ELEC5307 Assignment 2

Assignment 2

- Task 1: General Image Classification (Fruit dataset)
  - Train/val set (available, Both with labels). Test set (unavailable, marking).
  - 20 marks

- (Optional)
- Task 2: Domain adaptive image Classification in real-world scenario. (Office dataset)
  - Source/Target set available. (Source set with labels, Target set without labels)
  - Successfully applied Domain Adaptation techniques can get up to 5 bonus marks.
    - General techniques:
      - Reverse Gradient
      - Pseudo Labels
      - …

Task 1 General Image Classification

- Parameter Tuning
- Load ImageNet Pre-trained models and fine-tune
- Apply different tricks you found useful (Google it!)

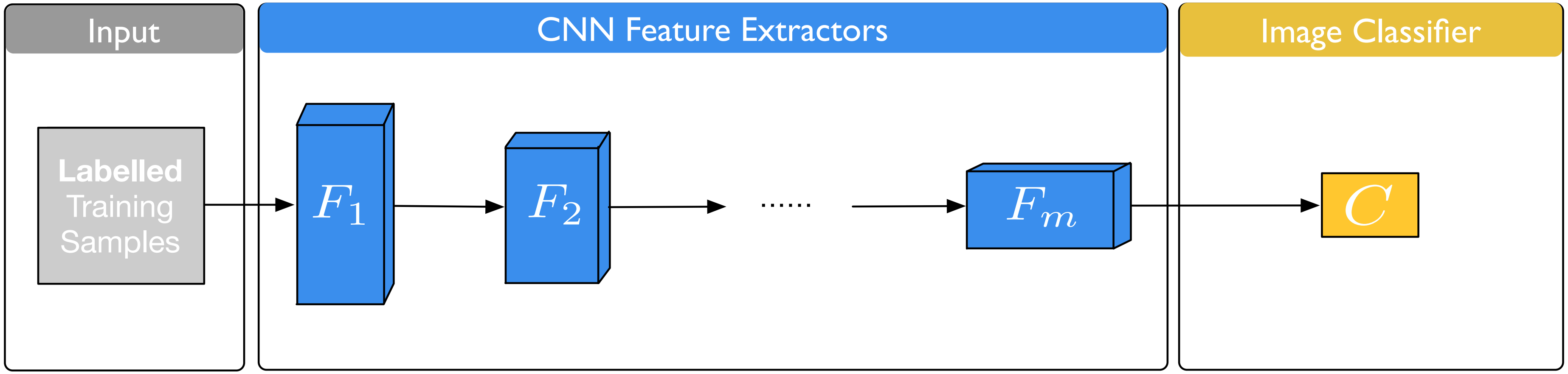
1.1 Load ImageNet Pre-trained models

- https://pytorch.org/docs/stable/torchvision/models.html

- Take AlexNet as an example:

```
import torchvision.models as models
alexnet = models.alexnet(pretrained=True)
```

- Modify classifier
  - self.classifier = nn.linear(1024,1000) -> self.classifier = nn.linear(1024,num_class)



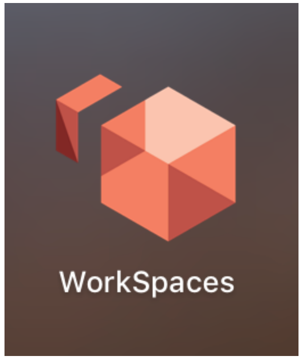
Pre-trained model Fine-tuning

1. Load Pre-trained Models (ImageNet)
2. Use Training samples to fine-tune

1. Re-build Classifier
2. Train Classifier

AWS Server

- 1.Installing and running
  - Check spam folder if you haven’t receive the email.



- 2. For someone having problems loading the models. Let me know. You might need to install older version Pytorch and torchvision.

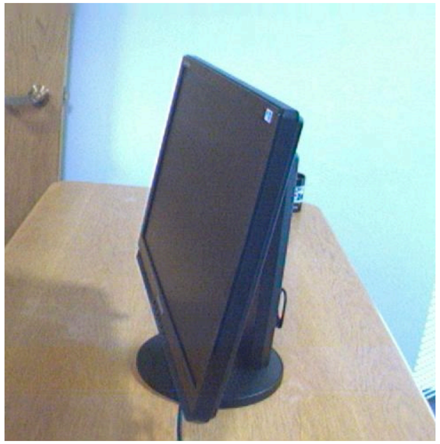
Task 2 Unsupervised Domain Adaptation

- Base on ImageNet Pre-trained models

Domain adaptation

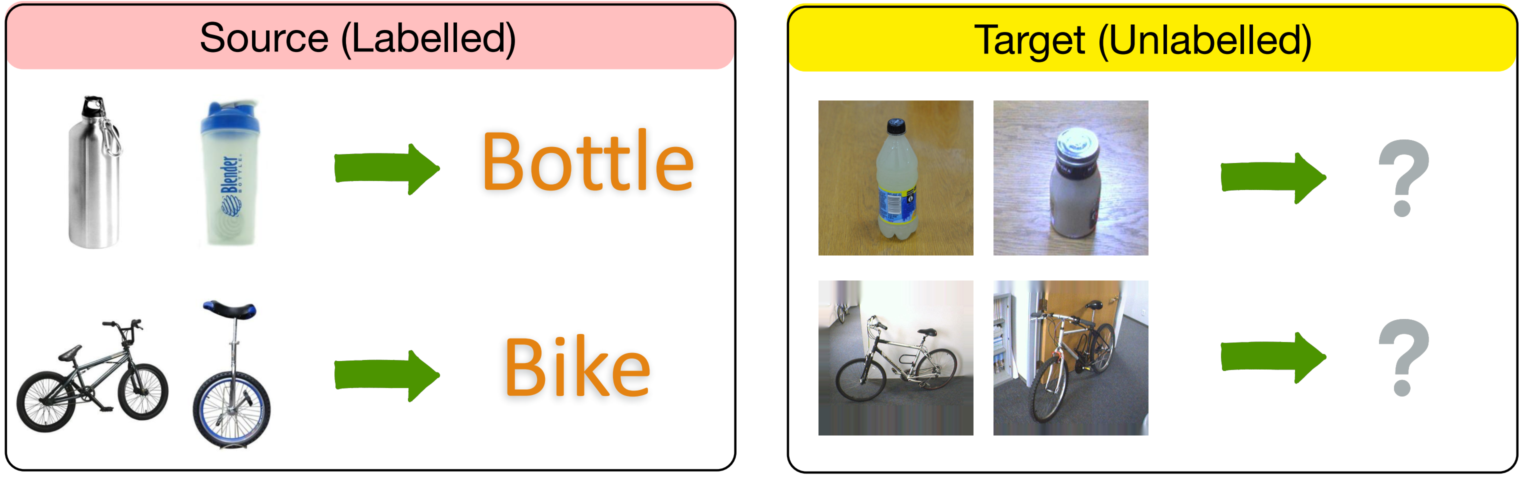
- In general, visual domains could differ in some combination of (often unknown) factors, including:
  - scene
  - intra-category variation
  - object location and pos
  - view angle
  - resolution
  - motion blur
  - scene illumination
  - background clutter
  - camera characteristics
  - etc…

Source Domain



Target Domain

Unsupervised Domain adaptation



Source (Labelled)

Bottle

Bike

Target (Unlabelled)

?

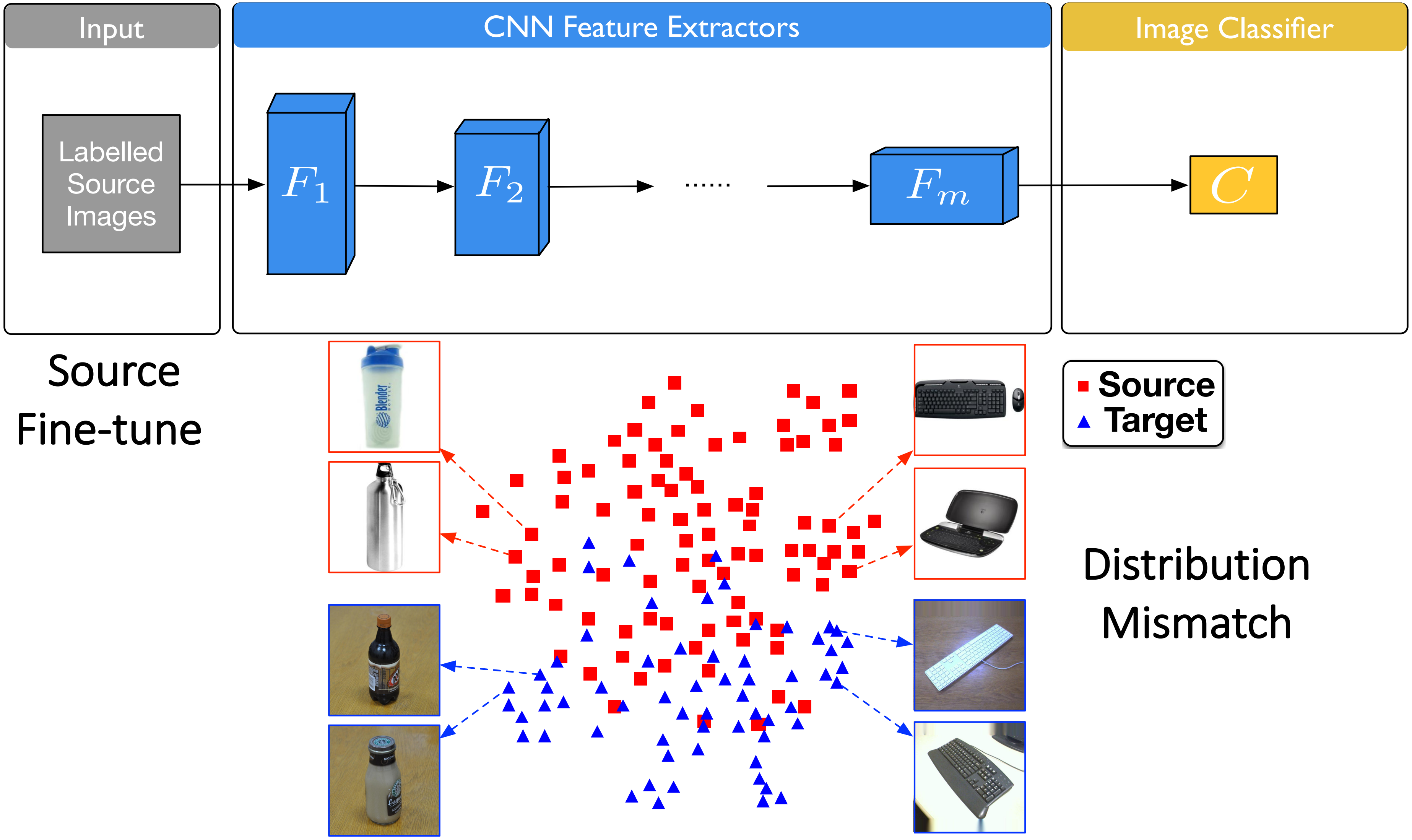
?

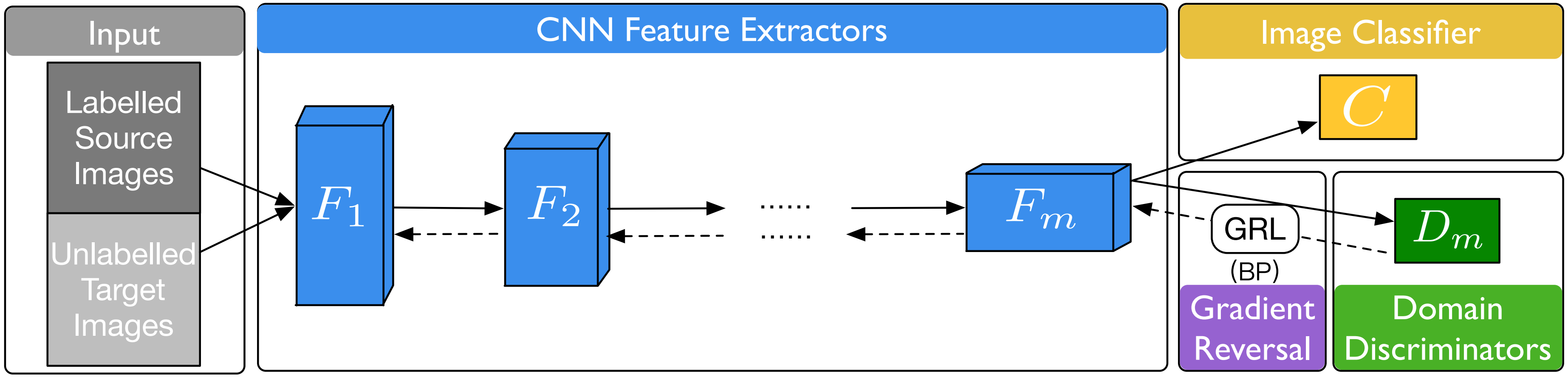
$$\mathcal{D}_s = \{(\mathbf{x}_i^s, y_i^s)|_{i=1}^{N_s}\}$$

$$\mathcal{D}_t = \{\mathbf{x}_i^t|_{i=1}^{N_t}\}$$

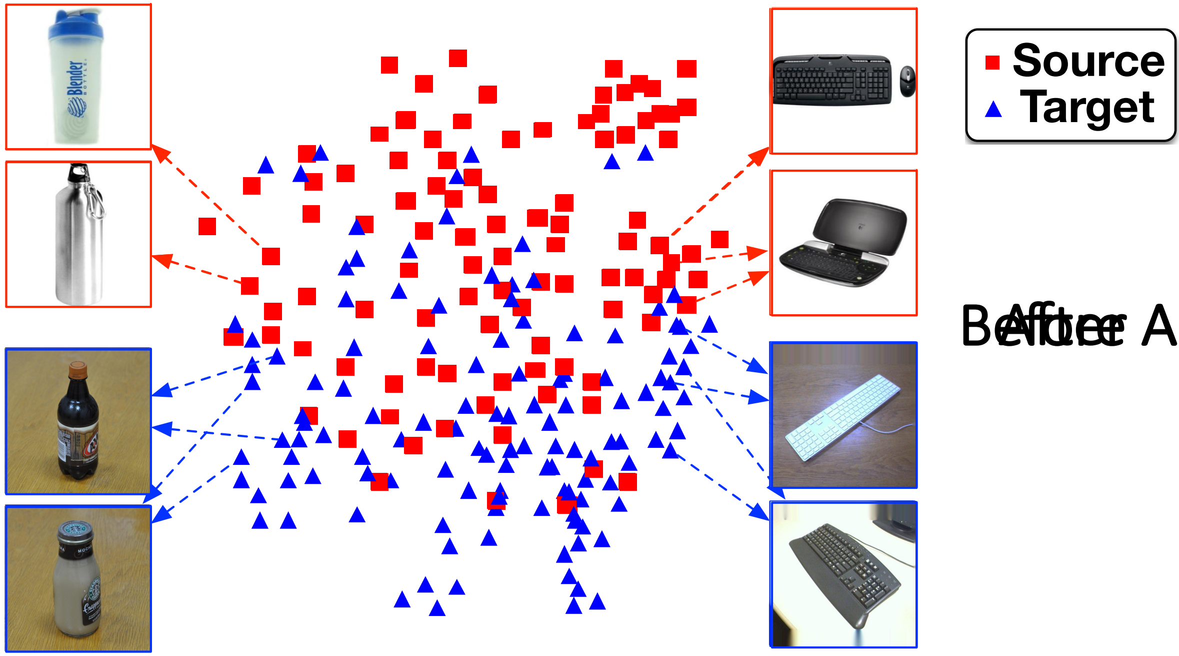
Domain Adaptation Techniques

- **1. Reverse Gradient (Domain Adversarial) [1]**
  - General Image Classification Network (Image Classifier)
  - + Domain Classifier (with Gradient Reversal Layer)


Input
Labelled Source Images
CNN Feature Extractors
F_1
F_2
......
F_m
Image Classifier
C
Source Fine-tune
Source
Target
Distribution Mismatch



Domain Adversarial Neural Network [1]



Before Adaptation
After Adaptation

Domain Adaptation Techniques

- **2. Retrain model with Pseudo-labels**

  In each epoch:

  - Train model 1 with Source samples
  - Evaluate Target samples with the trained model 1 and get the predicted class and probability of each target samples
  - Select High-confident Target samples (Define your own Selected Target Dataset)
  - Retrain model 1 with the Selected Target Dataset (using the predicted class and probability as labels).