School of Electrical & Electronic Engineering

IE4424 Machine Learning Design and Application

Academic Year 2024-2025

Image Classification Using Convolutional Neural Network (CNN)

Computer Engineering II (S2-B3b-08)

Laboratory Manual

1. Objective

The objective of this lab is to study the concepts and models for image classification using convolutional neural networks and transfer learning.

2. Introduction

The task of predicting the image category is called image classification, which is one of the most important tasks in computer vision. The purpose of training an image classification model is to recognize various types of images. For example, a model might be trained to recognize three animals: horses, cats, and dogs. When we provide a new picture to the model, it will output the probability that the picture contains one of these animals. With the development of deep learning techniques, Convolutional Neural Networks (CNNs) have been developed to handle the task of image classification.[1]

During training, an image classification model is fed with images and their corresponding labels. Each label is the name of a concept or category. The model will learn to recognize these labels. Given enough training data (usually each category label contains hundreds of images), the image classification model can learn to predict whether a new image belongs to a certain category. The prediction process is called inference.

In many practical applications, we often do not train a new model for a specific image classification task from the scratch since it requires a large amount of data and the training process is very time consuming. Instead, the common practice is to take a CNN model pretrained on a very large dataset (e.g. ImageNet, which contains 1.2 million images with 1000 categories), and then further finetune the pretrained model or use it as a feature extractor.

In this design experiment, we will use PyTorch [2], an optimized tensor library for deep learning using GPUs and CPUs to build CNN models. We will also perform experiments on popular CNN models such as VGGNet [3] and ResNet [4]. The lab is developed and modified using the models in Pytorch Tutorials [5].

3. Files

There are 4 files for this lab session.

1. IE4424_Image_classification_lab_manual.pdf
2. Part1_Image_Classification.ipynb
3. Part2_Transfer_Learning.ipynb
4. Ants-and-bees dataset

Note that the ants-and-bees dataset can be downloaded from https://download.pytorch.org/tutorial/hymenoptera_data.zip. You should create a folder in your desktop, download and put the dataset into the folder.

4. Cheat sheets and reading materials

The following cheat sheets and reading materials are optional, and they are useful for this lab. You are encouraged to browse through them.

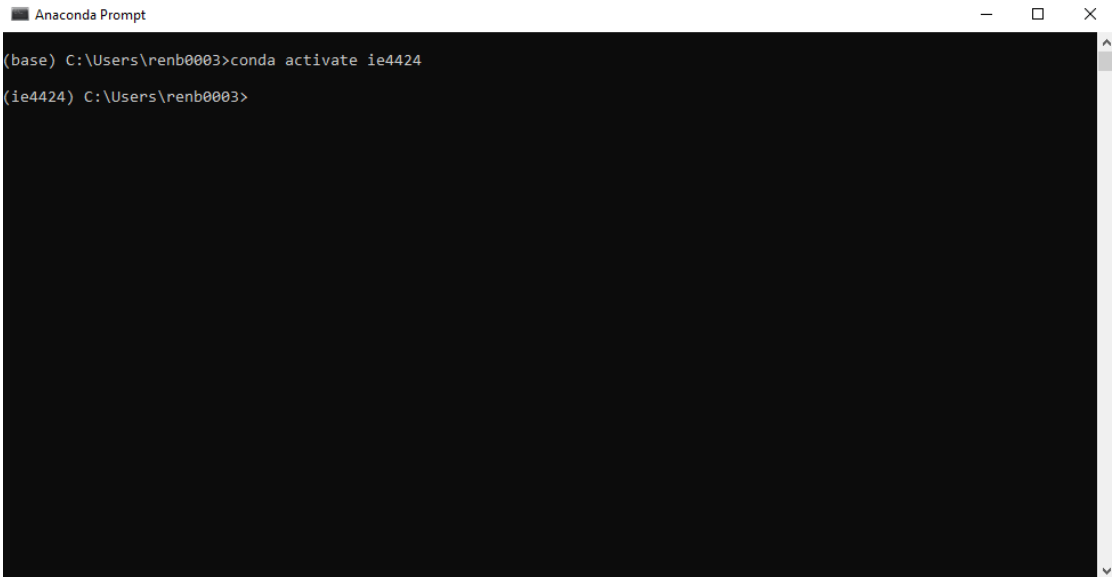
1. Jupyter Notebook:
   https://www.dataquest.io/blog/jupyter-notebook-tutorial/
2. PyTorch:
   https://pytorch.org/tutorials/beginner/ptcheat.html
   https://pytorch.org/tutorials/
3. Python:
   https://perso.limsi.fr/pointal/_media/python:cours:mementopython3-english.pdf
4. Deep Learning for Computer Vision
   http://cs231n.stanford.edu

5. A Quick Instruction to Jupyter Notebook

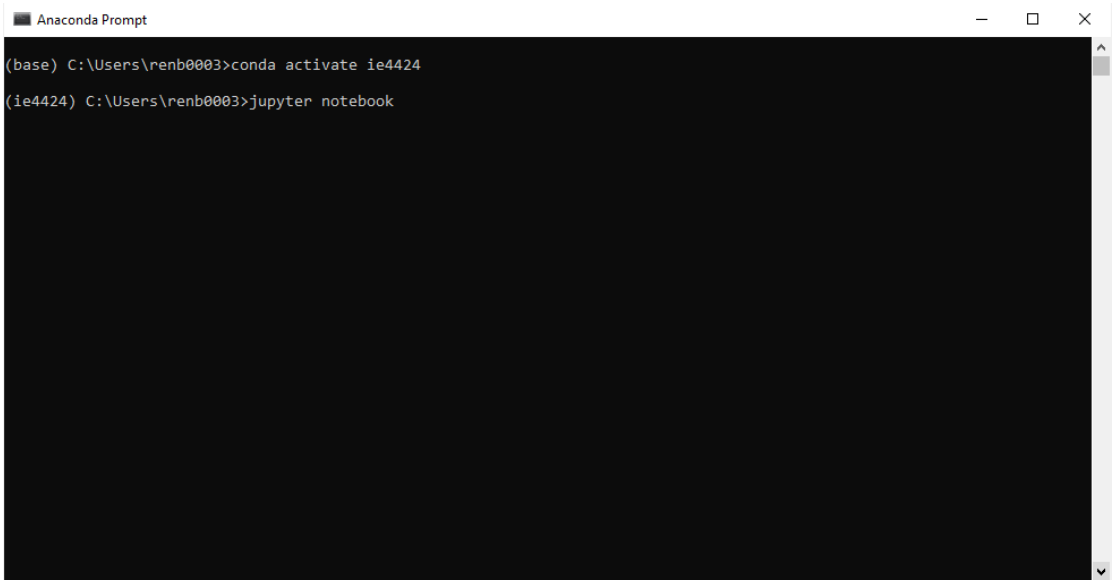
We will use Jupyter Notebook to conduct the lab experiments.

1. First open the anaconda prompt in the windows start menu.

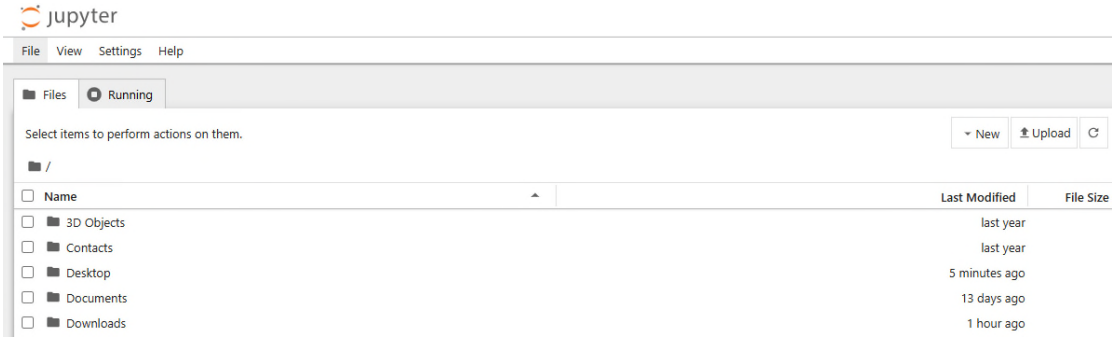
2. Type "conda activate ie4424" to activate the virtual environment.



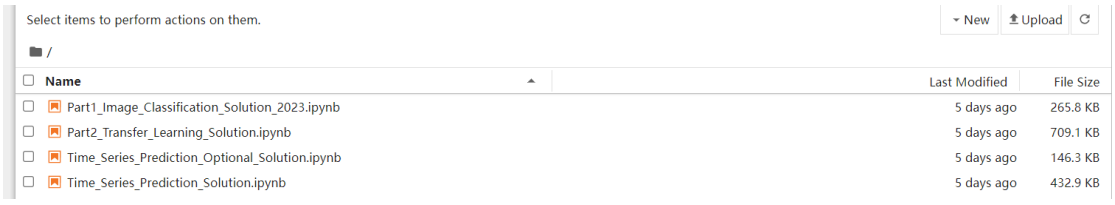
3. Make sure (ie4424) is the activated environment, and type “jupyter notebook” in the command window.



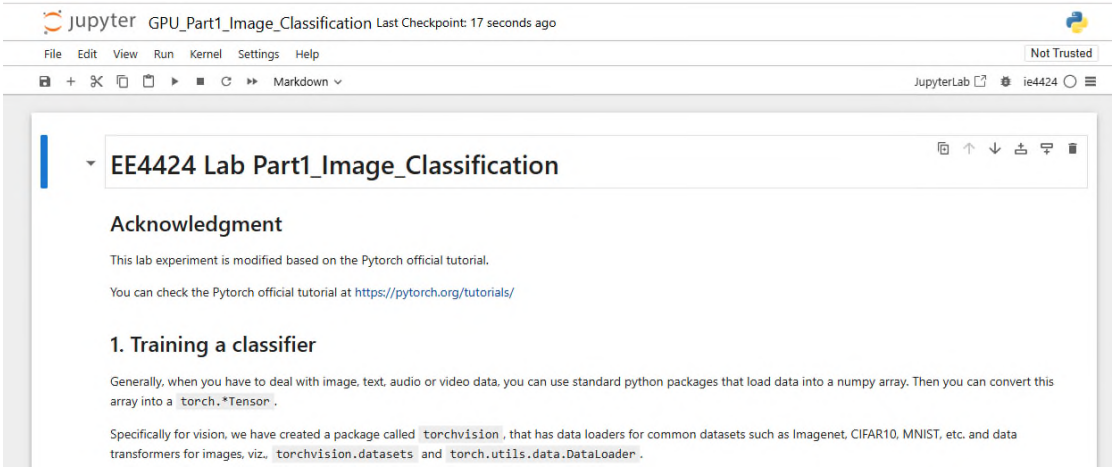
4. The Jupyter Notebook will start in your browser. If the Jupyter Notebook does not start automatically, you can copy the url in the command window and paste it into the browser. You will see the following.



5. Navigate to the folder that contains codes and files.



6. Click the Part1_Image_Classification.ipynb as this is the first part of this lab session.



7. Check if the kernel is ie4424 on the top right corner. If not, change the kernel to ie4424 using Kernel Tab.

8. Jupyter Notebooks work with cells. You can write your code in each cell and run each cell separately. You can directly see the output of each cell and all the variables are stored in the memory.

9. On the toolbar, you can create new cells using the + button.
   For example, we can run simple python codes in each cell:

```
In [1]: a = 1
        b = 2
        print(a+b)

        3
```

   The index [1] shows the sequence that the cells are operated. The variables a and b are stored in the memory, you can still use them in later cells.

6. Instructions

1. Read the instructions and complete the exercises in Part1_Image_Classification.ipynb and Part2_Transfer_Learning.ipynb.

2. Get the answer sheet from the lab staff. Follow the instructions and answer the questions in the answer sheet. Write your full name and other info clearly on the answer sheet. Submit the completed answer sheet at the end of lab.

References

[1] Abadi, Martín, et al. "Tensorflow: A system for large-scale machine learning." *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*. 2016.

[2] Paszke, Adam, et al. "Pytorch: An imperative style, high-performance deep learning library." *Advances in neural information processing systems*. 2019.

[3] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

[4] He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[5] Pytorch Tutorial. https://pytorch.org/tutorials/