**目錄**

[圖目錄 3](#_Toc66204961)

[表目錄 4](#_Toc66204962)

[摘要 5](#_Toc66204963)

[一、導論 5](#_Toc66204964)

[二、相關研究探討 8](#_Toc66204965)

[2.1 卷積類神經網路(Convolutional Neural Network, CNN) 8](#_Toc66204966)

[2.2 物件偵測(object detection) 9](#_Toc66204967)

[2.2.1 YOLOv4模型(You only look once) 9](#_Toc66204968)

[2.2.2 U-Net模型 9](#_Toc66204969)

[2.2.3 EAST模型(Efficient and Accuracy Scene Text detection) 10](#_Toc66204970)

[2.3 字串辨識(character recognition) 11](#_Toc66204971)

[2.3.1 RARE (Robust text recognizer with Automatic Rectification)的字串辨識 11](#_Toc66204972)

[2.3.2 CRNN(Convolutional Recurrent Neural Network)模型的字串辨識 12](#_Toc66204973)

[2.3.3 Rosetta模型的字串辨識 13](#_Toc66204974)

[2.3.4 StarNet模型的字串辨識： 14](#_Toc66204975)

[三、研究方法 15](#_Toc66204976)

[3.1字串區域的偵測(detection of string region) 15](#_Toc66204977)

[3.1.1字串區域的初步偵測 16](#_Toc66204978)

[3.1.2字串區域的雜訊處理 16](#_Toc66204979)

[3.1.3字串區域的偵測 17](#_Toc66204980)

[3.2字串區域方向的正規化 18](#_Toc66204981)

[3.3字串行間的切割(row segmentation of the text) 18](#_Toc66204982)

[3.4字串辨識(character recognition) 19](#_Toc66204983)

[3.5 表格定位與資料產生 19](#_Toc66204984)

[四、實驗結果 20](#_Toc66204985)

[4.1實驗環境 20](#_Toc66204986)

[4.2實驗用超參數 20](#_Toc66204987)

[4.3實驗用資料集 21](#_Toc66204988)

[4.3.1 測試資料集 21](#_Toc66204989)

[4.3.2 訓練資料集 22](#_Toc66204990)

[4.4實驗效能評估方式 23](#_Toc66204991)

[4.4.1 字元區域偵測的評估方式 24](#_Toc66204992)

[4.4.2 字串辨識的評估方式 24](#_Toc66204993)

[4.4.3 合併流程的評估方式 24](#_Toc66204994)

[4.5效能分析 25](#_Toc66204995)

[4.5.1字串區域偵測輸入模式分析 25](#_Toc66204996)

[4.5.2字串區域偵測分析 26](#_Toc66204997)

[4.5.3字串辨識分析 28](#_Toc66204998)

[4.6流程效能分析與最佳組合 29](#_Toc66204999)

[五、結論 31](#_Toc66205000)

[六、參考文獻 32](#_Toc66205001)

**圖目錄**

[圖(1)多種挑戰的圖像範例 5](#_Toc66126833)

[圖(2)YoloV4輸出範例 8](#_Toc66126834)

[圖(3)U-Net架構 8](#_Toc66126835)

[圖(4)EAST 偵測範例 9](#_Toc66126836)

[圖(5)EAST架構 9](#_Toc66126837)

[圖(6)RARE STN 網路架構 10](#_Toc66126838)

[圖(7)RARE STN 網路架構 11](#_Toc66126839)

[圖(8)CRNN 網路架構 11](#_Toc66126840)

[圖(9)Rosetta網路架構 12](#_Toc66126841)

[圖(10)StarNet 網路架構 13](#_Toc66126842)

[圖(11)複雜格式表格影像示意圖 14](#_Toc66126843)

[圖(12)本研究系統處理流程圖 14](#_Toc66126844)

[圖(13)字串區塊的初步偵測結果 15](#_Toc66126845)

[圖(14)字串區域的雜訊與處理結果 16](#_Toc66126846)

[圖(15)字串區域的偵測 17](#_Toc66126847)

[圖(16)字串區域方向之正規化處理流程 17](#_Toc66126848)

[圖(17)字串行間的切割處理流程 18](#_Toc66126849)

[圖(18) RNBLC網路架構 18](#_Toc66126850)

[圖(19)字串中心點連線 18](#_Toc66126851)

[圖(20)*x*與*y*軸的標題定位 18](#_Toc66126852)

[圖(21)測試資料及六種形式範例 20](#_Toc66126853)

[圖(22)字串辨識之資料測試集 21](#_Toc66126854)

[圖(23)訓練影像的參數 22](#_Toc66126855)

[圖(24)產生後的訓練影像示意圖 22](#_Toc66126856)

[圖(26)U-net訓練字串的區域(白色區域) 22](#_Toc66126857)

[圖(25)YOLOv4、EAST訓練區塊的區域(紅色框選區域) 22](file:///C:\Users\User\Desktop\大專生\成果報告_v12_修改.docx#_Toc66126858)

[圖(27)各種生成的字串辨識測試集範例 22](#_Toc66126859)

[圖(28)YOLOv4三種模式測試結果 24](#_Toc66126860)

[圖(29)U-net變形模式測試結果 25](#_Toc66126861)

[圖(30)EAST訓練資料(左)與EAST預測結果(右) 25](#_Toc66126862)

[圖(31)YOLOv4於圖編號13預測結果 26](#_Toc66126863)

圖(32)YOLOv4於圖編號13三種模式測試結果 26

[圖(33)U-net偵測結果範例 26](#_Toc66126864)

[圖(34)雜訊與不準確的預測框 29](#_Toc66126865)

**表目錄**

[表(1)軟體與硬體配置 19](#_Toc66127166)

[表(2)字串區域偵測模型超參數設定 19](#_Toc66127167)

[表(3)字串辨識模型超參數設定 19](#_Toc66127168)

[表(4)測試資料集統計資料 20](#_Toc66127169)

[表(5) 偵測策略於不同圖片之效能 27](#_Toc66127170)

[表(6) 各模型於第一至三類字串辨識資料集之準確率 27](#_Toc66127171)

[表(7)加入訓練後結果與RNBLC模型的準確率結果 28](#_Toc66127172)

[表(8) 合併模型準確率比較 28](#_Toc66127173)

[表(9) 辨識模型與偵測模型交插測試結果 29](#_Toc66127174)

[表(10)單一圖片最佳流程 30](#_Toc66127175)

**摘要**

在企業轉型的過渡階段，往往有許多紙本資料需要轉換至電腦系統中，若採用人工輸入會消耗大量的人力與成本，也無法保證其資料的正確性。若能藉由圖像辨識及字元解析等技術，實現一個能更即時且正確性高的圖像辨識方法，更能用於辨識各種不同的圖片與文檔格式，以取代傳統需依靠人眼實際辨識的工作方式，將可以降低人力成本之外，也能全面提升操作效益，以及減少人為辨識錯誤的發生。

本研究以複雜格式表格影像進行字串區域偵測、資訊提取及結構化。本研究提出的處理方法，能夠克服表格式非字串符號的干擾、多型式字串與字元以及複雜背景的影像，其處理流程分別有字串區域偵測、字串行的切割與字串區域方向的正規化、字串辨識以及表格定位與資料產生的處理。字串區域偵測係分別採用EAST、YOLOv4與U-net類神經網路的模型架構，分析優劣後合併模型，以最佳準確率之合併模型做為本研究的偵測結果。字串區域偵測後，再採用傳統的影像處理技術，對多行的字串，進行字串行的切割以及字串區域方向的正規化。字串辨識(string recognition)的部分，本研究將採用RARE、CRNN、Rosetta、StarNet與本研究提出RNBLC的五種模型，對已經完成切割的字串進行端與端(end-to-end)的辨識，並使用投票的方式作為最終的結果。在字串辨識完成後，再將影像做表格的定位與產生最後將所對應之字串，以文字方式記錄起來。

為了快速且精確地進行複雜格式表格影像進行字串區域偵測以及字串的辨識，字串區域偵測所採用EAST、YOLOv4與U-net的模型架構，以及字串辨識所使用RARE、CRNN、Rosetta、StarNet 與本研究提出RNBLC的模型架構，本研究提供一套新穎且多元化的方式，產生大量的訓練樣本影像。

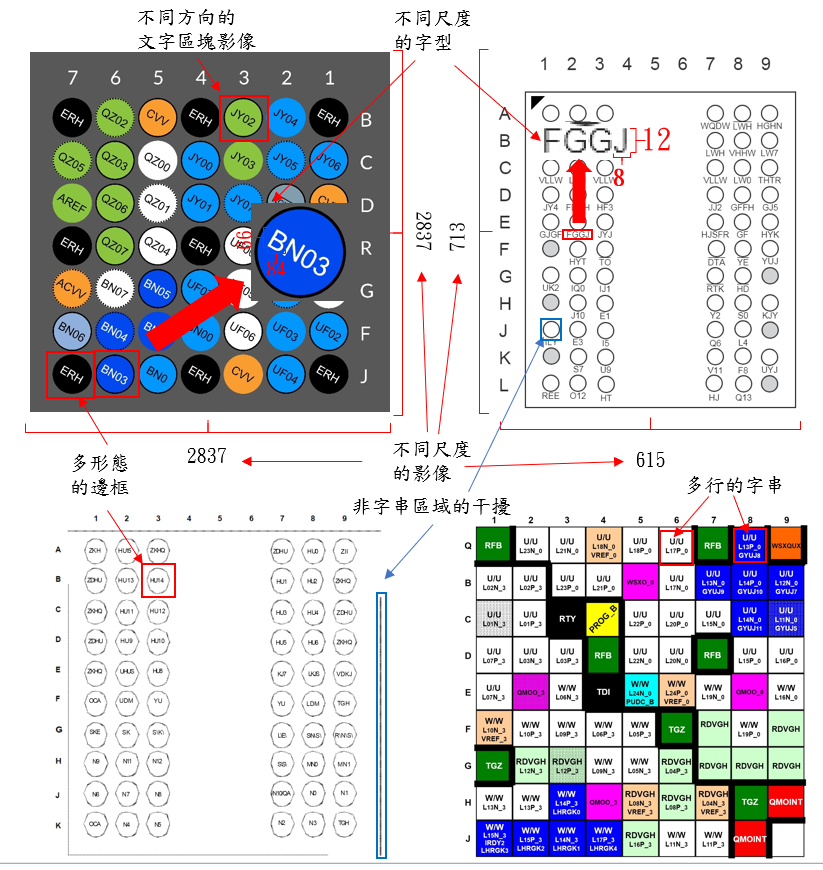
**關鍵字**：表格影像、字串區域偵測、字串辨識、EAST、U-net、YOLOv4、RARE、CRNN、Rosetta、StarNet

**一、導論**

在企業轉型的過渡階段，往往有許多紙本資料需要轉換至電腦系統中，例如：硬體的腳位圖、履歷表、打卡紀錄等文字檔案，若採用人工輸入會消耗大量的人力與成本，也無法保證其資料的正確性。隨著科技的進步，以及攝影設備的提昇，影像處理技術被廣泛應用在圖像辨識上，以解決含有文字影像的影像，並加以紀錄且分析。圖像辨識係利用辨識的結果加以編輯，如此可以省去人工輸入的時間，並可加速輸入編輯的速度。因此，若能藉由圖像辨識及字元解析等技術，實現一個能更即時且正確性高的圖像辨識方法，更能用於辨識各種不同的圖片與文檔格式，以取代傳統需依靠人眼實際辨識的工作方式，將可以降低人力成本之外，也能全面提升操作效益，以及減少人為辨識錯誤發生。

近年來，許多以視覺為基礎之深度學習技術應用在圖像的辨識，其技術也有卓越的提升，同時配合採用圖形處理器(Graphics Processing Unit, GPU)，使得處理模型可以朝更多層或更寬的方式去做學習與訓練，讓在訓練模型時有更多方法可以做選擇。因此，本研究將使用深度學習的方式，並且讓系統自動找到規律，也可以克服複雜的狀況，以便省下大量設計演算法的時間與成本。

微軟在2019年3月將android版的excel加入一項新功能，此功能可以將手機拍攝的圖轉成excel表。如此讓使用者在讀取圖片內圖表時，可以省下許多輸入資料的時間，使得攜帶裝置的資料處理能力，可以媲美個人電腦，同時也提升使用者利用拍照來收集資料的意願，但是對於複雜格式的表格上表現不佳。因此，為了處理複雜格式的表格影像，本研究將提出一套智慧型的字串辨識方式。其中複雜格式的表格影像包含六種不同形式的格式，導致偵測與識別的困難，第一種為多形態的邊框、第二種為不同方向的文字區塊影像、第三種是各種不同顏色的字與背景、第四種為不同尺度的影像與字型、第五種為非字串區域的干擾、第六種為多行的字串，如圖(1)所示。第一種格式字串受到邊框之干擾，需在字串區域偵測時排除，還需要在字串辨識時忽略，其中邊框包含了矩形、圓形與不規則形狀；第二種格式文字的方向不同，需校正後才進行端到端的字串辨識。第三種格式字體的顏色與字的背景顏色皆有不同，其中字體顏色包含了黑、白、淺灰、黃、橘、綠、紅七種顏色，字的背景顏色則包含了白、黑、藍、橘、綠等十七種顏色；第四種格式影像與字體的大小差距甚大，影像大小從長713、寬615至長5026、寬3956不等，字體大小從長8、寬13至長84、寬66不等，小型字體如隨意縮小，會導致字體消失，大型字體如不縮小會導致偵測與辨識上的困難，而不同尺度的影像則使偵測難以以統一縮放大小的辨識；第五種格式具有不同的非字串區域的干擾，可細分為表格外框、圓圈與噪點，需再字串區域偵測時排除，且在字串辨識時須使其辨識為空字串。第六種格式為多行的字串，需分別將每行字切開，才可進行端到端的字串辨識。若採用傳統的字串辨識方法，進行複合此六種格式之複雜表格影像之字串辨識，其辨識率速度與精確度並不理想，甚至無法辨識。



圖(1)多種挑戰的圖像範例

因此，本研究以複雜格式表格影像為研究對象，並找出表格座標，相對應於相對位置之字串做為辨識的字元，最後再將表格所對應之字串字元，以文字方式記錄起來。本研究提出之字串辨識方法，必須能夠克服表格式複合上述六種不同形式的影像，其辨識的處理流程依序有字串區塊的偵測(detection of string block)、字串切割(segmentation of string)與字元方向的正規化(normalization of character direction)、字串辨識(optical character recognition)以及表格定位(table positioning)。字串區塊的偵測將分別採用EAST[1](efficient accurate scene text detector)、YOLOv4[2](you only live once)與U-Net[3]的模型架構，字串辨識則分別採用RARE[4](Robust text recognizer with Automatic REctification)、CRNN [5](convolutional recurrent neural network)、Rosetta[6]、StarNet[7]與本研究提出RNBLC的模型架構。除此之外，為了能克服本文所要解決的問題，本研究也提出了產生大量的訓練樣本影像的方式。

從場景圖像中進行文字的偵測，在現實生活中已經有許多應用，包括基於內容的影像偵測、智慧型導航以及場景認知等。當然更受到許多研究領域的關注，例如：舉辦一系列有關文檔分析的國際會議、研討會與競賽[8-9]。近五年內也有一些對文檔中字串偵測和字串辨識的研究，有一些方法在特殊應用中也有不錯的結果[10-12]。 但是對於背景複雜、不同尺寸大小以及視角和方向的問題，字串的偵測仍然是一個具有挑戰性的問題[13,14]。

近年來對於場景影像的字串偵測的研究[15-17]，其中提供有關場景中字串偵測與字串辨識的研究[15]，以及檢測多方向文檔中的字串相關的研究，已經有大量研究被發表[11,14,16,17]。

字元產生的方式可以分為基於區域(region based)的方法[18]與基於連接組件(connected component based)的方法[19]。基於區域方法係採多尺度的區域來偵測所有可能的字串，再使用字元分類器進行辨識，由於此方法主要處理水平的字串，因此對於旋轉、縮放比例的字串，比較無法處理。基於連接組件方法係假設同一字元的像素具有相似的顏色與筆劃寬度，所以將像素分組為一個區域，然後再採用連通分量分析技術辨識字元。

結合基於區域與基於連接組件兩種方法的研究[20]，其中M. Basavanna等[14]提出了一種新的植基變動長度(run length based)的方法來偵測字串。此方法首先做Sobel邊緣偵測後，再計算其邊緣間變化數量，並以統計其變動長度，最後再進行分類。

近年來，卷積神經網路(Convolutional neural network, CNN)應用在圖像的辨識，也有卓越的成效[5,16,18]。其中Huang等人[5]設計的模型係基於遞歸神經網絡(recurrent neural network, RNN)和卷積類神經網路(convolutional neural network, CNN)的組合。Baoguang Shi等人[5]將卷積神經網絡(convolutional neural networks, CNN)與遞迴神經網路(recurrent neural networks, RNN)做結合，並對已經完成切割的字串進行端與端(end-to-end)的文字檢測和辨識。此卷積遞迴神經網絡的架構具有四種特點：（1）此模型不須要對字元單獨訓練與辨識，而是對字串端對端的訓練與辨識。（2）此模型能處理不同長度的字串，字元不須要切割或是字元成度的正規化。（3）可以有效取得任意字串的字串辨識。（4）此模型能產生一個有效且大小不一的模型，以符合現實的應用場景。

**二、相關研究探討**

在電腦視覺領域，深度學習技術廣泛應用於醫學影像、遙測影像等方面。本章節將依序介紹深度學習技術中的卷積類神經網路架構，以及本文研究方法運用之物質分類模型。

**2.1 卷積類神經網路(Convolutional Neural Network, CNN)**

卷積類神經網路(convolutional neural network, CNN)[21]是深度學習的一種架構，近年來廣泛應用於電腦視覺和語音辨識。以往要辨識或分類物件時，需要以人工方式提取目標物件的特徵，再經由分類器得到結果。然而深度學習則是將萃取特徵與分類器整合在一起，並由模型自行嘗試各種可能，訓練出合適的特徵。

卷積類神經網路主要包含卷積層(convolution layer)、池化層(pooling layer)和全連接層(fully connected layer)三個部分，其中卷積層用以提取特徵，池化層負責減少資料量並擴大感受野(receptive field)，全連接層則用來分類。一般架構存在多個卷積層及池化層，淺層網路提取的特徵通常為局部的細節。

**卷積層(convolution layer)：**卷積層(convolution layer)在卷積類神經網路中負責特徵萃取，透過濾波器(filter)對輸入影像進行卷積運算，並輸出特徵圖(feature map)。為了提取多個特徵，卷積類神經網路會使用多種濾波器。

**填補(padding)：**卷積運算會讓輸出的特徵圖尺寸變小，為了保持特徵圖與輸入的影像尺寸相同，在卷積運算之前會先做填補(padding)的動作。

**線性整流函數(Rectified Linear Unit, ReLU)：**線性整流函數(rectified linear unit, ReLU)是活化函數(activation function)的一種，一般在卷積運算之後使用，目的是做非線性的處理，避免輸入與輸出維持線性關係。

**Batch Normalization：**Batch Normalization[22]是2015年提出的演算法，主要希望克服深層神經網路訓練困難的問題。訓練資料會影響模型訓練的結果，當訓練資料之間的差距較大時，模型會變得不容易訓練，對於深層的網路更是如此。每層的參數更新都會影響下一層的資料分布，而每層的微小變化，可能會因為層數加深而變大，因此需要花費更多的時間調整，意即學習率(learning rate)必須設小。為了解決此問題，Batch Normalization被提出來，其概念為對每層的輸入資料進行正規化處理，降低資料分布的偏差。

**池化層(pooling layer)：**卷積類神經網路中的池化層(pooling layer)主要用來減少參數量，並保留重要特徵。另一方面，由於尺寸的縮減，使得感受野擴大，在池化層之後的卷積層能提取到的特徵也會較全面。池化層分為最大池化(max pooling)及平均池化(average pooling)。

**全連接層(fully connected layer)：**全連接層為一般的神經網路(neural network)，在卷積類神經網路中，一般包含多層，而全連接層係透過卷積層與池化層所提取出來的特徵，來判定此影像所屬的類別，並由最後一層輸出預測結果。特徵圖輸入至全連接層之前，係先將特徵圖的特徵值轉成一維，也就是攤平(flatten)的動作。

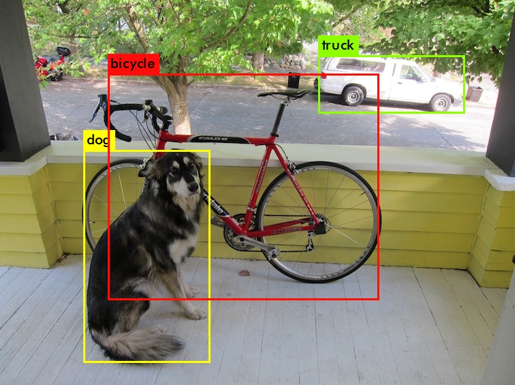
**損失函數(loss function)：**卷積類神經網路採用監督度式學習的方式，訓練過程需要提供正確標籤，協助模型評估訓練的程度來修正卷積層內濾波器的權重值，以及全連接層的權重，簡而言之，就是訓練出最合適的特徵萃取器及分類器，而判斷訓練程度的優劣，則是透過損失函數(loss function)作為評估標準。當輸出模型的預測結果之後，會經由損失函數計算出預測結果與實際標籤的誤差，也就是損失值(loss)，之後係透過反向傳播算法(backpropagation)計算與各權重參數相關的損失函數梯度，並由梯度下降法更新模型的權重參數。

**2.2 物件偵測(object detection)**

卷積類神經網路(convolutional neural network,CNN)是深度學習(deep learning)[23,24]的一種，它是目前在電腦視覺、語音分析、自然語言最為熱門的方法之一。卷積類神經網路在處理物件辨識有非常好的效果，由於它的神經元可以響應某些覆蓋範圍內的神經元，而且比其他深度神經網路使用更少的維度同時共用參數使得參數減少。

**2.2.1 YOLOv4模型(You only look once)**

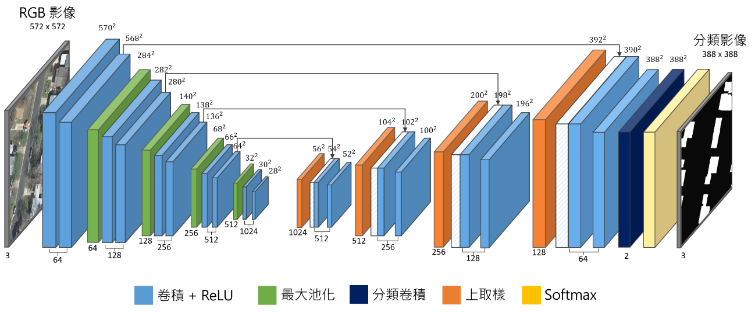
本研究字串的偵測係採用YOLO第三(you only look once,YOLOv3)版所提出的物件偵測架構，YOLOv4[2]使用CSPDarknet53做為特徵提取器，以及引入特徵金字塔網路(feature pyramid network,FPN)的概念，如此可以不同尺度的特徵圖(feature map)問題。YOLOv4將輸入影像分成S×S的區塊(grid cell)，分成三種尺寸大小分別是13、26與52。每個區塊同時預測三個結果，其結果為邊框(anchor box)，而每個邊框包含(*x*, *y*, *w*, *h*, confidence)以及B個類別的機率，其中(*x*, *y*)表示該邊框平面座標的中心點，(*w*, *h*)表示邊框的長與寬，confidence 表示邊框是否有物件的機率，若有物件，則再決定此邊框在各類別的機率。



圖(2) YoloV4輸出範例

**2.2.2 U-Net模型**

U-Net[3]為應用於生物醫學影像的物質分類模型，也被廣泛應用於醫學影像。U-Net由卷積層、池化層和上取樣層組成，其架構如圖(3)所示。架構分為編碼層和解碼層，編碼層包含10個卷積層和4個池化層，解碼層則有4個上取樣層、8個卷積層和1個分類卷積層。



圖(3) U-Net架構

**編碼層：**編碼層中，卷積層皆使用尺寸3🞨3的濾波器，且不使用填補方式，因此特徵圖尺寸會隨著卷積運算而越來越小。池化層採用最大池化，以尺寸2🞨2的濾波器進行運算。為了結合局部細節和全域的特徵，將第2、4、6、8層卷積輸出的特徵圖儲存，用以在解碼層中將特徵圖結合。

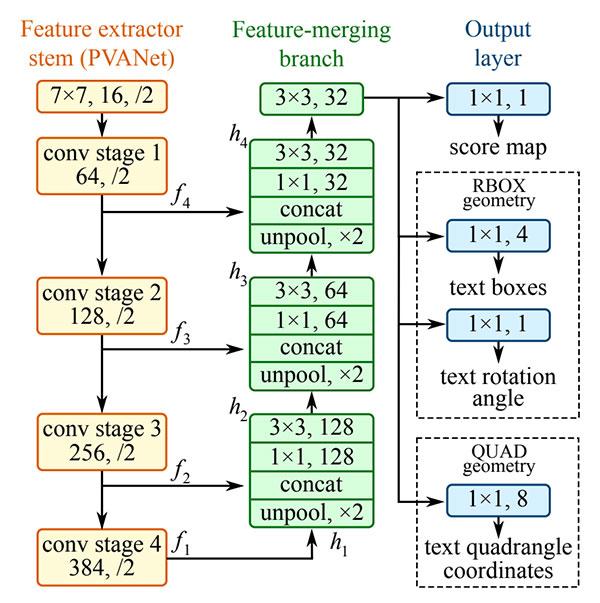
**解碼層：**解碼層方面，係將一個上取樣層、合併層，以及兩個卷積層視為一組，共有四組。在分類卷積層後的結果，再使用Softmax函數將預測分數轉換為機率，並獲得字串偵測的結果。

**2.2.3 EAST模型(Efficient and Accuracy Scene Text detection)**

EAST[1]為應用於自然場景文本檢測的偵測模型，可用用於不同材質、光線與角度的文本檢測。其模型參考U-Net來建置出的全卷積網路，共有4個編碼層與3個解碼層，輸出為特徵圖與幾何框，特徵圖為逐像素的分割預測結果，幾何框則是逐像素皆預測一個邊框，格式可以在訓練前選擇要使用(x, y, w, h, angle)的回歸，其中(x, y)表示該邊框平面座標的中心點，(w, h)表示邊框的長與寬，angle表邊框的選轉角度；或是採用(x1, y1, x2, y2, x3, y3, x4, y4)的回歸，意旨4個點的(x,y)座標。最後會使用非極大值抑制做框的合併，即可得到文本檢測框。EAST[1]架構由編碼層與解碼層組成，架構如圖(5)所示。



圖(4) EAST 偵測範例



圖(5) EAST架構

**編碼層**：編碼層中，卷積層皆使用尺寸3🞨3的濾波器，因此特徵圖尺寸會隨著卷積運算而越來越小。池化層採用最大池化，以尺寸2🞨2的濾波器進行運算。為了結合局部細節和全域的特徵，將第2、3、4、5層最大池化輸出的特徵圖儲存，用以在解碼層中將特徵圖結合。

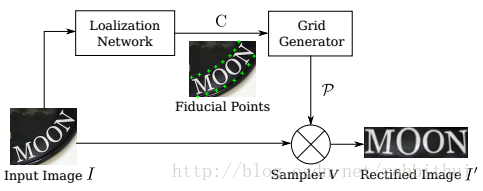
**解碼層**：解碼層係將一個上取樣層、合併層，以及兩個卷積層視為一組，共有四組，其中上取樣層採用的的是雙線性差值。最後將輸出的特偵圖，分別進行兩個輸出通道為一、一個輸出通道為四的卷積層並通過sigmoid層將數值收縮至零到一，取得特徵圖與幾何框之結果。

**2.3 字串辨識(character recognition)**

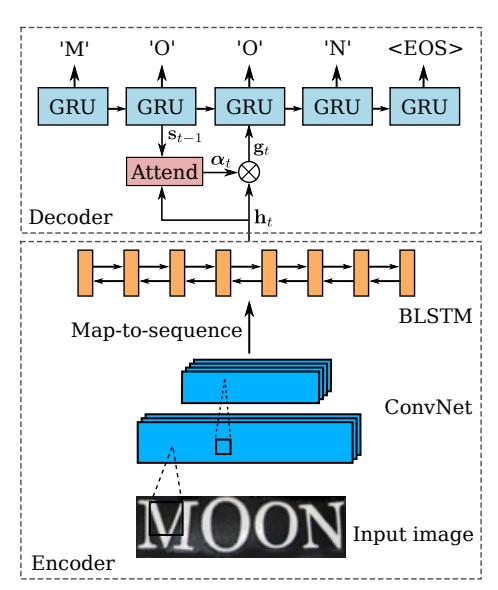
字串辨識領域對於市場上有眾多的需求，比如車牌辨識、文本數位化、招牌識別等等。本研究將採用4種現有的深度學習模型，分別為RARE[4]、CRNN[5]、Rosetta[6]、StarNet[7]，與本研究自行設計並提出的RNBLC模型進行實驗。

**2.3.1 RARE (Robust text recognizer with Automatic Rectification)的字串辨識**

RARE模型用以不規則文本端到端的字串辨識，由STN(Spatial Transformer Network)與SRN(Sequence Recognition Network)兩個部分組成。STN是使用卷機類神經網路生成多個參考點，神經網路包含了4個卷機層，與4格池化層，其中一、二、三層卷機層後使用最大池化層，第四層後則是接1個自適應平均池化層，最後則是2個全聯接層，輸出20個點的(x, y)座標共40個數值，在經由TPS轉換成易於辨識的圖像。SRN則是將STN之結果經編碼層與解碼層後，得到文本的預測。編碼層包含了７個卷機層、4個的最大池化層、2層批次正規化與雙向的LSTM。解碼層則是基於結合注意力機制與GRU的循環網路，會持續辨識直到輸出EOS符號作為字串結尾。



圖(6) RARE STN 網路架構



圖(7) RARE STN 網路架構

**2.3.2 CRNN(Convolutional Recurrent Neural Network)模型的字串辨識**

傳統的字串辨識方法，必須先將字串內的字元做切割，然後再進行字元的辨識。為了減去字元切割與字元單獨訓練與辨識的流程，本研究將採用卷積遞迴神經網路(convolutional recurrent neural network, CRNN)的模型[5]進行字元的辨識，此模型係將卷積神經網絡(convolutional neural networks, CNN)與遞迴神經網路(recurrent neural networks, RNN)做結合，並對已經完成切割的字串進行端與端(end-to-end)的文字檢測和辨識。

卷積遞迴神經網路的結構三個部分所組成：(1)卷積層(convolutional layers)係從輸入的字串影像中萃取特徵的序列，(2)遞迴層(recurrent layers)係預測每個序列特徵的標籤分佈，(3)轉錄層(transcription layer)係將每個序列的預測轉換為最終標籤序列。



圖(8) CRNN 網路架構

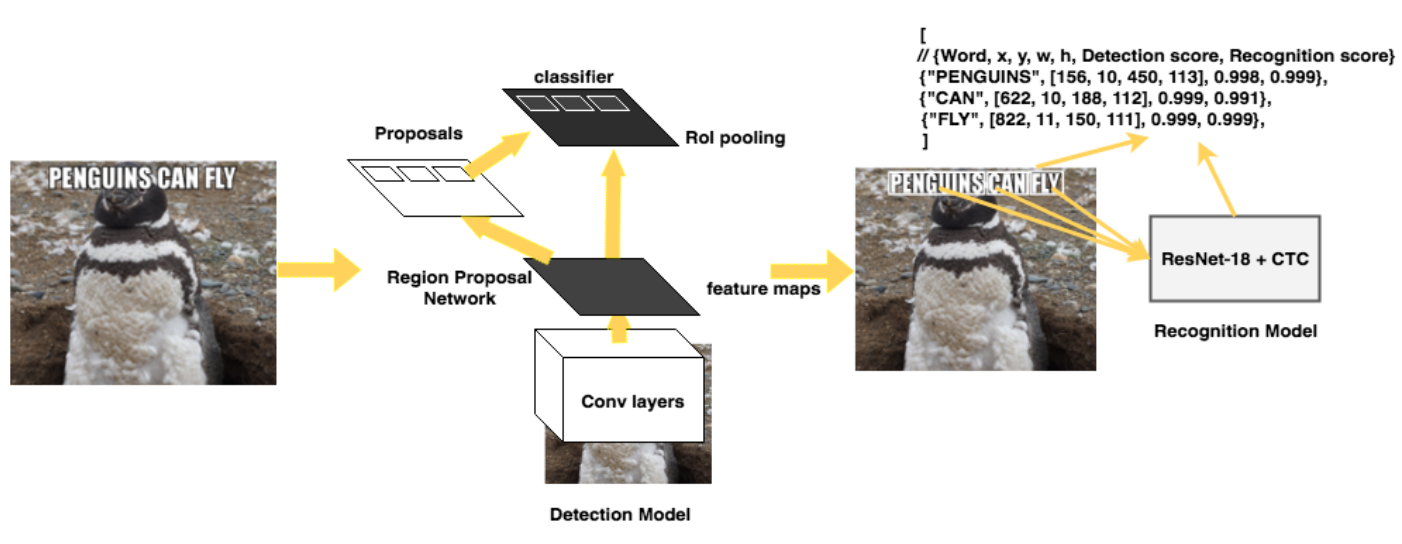
**卷積層**：在進入網路之前，所有字串影像必須先縮放到相同的高度。然後再從卷積層元件產生的特徵圖，並按列從左到右生成提取特徵的向量序列，每列的寬度固定為單個畫素，這些特徵向量序列將做為遞迴層的輸入。

**遞迴層**：長短期記憶網路(long short-term memory network, LSTM)[25]於1997年由Sepp Hochreiter和Jürgen Schmidhuber提出，LSTM由數個LSTM細胞(cell)所組成，一個LSTM細胞，其中包含3個輸入、3個元件以及2個輸出，3個輸入分別為t時刻的輸入()，時刻的輸出()，時刻的長期記憶()，3個元件分別為輸入閘(input gate,)、輸出閘(output gate,)及遺忘閘(forget gate,)，3個元件都使用Sigmoid Function做為激活函數使輸出值介於0～1來模擬閥門的開關。輸入閘根據t時刻的輸入()跟時刻的輸出()決定t時刻所產生的記憶()是否要加入長期記憶()中，輸出閘根據t時刻的輸入()跟時刻的輸出()決定t時刻所產生的長期記憶()是否要輸出，遺忘閘根據t時刻的輸入()跟時刻的輸出()決定時間點的長期記憶()是否要加入t時刻的長期記憶()中，2個輸出為t時刻的輸出()以及t時刻的長期記憶()，LSTM比一般的RNN多了一個輸出()做為長期記憶，來解決RNN遇到時序過多所產生的梯度問題接著說明LSTM細胞的處理流程首先透過t時刻的輸入()和時刻的輸出()及各自的權重(,)通過Sigmoid Function算出3個閥門的值，接著計算t時刻所產生的記憶()使用t時刻的輸入()和時刻的輸出()及各自的權重(W,U)通過tanh函數使輸入值介於-1～1，並將時刻的長期記憶()乘上遺忘閘的值加上t時刻的輸入()乘上輸入閘的值做為t時刻的長期記憶()，最後將t時刻的長期記憶()通過tanh函數並乘上輸出閘的值做為t時刻的輸出()。

**轉錄層**：轉錄是將遞迴層所做的每個序列預測，從中找到最高概率的標籤序列。此轉錄包括無詞典和有詞典的兩種轉錄模式，詞典是一組字元組成的字串標籤序列。在無詞典轉錄模式中，預測時沒有詞典可以對應。在有詞典的轉錄模式中，選擇最高概率的字元組成的字串標籤序列進行預測。雖然CRNN由CNN和RNN的模型架構組成，但可以採用一個損失函式來進行學習訓練。

**2.3.3 Rosetta模型的字串辨識**

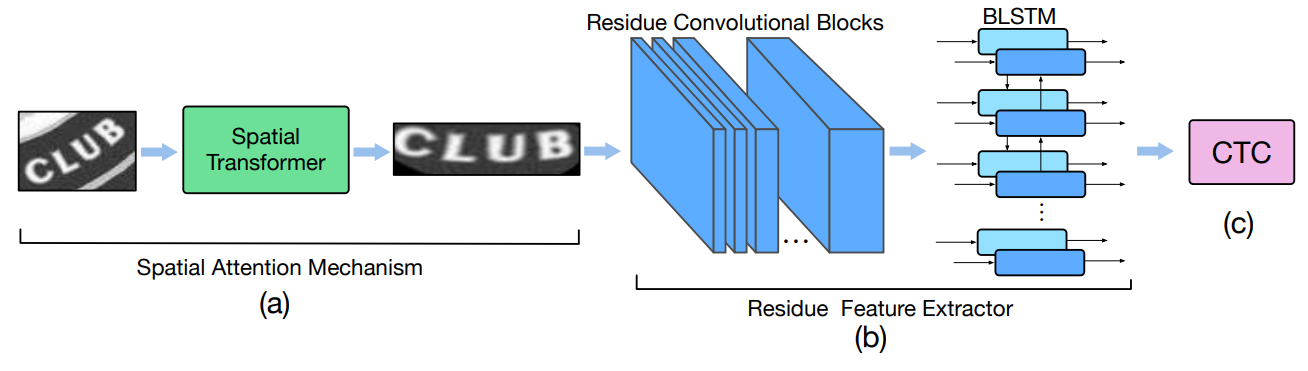
Rosetta[6]為facebook提出用以大規模系統的字串偵測與辨識的模型，辨識目標為非旋轉的文本。字串偵測的部分採用Faster-RCNN，其核心模型替代為SufferNet已達成加快偵測速度。字串辨識則是使用RestNet-18搭配CTC計算loss。最後產生多組辨識結果，每個辨識結果包含(word, x, y, w, h, Detection score, Recognition score)，其中word表示字串辨識的結果，(x, y)表示該邊框平面座標的中心點，(w, h)表示邊框的長與寬，(Detection score, Recognition score) 表示偵測與辨識的信心分數。本研究只採Rosetta字串辨識的部分與其他字串辨識模型進行比較。



圖(9) Rosetta網路架構

**2.3.4 StarNet模型的字串辨識：**

StarNet[7]是用以真實影像的字串辨識，影像包含扭曲、歪斜、藝術字等。由空間轉換、特偵抽取與CTC三個部分組成。空間轉換如同RARE的STN，使用卷機類神經網路找到特偵點，在交由TPS做轉換成易辨識的圖像。特偵抽取則是使用RestNet-18與雙向的LSTM，獲得文本內容。最後則是將文本內容經由CTC Loss計算後，反向優化空間轉換與特偵抽取的神經網路權重。



圖(10) StarNet 網路架構

**三、研究方法**

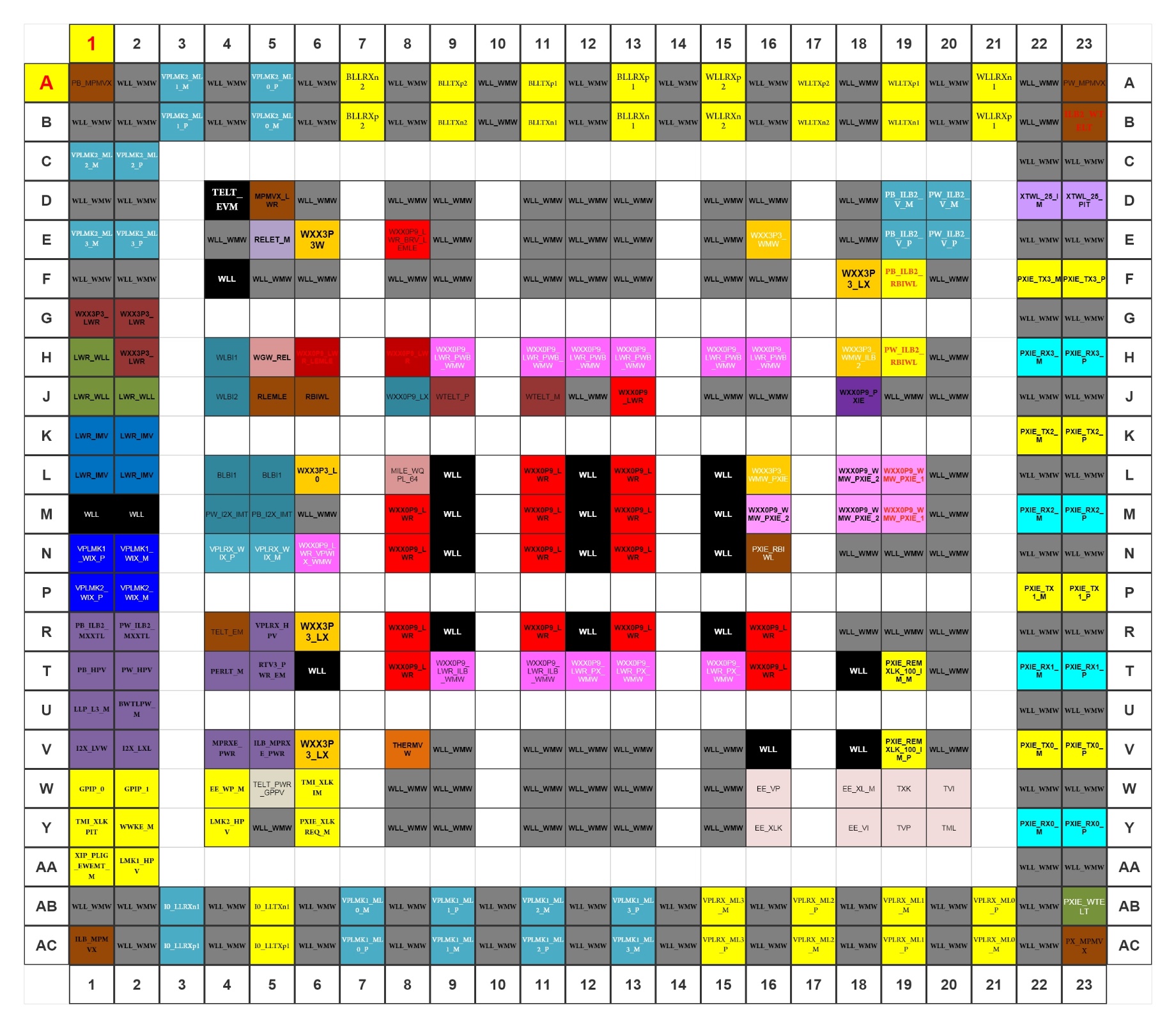
本研究針對複雜格式表格影像進行字元的辨識，並且以表格式方式記錄為目的。因此，本文提出表格式複雜背景影像之字串辨識方法之外，也提出字串區塊的偵測與字串辨識的類神經網路模型之樣本影像產生的方式，以及架構模型的參數設定。

本研究以表格式影像中的以*x*與*y*軸標題為主，然後再以兩軸標題相對應之字串做為字元的辨識，最後再將兩軸標題以及所對應之字串，以文字方式記錄起來，其說明如圖(11)所示。本研究提出之字串辨識方法，能夠克服表格式複雜背景影像，其辨識的處理流程分別有字串區塊的偵測、字串區域方向的正規化、字串的切割、字串的辨識及資料結構化，其處理流程如圖(12)所示。

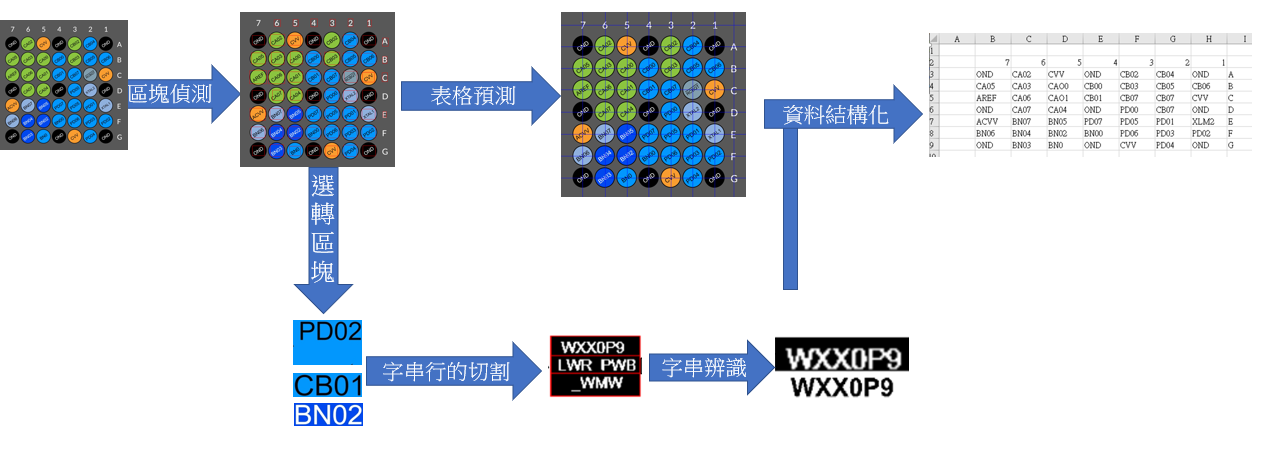
*x*軸區域

*y*軸區域

位於座標**(6,E)**的字串「**WXX3P3W**」



圖(11)複雜格式表格影像示意圖



圖(12) 本研究系統處理流程圖

**3.1字串區域的偵測(detection of string region)**

卷積類神經網路(convolutional neural network, CNN)是深度學習(deep learning)[23,24]的一種，它是目前在電腦視覺、語音分析與自然語言最為熱門的方法之一。卷積類神經網路在處理物件辨識有非常好的效果，由於它的神經元可以響應某些覆蓋範圍內的神經元，而且比其他深度神經網路使用更少的維度同時共用參數使得參數減少。

**字串區域偵測模型前處理**：由於本研究的表格影像格式複雜且變化多元，因此採用一種偵測方法，無法將字串區域精確地偵測。因此，本計畫採用YOLOv4[2]與U-net[3]進行字串區域的初步偵測，接著將對其偵測結果做雜訊處理，最後再對這些偵測的區域合併，以作為字串區域的偵測結果。

**3.1.1字串區域的初步偵測**

本計畫提出字串區塊的初步偵測，分別是YOLOv4[2] 與U-net[3]模型的字串區域偵測，其說明如下：

**YOLOv4的字串區域偵測：**由於本研究之影像尺寸從615×713像素至3956×5026像素，以及影像中的字型從8×9像素至 83×109像素，若固定模型輸入的影像大小，則字串無法完整的被偵測出來。因此，本模型首先將輸入的影像分成三種模式，第一種係將所有影像調整大小至608×608像素；第二種係將所有影像取大小608×608大小的子影像，再以重疊方式往右或往右304像素取大小608×608大小的子影像，以此類推；第三種係將所有影像調整大小至*m*×*m*像素，再取大小*n*×*n*大小的子影像，再以重疊方式往右或往右*n*/2像素取大小*n*×*n*大小的子影像，以此類推。本研究的*n*=608，m將實驗後討論適合的數值，其偵測的結果如圖(13.a)所示。

**U-net模型的字串區域偵測**： U-net模型主要是作為物件的切割，本研究將採用此模型偵測字串的區域。由於本計劃處的影像大小不能太大，因此將所有影像調整大小至*m*×*m*像素，再取大小*n*×*n*大小的子影像，再以重疊方式往右或往右*n*/2像素取大小*n*×*n*大小的子影像，以此類推。其中*m*與*n*的將於實驗時，再討論適合的數值，本研究假設*m*=608與*n*=608，其偵測的候選字串區域結果如圖(13.b)所示。

|  |  |
| --- | --- |
| G:\我的雲端硬碟\NSC\109\大專生\成果報告\sample\05yoloV4偵測結果.png  (13.a) YOLOv4字串處理結果 | G:\我的雲端硬碟\NSC\109\大專生\成果報告\sample\07Unet偵測結果.jpg  (13.b) U-Net字串偵測結果 |

圖(13)字串區塊的初步偵測結果

**3.1.2字串區域的雜訊處理**

經過初步偵測後可能會造成一些雜訊，這些雜訊有小面積雜訊、與其他字串區域面積差異大的雜訊、字串區域內的雜訊、信心分數過低的雜訊以及重複字串區域的雜訊，其雜訊如圖(14)所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 雜訊型態 | (14.a)小面積雜訊或與其他字串區域面積差異大的雜訊 | (14.b)字串區域內的雜訊 | (14.c)信心分數過低的雜訊 | (14.d)重複字串區域的雜訊 |
|  |  | 0.08  0.82  0.12 |  |
| 處理後 |  |  | G:\我的雲端硬碟\NSC\109\大專生\成果報告\151671117_739052863651382_10553323177463431_n.png |  |

圖(14)字串區域的雜訊與處理結果

為了解決去除這些雜訊，本研究提出五種雜訊處理的方法，並依序刪除雜訊，其處理的方式說明如下：

**小面積的雜訊處理**：經過統計字串區域的長寬，若候選字串區域邊框長度小於8像素，則視為雜訊即刪除，其處理結果如圖(14.a)所示。

**其他字串區域面積差異大的雜訊處理**：計算所有候選字串區域長寬的平均數與標準差，再計算候選字串區域長寬與平均值的差值，若差值大於或小於六倍的標準差之候選字串區域，將視為雜訊並刪除。重複執行此處理，直到所有候選字串區域都符合以上的規則為止，其處理結果如圖(14.a)所示。

**字串區域內的雜訊處理**：計算兩個候選字串區域的交集面積，若交集面積為其中一個區域的面積，則將小面積的區域刪除，其處理結果如圖(14.b)所示。K0

**信心分數過低的雜訊處理**：經過實驗統計後，本研究將候選字串區域的信心分數設為0.4以上，若候選字串區域的信心分數小於0.4則刪除，其處理結果如圖(???c)所示。

**重複字串區域的雜訊處理**：將所有候選字串區域間的IOU，若交集區大於等於*n*的區域，從中選擇最高信心分數的候選字串區域，其餘的候選字串區域視為重複雜訊即刪除。若交集區域大於0.1且小於*n*的區域，再取其聯集區域以取代此交集框。為了符合不同模型的特性，本研究的YOLOv4模型*n*採用0.32，U-net模型則採用0.5，其處理結果如圖(14.d)所示。

**3.1.3字串區域的偵測**

字串區域偵測係將YOLOv4與U-net偵測後的候選字串區域進行結合，其結果如圖(15.c)所示。由於結合後，候選字串區域會出現字串區域內的雜訊、其他字串區域面積差異大的雜訊以及重複的雜訊，因此再依序進行雜訊的處理，以作為字串區域偵測的結果，其結果如圖(15.d)所示。

|  |  |
| --- | --- |
| G:\我的雲端硬碟\NSC\109\大專生\成果報告\Y.png  (15.a) 採用YOLOv4偵測的字串區域結果 | G:\我的雲端硬碟\NSC\109\大專生\成果報告\U.png  (15.b) 採用Unet偵測的字串區域結果 |
| (15.c) 候選字串區域進行結合結果 | G:\我的雲端硬碟\NSC\109\大專生\成果報告\YU.png  (15.d) 字串區域的偵測結果 |

圖(15)字串區域的偵測

**3.2字串區域方向的正規化**

由於字串區域的字串，其方向有可能不是水平方向，因此為了讓字串的方向統一成水平方向，本研究將偵測此字串的方向，然後再正規化成水平方向。其正規化方式首先採用傳統影像處理，係將字串區域轉成灰階影像，然後再採用Otsu’s進行影像二值化，接著再進行形態學之閉運算與輪廓的偵測，如此即可取得此字串區域的最小外接矩形，以及此矩形與水平線的夾角，最後再依夾角做正規化處理。其正規化處理流程如圖(16)所示。



圖(16) 字串區域方向之正規化處理流程

**3.3字串行間的切割(row segmentation of the text)**

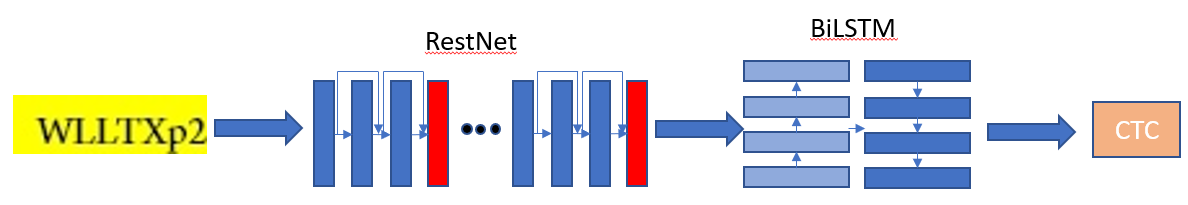
偵測後的字串區域有可能有兩行以上的字串，如圖(17)左圖所示，因此方便後續的字串辨識，本研究必須先將字串區域各行的字串進行切割。其方法首先將字串區域色階轉成灰階，然後再採用Otsu’s進行二值化處理，接著再進行水平的閉運算，讓同一行的字元連起來，最後再將不同行的字串切割出來。其處理流程與結果如圖(17)所示。



圖(17)字串行間的切割處理流程

**3.4字串辨識(character recognition)**

傳統的字串辨識方法，必須先將字串內的字元做切割，然後再進行字元的辨識。為了減去字元切割與字元單獨訓練與辨識的流程，本研究將採用深度學習模型進行字串的字串辨識。目前有一些以字串進行字串辨識的模型，本計畫為了能精確地做字元的辨識，本研究將結合RestNet[26]、BiLSTM以及CTC模型，並提出一套以字串進行字串辨識的模型，以上稱為RNBLC模型，架構如圖(18)。除此之外，本研究將產生大量訓練的字串影像，然後再分別採用RARE[4]、CRNN[5]、Rosetta[6]、StarNet[7]以及RNBLC模型進行訓練學習後，再做字元的辨識。這些模型係將卷積神經網絡(convolutional neural networks, CNN)、注意力機制(attention)與遞迴神經網路(recurrent neural networks, RNN)做結合，並對已經完成切割的字串進行端與端(end-to-end)的文字檢測和辨識。最後，對於不同的字串辨識模型，將以投票的方式決定字串辨識的結果。



圖(18) RNBLC網路架構

**3.5 表格定位與資料產生**

表格定位與資料產生，首先將做表格的定位，然後尋找*x*與*y*軸的標題位置，最後再將兩軸標題以及所對應之字串，以文字方式記錄起來，其說明如圖(19)所示。因此，本研究在字串辨識完成後，將對字串的中心點連成多條水平與垂直線，以紅線表示如圖(19)所示。然後再找出字串低於2個字元的所有水平直線，並挑選字元總數最少的直線並視為x軸，接著找出其字串低於2個字元的所有垂直直線，並挑選字元總數最少的直線並視為y軸，並以黃線表示如圖(12)所示。然後再依據*x*與*y*軸的字元中心，做水平與垂直的連線，其交點係此座標的字串位置，並完成定位。最後將兩軸標題以及所對應之字串，以文字方式記錄起來。

|  |  |
| --- | --- |
| FPK2_01  圖(19) 字串中心點連線 | FPK2_01  圖(20) *x*與*y*軸的標題定位 |

**四、實驗結果**

本章節將敘述實驗的相關設定與其結果，分別可分為4.1實驗環境：說明實驗的硬體與軟體環境；4.2實驗用超參數：說明字串區域偵測與字串辨識各模型訓練的訓練參數；4.3實驗用資料集：說明用以訓練與測試的資料集特性；4.4實驗效能評估方式：說明實驗效能評估的計算方式與相關公式；4.5 效能分析：展示字串區域偵測與字串辨識個別與合併的交叉測試結果，並說明三個結論，第一是電腦自動生成資料可提升模型準確率；第二是自定義模型較佳；第三是合併模型有助於提升準確率。

**4.1實驗環境**

本實驗使用的軟體與硬體配置如表(1)所示，硬體上的CPU處理器使用Intel Core i7-870，GPU處理器則是使用NVDIA GeForce RTX 2060 8GB，RAM大小為16G。軟體上的作業系統使用的Windows 10，程式語言係Python與C++語言，其框架則是Pytorch與Darknet，電腦視覺之相關的程式庫則分別使用了OpenCV-Python與OpenCV。

表(1)軟體與硬體配置

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 硬體配置 | | | | |
| CPU處理器 | Intel Core i7-870 | | | |
| GPU處理器 | NVDIA GeForce RTX 2060 8GB | | | |
| RAM記憶體 | 16G | | | |
| 軟體配置 | | | | |
|  | 字串辨識 | 字串區域偵測(Unet、EAST) | 字串區域偵測(YOLOV4) | 合併系統 |
| 作業系統 | Windows 10 | | | |
| 程式語言 | Python 3.7.6 | Python 3.7.6 | C++ | Python 3.7.6 |
| 神經網路框架 | Pytorch 1.5.0+cu101 | Pytorch 1.5.0+cu101 | Darknet | Pytorch 1.5.0+cu101 |
| 電腦視覺程式庫 | OpenCV-python 3.4.2 | OpenCV-python 3.4.2 | OpenCV 3.0.0 | OpenCV-python 3.4.2 |

**4.2實驗用超參數**

本節將說明字串區域偵測與字串辨識各模型訓練的訓練參數。字串區域偵測共使用了三個模型，分別為EAST[1]、YOLOv4[2]、U-net[3]與字串辨識則使用了4個模型RARE[4]、CRNN[5]、Rosetta[6]、StarNet[7]與一個自定義的模型，訓練參數可分為批次大小(Batch Size)、優化器(Optimizer)、學習率(Learning rate)與圖片輸入的大小(Intput size)，詳細資料如表(2)與(3)所示。

表(2)字串區域偵測模型超參數設定

|  |  |  |  |
| --- | --- | --- | --- |
| 模型  參數 | EAST[1] | YOLOv4[2] | U-net[3] |
| Batch size | 8 | 32 | 1 |
| Optimizer | Adam |  | Adam |
| Learning rate | 0.001 | 0.0001 | 0.0001 |
| Intput size | 512× 512 | 608×608 | 800×800~1500×1500 |

表(3)字串辨識模型超參數設定

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 模型  參數 | RARE[4] | CRNN[5] | Rosetta[6] | STAR-Net[7] | RNBLC |
| Batch size | 192 | 192 | 192 | 192 | 192 |
| Optimizer | Adadelta | Adadelta | Adadelta | Adadelta | Adadelta |
| Learning rate | 1 | 1 | 1 | 1 | 1 |
| Intput size | 32× 100 | 32× 100 | 32× 100 | 32× 100 | 32× 100 |

**4.3實驗用資料集**

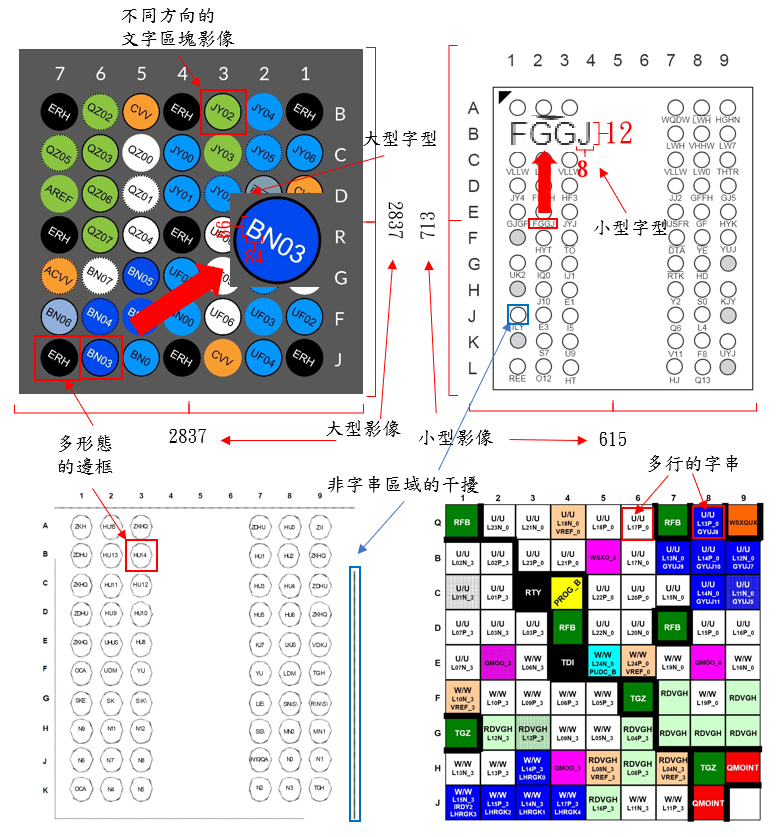
為了清楚描述實驗結果，本節將分別說明訓練與測試的資料集特性。

**4.3.1 測試資料集**

本實驗之測試資料集來至富比酷舉辦的「2019富比庫校園達人秀 - 以 i 為名，AI動起來 !」之競賽資料，總共包含18張圖像，且各自複合六種形式的格式，細項如表(4)所示，分別為多形態的邊框、不同方向的文字區域影像、不同顏色的字與背景、不同尺度的影像與字型、非字串區域的干擾與多行的字串，如圖(21)所示。

表(4)測試資料集統計資料

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 格式  圖檔  檔名 | 圖編號 | 多形態的邊框 | 不同方向的文字區域影像 | 不同顏色的字與背景 | 大型圖像 | 大型字體 | 小型字體 | 非字串區域的干擾 | 多行字 |
| FPK\_01.jpg | 01 | 🗸 | 🗸 | 🗸 | 🗸 | 🗸 |  |  |  |
| FPK\_02.jpg | 02 |  |  |  |  |  | 🗸 | 🗸 |  |
| FPK\_03.jpg | 03 | 🗸 |  | 🗸 |  |  | 🗸 |  | 🗸 |
| FPK\_04.jpg | 04 | 🗸 | 🗸 | 🗸 |  |  |  | 🗸 | 🗸 |
| FPK\_05.jpg | 05 | 🗸 |  |  |  |  | 🗸 |  |  |
| FPK\_06.jpg | 06 | 🗸 |  |  |  |  | 🗸 |  | 🗸 |
| FPK\_07.jpg | 07 | 🗸 | 🗸 | 🗸 | 🗸 | 🗸 |  |  |  |
| FPK\_08.jpg | 08 | 🗸 | 🗸 | 🗸 |  |  |  | 🗸 | 🗸 |
| FPK\_09.jpg | 09 | 🗸 |  |  |  |  | 🗸 |  |  |
| FPK\_10.jpg | 10 | 🗸 |  | 🗸 | 🗸 |  | 🗸 |  | 🗸 |
| FPK\_11.jpg | 11 |  |  |  |  |  | 🗸 | 🗸 |  |
| FPK\_12.jpg | 12 | 🗸 |  | 🗸 |  |  | 🗸 |  | 🗸 |
| FPK\_13.jpg | 13 |  |  |  |  |  | 🗸 | 🗸 |  |
| FPK\_14.jpg | 14 | 🗸 |  | 🗸 |  |  | 🗸 |  | 🗸 |
| FPK\_15.jpg | 15 | 🗸 |  |  |  |  | 🗸 |  |  |
| FPK\_16.jpg | 16 |  |  |  |  |  | 🗸 | 🗸 |  |
| FPK\_17.jpg | 17 | 🗸 |  |  |  |  | 🗸 |  |  |
| FPK\_18.jpg | 18 | 🗸 |  |  |  |  | 🗸 |  |  |



圖(21) 測試資料及六種形式範例

**4.3.1.1 字串區域偵測之測試資料集**

字串區域偵測實驗之測試資料是表(4)的測試資料集，總共18張的測試影像。

**4.3.1.2 字串辨識之測試資料集**

由於本實驗的資料集有限，因此為了增加字串辨識的資料量。本實驗將分為三類測試集，第一類測試集是直接將偵測後的字串區域做為測試集，第二類測試集則是將第一類測試集經過正規化後的字串區域，第三類測試集則是將第二類測試集轉成二值化的字串區域，如圖(22)所示。





第三類

測試集

第二類

測試集

第一類

測試集



圖(22) 字串辨識之資料測試集

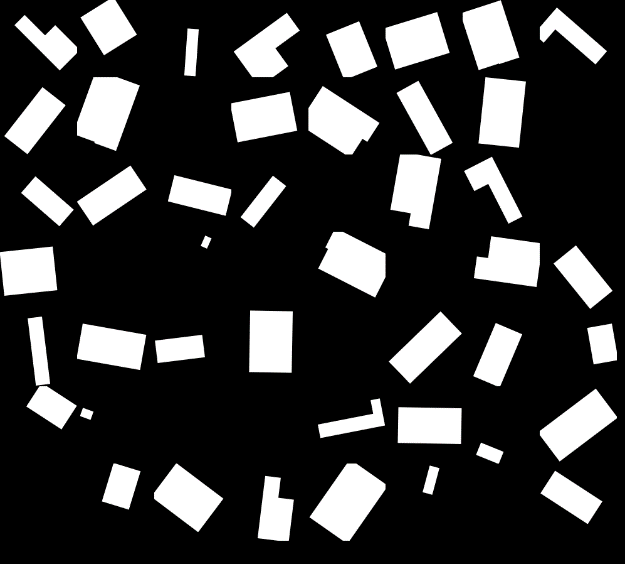
**4.3.2 訓練資料集**

本實驗的訓練資料集分為字串區域偵測與字串辨識的資料訓練集，由於實驗資料集的資料稀少且具獨特性，因此為了能克服本研究要解決的問題，本實驗採用現有的資料集特性，並大量生成字串區域偵測與字串辨識的訓練資料集。

**4.3.2.1 字串區域偵測模型之訓練資料集**

為了能對複雜背景、字體大小不一、字型不同、其他物件的干擾以及不同的旋轉角度等問題，本研究針對要解決的影像，統計分析後歸納為影像的長與寬、表格大小、每格不同的顏色、有無邊框、字串中心上下方有無圓框等變數。因此，本研究各種參數隨機產生後組成一張訓練影像，其參數分別為：(1)影像的高度從800至1500像素(pixel)之間。(2)影像的寬度從800至1500像素之間。(3)每個網格大小從50至600像素之間。(4)每個網格的顏色(R、G、B)有1/2的機率不相同，有1/2的機率相同。(5)網格的樣式分為7種：無邊框、有邊框、四邊的邊框有1/2的機率顯示或不顯示、以字串中心為圓心的大圓框、以字串中心為圓心的小圓框、在字串中心上方有小圓框、在字串中心下方有小圓框。(6)字串長度從0至13個字元，其中字串長度會受到網格空間影響，而必須調整。(7)字串旋轉角度從1至360度之間。 (8)178種字型與字體大小為10至80間，本文選擇509種大小與字型不同的字型。其影像中的參數如圖(23)所示。依據以上變數隨機產生100000張的訓練影像，其產生後的訓練影像如圖24所示。再將選取訓練影像中的字串，並且進行U-net[3]類神經網路模型的學習，其訓練字串的區域如圖25所示；與選取訓練影像中的區塊，進行YOLOv4[2]、EAST[1]類神經網路模型的學習，其訓練區塊的區域如圖26所示。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  | |  |  |  | abc |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  | bcd |  |  |  | |  |  |  |  |  |  |  |  |  |  | |  |  | cde |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |   圖(23)訓練影像的參數 | 6  圖(24)產生後的訓練影像示意圖 |



高

寬

網格且有框

字串中心大圓框

字串上方小圓框

圖(26) U-net訓練字串的區域(白色區域)

圖(25) YOLOv4、EAST訓練區塊的區域(紅色框選區域)

**4.3.2.2 字串辨識資料集**

字串辨識資料集包含的圖像為單行字串無角度之圖像。在生成時採用了4條生成的規則：第一條為採用178種字體，搭配上字體大小10至80後會有509種字體，每張圖會隨機採用一種字體做生成，以兼容不同字體的辨識；第二條為不同的背景色與字顏色，顏色範圍為RGB三通道，背景色與字顏色為互斥，以兼容不同顏色的辨識；第三條為不同的邊界大小，寬邊界範圍為20~，高邊界為0~，以增加對於辨識的強健性。總共生產了一百萬張圖像作為訓練集。



圖(27)各種生成的字串辨識測試集範例

**4.4實驗效能評估方式**

本實驗將詳細說明字元區域偵測、字串辨識、合併流程三大部分的準確率評估方式。合併流程為字元區域偵測與字串辨識後的評估方式，需同時評估偵測與辨識的準確性。

**4.4.1 字元區域偵測的評估方式**

為了能以量化方式，並且更精確評估實驗方法的效能，本研究採用混淆矩陣及IoU(intersection over union)，以數據化的方式呈現，評估準則如下所示：

1. IoU(intersection over union)：。
2. 真陽性 (true positive；TP)：預測邊框與實際邊框 之數量。
3. 偽陽性 (false positive；FP)：。
4. 偽陰性 (false negative；FN)：。
5. 精確率 (precision；PPV)：。
6. 召回率 (recall；TPR)：。
7. F1分數 (F1-Score)：。

F1分數即為本研究字元區域偵測的評估方式。

**4.4.2 字串辨識的評估方式**

為了能以量化兩字串的相似度，本研究採用數學家Vladimir Levenshtein提出的編輯距離(Levenshtein distance)，以此計算字串的準確率。另外一張圖像具有多條字串，故須計算平均準確率，評估準則如下所示：

1. 編輯距離(Levenshtein distance；LD)：預測字串編輯成實際字串所需最小次數，單次編輯包含三種操作，分別為插入一個字元、刪除一個字元與將一個字元取代成另一字元。
2. 準確率 (accuracy；ACC)：。
3. 平均準確率：

平均準確率即為本研究字串辨識的評估方式。

**4.4.3 合併流程的評估方式**

為同時評估字元區域偵測與字串辨識，本研究將結合編輯距離(Levenshtein distance)來修改混淆矩陣，評估準則如下所示：

1. 真陽性 (true positive；TP)：預測邊框與實際邊框 之數量。
2. 偽陽性 (false positive；FP)：。
3. 偽陰性 (false negative；FN)：。
4. 精確率 (precision；PPV)：。
5. 召回率 (recall；TPR)：。
6. F1分數 (F1-Score)：。

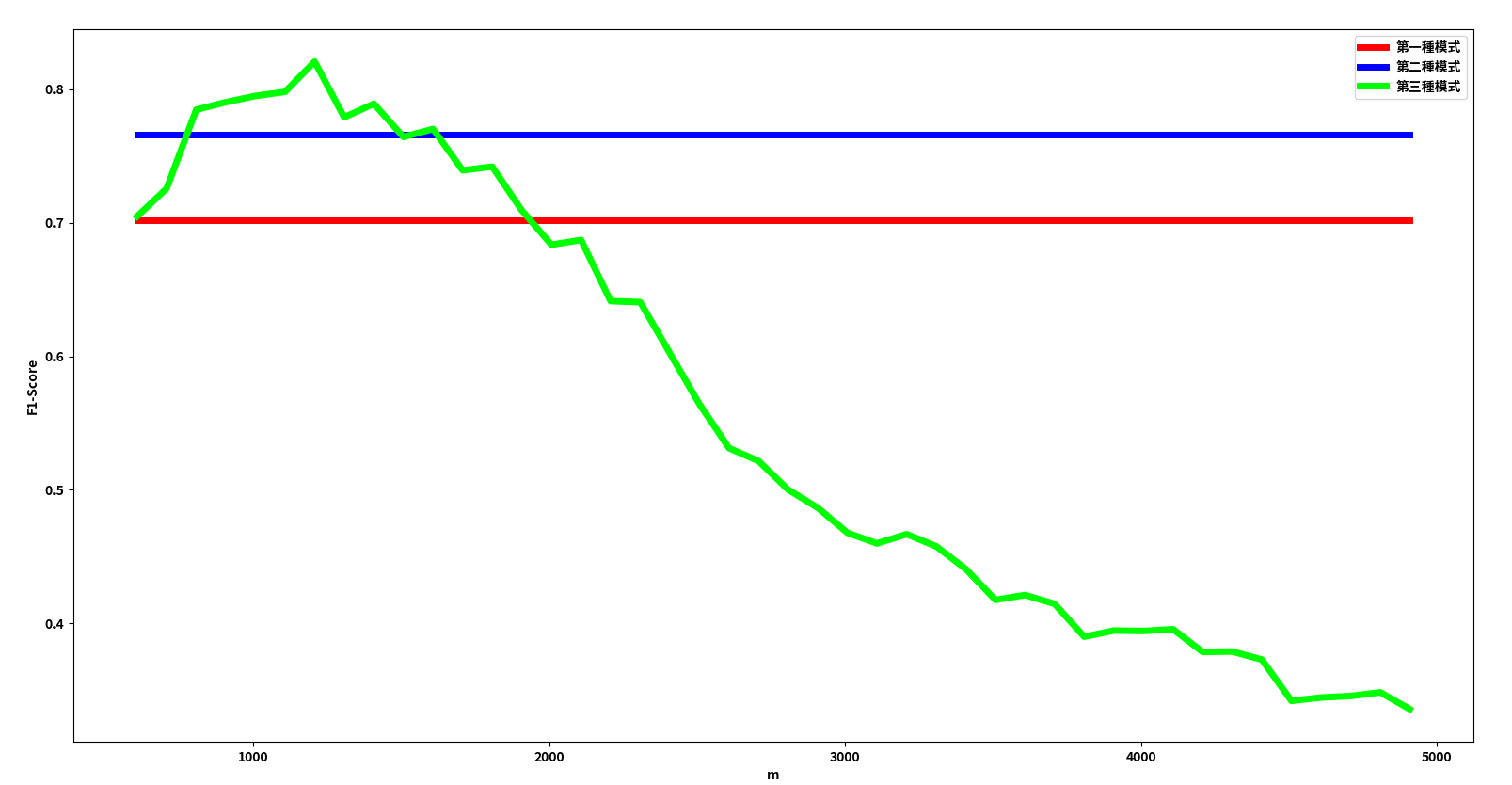
F1分數即為本研究合併流程的評估方式。

**4.5效能分析**

本實驗之效能分析分成四個部分進行，第一部分為實驗並討論最佳的字串區域偵測最佳輸入方式，第二部分為比較字串區域偵測的不同模型之實驗結果，第三部分是比較字串辨識各模型準確率，第四部分則是進行不同模型的字串辨識交叉實驗。

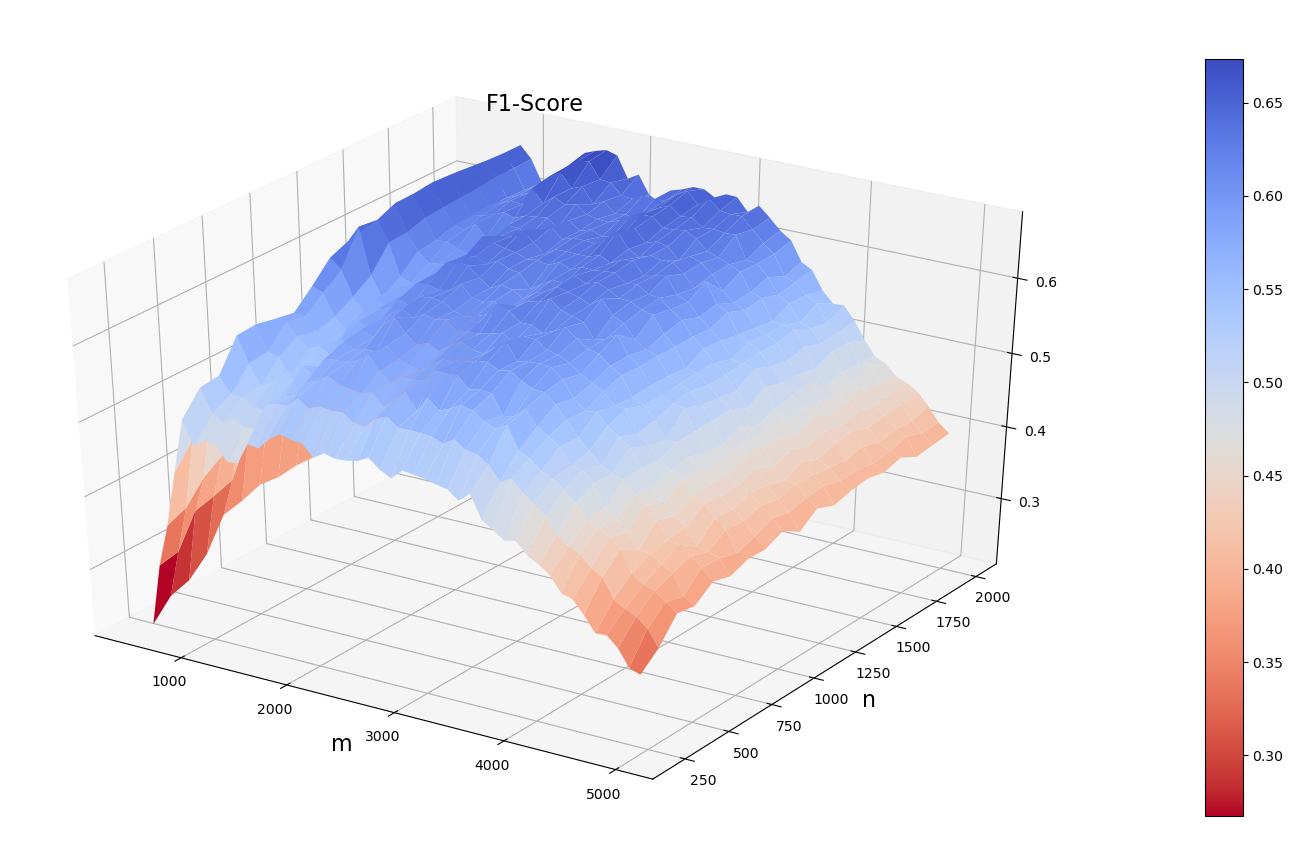
**4.5.1字串區域偵測輸入模式分析**

此節將分別探討YOLOv4[2]與U-net[3]的輸入模式，並提出一個可參考的最佳模式方式。

在YOLOv4[2]辨識前會分析三種模式，第一種係將所有影像調整大小至608×608像素；第二種係將所有影像取大小608×608大小的子影像，再以重疊方式往右或往右304像素取大小608×608大小的子影像，以此類推；第三種係將所有影像調整大小至*m*×*m*像素，再取大小608×608大小的子影像，再以重疊方式往右或往右304像素取大小608×608大小的子影像，以此類推，*m*需要經過測試後。上述的測試結果如圖(28)，紅色線段與藍色線段為第一種與第二種模式的偵測F1-Score，分別為0.7659與0.7017，置於圖上與第三種模式比較，綠色線段則是測試第三種模式在不同*m*值下的偵測F1-Score，測試範圍從608至4908，測試級距為100。在測試下第三種模式m於1208時可達最高的偵測F1-Score，其值為0.8208，因此我們會建議YOLOv4[2]在此測試集下採用第三種模式，m值取1208。 

圖(28)YOLOv4三種模式測試結果

在U-net[3]辨識前會將所有影像調整大小至*m*×*m*像素，再取大小*n*×*n*大小的子影像，再以重疊方式往右或往右*n*/2像素取大小*n*×*n*大小的子影像，以此類推。測試結果如圖(29)，*m*值測試範圍從500至5000，測試級距為100，n值測試範圍因受限於GPU容量大小因此範圍從200至2000，測試級距為100。其中*m*與*n*分別於2300與1500時達到最高的偵測F1-Score，故我們建議在測試U-net[3]在此測試集下需採用此二值作為偵測前的調整。

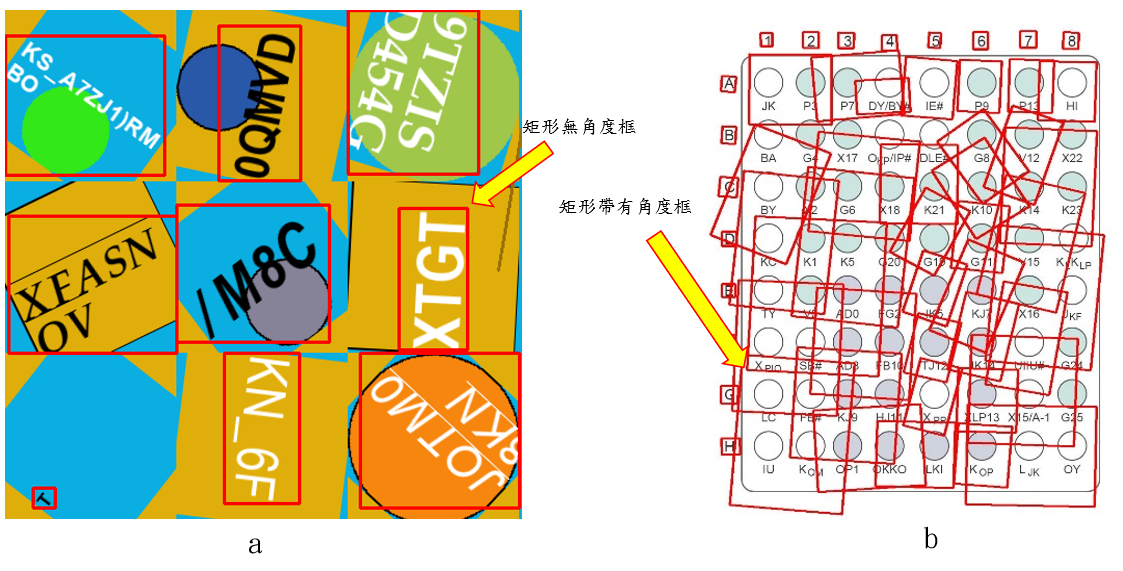


圖(29) U-net變形模式測試結果

**4.5.2字串區域偵測分析**

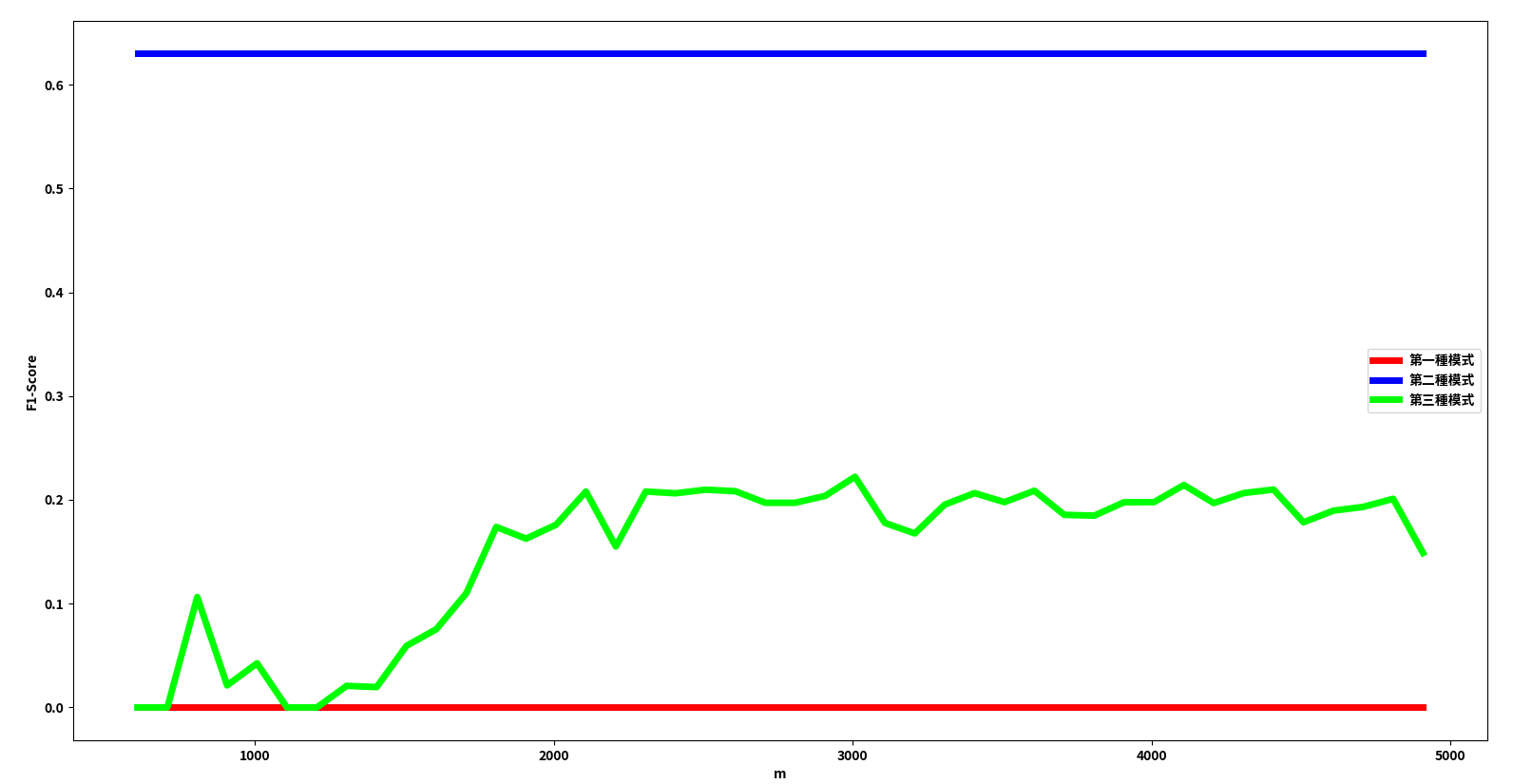
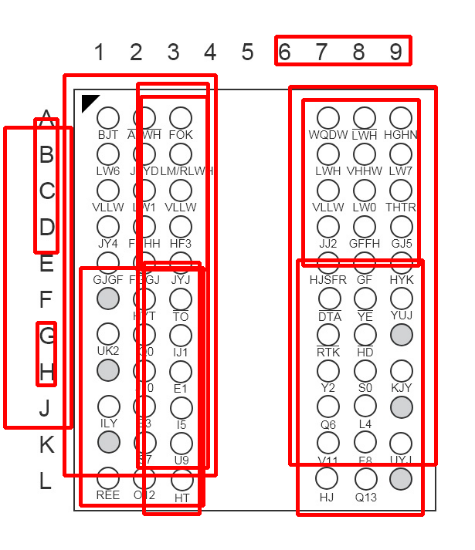
此節實驗了EAST[1]、YOLOv4[2]、U-net[3]對應初步偵測、加上後處理與合併模型三種策略下對於測試資料集的偵測F1-Score如表(5)。橫軸是採用的偵測策略，縱軸是圖片的編號與18張圖像的平均。

其中EAST[1]經由訓練數據集訓練過後，在測試數據集上各圖片的偵測F1-Score有17張圖像皆小於0.7，其中兩張甚至為0。經觀察後推測，EAST[1]是一個偵測形變字串的模型，但本研究採用的訓練資料集皆為矩形無角度的標籤如圖(30.a)，且經訓練過後EAST[1]依然會預測出帶角度的矩形框圖(30.b)，以致無法成功辨識，固認為EAST並不適合以本研究的訓練集訓練。



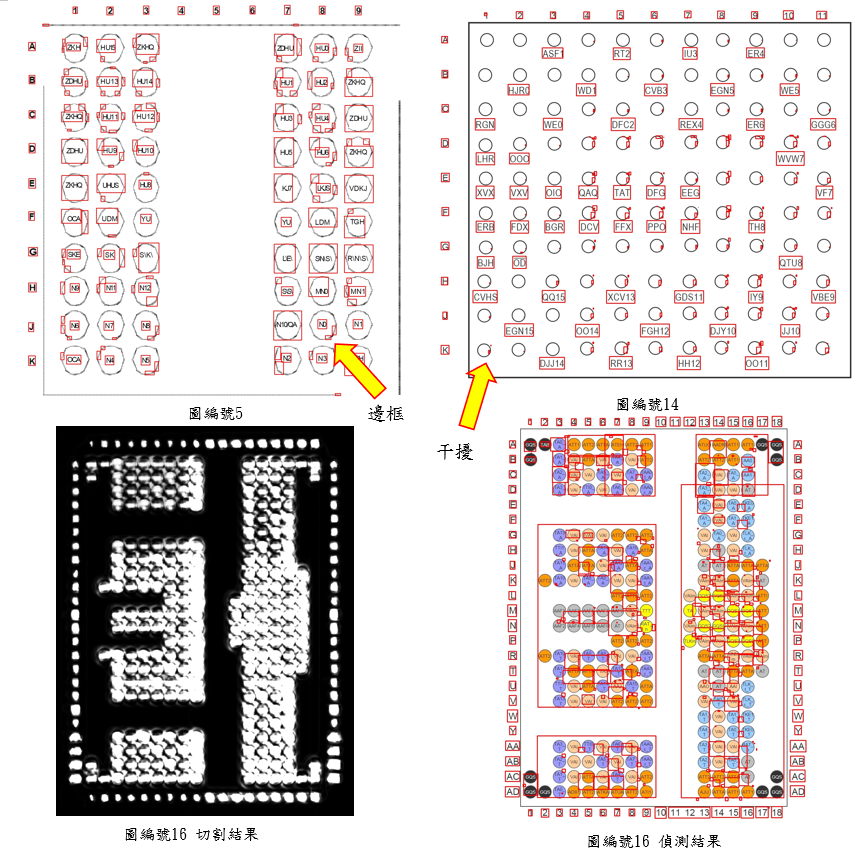
圖(30) EAST訓練資料(左)與EAST預測結果(右)

YOLOv4[2]於17張圖皆有達到偵測F1-Score大於0.8，唯一在圖標號13上偵測F1-Score為0，完全無法預測。根據4.5.1測試結果中發現，如使用第二種模式辨識圖標號13，其偵測F1-Score可達0.6299，採用第一種模式一樣為0，第三種模式則是最高為0.22，固推測，此圖因圖片與字型過小，導致經變形後會使YOLOv4[2]於本研究訓練集訓練後的模型，無法辨識。



圖(31) YOLOv4於圖編號13預測結果 圖(32) YOLOv4於圖編號13三種模式測試結果

U-net[3]則是於7張圖片達到偵測F1-Score大於0.8，為表現較佳，於圖編號05、06、14、16小於0.5，表現較差。其中圖編號05、06、14因邊框與干擾，導致其偵測出許多非字串之區域，預期於字串辨識時需辨識空字串，以排除干擾。圖編號14則是字與邊框的距離過小，以致切割時產生許多沾黏，無法有效切開成不同的邊框如 圖(33)。



圖(33)U-net偵測結果範例

U-net[3]與 YOLOv4[2]各自於18張圖像都具有最好的結果，固我們合併了U-net[3]與 YOLOv4[2]的模型的預測結果，雖說平均偵測F1-Score並非最高，但卻可以在多數圖片上皆有最佳的偵測F1-Score，固認為合併模型有一定的助益，4.5.4會分析不同偵測模型策略搭配不同字串辨識模型的交叉比對結果，進一步證明合併模型的有效性。

表(5) 偵測策略於不同圖片之效能

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **偵測策略**  **圖片**  **標號** | **U-net[3]** | **U-net[3]+**  **後處理** | **YOLOv4[2]** | **YOLOv4[2]+後處理** | **EAST[1]** | **EAST[1]+後處理** | **U-net[3]+ YOLOv4[2]+後處理** |
| 01 | 0.52 | 0.77 | 0.90 | 0.94 | 0.21 | 0.21 | 0.98 |
| 02 | 0.48 | 0.61 | 0.91 | 0.93 | 0.22 | 0.20 | 0.57 |
| 03 | 0.92 | 0.96 | 0.81 | 0.87 | 0.70 | 0.70 | 0.98 |
| 04 | 0.80 | 0.86 | 0.81 | 0.93 | 0.86 | 0.86 | 0.92 |
| 05 | 0.26 | 0.26 | 0.96 | 0.99 | 0.07 | 0.07 | 0.54 |
| 06 | 0.16 | 0.48 | 0.95 | 0.93 | 0.10 | 0.10 | 0.79 |
| 07 | 0.64 | 0.72 | 0.78 | 0.81 | 0.25 | 0.25 | 0.84 |
| 08 | 0.66 | 0.79 | 0.71 | 0.85 | 0.74 | 0.74 | 0.87 |
| 09 | 0.88 | 0.94 | 0.92 | 0.93 | 0.07 | 0.07 | 0.96 |
| 10 | 0.84 | 0.91 | 0.82 | 0.83 | 0.61 | 0.62 | 0.91 |
| 11 | 0.98 | 1.00 | 0.93 | 0.99 | 0.15 | 0.12 | 1.00 |
| 12 | 0.72 | 0.72 | 0.93 | 0.96 | 0.60 | 0.60 | 0.73 |
| 13 | 0.78 | 0.87 | 0.00 | 0.00 | 0.00 | 0.00 | 0.88 |
| 14 | 0.23 | 0.31 | 0.82 | 0.80 | 0.00 | 0.00 | 0.72 |
| 15 | 1.00 | 1.00 | 0.99 | 0.99 | 0.72 | 0.72 | 1.00 |
| 16 | 0.45 | 0.78 | 0.79 | 0.86 | 0.14 | 0.14 | 0.97 |
| 17 | 0.21 | 0.21 | 0.82 | 0.81 | 0.10 | 0.10 | 0.37 |
| 18 | 0.91 | 0.98 | 0.95 | 0.98 | 0.13 | 0.13 | 0.99 |
| 平均 | 0.63 | 0.73 | 0.82 | 0.86 | 0.31 | 0.31 | 0.83 |

**4.5.3字串辨識分析**

此節先於4.5.3.1分析RARE[4]、CRNN [5]、Rosetta[6]、StarNet[7]之原論文模型，在未經本實驗資料集訓練前於第一至三類字串辨識測試資料集的準確率。再來於4.5.3.2實驗RARE[4]、CRNN [5]、Rosetta[6]、StarNet[7]與RNBLC、經本實驗資料集訓練後於第三類字串辨識測試資料集上的準確率。最後於4.5.3.3分析合併模型後的結果。

**4.5.3.1原論文模型分析**

此節先測試了RARE[4]、CRNN [5]、Rosetta[6]與StarNet[7]四個模型在未經本實驗訓練集訓練前於第一至三類字串辨識資料集的準確率如表(6)。其中RARE[4]於第一類至三類字串辨識資料集上準確率皆最高，固認為RARE[4]比較適合本實驗的測試資料集。另外第二類資料集在各模型準確率皆最高，這影響本實驗在製作字串辨識訓練集時，是參考至第二類資料集，認為類第二類資料集可以使訓練結果最好。

表(6) 各模型於第一至三類字串辨識資料集之準確率

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **模型**  **資料集** | **RARE[4]** | **CRNN[5]** | **Rosetta[6]** | **Star-Net[7]** |
| 第一類資料集 | 84.05 | 79.75 | 79.14 | 81.13 |
| 第二類資料集 | 84.57 | 80.82 | 80.02 | 81.66 |
| 第三類資料集 | 82.12 | 77.37 | 74.83 | 80.03 |

**4.5.3.2訓練後模型分析**

本節將探討RARE[4]、CRNN [5]、Rosetta[6]、StarNet[7]與RNBLC經本實驗資料集訓練後於第三類字串辨識測試資料集上的準確率，並與未訓練之模型相互比較，結果如表(7)。可得出兩個結論，一是經過本實驗訓練集訓練後於各模型皆可提升準確率，以此證明本研究之字串辨識訓練集對於訓練有助益；二是本研究提出的RNBLC準確率高達88.35，在各模型之中為最高。

表(7)加入訓練後結果與RNBLC模型的準確率結果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **模型**  **資料集** | **RARE[4]** | **CRNN[5]** | **Rosetta[6]** | **StarNet[7]** | **RNBLC** |
| 第一類資料集 | 84.05 | 79.75 | 79.14 | 81.13 |  |
| 第二類資料集 | 84.57 | 80.82 | 80.02 | 81.66 |  |
| 第三類資料集 | 82.12 | 77.37 | 74.83 | 80.03 |  |
| 訓練後 | 86.10 | 87.45 | 81.59 | 84.59 | 88.35 |

**4.5.3.3合併模型分析**

得到17種模型的結果後，本研究採用經過本實驗訓練過的5種模型的參與投票，其結果如表(8)，會使準確率比最高單一模型提高了約0.5個百分點，對於模型的幫助有限且所花時間將近五倍，但這是以裁切好的圖像計算的準確率，對於偵測出來的圖像比較將在4.6說明。

表(8) 合併模型準確率比較

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **模型**  **資料集** | **RARE[4]** | **CRNN[5]** | **Rosetta[6]** | **StarNet[7]** | **RNBLC** | **訓練後模型投票** |
| 第一類資料集 | 84.05 | 79.75 | 79.14 | 81.13 |  |  |
| 第二類資料集 | 84.57 | 80.82 | 80.02 | 81.66 |  |  |
| 第三類資料集 | 82.12 | 77.37 | 74.83 | 80.03 |  |  |
| 訓練後 | 86.10 | 87.45 | 81.59 | 84.59 | 88.35 | 88.80712 |

**4.6流程效能分析與最佳組合**

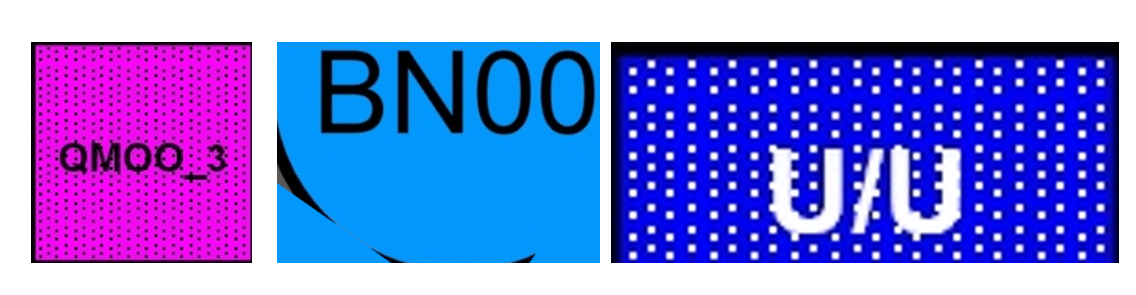
本節將分析本實驗流程之效能，包含字串區域偵測、字串區域方向的正規化、字串行間的切割與字串辨識。其中字串區域偵測有YOLOv4[2]、U-net[3]與YOLOv4[2]和U-net[3]的合併模型三種選擇。字串辨識有RARE[4]、CRNN[5]、Rosetta[6]與STAR-Net[7]四種模型搭配上三類輸入格式共12種選擇，與加上經本實驗訓練集訓練過後RARE[4]、CRNN[5]、Rosetta[6]、STAR-Net[7]與RNBLC共5種選擇，再加上以訓練後5種模型合併的模型，種共有18種選擇。綜合所有流程與選擇，為每個選擇下的流程都計算一個修改過的F1-Score如表14、15。表中第一行是字串辨識使用的模型名稱，其中重訓練只經過本實驗設計的訓練集訓練過的模型，投票1是分析字串辨識時找出的最佳合併模型結果，投票2則是使用單一字串辨識模型搭配不同字串區域偵測模型，取平均準確率最好的三個模型進行合併的結果；第二行則是字串辨識使用三種類型的輸入資料，第一類為只經過字串的行切割後的圖像、第二類為經過區塊選轉與字串的行切割後的圖像、第三類則是經過字串區域的正規化、字串行間的切割後並二質化後的圖像；第一欄則是使用的偵測模型。紅色字體表示橫向最高值，也就是選擇使用字串辨識模型搭配不同字串區域偵測模型之最高準確率；黃色底色表示縱向最高值，也就是選擇使用字串區域偵測模型搭配不同字串辨識模型之最高準確率。

表(9) 辨識模型與偵測模型交插測試結果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 辨識模型  偵測模型 |  | YOLOv4[2] | U-net[3] | U-net[2]+ YOLOv4[3] | 平均 |
| RARE[4] | 第一類資料集 | 0.53 | 0.50 | 0.55 | 0.53 |
| RARE[4] | 第二類資料集 | 0.59 | 0.55 | 0.60 | 0.58 |
| RARE[4] | 第三類資料集 | 0.56 | 0.52 | 0.57 | 0.55 |
| 重訓練RARE[4] | 第二類資料集 | 0.66 | 0.66 | 0.71 | 0.67 |
| CRNN[5] | 第一類資料集 | 0.45 | 0.41 | 0.45 | 0.44 |
| CRNN[5] | 第二類資料集 | 0.51 | 0.46 | 0.50 | 0.49 |
| CRNN[5] | 第三類資料集 | 0.47 | 0.43 | 0.48 | 0.46 |
| 重訓練CRNN[5] | 第二類資料集 | 0.66 | 0.62 | 0.66 | 0.65 |
| Rosetta[6] | 第一類資料集 | 0.46 | 0.43 | 0.46 | 0.45 |
| Rosetta[6] | 第二類資料集 | 0.52 | 0.48 | 0.52 | 0.51 |
| Rosetta[6] | 第三類資料集 | 0.47 | 0.44 | 0.49 | 0.46 |
| 重訓練Rosetta[6] | 第二類資料集 | 0.60 | 0.59 | 0.63 | 0.61 |
| Star-Net[7] | 第一類資料集 | 0.47 | 0.44 | 0.48 | 0.46 |
| Star-Net[7] | 第二類資料集 | 0.55 | 0.50 | 0.55 | 0.53 |
| Star-Net[7] | 第三類資料集 | 0.52 | 0.47 | 0.51 | 0.50 |
| 重訓練StarNet[7] | 第二類資料集 | 0.64 | 0.60 | 0.65 | 0.63 |
| RNBLC | 第二類資料集 | 0.69 | 0.66 | 0.70 | 0.69 |
| 投票 | 第二類資料集 | 0.68 | 0.64 | 0.69 | 0.67 |
| 平均 |  | 0.56 | 0.52 | 0.57 |  |

對於字串區域偵測模型YOLOv4[2]與U-net[3]加的合併模型在搭配上不同字串辨識模型時，對比單一用YOLOv4[2]與U-net[3]在大部分情況與平均準確率都較高，這與偵測模型的分析相符，以此證明了合併偵測模型的有效性。

對於字串辨識模型，投票的方式並沒有比其他單一模型有較高的準確率，主因在真實偵測的偵測框，經常帶有許多雜訊且會有不準確的預測框如圖(34)，以致於與4.5.3.3不一致。另外對於字串辨識的單一模型，RNBLC的模型準確率在搭配不同字串區域偵測模型下都可以達到最佳與平均準確率；在使用本實驗的字串辨識訓練資料集訓練，皆可以大幅提高準確率。



圖(34) 雜訊與不準確的預測框

綜合上述所述，字串區域偵測與字串辨識分別採合併模型與RNBLC的方式，可以達到最佳準確率。電腦生成的資料集作為訓練集，是可以順利訓練並辨識，並不會遜於真實情況的訓練集，且可以為不同情況做客製化。

**五、結論**

本文提出了對於複雜的表格影像的辨識流程、字串區域偵測與字串辨識的多模型合併方式、字串區域偵測與字串辨識的訓練資料大量生成方式與一個獨創的字串辨識模型RNBLC。並分析了3種字串區域偵測模型與5種字串辨識模型，在「2019富比庫校園達人秀 - 以 i 為名，AI動起來 !」之競賽資料上，獨自辨識與整合流程後的準確率，並找出最佳組合為U-net+YOLOv4搭配RNBLC，平均F1-Score達0.71。

另外因為此資料集，每張圖像皆各具代表性，如後續有大量表格辨識的需求，且與18張任一圖像中相似，會建議參照表(10)去選擇最好的組合去調整。

表(10)單一圖片最佳組合流程

|  |  |  |  |
| --- | --- | --- | --- |
| 圖片編號 | 偵測模型 | 辨識模型 | F1-Score |
| 01 | YOLOv4 | RNBLC | 0.80 |
| 02 | U-net | 重訓練RARE | 0.83 |
| 03 | U-net | 重訓練RARE | 0.64 |
| 04 | YOLOv4 | 重訓練RARE | 0.92 |
| 05 | YOLOv4 | StarNet+第一類資料集 | 0.82 |
| 06 | YOLOv4 | 重訓練StarNet | 0.66 |
| 07 | U-net | RNBLC | 0.68 |
| 08 | U-net+YOLOv4 | RNBLC | 0.86 |
| 09 | U-net+YOLOv4 | 投票 | 0.90 |
| 10 | U-net+YOLOv4 | 重訓練StarNet | 0.90 |
| 11 | U-net+YOLOv4 | RARE+第二類資料集 | 0.88 |
| 12 | YOLOv4 | 投票 | 0.78 |
| 13 | U-net+YOLOv4 | RARE+第二類資料集 | 0.54 |
| 14 | U-net+YOLOv4 | StarNet+第一類資料集 | 0.51 |
| 15 | U-net+YOLOv4 | RARE+第二類資料集 | 0.95 |
| 16 | U-net | 重訓練CRNN | 0.90 |
| 17 | YOLOv4 | 投票 | 0.78 |
| 18 | U-net+YOLOv4 | 投票 | 0.85 |

對於字串區域偵測與字串辨識的訓練資料大量生成方式，在偵測模型下證明其可順利訓練並偵測，辨識模型則是可以提升模型準確率，並不亞於真實圖片。往後會建議細部的修改，以貼合辨識目標，預期可進一步提升準確率。

**參考文獻**

[1]Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., & Liang, J. (2017). East: An efficient and accurate scene text detector. arXiv:1704.03155 [cs]. <http://arxiv.org/abs/1704.03155>

[2] Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv:2004.10934 [cs, eess]. http://arxiv.org/abs/2004.10934

[3]Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. arXiv:1505.04597 [cs]. <http://arxiv.org/abs/1505.04597>

[4] Shi, B., Wang, X., Lyu, P., Yao, C., & Bai, X. (2016). Robust scene text recognition with automatic rectification. arXiv:1603.03915 [cs]. http://arxiv.org/abs/1603.03915

[5] B. Shi, X. Bai and C. Yao, An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition, in: IEEE Transactions on Pattern Analysis and Machine Intelligence, Nov 1, 2017, pp. 2298–2304.

[6] Borisyuk, F., Gordo, A., & Sivakumar, V. (2018). Rosetta: Large scale system for text detection and recognition in images. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 71–79. https://doi.org/10.1145/3219819.3219861

[7] Liu, W., Chen, C., Wong, K., Su, Z., & Han, J. (2016). Star-net: A spatial attention residue network for scene text recognition. BMVC. https://doi.org/10.5244/C.30.43

[8] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V.R. Chandrasekhar, S. Lu, ICDAR 2015 competition on robust reading, in: International Conference on Document Analysis and Recognition, 2015, pp. 1156–1160.

[9] A. Shahab, F. Shafait, A. Dengel, ICDAR 2011 robust reading competition challenge 2: Reading text in scene images, in: International Conference on Document Analysis and Recognition, 2011, pp. 1491–1496.

[10] L. Neumann, J. Matas, Efficient scene text localization and recognition with local character refinement, in: International Conference on Document Analysis and Recognition, 2015, pp. 746–750.

[11] V. Khare, P. Shivakumara, P. Raveendran, M. Blumenstein, A blind deconvolution model for scene text detection and recognition in video, in: Pattern Recognit. 54, 2016, pp. 128–148.

[12] M. Liao, B. Shi, X. Bai, X. Wang, W. Liu, TextBoxes: A fast text detector with a single deep neural network, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, Feb 4–9, 2017, San Francisco, California, USA, pp. 4161–4167.

[13] S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu, C.L. Tan, Text flow: A unified text detection system in natural scene images, in: CoRR, arXiv:1604.06877, 2016, pp. 4651–4659.

[14] M. Basavanna, P. Shivakumara, S.K. Srivatsa, G. Kumar, Multi-oriented text detection in scene images, in: International Journal of Pattern Recognition and Artificial Intelligence 26(07), Feb, 2013, pp. 19–36.

[15] Y. Zhu, C. Yao, X. Bai, Scene text detection and recognition: recent advances and future trends, in: Frontiers of Computer Science 10(1), 2016, pp. 19–36.

[16] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, X. Bai, Multi-oriented text detection with fully convolutional networks, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, June 27–30, 2016, Las Vegas, NV, USA, pp. 4159–4167.

[17] C. Tian, Y. Xia, X. Zhang, X. Gao, Natural scene text detection with MC-MR candidate extraction and coarse-to-fine filtering, in: Neurocomputing. 260, 2017, pp. 112–122.

[18] T. Wang, D.J. Wu, A. Coates, A.Y. Ng, End-to-end text recognition with convolutional neural networks, in: International Conference on Pattern Recognition, 2013, pp. 3304–3308.

[19] L. Neumann, Real-time scene text localization and recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3538–3545.

[20] Y.F. Pan, X. Hou, C.L. Liu, A hybrid approach to detect and localize texts in natural scene images, in: IEEE Transactions on Image Processing, Mar, 2011, pp. 800–813.

[21] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. nature, 521(7553), 436.

[22] Ioffe, S., and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.

[23] J. Patterson, A. Gibson, Deep Learning: A Practitioner's Approach, O'Reilly Media, Inc, 2017.

[24] L. Kang, P. Ye, Y. Li, D. Doermann, Convolutional neural networks for no-reference image quality assessment, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 1733-1740.

[25] S. Hochreiter, J. Schmidhuber, Long short-term memory, in: Neural computation 9.8 , 1997, pp. 1735-1780.

[26] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. arXiv:1512.03385 [cs]. http://arxiv.org/abs/1512.03385