Machine Learning HW3 - Image Classification

ML TAs
ntu-ml-2021-spring-ta@googlegroups.com

Objective

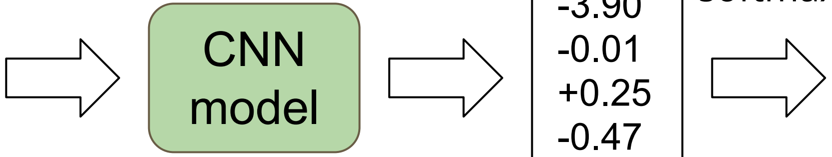
1. Solve image classification with **convolutional neural networks**.
2. Improve the performance with **data augmentations**.
3. Understand how to utilize **unlabeled data** and how it benefits.

Task - Food Classification

Task - Food Classification

- The images are collected from the food-11 dataset classified into 11 classes.
- The dataset here is slightly modified:
- Training set: 280 * 11 **labeled** images + 6786 **unlabeled** images
- Validation set: 60 * 11 **labeled** images
- Testing set: 3347 images
- **DO NOT** utilize the original dataset or labels.
    - This is cheating.

Task - Food Classification



logits (not normalized)	probability (normalized)
-0.25	0.004
+3.02	0.114
+0.56	0.009
-3.90	0.001
-0.01	0.005
+0.25	0.007
-0.47	0.003
+5.00	0.831
-1.32	0.001
+1.14	0.017
-0.28	0.004

Kaggle link: here

Requirements

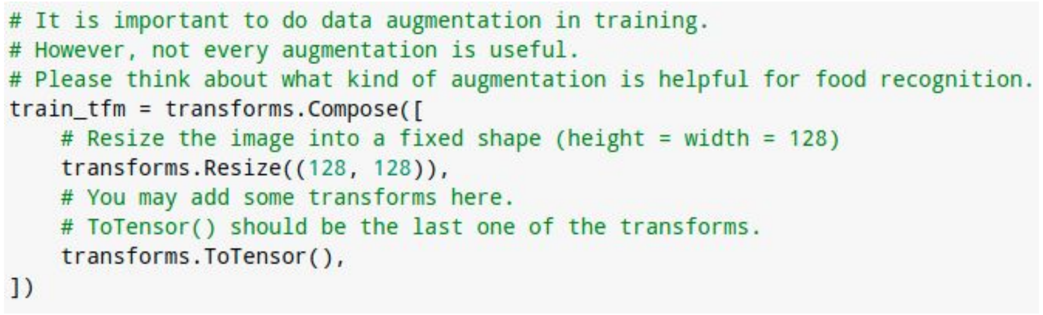
- This homework is in three levels:
  - Easy
  - Medium
  - Hard
- You can easily finish the easy level by running the example code.
- For the rest, we recommend you start with the same code.
  - We already prepared some TODO blocks for you.
- **DO NOT** pre-train your model on other datasets.
- If you use some well-known model architecture (e.g., ResNet), make sure **NOT** to load pre-trained weights as initialization.

Requirements - Easy

- Build a **convolutional neural network** using **labeled images** with provided codes.
- Public simple baseline: 44.862 (accuracy, %)

Requirements - Medium

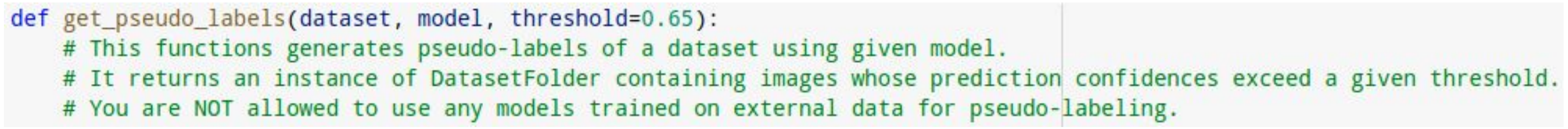
- Improve the performance using **labeled images** with different model architectures or data augmentations.
- Public medium baseline: 52.807 (accuracy, %)
- You can achieve the baseline by adding a few lines to the example code.



```
# It is important to do data augmentation in training.
# However, not every augmentation is useful.
# Please think about what kind of augmentation is helpful for food recognition.
train_tfm = transforms.Compose([
    # Resize the image into a fixed shape (height = width = 128)
    transforms.Resize((128, 128)),
    # You may add some transforms here.
    # ToTensor() should be the last one of the transforms.
    transforms.ToTensor(),
])
```

Requirements - Hard

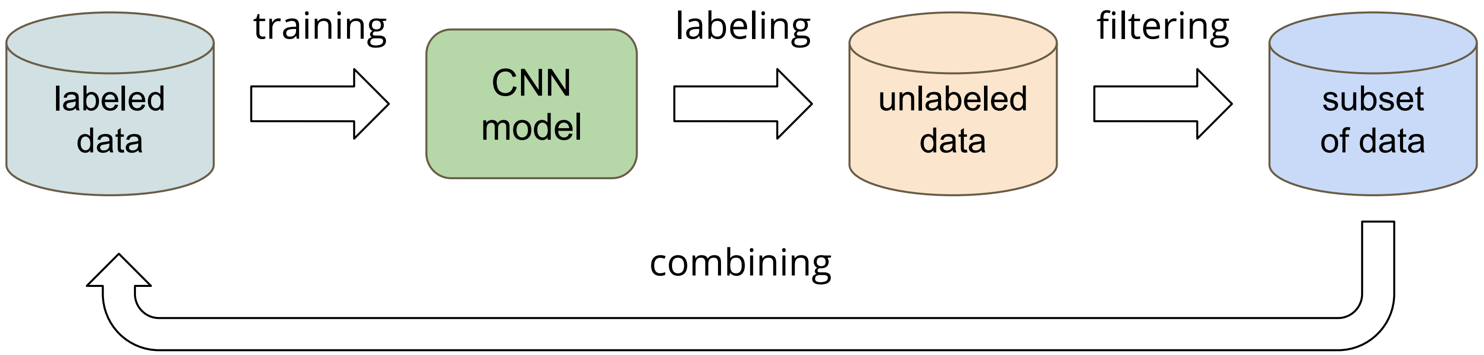
- Improve the performance with **additional unlabeled images**.
- Public strong baseline: 82.138 (accuracy, %)
- Do it on your own (by finishing TODO blocks in the example code).
- Using unlabeled testing data here is allowed.
- **Hint:** semi-supervised learning, self-supervised learning



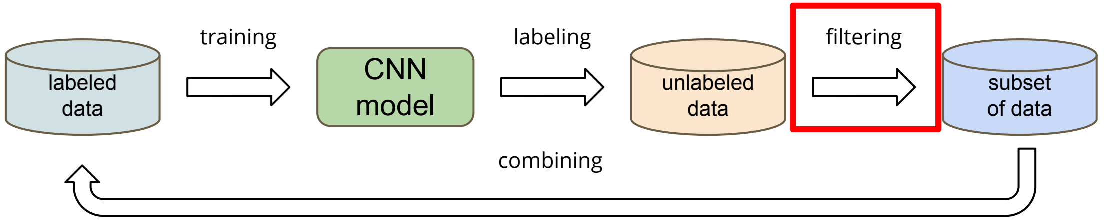
```
def get_pseudo_labels(dataset, model, threshold=0.65):
    # This functions generates pseudo-labels of a dataset using given model.
    # It returns an instance of DatasetFolder containing images whose prediction confidences exceed a given threshold.
    # You are NOT allowed to use any models trained on external data for pseudo-labeling.
```

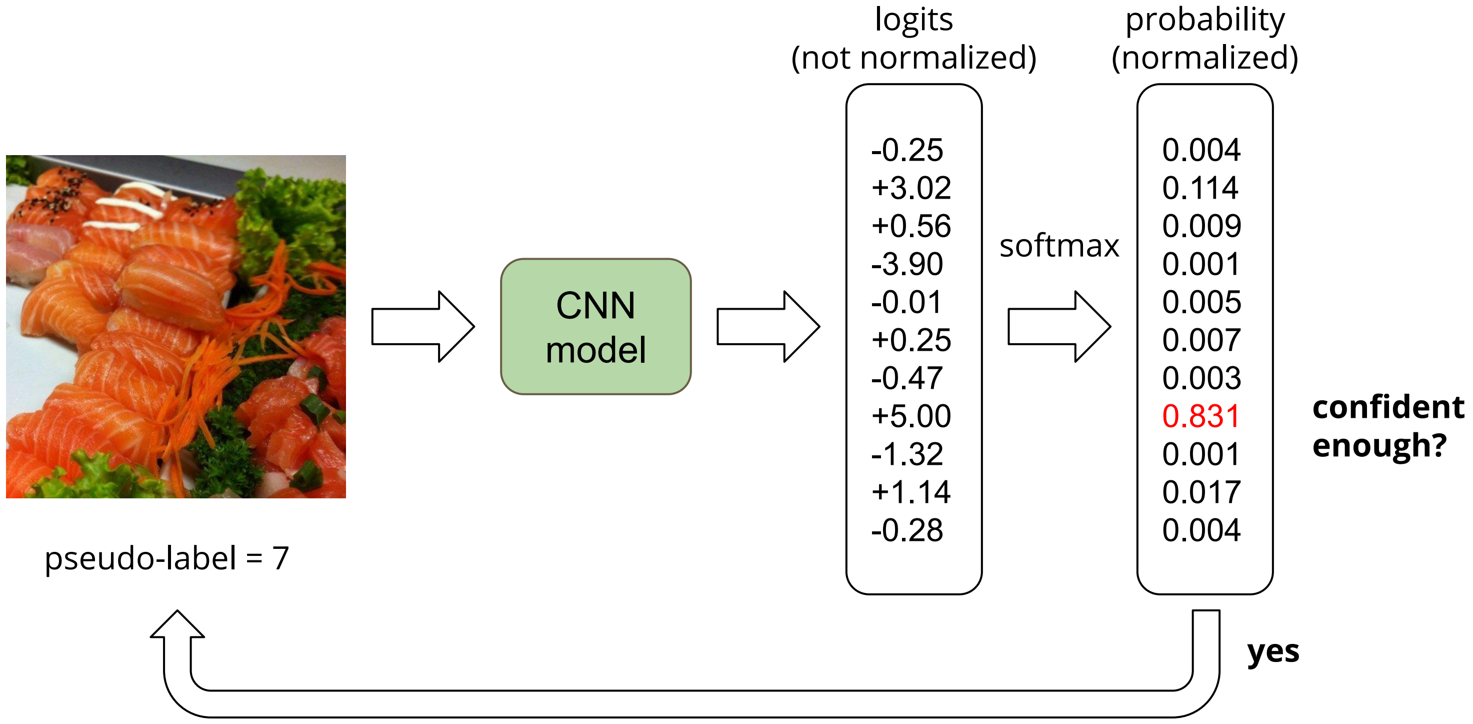
Semi-supervised Learning

- There are many ways to do semi-supervised learning.
- E.g., generate pseudo-labels for unlabeled data and train with them.



Pseudo-labels





Kaggle Submission Format

- The submitted predictions should be in **CSV** format.
- The first row is **“Id, Category”**
- The rest of rows are “{id}, {prediction}” (e.g., 0005, 8)
- There should be (3347 + 1) rows in total.

Id	Category
0001	0
0002	9
0003	4
0004	5

Grading Policy

Public simple baseline:	+1pt
Public medium baseline:	+1pt
Public strong baseline:	+1pt
Private simple baseline:	+1pt
Private medium baseline:	+1pt
Private strong baseline:	+1pt
Submit your code:	+4pt

Code Submission

- Submit your code via NTU COOL.

  <student_id>_hw3.zip
- **DO**
  - Specify the source of your code. (You may refer to Academic Ethics Guidelines)
  - Organize your code and make it easy to read (not necessary).
- **DO NOT**
  - Submit empty or garbage files.
  - Submit the dataset or model.
  - Compress your codes into other formats like .rar or .7z and simply rename it to .zip.

- If we find you cheating or your code problematic, you will be punished.
  - Course final score * 0.9 for the first time, or fail the course otherwise.

Bonus

- If you successfully get 10 pts:
  - Your code will be made public to students.
  - You can submit a report in **PDF** format briefly describing what you have done (in English, less than 100 words) for **extra 0.5 pts**.
  - Reports will also be made public to students.

- Report template

Deadline

- Kaggle deadline: 2021/04/16 23:59:59
- Code submission: 2021/04/18 23:59:59
- Late submissions are **NOT** accepted.

Should You Have Any Questions...

- NTU COOL (recommended)
  - https://cool.ntu.edu.tw/courses/4793
- E-mail
  - ntu-ml-2021spring-ta@googlegroups.com
  - The title **must** start with **[hw3]**.
- TA hour
  - Fri. 14:00 - 18:00

Useful Resources

- Semi-supervised learning
  - https://speech.ee.ntu.edu.tw/~tlkagk/courses/ML_2016/Lecture/semi%20(v3).pdf
  - https://www.youtube.com/watch?v=fX_guE7JNnY&ab_channel=Hung-yiLee
  - MixMatch: https://arxiv.org/abs/1905.02249
  - Noisy student: https://arxiv.org/abs/1911.04252
- PyTorch
  - https://pytorch.org/
- Torchvision
  - http://pytorch.org/vision/stable/index.html