科別:

組別:高中組

作品名稱:GAN你又亂輸學號了

關鍵詞:CNN、VGG

編號:

**摘要：**

**壹、研究動機：**

　　本校曾經施行過遲到刷學生證來記錄遲到的機制，並且對於未帶悠遊卡的學生以輸入學號為替代方法，但會有部分學生輸入錯誤的學號來做驗證，希望能開發一款具有實際辨識意義的系統來防範，由於臉部辨識會牽扯到個資安全，因此決定改以辨識校服為替代來當作辨識依據。由於本校校服並未強制做字體、位置等規範因此無法用單一簡單的演算法做識別，於是使用目前圖形識別架構ＣＮＮ來完成功能，並在之後能以ＧＡＮ來做輔助達到ＣＮＮ無法做到的圖形清理。

**貳、研究目的：**

　　為了能在有限測資下完成功能因此規劃了以下步驟：

1. 使用open-cv將電腦字體複製到校服背景的圖片上做模擬
2. 用python將讀進來的做清理和整理
3. 使用CNN架構下做預測
4. 探討各種架構下的成效
5. 處理overfitting問題
6. 實際實驗真實校服的比較
7. 分析出何種圖片對辨識最有利，並提出ＧＡＮ的模型

**參、研究設備及器材：**

一、硬體

(一)桌上型電腦(ＣＰＵ：i7-7700K；ＧＰＵ：ＲＸ５８０；記憶體：16G)

二、軟體

(一)ubuntu 18.04 ；windows 10(作業系統)

(二)python 3.6 (程式語言)

(三)keras (深度學習框架)

(四)tensorflow (深度學習框架)

(五)google colab (google 運算資源)

(六)相關套件：opencv；matplotlib；

**肆、研究過程與方法：**

**一、**模擬測資：

（一）製作原因

　　因為Deep learning的精確度會反映在訓練集大小上面，因此使用Deep leaning時需要大量的訓練圖片，但因取得困難因此使用模擬的方式製作。

　　ＣＮＮ是由捲積層（convolution）後面連ＮＮ層（neural network）所完成的。捲積的做法是先定義一組filter由多個或單一kernel 組成並且對圖片做向量內積，將符合特徵的圖片區塊做保留此外的做移除。因此如果此模型能成功運行相信在有真實的測資時一樣能訓練起來。

（二）製作方法

　　先將本校校服做背景拍照，製作三張原始圖片，之後使用

ｏｐｅｎｃｖ讀入圖片並且以最近鄰近點插值法來做圖片尺寸縮放，在圖片上隨機位置插入隨機學號並將結果和答案標籤存起來參考圖１。

二**、**CNN１號模型：

（一）模型架構

參考圖３，１號模型使用ＶＧＧ做改良，將輸入層改為邊長１００的正方形，並將輸出層改為９ X １１的輸出標籤，代表學號的１～９個數字中０～９和沒有學號的標籤，系統使用windows10並使用tensorflow 13.0.0做深度學習架構 。

（二）定義Ｌｏｓｓ函數、優化器和單次迭代數量

本模型使用Cross-Entropy做Loss函數並搭配tensorflow 裡面的AdamOptimizer 做優化器，單次迭代數量（batch size）由於本身的數據數量不多加上記憶體限制設定為64。

（三）參數初始化與激活函數

本模型使用Xavier做初始化將所有的同層的wieght以做正態分布，以Relu做激活函數。

（四）圖形處理

將輸入進來的圖形做灰階化和縮放到適合的尺寸，並對讀入的圖片做零均值化（zero-mean）。

三**、**CNN２號模型：

1. 針對１號模型提出的解決

１：將系統轉移到ubuntu 18.04上面並為了搭配AMD顯示卡轉為keras為深度學習架構使用Rocm做優化。

２：將輸出層從單一輸出層改為9個輸出層分別代表學號上9個數字，每個輸出層大小為11代表數字1~9和沒有數字。

1. 模型架構

參考圖４，除了上述的輸出層更改，由於本機記憶體有限因此將fully connected層（以下簡稱ＦＣ層）改為１０２４並將convolution層（以下簡稱CN層）縮小為0.5倍。

1. 定義Ｌｏｓｓ函數和優化器

跟上述tensorflow使用相同的優化器和模型，但由於keras自帶參數輸入因此並無調整learning rate等部分。

1. 參數初始化與激活函數

　　將每層的weight正態分布在正負0.1並且總和為0，激活函數在CN層使用relu，防止網路過深失去梯度作用，FN層一樣使用relu，並在輸出層使用softmax 。

1. 處理過擬合問題

１：在圖片輸入前增加batch normalization (以下簡稱BN層)

２：在進入ＦＮ層以前先經過Dropout切斷５０％的連結層

３：設定正規化(regularization)L2 norm

伍、研究成果：

**一**、CNN1號模型：

　　從圖５可以發現訓練15個epoch後精確度卡在3.9 並持續無明顯上升因此判斷模型1很可能遭遇到了未擬合的問題並因為google colab的限制無法再做大變動的修改因此研發模型二。

二、CNN1號模型：

　　從圖7可以發現在10個epoch後訓練資料的精確度仍在上升並且在訓練超過40個epoch後達到了接近95%的準確度，但測試資料卻卡住無法繼續上升，由此判斷遇到了過擬合的問題因此插入了ＢＮ層並且在ＦＮ層時實作了Droput和regularization，參考圖8在同樣的epoch下確實將訓練資料和測試資料同時提升但精確度卻卡在45%，而在持續的訓練到100 epoch時訓練資料精確度卻因為正規畫被卡在54%而測試資料被卡在48%無法在提升。

三、分類結果

　　由訓練結果可得知在CNN的辨識狀況下前兩碼也是班級號碼再整體的預測是最為精準的，而在測試資料裡前兩碼也將進達到90%，在而後面的數字依序下降，因此在真實環境實作時可以考慮將前兩碼做忽略不計算的操作，縮短整體模型大小。

四、資料簡單化

　　原始測試圖片請參考圖１，而資料簡單化後圖片請參考圖２，在訓練同樣40epoch整體精確度請參考圖9 ，整體正確性達到了接近100%，不管是訓練集或是測試集的精確度都有極大的提升，因此在未來實作ＧＡＮ模擬時也能標示出邊框回歸（bounding box）或是直接做裁切的模擬。

五、ＧＡＮ模型

　　由上述等問題和現實中可能愈到的資料模糊或是遮蔽時需要以極少數的特徵來模擬完整學號，因此建議使用DCGAN或是cycle-GAN來做資料的清洗，或是使用autoencoder來做模糊化的消除。

陸、討論：

一、整理CNN1號模型問題原因：

(一)　　VGG模型所吃的資源過大在沒有顯示卡支援的windows系統下無法運行完畢，因此搬到Google colab上運行但，。

Google colab 的運行也有限因此在當次訓練時整個epoch所包含的資料量太小導致underfitting的狀態。

（二）　　將所有標籤儲存在同一個tensor裡面會導致輸出層無法做到softmax的功能並且loss值會受到整體結果的影響加上上述原因使得單個batch size過小而有underfitting的問題。

二、整理CNN2號模型問題原因：

（一）　　模型深度高達１６層導致整體model過於擬合於測試資料。

（二）　　生成圖片按照特殊特徵排序導致訓練資料規則分布，產生過擬合的問題。

（三）　　輸出層精元過多導致計算梯度時影響了整體的model，所以導致在前幾個的數字預測時能達到極高的預測水準但之後的確差強人意。

三、過擬合的處理

　　１：在圖片的輸入和進入ＦＮ層前和一開始都先經過了ＢＮ層的修正，ＢＮ層的實作原理為將整次跌代的資料做減去平均在除平均值的長度，也就是將整個資料求單位化。

　　２：在所有FN層使用Dropout，實作為將整層網路中挑出50%的神經元做運算，由每次不一樣來做到隨機更動降低過擬合的可能。

　　３：時做regularization時將所有最後做梯度的loss值加上regularization term，類似lagrange 乘數法將整體求loss值函數最小值限定在一定範圍內防止整體模型過於複雜。

四、資料簡化後表現良好原因

　　從ＣＮＮ的基礎來看模型預測是先經過捲積在進入全連接層做預測，而大小不一的圖形會導致整張圖出現在同一個捲積或是一個捲積裡，進而導致單一捲積無法維持一個特徵的大小，所以在觀察整訓練一個epoch時時常可以發現loss值晃動極大，因為一個捲基層無法預測特適合所有圖片的特徵做預測。

五、修正誤差

（一）　　數字大小不規則：在輸入前先將整體的數據做標示，並且從新標定尺寸，或是使用ＹＯＬＯ的模型做預先的bounding box預測在進行捲積。

（二）　　模糊遮蔽：使用DCGAN或是cycle-GAN來做預測被遮蔽的校符實際學號應該是多少，或是可以使用autoencander減少因為相機模糊而產生辨識度過低的照片。

**柒、結論：**

　　本研究想找出一個模型能辨識出校服，由於此問題屬於影像處理所以使用ＣＮＮ架構來完成，本模型使用原本用來標示圖片種類的模型ＶＧＧ來做更改，並因應電腦性能和測試資料更改層數和大小並新增優化。

　　並因為實際無法得到足夠的測試資料和個資問題，因此使用模擬資料來完成模型，相信在擁有足夠大的真實資料時也能擁有跟模擬資料一樣的表現。

由最終的模型可在圖像清晰時辨識高達９８％並且在圖像隨機化時辨識度達到４８％，並且在未來希望能架設出ＧＡＮ模型幫助資料做初始化。

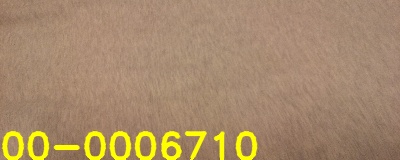


圖１隨機學號圖



圖２精簡化學號圖

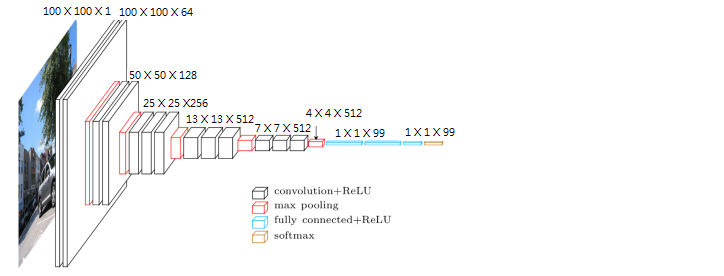


圖３ＶＧＧ架構修改後的架構圖

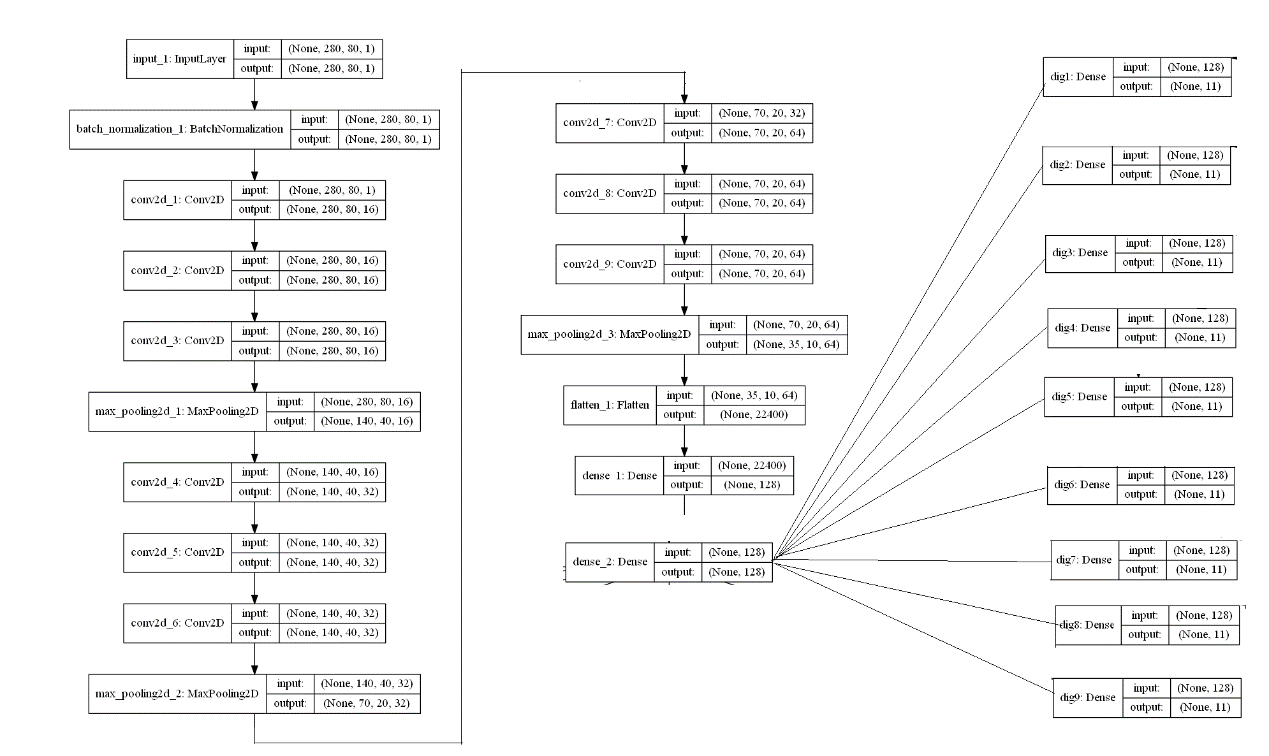


圖４ＣＮＮ模型二的架構圖

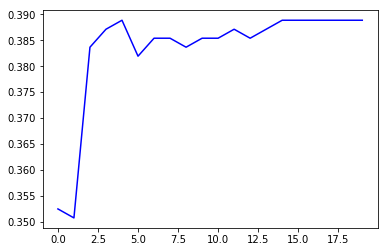


圖５ＣＮＮ模型一的訓練結果

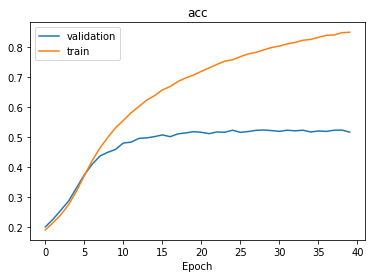


圖６未經過處理的ＣＮＮ模型二訓練結果

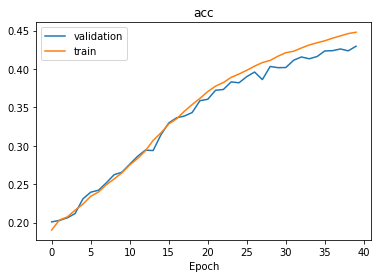


圖７經過處理的ＣＮＮ模型二訓練結果0~40epoch

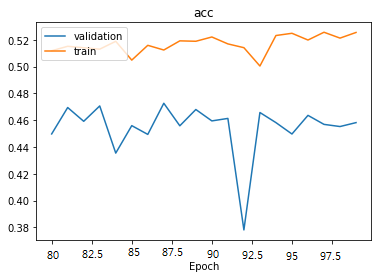


圖8經過處理的ＣＮＮ模型二訓練結果80~100epoch

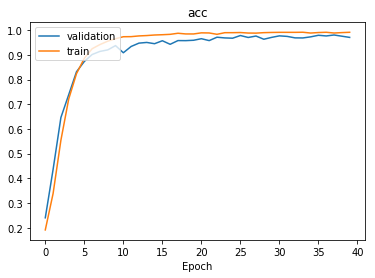


圖9經過資料精簡化的CNN模型二訓練結果

**捌、參考資料：**