Data Science HW3 Few-Shot Learning

Can machines learn fast as humans?

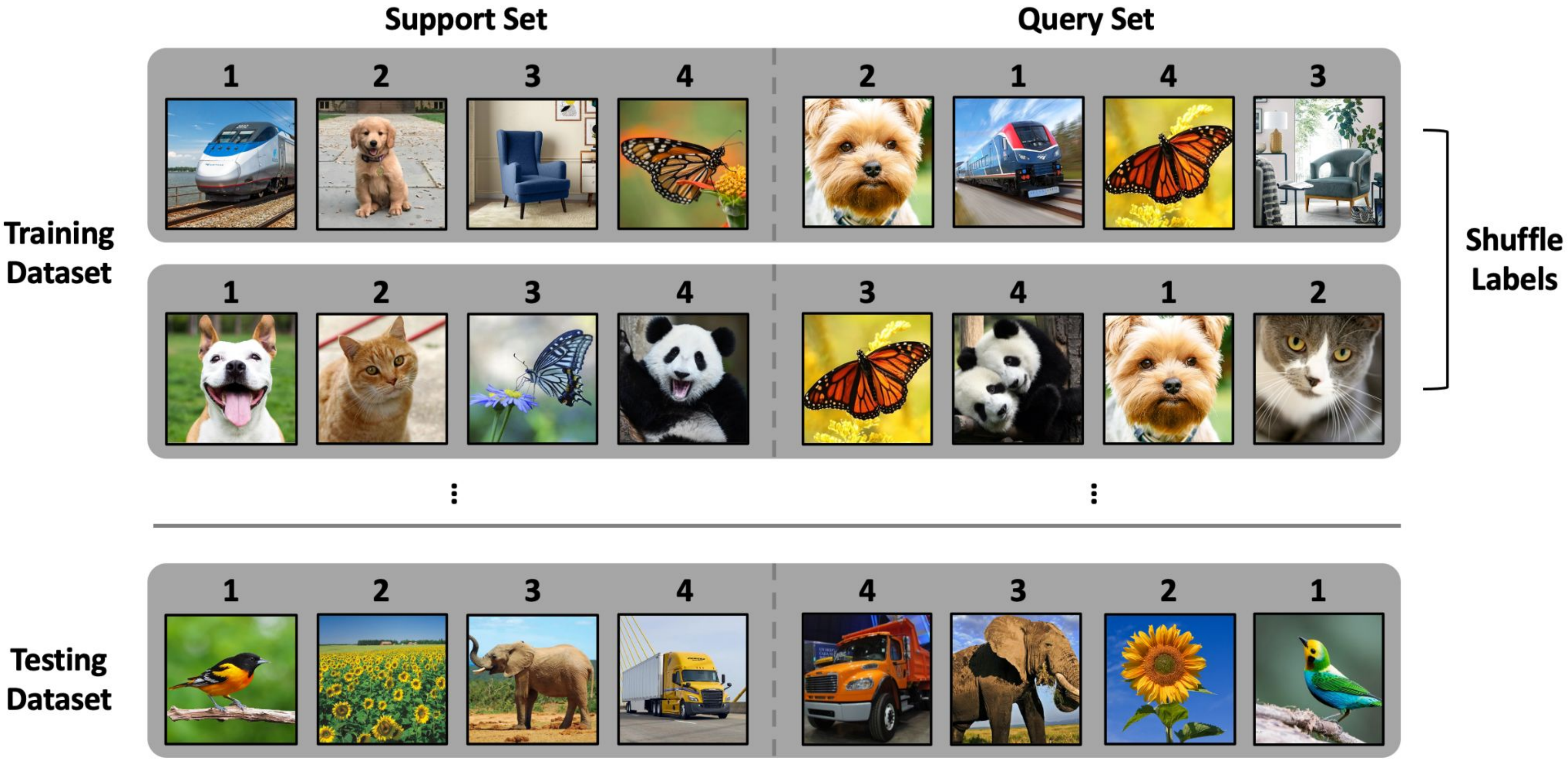
Objectives

- In this homework, you need to implement few-shot learning algorithms to solve the given tasks.
- In order to successfully complete this homework, you need to pay close attention to two key points:
  - Data formats of few-shot learning
  - Basic algorithms for few-shot learning

Introduction

- In few-shot learning, we have **training**, **validation**, **testing datasets** as the conventional learning paradigm.
- In each split, we further have **support sets** for adaptation and **query sets** for evaluation.

Introduction



Introduction

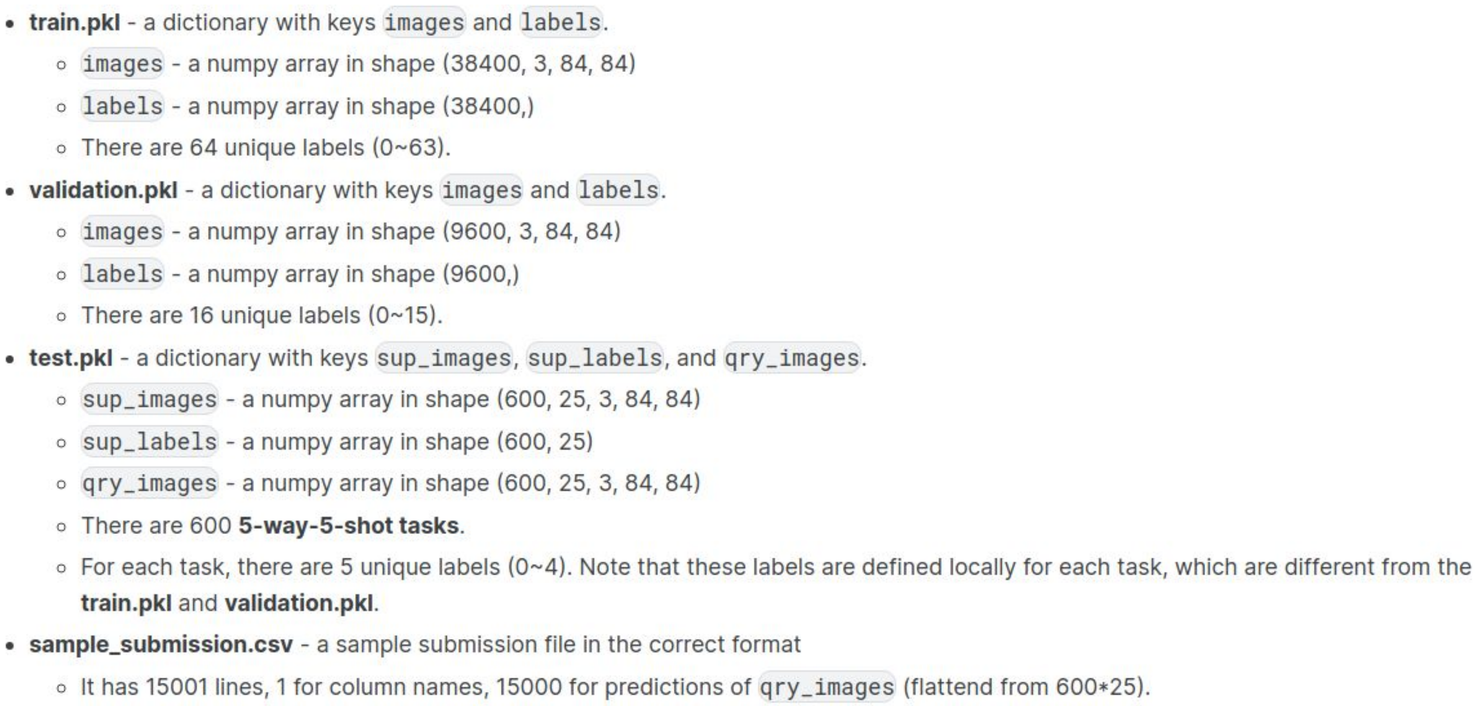
- A support set and a query set jointly define a **task**.
- Note that the labels are **randomly shuffled across different tasks**. Therefore, models are generally not able to solve new tasks simply through memorization. It turns out that models need to leverage support sets to make predictions for query sets.
- We usually use the term **N-Way-K-Shot** to describe a task. That means the support and query sets both have N classes and each class has K images. In other words, we have totally N * (2 * K) images for this task.

Introduction

- We use the **support/query sets in training dataset** to learn a model, and validate it with **support/query sets in validation datasets**. Finally, we test the model with **support/query sets in testing datasets**.
- The classes in training, validation, and testing datasets are typically **disjoint** to evaluate the model's generalization ability.
- To put the whole thing simply, we ultimately need the predictions on **query sets in testing datasets**. Note that the labels for these query sets are defined according to the corresponding support sets as aforementioned.
- There are numerous algorithms designed for this purpose. Pick one and implement it. Enjoy!

Datasets

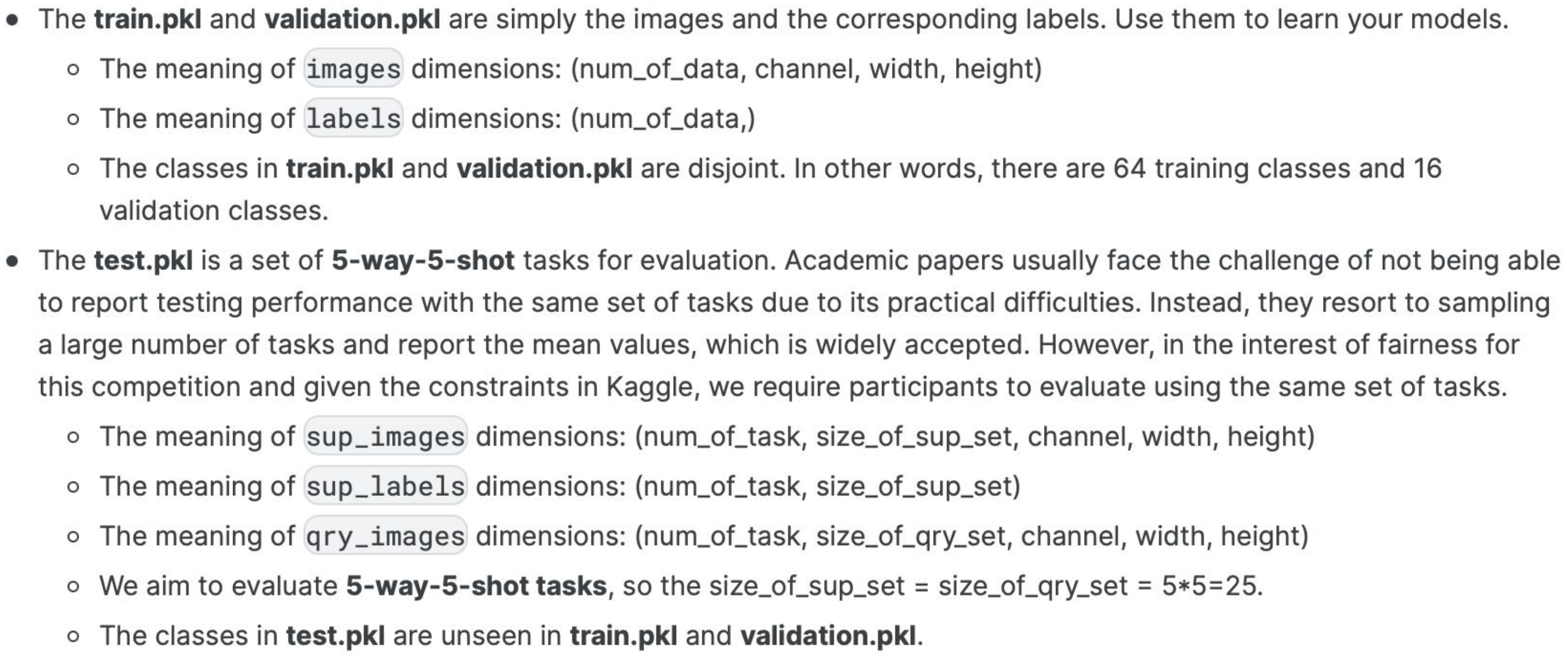
- Files



- **train.pkl** - a dictionary with keys `images` and `labels`.
  - `images` - a numpy array in shape (38400, 3, 84, 84)
  - `labels` - a numpy array in shape (38400,)
  - There are 64 unique labels (0~63).
- **validation.pkl** - a dictionary with keys `images` and `labels`.
  - `images` - a numpy array in shape (9600, 3, 84, 84)
  - `labels` - a numpy array in shape (9600,)
  - There are 16 unique labels (0~15).
- **test.pkl** - a dictionary with keys `sup_images`, `sup_labels`, and `qry_images`.
  - `sup_images` - a numpy array in shape (600, 25, 3, 84, 84)
  - `sup_labels` - a numpy array in shape (600, 25)
  - `qry_images` - a numpy array in shape (600, 25, 3, 84, 84)
  - There are 600 **5-way-5-shot tasks**.
  - For each task, there are 5 unique labels (0~4). Note that these labels are defined locally for each task, which are different from the **train.pkl** and **validation.pkl**.
- **sample_submission.csv** - a sample submission file in the correct format
  - It has 15001 lines, 1 for column names, 15000 for predictions of `qry_images` (flattend from 600*25).

Datasets

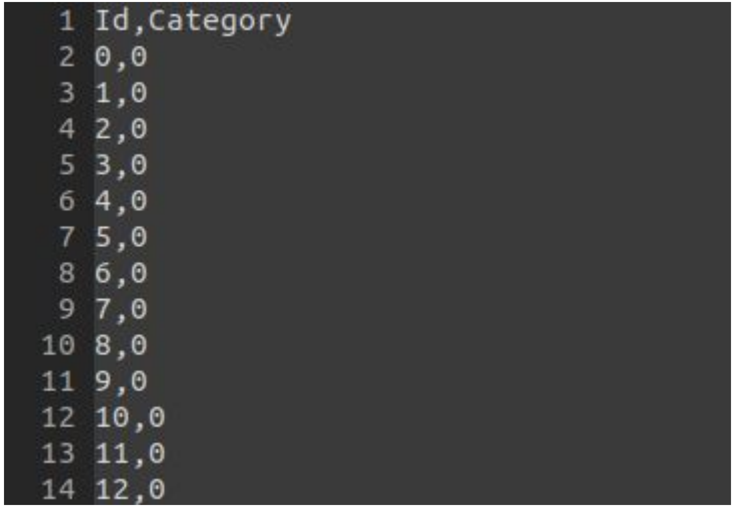
- Descriptions



- The **train.pkl** and **validation.pkl** are simply the images and the corresponding labels. Use them to learn your models.
  - The meaning of `images` dimensions: (num_of_data, channel, width, height)
  - The meaning of `labels` dimensions: (num_of_data,)
  - The classes in **train.pkl** and **validation.pkl** are disjoint. In other words, there are 64 training classes and 16 validation classes.
- The **test.pkl** is a set of **5-way-5-shot** tasks for evaluation. Academic papers usually face the challenge of not being able to report testing performance with the same set of tasks due to its practical difficulties. Instead, they resort to sampling a large number of tasks and report the mean values, which is widely accepted. However, in the interest of fairness for this competition and given the constraints in Kaggle, we require participants to evaluate using the same set of tasks.
  - The meaning of `sup_images` dimensions: (num_of_task, size_of_sup_set, channel, width, height)
  - The meaning of `sup_labels` dimensions: (num_of_task, size_of_sup_set)
  - The meaning of `qry_images` dimensions: (num_of_task, size_of_qry_set, channel, width, height)
  - We aim to evaluate **5-way-5-shot tasks**, so the size_of_sup_set = size_of_qry_set = 5*5=25.
  - The classes in **test.pkl** are unseen in **train.pkl** and **validation.pkl**.

Evaluation

- We evaluate on 600 **5-way-5-shot tasks**. The evaluation metric is simply the **category accuracy** for **predictions on query sets**. Please refer to the Kaggle pages for details and be sure you understand the submission format.



```
Id,Category
0,0
1,0
2,0
3,0
4,0
5,0
6,0
7,0
8,0
9,0
10,0
11,0
12,0
```

Grading Policy

- Top 10%: 100 points
- Top 25%: 90 points
- Top 50%: 80 points
- Top 75%: 75 points
- Others: 70 points
- **Bellow baseline (shown in Kaggle leaderboard): 0 point**
- **Public 30%, Private 70%**

Rules

- Use your student ID as the team name on Kaggle.
- **Do not** use additional accounts to get more submission quota.
- **Do not** plagiarize. Write your own codes.
- Since the implementation of few-shot learning algorithms may involve some tricky steps, we **do not constrain** the usage of libraries.
- You are **not allowed** to use additional datasets for learning. In other words, you can only use the datasets provided in this competition to learn your model.
- **Do not** attempt to recognize the datasets we used and hack the testing performance. You will not obtain scores for this homework if you violate this rule (we will re-implement your results).

Submissions

- Submit your results to Kaggle:

  https://www.kaggle.com/t/c985e5ad3c364f77ab00c201ecb2a3f4
- Submit your zipped source code {student_id}.zip to E3. After unzip the file, it should appear a folder {student_id}:
  - {student_id}
    - {student_id}.sh: run this script should regenerate your final submission result.
    - requirements.txt: list the required libraries.
    - Other files
    - No need to include the datasets. Therefore, please clearly specify the expected location of the datasets.

Information

- **Deadline: 2023/4/11 (Tuesday) 23:59**
- If you have any questions, please post on the **E3 forum**. TA will response ASAP.
- [TA] Yi-Syuan Chen: yschen.ee09@nycu.edu.tw