

Task 1B – Fruit Classification with a CNN

NOTE: Please go through [Extra_Code_Instructions.pdf](#) for the Coding Instructions.

This document explains the problem statement to be solved in the sub-task **Task 1B**, the organization of sub-folders, how to use the folder content and programming instructions.

Problem Statement:

To classify fruits into 5 categories given images using a CNN.

Objective:

We have 3 objectives for this sub-task. First, we will learn about CNN architecture. Second, we will tune model's hyper-parameters to get good results ourselves. And third we will solve the problem (fruit classification) at hand using best techniques for the given task. In our case, we do this with PyTorch.

This sub-task is different from the previous sub-task. In previous sub-task, we implemented algorithms from scratch. But in this sub-task, we will use algorithms and techniques, already implemented in PyTorch (high-level) to best solve the given task. Also, in previous sub-task our focus was not on achieving best possible accuracy but rather getting our algorithms right. In this task, we want good results. This task, thus, concentrates on your learning of advanced techniques and algorithms mostly by intuitive examples and simultaneously skipping some (advanced) maths.

Machine Learning is advancing very fast and it's difficult to learn everything. But you may still want to use the best techniques to get good results. To conclude, in this task you'll demonstrate knowledge of **CNN**, knowledge of **tuning hyper-parameters** and, ability to search for and implement more advanced techniques in Machine Learning using PyTorch high level API.

NOTE: Use of Transfer Learning is not allowed in this task.

Where to write the code ?

In this task, you will write **Python programs in files** present inside folder named **Code**. But unlike before, in this task we want you to write more code by yourself and think about how you will structure it. You should structure your code like you have seen previously and divide the code into modular blocks. Extra marks for test cases and good coding style.

There are not many functions, modules and classes in this exercise (since we want you to create them yourself when you find a need to) but still you should complete their objectives.

We assume that, by now, you **definitely** understand the steps performed when building a Machine Learning solution. The steps will be the same here. Only the model architecture (here, a convolutional neural network) and its training will differ. Checkout **Code** folder.

Code Structure:

Now, that you are somewhat aware of the modules of the application, we will brief them here.

- *dataset.py*

This module should take care of all the data-handling tasks like facilitating loading of data (images) in memory efficiently, splitting into train and test set, shuffling, etc. We have provided scaffolding and instructions for this inside this file so that it becomes easy for you to know what has to be done.

- *model.py*

This module will contain code for the model architecture. You'll use PyTorch's **nn** module to build a net.

- *main.py*

This is the actual entry point to your application. It will use parts from above to pre-process data and then train a model on it. On running this file, it should start training your model and obey additional instructions, like an option to save model, which are mentioned inside this file.

- *Task_1B.ipynb*

This notebook file will have to present all the work that you have done. You will use the above modules (except *main.py*) in this file. Additional instructions as to what you have to do are given in this file.

- *test package*

It contains *test_dataset.py* to verify implementation of *dataset.py*.

You are allowed to create more modules and packages to structure your code better but **the functions, modules, etc. that we have documented should return exactly what is asked unless specified otherwise.**

Dataset:

The data set will be in the Data folder. It is a data set of 5 fruits, namely, **Apple, Banana, Orange, Strawberry and Pineapple**. You will use it to train your neural network. Go checkout the *Data* folder.

Resources:

Whatever resources you will need for completing this task are present in the **Task 1B Resources** folder. The resources can be links to some of the best tutorials available on the web like videos, notes, etc. If required, we will also provide our own notes and videos.

Please note that we assume that you have completed all tasks before this task. So we may not include resources from previous tasks again. If you think something that you cannot complete the task without, is not present as a resource in this folder, checkout previous tasks resources. We will mark wherever required but if we don't please do as said above.

Where to go next from here ?

The most important thing you need to know to complete the task is the **theory of convolutional neural networks**. Checkout the **Resources** folder.

If you understand **Neural Networks**, CNN will be easier to understand. You'll come across new concepts like parameter sharing, etc. In my opinion, **CS231n** notes are the best. Once you understand from it, go through PyTorch docs and see examples on implementing a CNN. You can then start working on the code. Since you have to create a small CNN with PyTorch, you'll only need to work on finding a good architecture parameters. How many filters, their sizes, number of feature maps, etc. On the other hand, **dataset.py** module uses mostly plain Python programming, Pandas and Torchvision's **dataset** module. Try to implement the functions and check if its right by executing tests in **test_dataset.py**.

Use resources as reference or take notes when doing this. One by one, complete each part and in no time you'll be done with complete task.

...Best Wishes ! ...