

ML Final Report

學號：B05902017系級：資工四姓名：王盛立

學號：B05902104系級：資工四姓名：王普禾

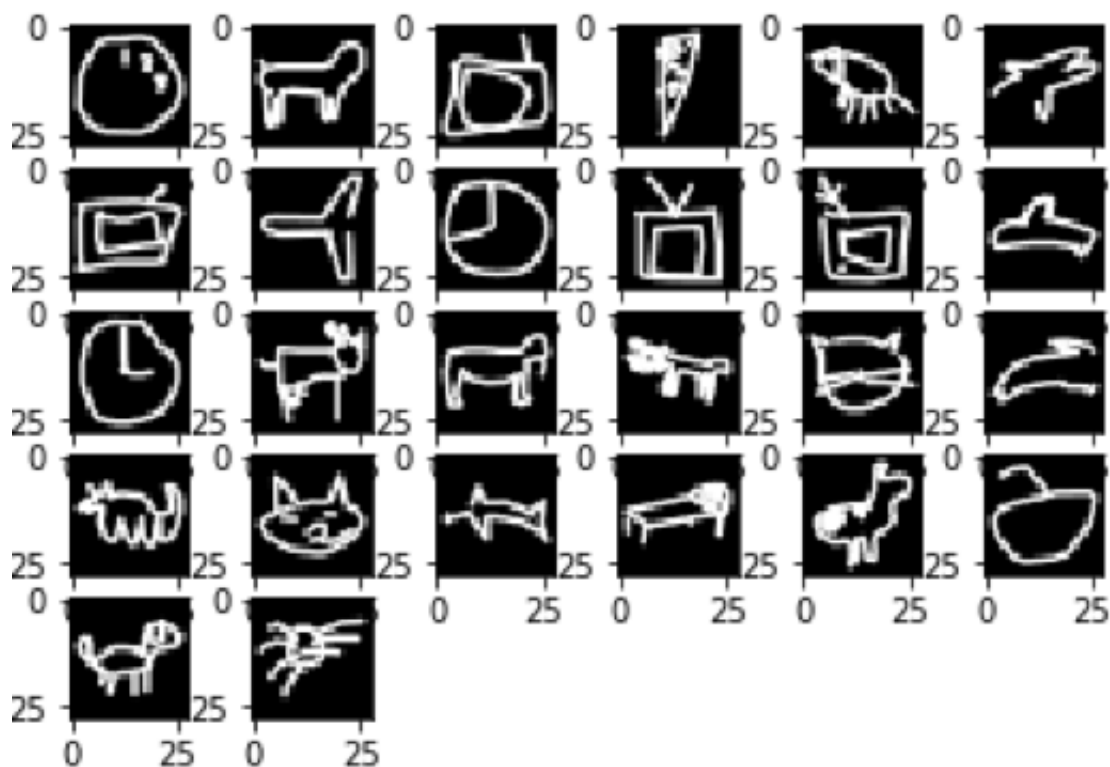
學號：B05902105系級：資工四姓名：余友竹

Introduction & Motivation

在經典的機器學習問題中，我們往往假設訓練集和測試集分佈一致，在訓練集上訓練模型，在測試集上進行測試。然而在實際問題中，測試的場景往往無法控制，測試集和訓練集分佈有很大差異，這時候就會出現模型在測試集上效果不理想的狀況。所以此次Final作業想探討的Domain Adaptation就是這個問題。



training的資料是彩色完整的圖片



testing的資料是只有邊緣的黑白圖片

我們會選擇這個主題的原因是我們對影像的實作是比較有興趣的，像是作業三就讓我們感到比較有趣。而影像辨識在未來也是重要且熱門的，所以希望能利用final這個作業來多認識機器學習在CV上的應用。

Data Preprocessing/Feature Engineering

我們將testing的資料補齊到和training相同大小。另外由於training是彩色的，因此我們將testing的部分複製成3個channel，令training和testing在大小是一致的。我們也有做一些data augmentation，讓他隨機的做水平的翻轉和一點點角度的偏移。一開始的時候我們有做canny edge detection來試圖讓training的資料和testing的資料比較相似，不過後來我們實作了一篇論文的方法，就沒有再使用canny了，因為它反而會使結果更差，詳細關於那篇論文的說明我們會在接下來講。

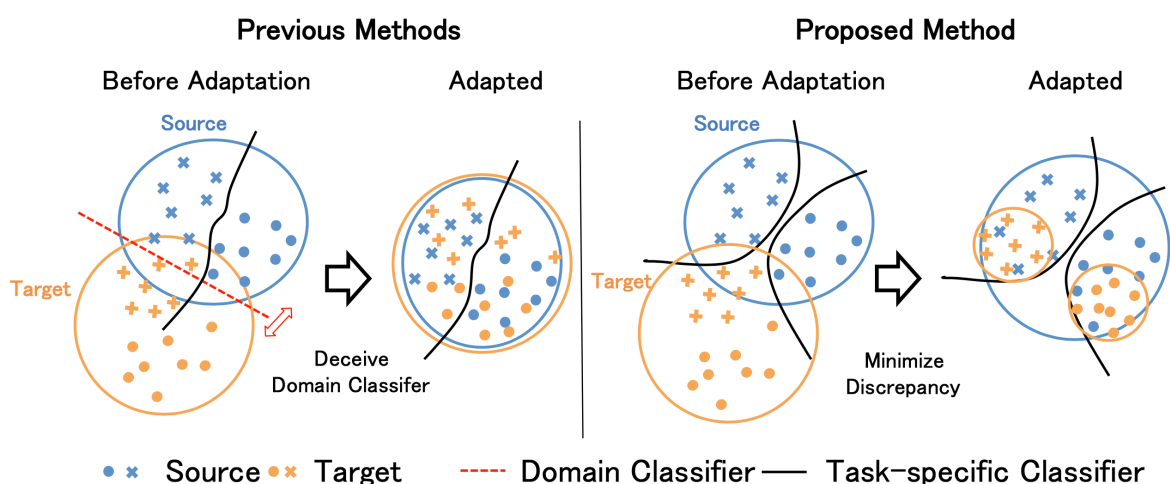
Model Description

首先我們有試過先train一個簡單的CNN model，並在testing上先用autoencoder+pca的方式將資料分成十份。接下來將剛剛簡單的model拿來預測這十類分別是什麼，這個簡單的方法只能勉強過simple，效果並不是很好。我們也有嘗試過使用canny edge detection。首先他是先將圖片轉成灰階，並利用 Gaussian Blur 去除雜訊。之後取得圖片每個 pixel 的梯度值和梯度方向，利用Non-maximum suppression尋找可能的邊緣後，根據兩個閾值選取 strong edge（確定的）和 weak edge（進一步判斷）。最後選取和 strong edge 相連的 weak edge 當作確定的 edge。這部分的演算法其實已經在cv2中有完整的套件了，實做出來的圖會長得像下圖，可以看到我們取得了圖的邊緣。在這次的資料中，可以看到training是完整的圖片，而testing是黑白的輪廓圖。其實就有一點像training的資料去取得他的邊緣後所得到的圖。不過這個方式表面上看

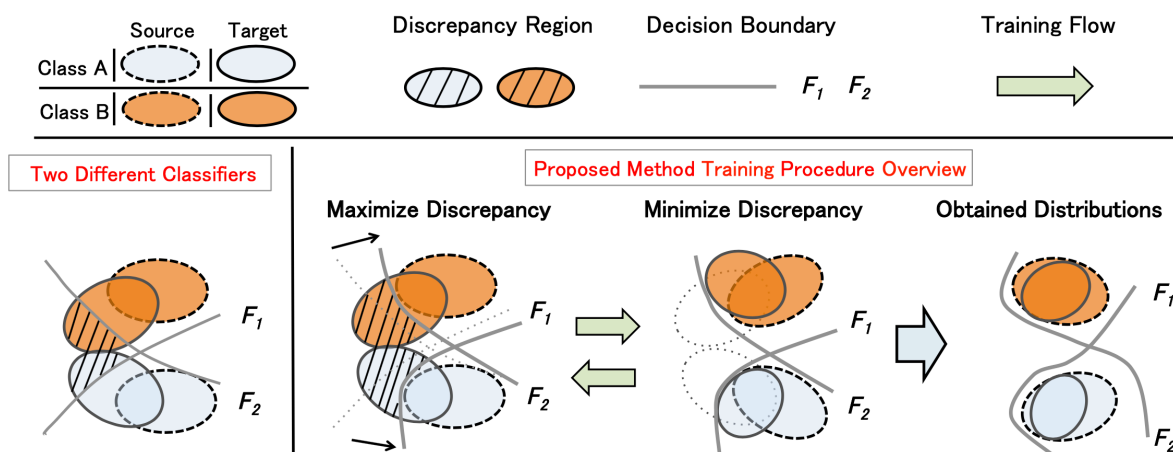
起來很合理，但最後的結果依舊沒有很好。



後來我們實做了**Maximum Classifier Discrepancy for Unsupervised Domain Adaptation**這篇論文的方法。他的主要概念是這樣的，在許多做domain adaption的方法中，其實都只是將兩個domain的特徵盡可能地對齊，但其實忽略了每個domain的特點。

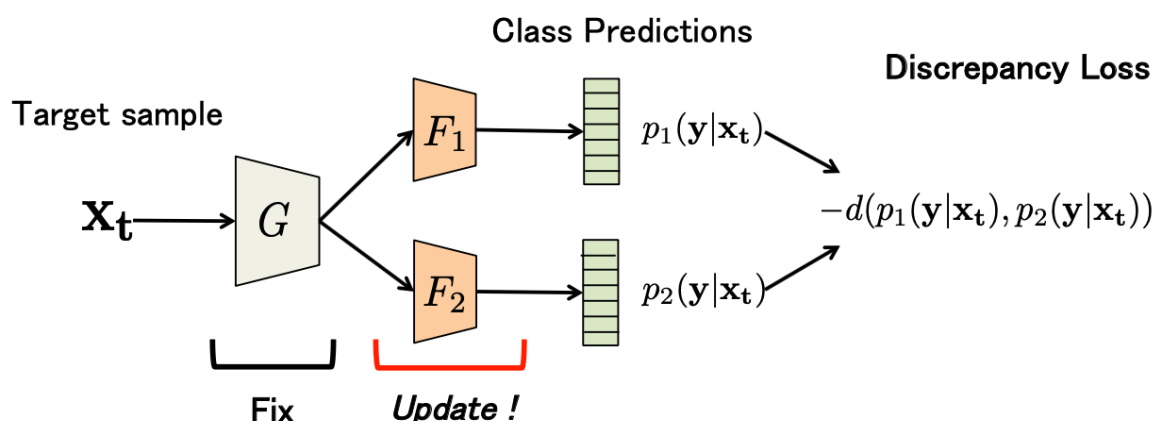


而他的訓練過程是這樣的。我們首先會有generator和classifier兩種model，generator的工作是提取資料中好的特徵，classifier則負責把資料做正確的分類。如下圖所示我們有兩種分布的資料Source和Target，在我們的例子裡就是training和testing。他們分別會有好多個類別，我們先假設只有兩個類別Class A和Class B。首先我們先訓練兩個Classifier F_1 和 F_2 ，這兩個Classifier都要先滿足在source也就是training的部分做得好。接下來我們把兩個classifier應用在target也就是testing的部分，如果兩個classifier對於testing都預測相同我們就不管他，重點是兩個classifier預測不同的部分，也就是下圖有斜線的區域。兩者有分歧代表可信度不高，需要進一步的訓練。透過提取比較好的特徵使得這兩個classifier能更加正確的分類在testing上的資料。

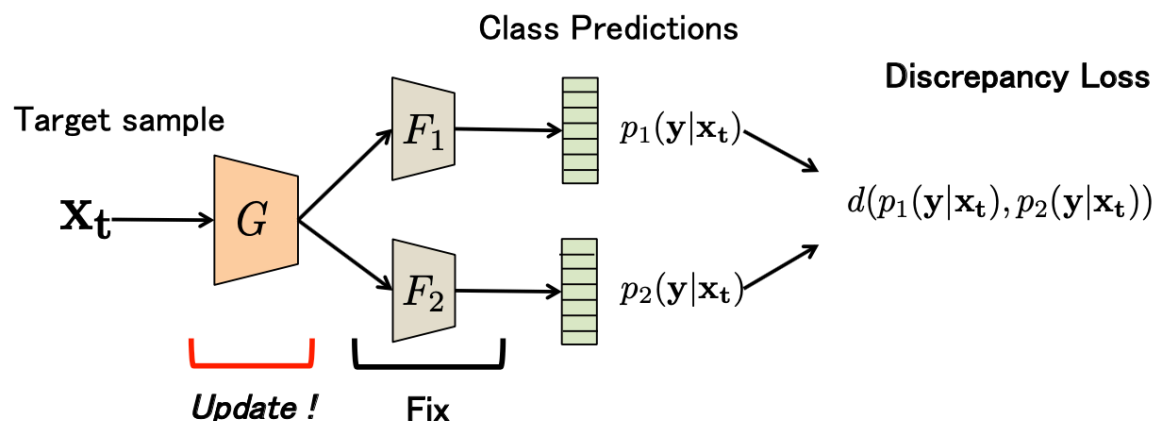


在下面這張圖中我們可以更明確的看到訓練的過程，Step A是先訓練好兩個好的分類器 F_1 和 F_2 ，接下來Step B我們先固定提取特徵的generator，去改善兩個分類器 F_1 和 F_2 ，令他們輸出的預測越不像。越不像代表那些資料都有可能是錯的可能需要進一步的分類，Step C就是改進generator讓我們能抽出比較好的特徵，讓兩個分類器 F_1 和 F_2 能夠預測出正確的類別。

Step B : **Maximize** discrepancy on target (Fix G)



Step C : **Minimize** discrepancy on target (Fix F_1, F_2)



而具體在訓練的過程中，他是將兩者差異叫做discrepancy loss，定義為

$$d(p_1, p_2) = \frac{1}{k} \sum_{k=1}^K |p_{1k} - p_{2k}|$$
。 p_{1k} 代表 p_1 對於被分類為class k的機率。而之所以選擇 L_1 distance則是實驗的成果。

Experiment and Discussion

我們一開始的方法包括了使用簡單的CNN，之後加了像是Canny的邊緣偵測，和用Pca先分類testing上的結果。這些方法的正確率都不高，約在二十幾個百分點上下。

後來我們就改成了使用上述那篇論文MCD的方法，在我們實作的方法中有幾項參數是可以做調整的。首先是我們使用SGD，因此我們可以去調整他的learning rate。當然我們也可以改成使用adam等等方式。另外在他的演算法當中可以分成三個步驟，而第三個步驟中也就是上圖minimize discrepancy那邊，可以調整要他更新幾次。當然還有epoch跟batch的大小也是可以做調整的。在我們實驗多次過後我們發現要使learning rate稍微大一些些，會得到比較好的效果。不過整體而言我們實做的這個方法能調整的參數其實不算多，調整過後也頂多差一點點而已。整體的正確率大概是50幾個百分點左右。

差距比較大的部分就是最後我們調整了模型的架構，在許多層間加了**dropout**。在domain transfer中，我們可以很輕易的在training的過程中得到很好的結果，但有可能是學到了我們不想要的東西因此在testing上得到的效果會低很多，我們認為這就像是某種程度上的overfitting，因此加dropout後能使結果變好一些。在實際實作上也得到了很好的效果，比沒有加dropout的版本好了十幾個百分點。最後約可以達到70%的正確率。

Conclusion

這次作業讓我們認識到了domain transfer這個領域，我們可能無法取得大量我們想要的資料來做訓練，但透過domain transfer我們可以從另一個已經有大量資料的領域，來推廣到我們想要的領域。我們覺得這個主題是很吸引人的，在實做的過程當中一開始我們是毫無頭緒，不過後來當我們看了幾篇相關論文後，才慢慢瞭解到這個領域的博大精深，以及一些相關的技巧像是對抗網路等等。最後我們選擇了MCD這個方法，實做出來的效果也真的不錯。不過我們還是有許多可以改進的空間，例如我們其實沒有去修改他model的結構，以及由於我們的狀況是有十類，或許我們可以先將圖片分成兩大類再不停地去細分等等。另外就是這個領域其實有非常多的論文，如果我們有更多的時間或許我們還能嘗試看看別種做法，不同的方法ensemble起來或許能達到更好的成果。

Reference

Maximum Classifier Discrepancy for Unsupervised Domain Adaptation

Kuniaki Saito¹, Kohei Watanabe¹, Yoshitaka Ushiku¹, and Tatsuya Harada^{1,2}

¹The University of Tokyo, ²RIKEN

{k-saito,watanabe,ushiku,harada}@mi.t.u-tokyo.ac.jp

http://openaccess.thecvf.com/content_cvpr_2018/papers/Saito_Maximum_Classifier_Discrepancy_CVPR_2018_paper.pdf

https://github.com/mil-tokyo/MCD_DA