台北市立大學 機器學習概論 期末專題計畫書 **Handwritten Equation Recognition 手寫數學方程式影像識別**

U11003008 物化系三 邱月之

**Introduction引言**

這項工作致力於研發能夠辨識手寫數學公式的程式。訓練了一個卷積神經網絡模型 (Convolutional Neural Network , CNN)。使用Tkinter模組來設計使用者介面，使用者可以在介面上書寫文字或上傳照片，程式能成功地提取影像中每個字元的輪廓與特徵，並輸入模型中進行辨識。目前程式能夠辨識加減與數字等基本符號，還能透過計算相鄰字元中點的距離，判斷上標、下標與分數等數學格式，並將這些特殊數學格式轉換為LaTeX代碼輸出。

**Motivation動機**

手寫數學公式辨識技術能夠將傳統紙筆手寫公式轉換為文件或程式代碼等電腦可以識別的格式，在科學教育、學術研究等領域中，有著潛在的應用價值。在數位教育方面，由於線上教學的普及，數學公式辨識技術能夠讓研習理工科目的學生在網路上更快地搜索到相應的學習資源與課外教材，提高教學效率和學習成效；在學術研究方面，有些科學研究工作會涉及數學公式分析，研究工作中留下大量的手寫紙筆資料，若能使用手寫數學公式辨識技術進行後續的文書與數據處理，將能大大節省時間並提高工作效率。

LaTeX是一種強大的排版系統，對於技術文件特別有用，尤其是在工程領域中重現高質量數學方程式的能力。對於研究生或教授來說，能夠從教科書、技術論文、家庭作業等中復制現有的方程式到他們自己的作業、解答或論文中，非常實用與方便。然而，一旦文檔以這種格式呈現，如果沒有原始製作者的代碼，就無法重新創建它。自己重新編碼一個冗長的方程式既耗時又容易出錯。

市面上已經發行了非常多軟體，如：MyScript軟體可以線上書寫與編輯數學公式，轉換成輸入文字，還能即時計算出結果；Mathpix Snipping Tool的螢幕截圖工具方便用戶截取文件或照片上的數學方程式，複製成LaTex指令或是Microsoft Word文件，方便用戶進行文書處理；甚至連Google Len的也有手機相機辨識數學方程式，轉換成LaTex代碼並在網路上搜尋的功能，方便學子在網路上搜尋解答與學習資源。然而，就算目前市場已經有相當的發展，此技術仍存在一些不足之處，包括不同書寫風格、字跡與多行公式之間的重疊等問題，會使軟體會出現辨識上的失誤。

**Objectives目的**

本專題的目標是利用Tkinter模組開發一款能夠直接在視窗上書寫方程式的視窗，還能夠支援使用者上傳照片，並具備去除照片背景雜訊的能力。該程式能夠截取照片上的字元與數字，分別辨識這些字元與數字並標註其位置。最後，程式將這些數學方程式影像轉換為LaTeX代碼，以便於使用者進一步編輯和使用。

程式要具備辨識上標、下標、分數等常用排版，若是成功，程式要進一步能辨識積分、求和符號、連乘符號等數學上的特殊用法。有了上述這些功能，程式在處理數學方程式時能將能夠貼近人類書寫的要求。

**文獻探討**

Jim Brewer , James Sun . *LATEX Generation from Printed Equations* . Department of Electrical Engineering , Stanford University . Retrieved from <https://web.stanford.edu/class/ee368/Project_Autumn_1516/Reports/Brewer_Sun.pdf>

文獻中的工作能夠將印刷體方程式的照片或截圖轉換成LaTeX代碼表示。同樣使用了二值化方法對圖像進行預處理，也利用提取輪廓用以對圖像的每個符號進行分割。不同之處是，文獻中的工作多了對傾斜的圖像進行旋轉校正的功能。大多數等式都有多條水平線，例如分數線、等號和負號，他們使用霍夫變換來求得方程式的水平方向。

關於單一符號影像辨識，使用了多種特徵提取的方式，其中包括：標準化中心慣量矩、圓形拓撲結構和Hu不變矩，每個符號會表達成二十二個元素的特徵向量。事先建立了模板數據庫儲存了標準字符的資訊，使用K-近鄰演算法 (KNN) 讓每個輸入符號對模板數據庫中的字符進行匹配和分類。

這些結果以及輪廓位置資訊一起傳遞給方程式組裝函數。此方程式組裝函數使用原理與我們的工作相當類似，使用輪廓邊界框的位置，計算相鄰字元中點的距離。判斷流程也包含了：(1) 上下標判斷；(2) 字符若是“”，則需判斷是否為“”或著分數格式。若是字符 “”，若輪廓邊界框與另一個字符 “”重疊，則判斷為“”；輪廓邊界框與很多其他字符的邊界重疊，將會判斷其為分數與它的分子和分母。

優點：

1. 這項技術對於需要將印刷文本轉換為可編輯代碼的情況非常有用，特別是對於學術文件中的數學方程式。
2. 大多數等式都有多條水平線（如分數線、等號和負號），因此利用霍夫變換來進行旋轉校正方法廣泛適用於大多數的數學方程。

缺點：

1. 報告中提到的系統主要針對印刷方程式，對於手寫方程式的識別能力可能有限。
2. 對傾斜的圖像進行旋轉校正，需要取得水平線，對於某些特定情況，比如符號“ ” ，算法可能會受到干擾，得到錯誤的判斷結果。
3. 字符匹配依賴於預先建立的模板數據庫，如果數據庫不夠全面，可能會影響識別效果。
4. 並非所有印刷體格式都是一樣的，使用輪廓邊界框重疊的方式進行分數或等於符號的判斷，將會有出錯風險。

**方法**

A. 使用者介面

本程式的使用者介面設計參考了GitHub 「Handwritten-Equation-Solver」[1]，設計了一個Tkinter視窗，使用者可以在其上書寫式子，按下 "save" 按鈕，程式讀取目前的數學式影像；再按下 "predict" 按鈕，即可顯示出預測結果。目前程式可以辨識包含數字、加減、等於符號的單行數學方程式，如圖一所示。對原程式進行修改，使程式可以標註各個數字的位置 [2]，如圖二所示。還增加照片上傳的功能 [3]，使用者按下 "開啟圖片" 按鈕，即可從本機選擇照片，程式將會對圖片進行預測，並將預測結果打印在右下角。

B. 訓練CNN模型

訓練資料集是從Kaggle 「Handwritten Math Symbol Dataset」[4] 下載，該項目有數字 0 到 9與其他常用數學符號，估計約有 47,000 張圖像。每張影像的大小為 像素，影像格式為 JPG。使用卷積神經網絡 (Convolutional Neural Network , CNN) 演算法訓練模型，使用 Keras 的 Sequential 類建立了一個卷積神經網絡，包含兩個卷積層 (Conv2D)，用於提取圖像特徵；兩個池化層 (MaxPool2D)，用於減少模型的計算量和過擬合風險；兩個全連接層(Dense)，用於分類。詳細架構如下：

1. 第一個卷積層，具有 30 個濾波器，每個濾波器的大小為 5x5，使用 ReLU 激活函數。
2. 第一個池化層，池化窗口大小為 2x2。
3. 第二個卷積層，具有 15 個濾波器，每個濾波器的大小為 3x3，使用 ReLU 激活函數。
4. 第二個池化層，池化窗口大小為 2x2。
5. 一個 Dropout 層，隨機丟棄 20% 的神經元以防止過擬合。
6. 一個展平層，將多維輸入展平為一維。
7. 第一個全連接層，具有 128 個神經元，使用 ReLU 激活函數。
8. 第二個全連接層，具有 50 個神經元，使用 ReLU 激活函數。
9. 最後一層輸出層，具有 13 個神經元（對應於 13 個分類），使用 softmax 激活函數，將輸出轉換為概率分佈。

C. 圖像二值化與單一字符輪廓提取

使用者輸入的影像會先進行預處理，以使其適合讓電腦識別。首先，將RGB彩色影像轉換為灰階影像，並透過單一矩陣表示。將矩陣二值化，其值調整為0和1，在本次程式中，1 代表黑色像素，而 0 代表白色像素。接著使用輪廓提取指令，將影像分割成各個數字的子影像。對各個子影像進行特徵提取，模型將子影像對進行預測，得到方程式中每個符號的標籤。

D. 數學格式辨識與LaTeX代碼輸出

接下來要使程式判斷上標、下標與分數等數學格式。程式使用cv2.findContours 指令提取每個符號的輪廓，cv2.boundingRect指令提取每個輪廓的位置，計算相鄰字元中點的距離。

判斷出的減號“”，也有可能是方程式中的等於“”，或者分數中區隔分子與分母的長槓。

標籤與輪廓資訊的陣列首先會先經過判斷等於符號的函式。兩個長度相當、距離相近的減號為等於。

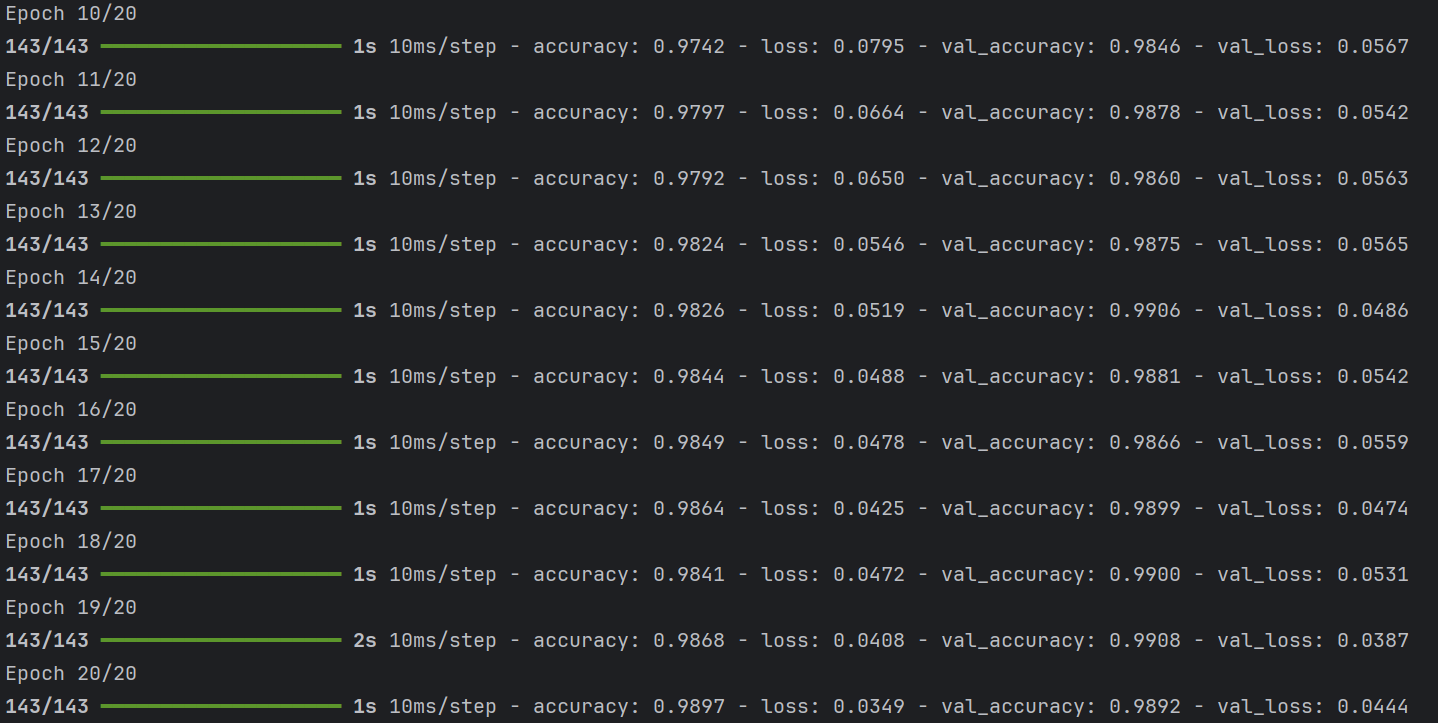
接著標籤與輪廓資訊陣列會再經過判斷分數格式的函式。每遇到一個減號，計算每個符號與此減號的距離，判斷數字是否隸屬於分數的分子或分母，將分數之前的數字串、分數的分子、分數的分母這三項當作一個群體，將其標籤和輪廓資訊分別暫存在陣列中。

三個陣列分別經過判斷上下標的函式，此函式會計算位於其後的輪廓的垂直距離是否符合上下標的標準，最後輸出包含上下標表示的LaTeX代碼。再將其與包含分數表示法的LaTeX代碼組合，將判斷結果輸出到要印出結果的陣列中。尋找下一個包含數字串、分子、分母群體，重複上述流程，最終即可得到一條方程式的完整LaTeX代碼。

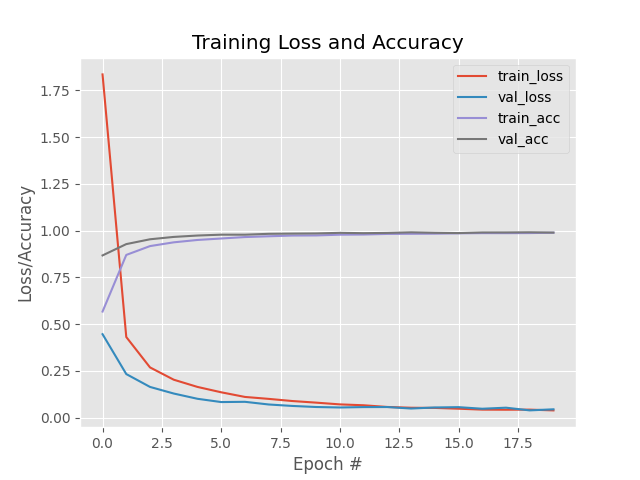
**實驗結果**

A. 模型訓練結果

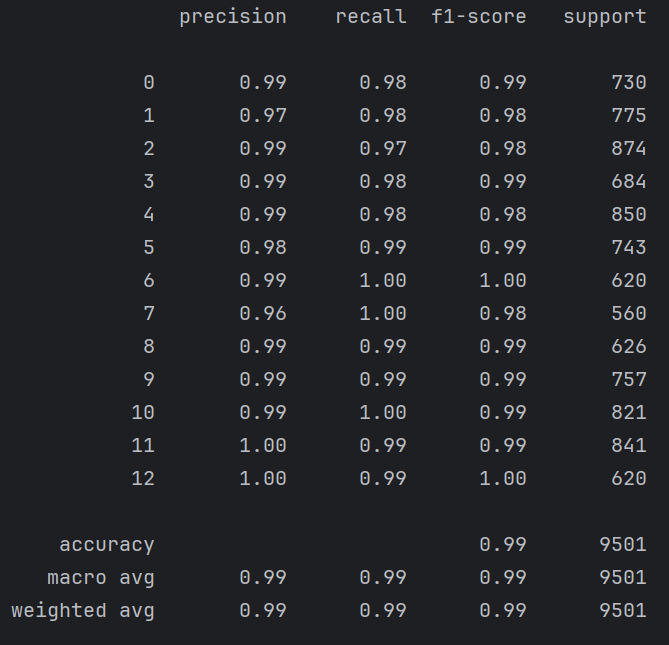
epochs為20。最終的辨識精確度 (accuracy) 為 0.9897，損失函數 (loss function) 為 0.0349，如圖一所示。圖二為辨識精確度與損失函數對 epochs作圖，可以看到曲線趨於平緩，且沒有明顯凸起的地方，意味著沒有出現過度擬合 (overlap) 的情況。



圖一 模型訓練過程



圖二 辨識精確度 (accuracy) 與損失函數 (loss function)對 epochs作圖



圖三 classification report

B. 使用者介面與辨識結果

目前程式能辨識 “0”, “1”, “2”, “3”, “4”, “5”, “6”, “7”, “8”, “9”, “”, “”, “” 等符號，能將方程式影像轉換為LaTeX代碼輸出。

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

圖四 使用者介面

一張含有 圖表, 文字, 螢幕擷取畫面, 行 的圖片

自動產生的描述

圖五 判斷結果

最常出現的錯誤就是將“”判斷為“”，這是一個特別麻煩的情況，因為“”還涉及了等於符號與分數格式的判斷。一但模型沒有成功預測將影像中的等於符號與分數格式的“”符號，將會導致格式辨識的演算法錯誤。

一張含有 螢幕擷取畫面, 文字, 軟體, 多媒體軟體 的圖片

自動產生的描述

圖六 錯誤範例

**參考資料**

[1] <https://github.com/prateeek1/Handwritten-Equation-Solver>

[2] <https://www.kaggle.com/code/pranavdhawan/calculator>

[3] <https://steam.oxxostudio.tw/category/python/tkinter/open-and-show-image.html#google_vignette>

[4] <https://www.kaggle.com/datasets/xainano/handwrittenmathsymbols/code>

<https://github.com/bennokr/latex-equation-from-image/blob/master/position_log/equation-symbol-locations.ipynb>

<https://github.com/bennokr/latex-equation-from-image/blob/master/position_log/equation-symbol-locations.ipynb>

<https://jianjiesun.medium.com/dl-ml%E7%AD%86%E8%A8%98-%E4%B8%83-clustering-algorithm-k-means-meanshift-dbscan-1ef8dbdeed7c>