, fork the repo: `https://github.com/rakshaksoftware/Assignments_2025.git`
- Create a Conda environment using `conda env create -f environment.yml` and use it to run your code. That is, select this (`cv_ml_assignment`) conda environment when selecting Python Kernel inside the Jupyter Notebook, don't run the Notebook on your base python installation.
- Add your code to `ML_Assignments/<your_name>.ipynb`.
- Deadline: 25th March, Tuesday

Good luck!

Anaconda and Miniconda: Overview and Installation

Anaconda vs Miniconda

Anaconda is a full-featured Python distribution for data science, including pre-installed libraries like NumPy, Pandas, and TensorFlow, along with Jupyter Notebook and Spyder. Miniconda, on the other hand, is a lightweight alternative that includes only Conda and Python, allowing users to install packages as needed.
Choose Anaconda if: You want a complete setup with pre-installed tools.
Choose Miniconda if: You prefer a minimal installation and manual package management.

Installation Instructions

Installing Anaconda

1. Download Anaconda from official site.
2. Run the installer:
   - Windows: Select "Add Anaconda to PATH" (optional but recommended).
   - macOS/Linux: Follow terminal instructions.
3. Verify installation:

```
conda --version
```

Installing Miniconda

1. Download Miniconda from official site.
2. Run the installer:
   - Windows: Execute the ‘.exe‘ file.
   - macOS/Linux: Run in terminal:

     ```
     bash Miniconda3-latest-Linux-x86_64.sh
     ```

3. Verify installation:

   ```
   conda --version
   ```