Machine Learning HW3 - Image Classification

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

Objective

1. Solve image classification with **convolutional nueral network**.

2. Improve the performance with **data augmentations**.

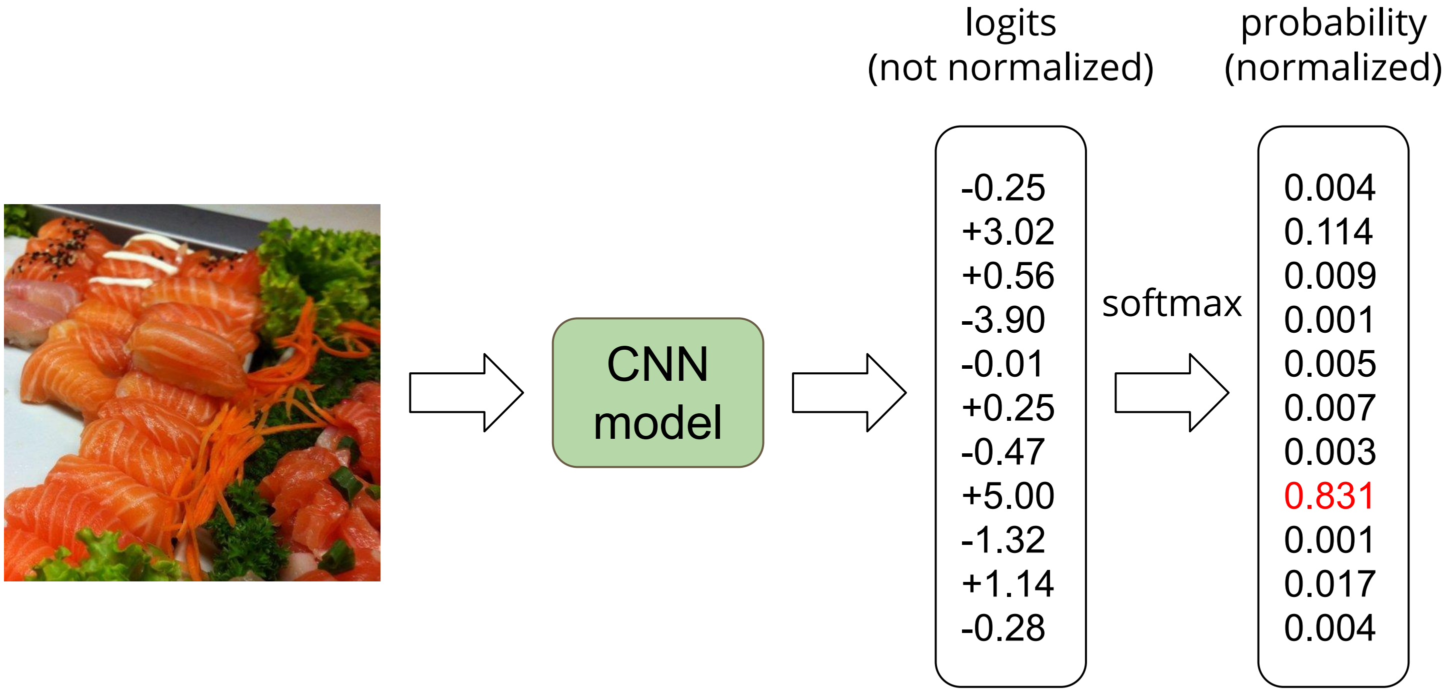
3. Understand how to utilize **unlabeled data** and how it benefits.

Task - Food Classification

Task - Food Classification

- The images are collected from food-11 dataset, all of which are classified into 11 classes.
- The dataset here is slightly modified:
- Training set: 280 * 11 **labeled** images + 6786 **unlabeled** images
- Validation set: 30 * 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:
https://www.kaggle.com/c/ml2021spring-hw3

Requirements

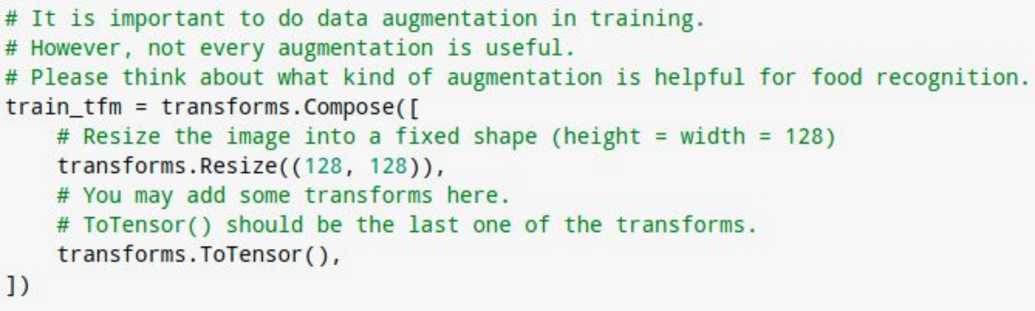
- This homework is in three levels:
  - Easy
  - Medium
  - Hard
- You can easily finish easy level by running the example code.
- For the rest, we recommend you start from the same code.
  - We already prepared some TODO blocks for you.
- **DO NOT** pre-train your model on other dataset.
- If you use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

Requirements - Easy

- Build a **convolutional nueral network** using **labeled images** with provided codes.
- Simple public baseline: 44.862 (accuracy, %)

Requirements - Medium

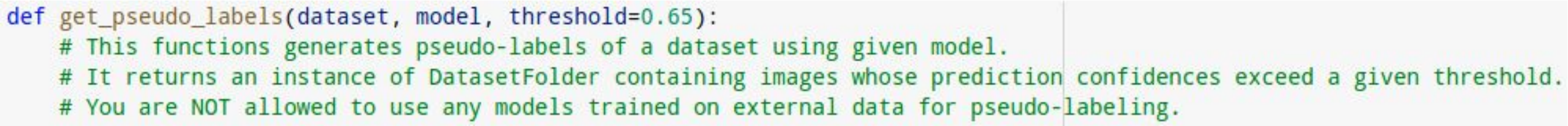
- Improve the performance using **labeled images** with different model architectures or data augmentations.
- Public normal baseline: 52.807 (accuracy, %)
- You can achieve the baseline by adding a few lines to example code.



```
# 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

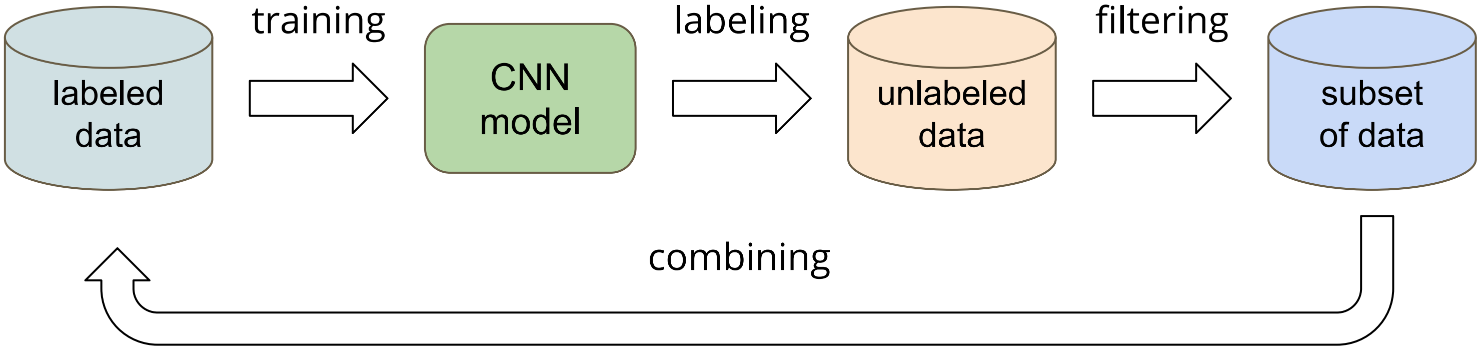
- Improve the performance with **additional unlabeled images**.
- Public strong baseline: 82.138 (accuracy, %)
- Do it on your own (by finishing TODO blocks in the example code).
- Using unlabeled testing data here is allowed.
- **Hint:** semi-supervised learning, self-supervised learning



```python
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 variants of semi-supervised learning.
- E.g., generate pseudo-labels for unlabeled data and train with them.



labeled data

training

CNN model

labeling

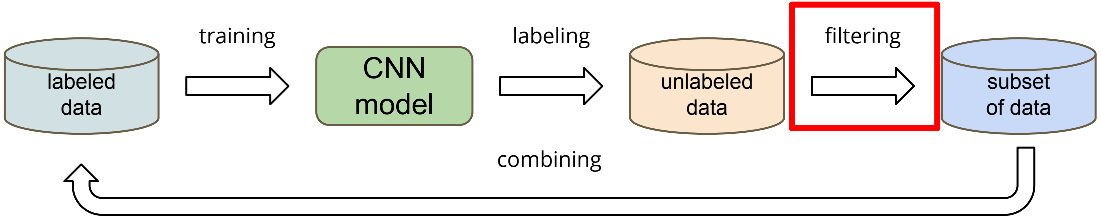
unlabeled data

filtering

subset of data

combining

Pseudo-labels



labeled data

training

CNN model

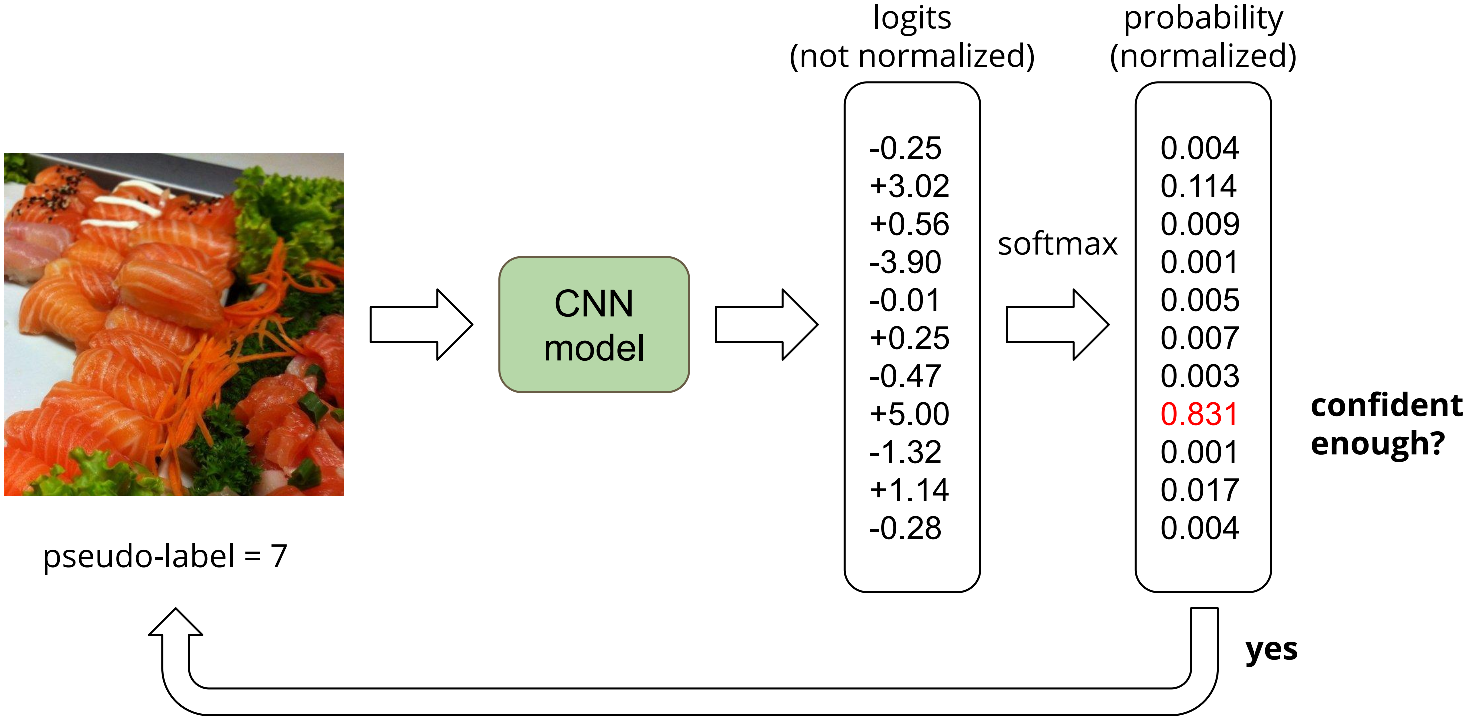
labeling

unlabeled data

filtering

subset of data

combining



CNN model

softmax

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

confident enough?

yes

pseudo-label = 7

Kaggle Submission Format

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

id	label
0001	0
0002	9
0003	4
0004	5

Code Submission

- Submit your code via NTU COOL.

  <student_id>_hw3.zip

- **DO**
  - specify the source of your code.
  - organize your code and make it easy to read (not necessary).
- **DO NOT**
  - submit an 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 first time, or fail the course otherwise.

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