Lap Project – Deep Learning

1. Objective

This experiment aims to help students understand Convolutional Neural Networks (CNNs) and their applications in deep learning by implementing an image recognition model. Students will use the Combined COCO dataset, download link also provided on course Moodle page, for object detection, and complete the entire process of data preprocessing, model training, evaluation, and performance analysis.

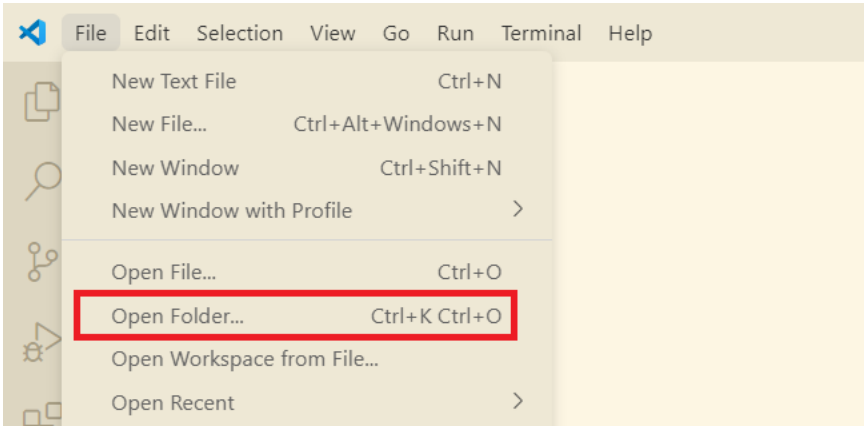
Note：The project provides a complete project package, according to the following steps to open the script file in VScode or Jupyter Notebook, complete the ***experiment.ipynb*** file in the package (download the ***ML-Project_Autlab.zip*** from course Moodle page) .

2. Integrated development Environment

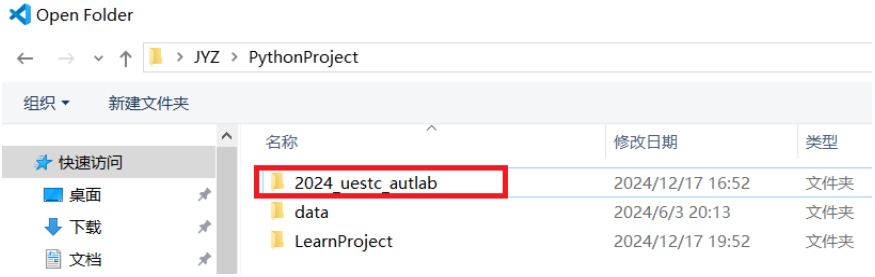
VScode:

Follow the following steps if you want to use the Visual Studio Code (VScode) integrated development environment (IDE).

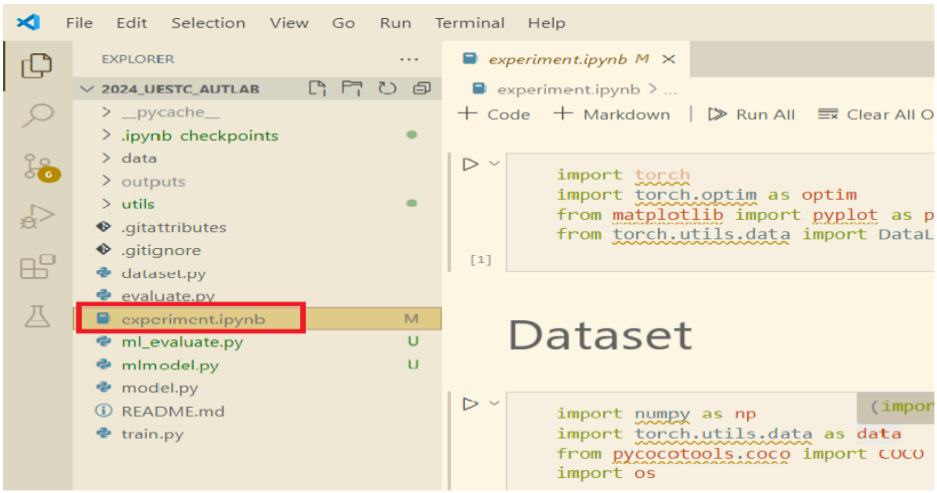
Step1:



Step2:



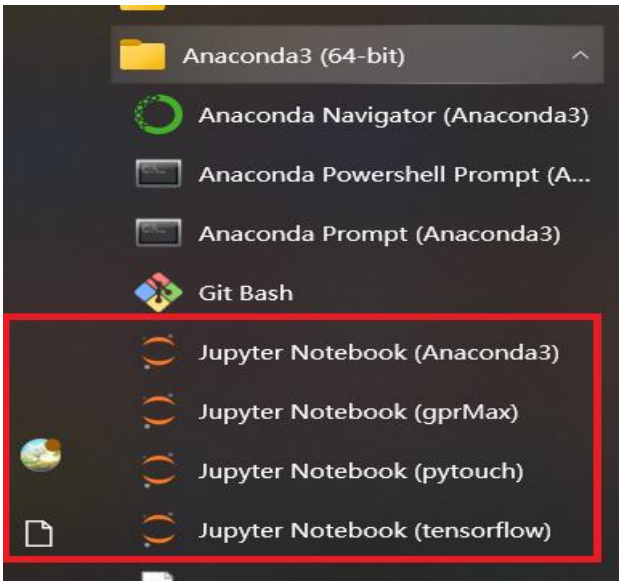
Step3:



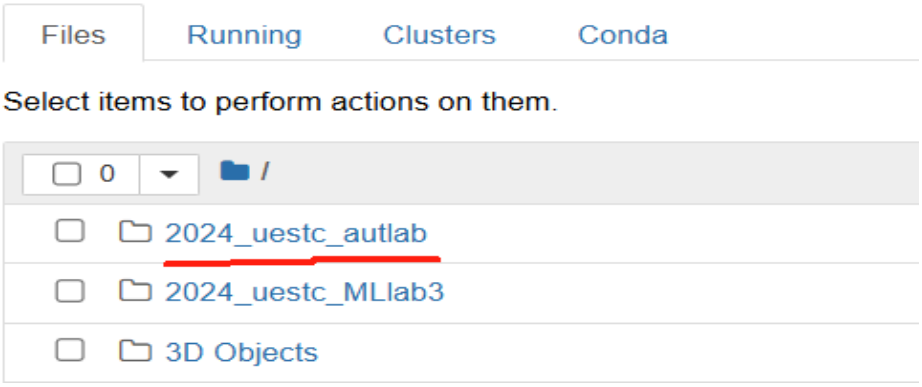
Jupyter Notebook:

Follow the following steps if you want to use the Jupyter notebook integrated development environment.

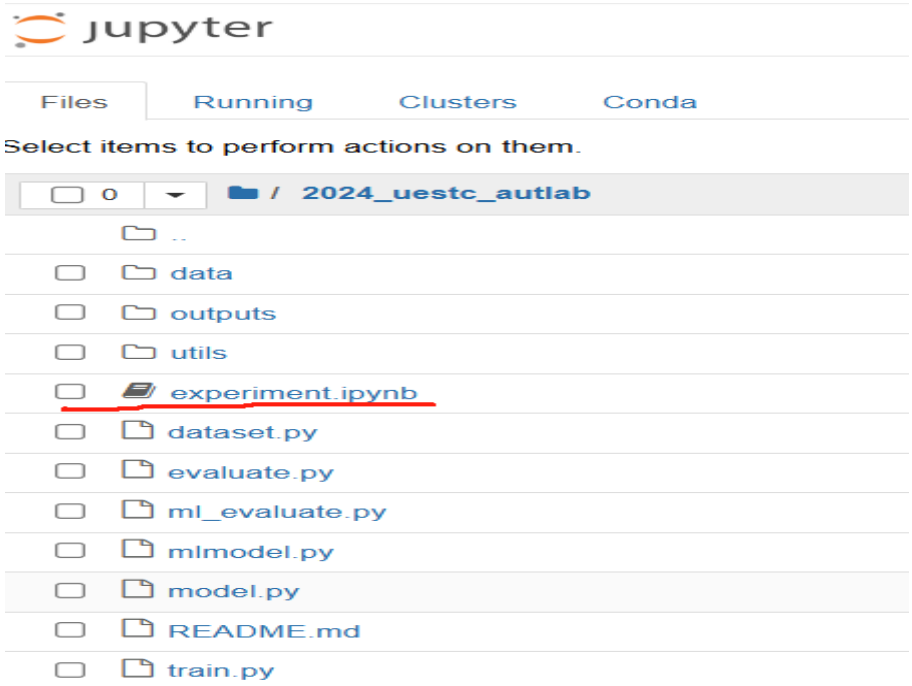
Step1:



Step2:



Step3:



3. Experiment Tasks

Figure 1 illustrated the complete framework of the experimental tasks. You need to choose the deep learning model according to your choice. You are required to complete the following tasks:

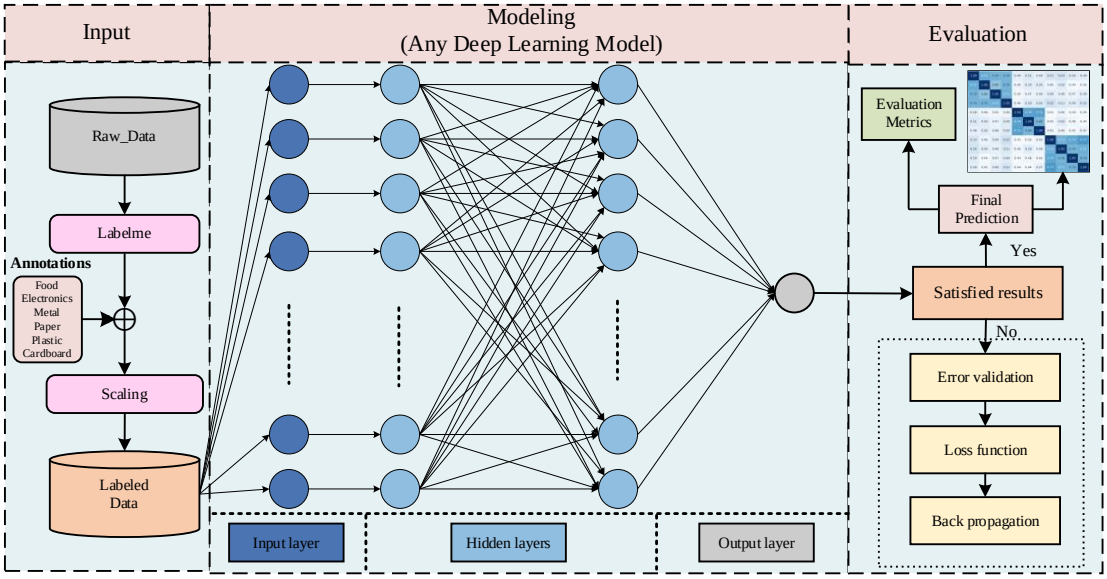


Figure 1: A complete framework diagram of the final project of machine learning.

(1) Dataset Preparation and Processing

- Load and use the COCO dataset for object detection, extracting images and labels for each instance.
- Use the *MyCOCODataset* class to load the data into PyTorch's DataLoader and perform necessary image processing steps (such as cropping, resizing, and normalization).

(2) Model Implementation

- Complete the implementation of the deep learning network, adjusting the input and output layers to match the number of classes in the COCO dataset (7 object categories).
- Ensure that convolutional layers, fully connected layers, and activation functions (ReLU) are correctly implemented. Make sure the network performs forward propagation properly.

(3) Model Training

- Train the model using the Cross-Entropy loss function (*CrossEntropyLoss*) and the Adam optimizer (*optim.Adam*).
- Complete the training process and save the model weights to best_model.pth.

(4) Evaluation and Performance Analysis

- Load the trained model and evaluate it on the test set.
- Compute and output the accuracy of the model on the test set.
- Calculate and display the confusion matrix for further analysis of the model's performance on each category.

(5) Visualization

- Use *matplotlib* to plot the confusion matrix and analyze the model's prediction performance across different categories.
- Observe and discuss the model's classification results, identifying potential weaknesses and areas for improvement.

4. Tasks

You are required to complete the following tasks and write a report, which you need to upload to the Moodle.

- **Data Loading and Processing:**
  - Correctly implement the image cropping, resizing, and other preprocessing steps in the *MyCOCODataset* class.
  - Load the COCO dataset and ensure it returns images and corresponding category labels correctly.
- **Network Implementation:**
  - Complete the implementation of the deep learning model, ensuring it is adapted for the 7-class classification task.
  - Understand and implement the construction of convolutional layers, pooling layers, and fully connected layers.
- **Model Training:**
  - Implement the training process for the model correctly, using the Cross-Entropy loss function and Adam optimizer.
  - Ensure the model can be saved and loaded correctly.
- **Performance Evaluation:**
  - Evaluate the model on the test set, compute the accuracy, and display the confusion matrix.
  - Analyze the results and identify how well the model performs on different categories.
  - Discuss about the comparison between your deep model mechanism and machine learning results (lab4 task).

5. Marks Distribution and Criteria

The submitted report and code will be marks against the following marking criteria.

Task	Weight	Description
Data Loading and Processing	20%	Correctly load the COCO dataset and complete image preprocessing (e.g., cropping, resizing, tensor conversion). Ensure that images and labels match the dataset.
Network Implementation	25%	Complete the implementation of the deep learning model. Ensure correct configuration of convolutional, pooling, and fully connected layers to fit the 7-class classification task.
Model Training	20%	Correctly implement the training process using Cross-Entropy loss and Adam optimizer. Ensure the model can save and load weights properly.
Performance Evaluation and Comparison	25%	Evaluate the model on the test set, calculate the accuracy, and plot the confusion matrix. Analyze model performance across different categories including decision tree and deep learning model.
Code Clarity and Reproducibility	10%	The code should be well-structured, with clear variable names and proper documentation. The experiment should be reproducible.

Note that you should include a detailed description of the implementation, results and discussion in of the following parts in your report.

- **Introduction:**
- **Data Loading and Processing (report and code):** This includes correctly implementing image preprocessing steps (cropping, resizing, normalization), ensuring the dataset loads correctly, and the integrity and consistency of data. You should explain the loading and processing parts including some sample outputs in your report.
- **Network Implementation (report and code):** You must complete the model architecture, ensuring each layer is properly defined and matches the task requirements (7-class classification). You are expected to include network diagram and discussion on the proposed model architecture in your report.
- **Model Training (report and code)**: Ensure the training process runs smoothly, with the correct use of the loss function and optimizer. The model should be correctly optimized and able to save and load weights. Discuss the model training process including the loss function and optimizer in tour report.
- **Performance Evaluation and Comparison (report and code)**: Evaluate the proposed model's accuracy on the test set, plot and analyze the confusion matrix, and discuss the model's performance on different categories including decision tree (from Lab 4) and deep learning model. Also include heatmap confusion matrix plots and the evaluation metrics results of both models on the test set. You are expected to compare the performance of these two models and include a critical analysis of their performance comparison in your report.
- **Code Clarity and Reproducibility (Code)**: Ensure that the code is well-structured, easy to understand, and the experiment is reproducible.

6. Supplementary Material

The following supplementary material is provided in the zip fila on course Moodle page.

- **Dataset**: Combined COCO dataset with images and annotations.
- **Code**: Provided experiment code, including dataset loading, model definition, training, and evaluation.
- **Environment**: Python 3.x, PyTorch as a backend library, and the required deep learning frameworks.

7. Report and Code Submission

You are required to submit the following items on Moodle before the due date.

- A complete PDF report, including all the details listed in section 5, using the following naming format: “GUID_FullName_ML-Report.
- A zip folder containing all the code necessary to reproduce the experiment, using the following naming format: “GUID_FullName_ML-Code.