Machine Learning - Final Project

Domain Adaptation

資工四 B05902021 徐祐謙
資工四 B05902023 李澤諺
資工四 B05902120 曾鈺婷

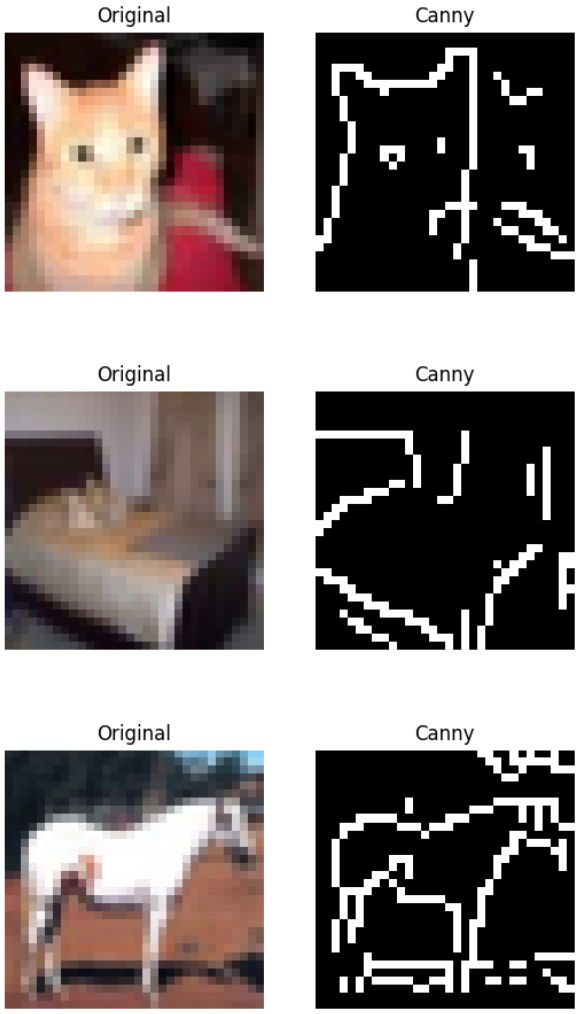
January 16, 2020

Introduction and Motivation

本次 final project 的題目爲：給定 10 個 class 的眞實照片，試著用其訓練出一個 model，以此將手繪的圖片進行分類。之所以選擇這個題目，是因爲 domain adaptation 爲 transfer learning 中一個很重要的問題：當我們想要處理的 domain (稱爲 target domain) 中沒有足夠多的 data，但是在另一個 domain (稱爲 source domain) 中的 data 充足，可以讓我們在 source domain 中解決相同的問題，那我們就可以使用 domain adaptation，透過在 source domain 中解決相同的問題，來幫助我們解決在 target domain 中的原問題。考慮到現實中蒐集 data 的困難度，我們不見得能在欲處理的 target domain 上蒐集到足夠多的 data，不過網路上已有許多現成的大型 dataset，其 domain 雖然可能會和我們想要處理的 target domain 不同，但是只要利用 transfer learning 中的 domain adaptation，仍能幫助我們解決原本的問題，由此可見 transfer learning 在現實中的重要性。

Data Preprocessing/Feature Engineering

(1) 因爲 testing data 爲手繪的黑白圖片，圖片中僅有物體的輪廓，而 training data 則是眞實物體的彩色照片，爲了讓 training data 盡可能和 testing data 相似，我們對 training data 使用了 Canny edge detection，將 training data 中物體的 edge 找出，如此一來 training data 便會在外觀上和 testing data 相似 (如下圖)，此時我們再用 training data 去訓練 model，便有可能提高 accuracy。



由於 training data 中除了欲分類的物體外，還有背景或其它物體等等，其經過 Canny edge detection 後也會留下輪廓，爲一種 noise，其對 model 的訓練會造成影響，爲了減少背景或不相關的物體所產生的輪廓，我們試著將 Canny edge detection 的 threshold 調大 (cv2.Canny(image, 250, 300))，將大部分的輪廓去掉，以減少雜訊。

(2) 由於 training data 的大小爲 32×32，而 testing data 的大小爲 28×28，因此我們有將 testing data 放大爲 32×32，再進行訓練與測試。

(3) 我們有將 training data 和 testing data 的值皆除以 255，使得 pixel 的值 normalize 到 0 和 1 之間。

(4) 在訓練過程中，我們會將 training data 和 testing data 皆經過 RandomAffine(10, translate = (0.1, 0.1), scale = (0.9, 1.1)) 和 RandomHorizontalFlip()，以增加 data 的數量。

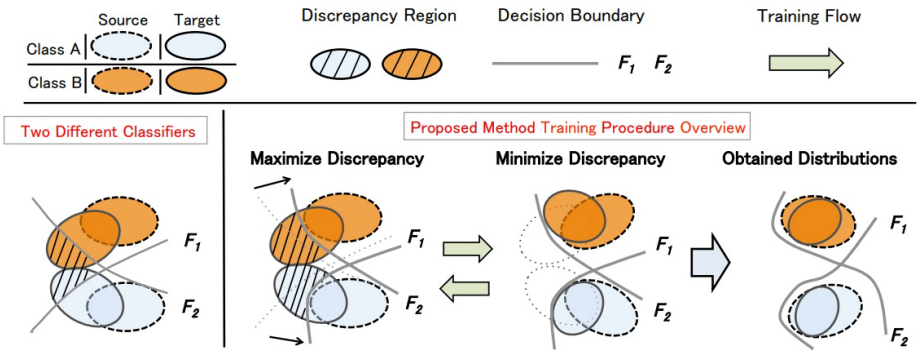
Model Description

我們所使用的 model 爲 MCD [ref]，其架構爲：前半部爲一個 feature extractor，其後接有兩個 classifier。訓練過程中，每一個 epoch 皆分爲三個步驟 (如下圖所示)：

(1) 訓練 feature extractor 和兩個 classifier，使得 feature extractor 要能從 source data 中找出夠好的 feature，並且兩個 classifier 皆要能以此將 source data 做出正確的分類。

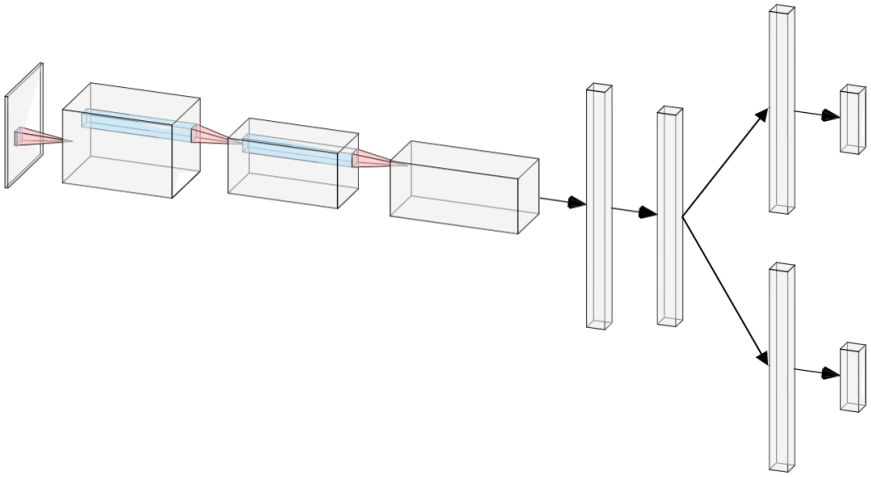
(2) 在固定 feature extractor 的 parameter 之下，訓練兩個 classifier，使得兩個 classifier 要在依然能正確分類 source data 的條件下，還要使得兩個 classifier 在 target data 上預測的 distribution 相差越多，增加兩個 classifier 的 decision boundary 之間的距離。

(3) 在固定兩個 classifier 的 parameter 之下，訓練 feature extractor，使得 feature extractor 在 target data 上所找出的 feature，經由兩個 classifier 預測後的 distribution 相差越多。



在經過如此的訓練過程後，可以期望 feature extractor 在 source domain 和 target domain 上所找出的 feature 可以彼此 align，如此一來只要 classifier 能夠將 source data 正確地分類，便也有可能將 target data 進行正確地分類。

接著，我們所實作的 MCD 架構如下：



MCD	
feature extractor	Conv2d(1 , 64 , kernel_size = (5 , 5) , stride = (1 , 1) , padding = (2 , 2))
	BatchNorm2d(64)
	ReLU()
	MaxPool2d(kernel_size = (3 , 3) , stride = (2 , 2) , padding = (1 , 1))
	Conv2d(64 , 64 , kernel_size = (5 , 5) , stride = (1 , 1) , padding = (2 , 2))
	BatchNorm2d(64)
	ReLU()
	MaxPool2d(kernel_size = (3 , 3) , stride = (2 , 2) , padding = (1 , 1))
	Conv2d(64 , 128 , kernel_size = (5 , 5) , stride = (1 , 1) , padding = (2 , 2))
	BatchNorm2d(128)
	ReLU()
	Linear(8192 , 3072)
	BatchNorm1d(3072)
	ReLU()
classifier	Linear(3072 , 2048)
	BatchNorm1d(2048)
	ReLU()
	Linear(2048 , 10)

我們使用了 Adam 訓練 MCD，其中 learning rate 爲 0.00002，weight decay 爲 0.0005，batch size 爲 128，訓練了 2000 個 epoch，以此得到 MCD。

Experiment and Discussion

除了使用 MCD 之外，我們也有試過將 training data 經過 Canny edge detection 處理之後直接給 CNN 進行訓練，以下爲 CNN 和 MCD 在 Kaggle 上分別所得到最高的 accuracy：

	Public	Private
CNN	0.41526	0.42044
MCD	0.80093	0.80145

由此可以看出直接使用 CNN 所得到的 accuracy 較差，其原因大致是因爲雖然我們讓 training data 和 testing data 在外觀上盡可能地相似，但兩者之間在 feature 上的差異仍然很大，必須使用 transfer learning 去找出 training data 和 testing data 在 feature 上的相同與相異之處，讓 training data 和 testing data 在 feature 上相似，才能讓 model 正確地將 testing data 進行分類。

Conclusion

Transfer learning 在現實中其實很常發生，經由本次的 final project 我們也對其學習到了不少，當 source domain 和 target domain 有所差異時，僅僅只是使用 data preprocessing 讓兩者的外觀相似無法得到好的 accuracy，必須使用 transfer learning 中的 domain adaptation 才能眞正學到兩個 domain 之間 feature 上的相同與相異之處，進而做出正確的分類，而在前三名組別的分享中，皆有使用到 pseudo-label (例如：若 model 在作測試時，認爲某一筆 testing data 爲某一個 class 的機率高於 threshold，就將該 class 當作該筆 testing data 的 pseudo-label，或是有組別將 testing data 進行 cluster，分爲 1000 個 group，若 model 預測某一個 group 之中某一個 class 所佔的比例高於 threshold，就將該 class 當作這個 group 之中所有 testing data 的 pseudo-label)，使用 pseudo-label 訓練 CNN，可以得到更高的 accuracy，我們認爲其原因爲：transfer learning 在 target domain 上是完全沒有 label 的，對 target domain 是一無所知，而若我們相信 pseudo-label 的正確性，則此時我們在 target domain 上就有 label，知道一部分的正確答案，就可以進行 supervised learning，比其它的學習方法略勝一籌，由此可見 pseudo-label 的重要性。總之，經由本次的 final project，我們學習到了 transfer learning 的基本流程，收穫著實頗豐。