



IMG & PDF OCR WEBUI

技術手冊

中央大學 MIAT 實驗室

鄧祺文

目錄

- 系統架構設計
- 辨識模型
- 專案代碼（後端）
- 專案代碼（前端）
- 網頁功能入口
- 網頁結果輸出

系統設計架構

- ✧ 功能需求：讀取 IMG 和 PDF 文件，將其中的內容透過光學字元辨識（OCR, Optical Character Recognition）進行擷取，最後輸出成可運用的文字檔案格式。
- ✧ 開發環境：函式 Python（IMG & PDF 之前處理、OCR 模型相關功能調用），後端 Flask 和 JavaScript（功能於網頁上的串接、跳轉，以及動態部件控制），前端 HTML（網頁實際展示 UI）
- ✧ 專案代碼開源地址：
<https://github.com/toby0622/IMG-Optical-Character-Recognition-Tool>
- ✧ Docker 專案自動化部屬：
<https://github.com/toby0622/IMG-Optical-Character-Recognition-Tool-Docker>

```
# main environment deploy
FROM nvidia/cuda:11.7.1-cudnn8-devel-ubuntu20.04

ARG DEBIAN_FRONTEND=noninteractive

# set the working directory to /app
WORKDIR /app

# copy the current directory contents into the container at /app
COPY . /app

RUN apt-get update && apt-get install --fix-missing -y python3.9 python3.9-dev python3.9-venv python3-pip python3-wheel build-essential libgl1 ffmpeg libsm6 libxext6 wget git

RUN wget http://nz2.archive.ubuntu.com/ubuntu/pool/main/o/openssl/libssl1.1_1.1.1f-1ubuntu2.19_amd64.deb
RUN dpkg -i libssl1.1_1.1.1f-1ubuntu2.19_amd64.deb

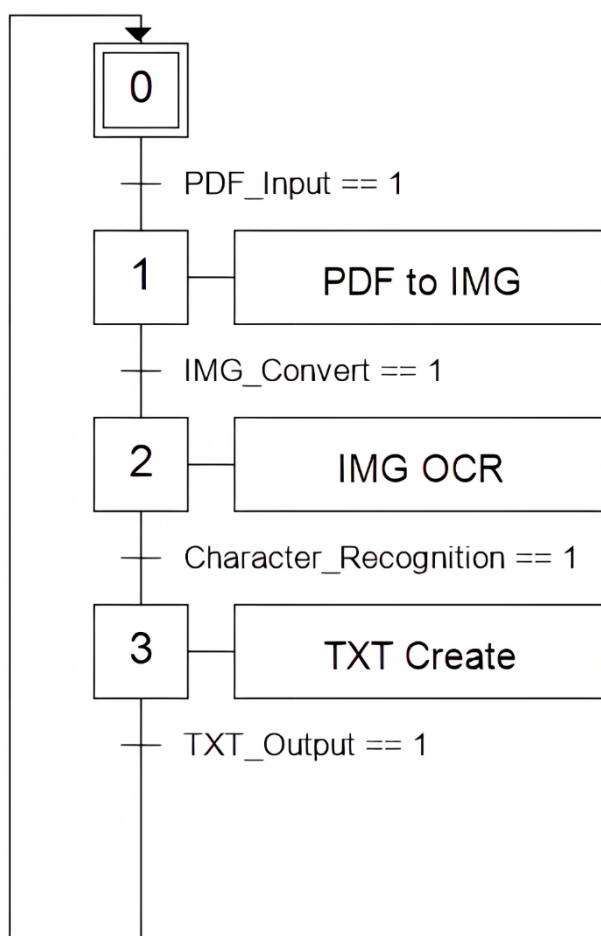
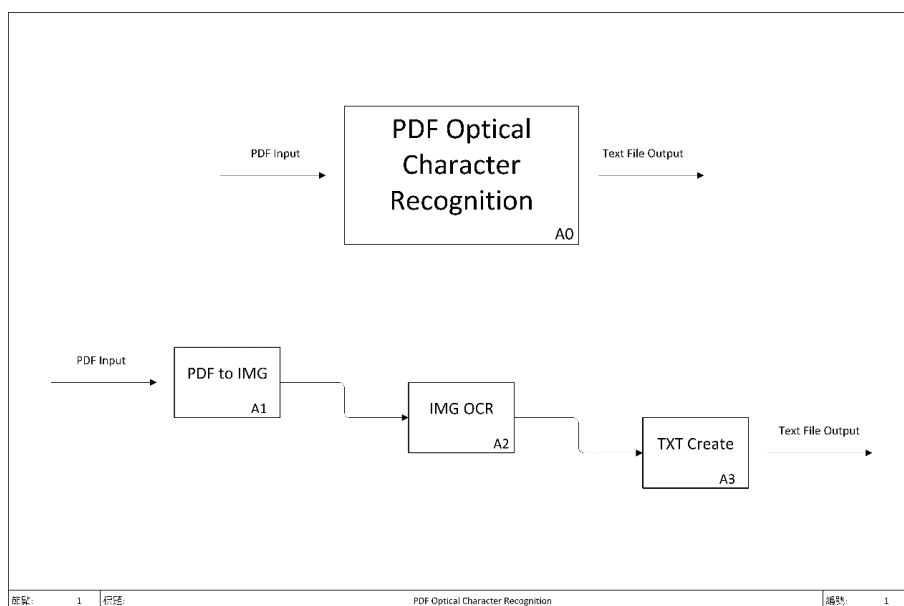
# install any needed packages specified in requirements.txt
# RUN pip install --trusted-host pypi.python.org -r requirements.txt

RUN pip install flask
RUN pip install opencv
RUN pip install paddlepaddle-gpu==2.5.1
RUN pip install paddleocr

# make port 9487 available to the world outside this container
EXPOSE 9487

# run app.py when the container launches
CMD ["python3", "app.py"]
```

✧ IDEF0 和 Grafcet



辨識模型

✧ 論文主題：

SVTR: Scene Text Recognition with a Single Visual Model

✧ 論文地址：

<https://arxiv.org/abs/2205.00159>

✧ 模型代碼開源地址：

<https://github.com/PaddlePaddle/PaddleOCR>

✧ 技術簡介：

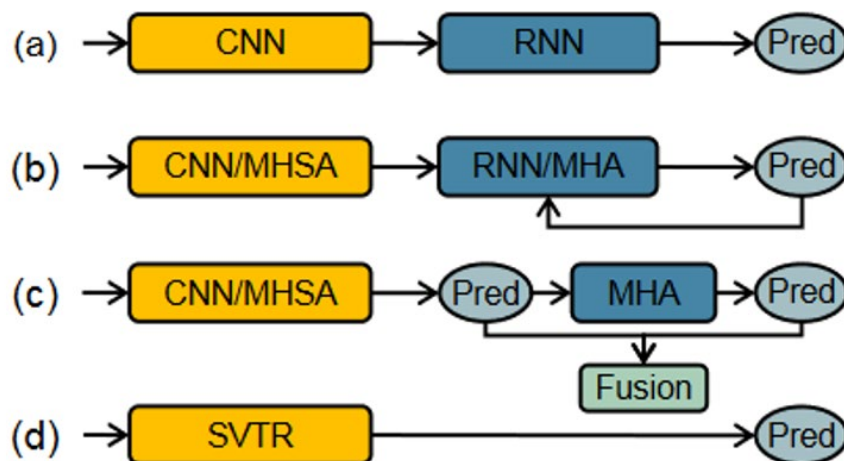


Figure 1: (a) CNN-RNN based models. (b) Encoder-Decoder models. MHSA and MHA denote multi-head self-attention and multi-head attention, respectively. (c) Vision-Language models. (d) Our SVTR, which recognizes scene text with a single visual model and enjoys efficient, accurate and cross-lingual versatile.

場景文字識別可以看作是一個從圖像映射到序列的任務。大多數的識別算法通常由兩個模塊構成，分別包含用於特征提取的視覺模塊，以及用於文本輸出的序列模塊。比如早期基於 CNN-RNN 的 CRNN，和現在一些基於注意力機制，進行自回歸式解碼

的算法，如上圖（圖一）所示。但是這樣設計出之雙階段算法的推理速度往往較慢，難以滿足工業應用的需求。因此該論文從推理速度和模型性能的雙重角度出發，提出了只由 Transformer 構成的純視覺模塊網絡 SVTR，在消費級顯示卡上達到了毫秒級的推理速度，並且參數量僅有 6 M。

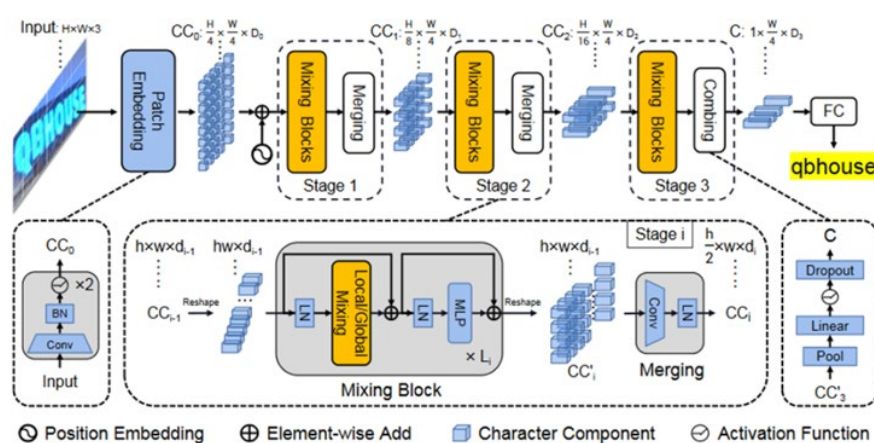


Figure 2: Overall architecture of the proposed SVTR. It is a three-stage height progressively decreased network. In each stage, a series of mixing blocks are carried out and followed by a merging or combining operation. At last, the recognition is conducted by a linear prediction.

上圖（圖二）為該論文所提出模塊網路 SVTR 的整體結構，採用類似於 SwinTransformer 的視覺模型和一個全連接層以及 CTC 解碼器進行文本序列預測。

首先和 ViT 類似，將輸入尺寸為 $H \times W \times 3$ 圖像按照 Patch 進行劃分，得到 $\frac{H}{4} \times \frac{W}{4} \times D_0$ Embeddings。本文采用的 Patch Embedding 操作和 ViT 中的有些許差異，其由兩層步距為 2，卷積核大小為卷積層 3×3 ，以及 BN 層構成。這樣不同的 Patch 之

間是存在著重疊的，如下圖（圖三）所示。經過 Patch Embedding 後的序列將經過一系列的 Stage，每一個 Stage 都由一系列的 Mixing Block 和 Merging Layer 構成。

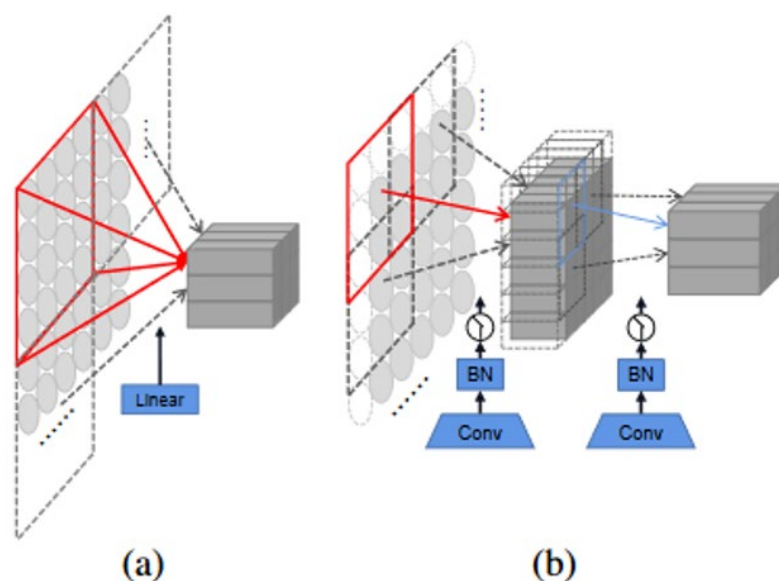


Figure 3: (a) The linear projection in ViT[Dosovitskiy *et al.*, 2021]. (b) Our progressive overlapping patch embedding.

作者認為文本識別需要兩種特征。第一種是局部特征，如筆畫特征。它編碼了字符的不同部分之間的形態特征和相關性。第二種是字符間的依賴性，如不同字符之間或文字與非文字成分之間的相關性。因此，作者設計了兩個混合模塊，即 Global Mixing 和 Local Mixing，通過使用不同大小感受的自注意層來實現。如下圖所示。Global Mixing 層本質上就是一個 Transformer Block，由一個多頭自注意層，一個 Layer Norm 層，以及一個 MLP 層構

成。通過自注意力機制的全局建模特性來進行全局字符建模。

Local Mixing 則是採用了帶窗的自注意層，窗大小設置為了 7×11 。

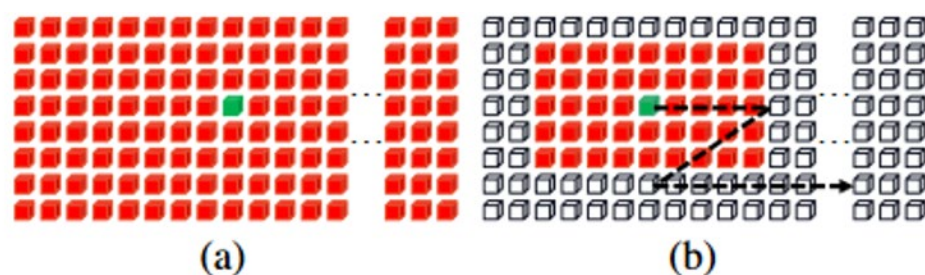


Figure 4: Illustration of (a) global mixing and (b) local mixing.


Merging 層扮演著將輸入序列進行下采樣的角色。其由高度方向步距為 2，寬度方向步距為 1，卷積核大小為 3×3 的卷積層構成。將輸入序列的尺寸由 $h \times w \times d_{i-1}$ 縮小為 $\frac{h}{2} \times w \times d_{i-1}$ 。同時每經過一次 Merging 層，序列的 Channel 維度也會增大，從而彌補在高度上的信息損失。

專案代碼 (後端)

✧ app.py 關鍵部分

下圖一為 IMG 上傳之 Flask Route，一開始是各項變數的宣告和用以進行檔案上傳的 Web Request。得到檔案之後先對副檔名進行檢查，確認無誤後即可進行 OCR 之操作並對輸出進行調整，最後打包成所需之檔案格式（PDF 檔案同理，只是多做轉檔相關前處理）。

下圖二則是下載功能的實現，通過使用 Flask 所提供之 `send_file` 功能，即可於網頁中將前述所提及之打包檔案進行下載。



```
@app.route('/downloadtxt')
def download_txt():
    return send_file(
        'download/output.txt',
        mimetype='text/plain',
        download_name='result.txt',
        as_attachment=True)

@app.route('/downloadjson')
def download_json():
    return send_file(
        'download/output.json',
        mimetype='application/json',
        download_name='result.json',
        as_attachment=True)
```

✧ ocr.py 關鍵部分

下圖三為 OCR 實際進行辨識的過程，一開始先載入 OCR 辨識所需的 SVTR 模型（於 GitHub 上開源為 PaddleOCR）。接下來針對上傳或經過轉換的 PDF 圖檔使用 OpenCV 進行影像前處理來提升識別率。最後再將前處理完成之圖片送至辨識模型，即可得到文字輸出結果。

為了方便進行確認辨識情況，還額外進行了辨識視覺化的展示，實際產出的圖像可於本技術手冊之「網頁結果輸出」章節進行查看，其中將會包含輸入的文件圖片、模型判定出之辨識框，以及實際文字的輸出結果和該段文字辨識的置信率。

```
def image_ocr_match(image_path, counter_number):
    process_start = datetime.datetime.now() # process starting time

    cc = OpenCC('s2twp')

    ocr_model = PaddleOCR(use_angle_cls=True, lang="ch",
                           use_gpu=True, enable_mkldnn=True,
                           ocr_version="PP-OCRv4",
                           det_model_dir="models/det",
                           cls_model_dir="models/cls",
                           rec_model_dir="models/rec")

    image = cv2.imread(image_path, cv2.IMREAD_COLOR)
    gray_image = cv2.cvtColor(image, 7)
    inverted_image = cv2.threshold(gray_image, 95, 255, cv2.THRESH_BINARY_INV)[1]

    kernel = np.ones((2, 2), np.uint8)

    dilation_image = cv2.dilate(inverted_image, kernel, iterations=1)

    recognition_result = ocr_model.ocr(dilation_image)

    data = recognition_result[0]

    # result static
    visual = Image.open(image_path).convert('RGB')
    rec_boxes = [line[0] for line in data]
    rec_texts = [cc.convert(str(line[1][0])) for line in data]
    probability = [line[1][1] for line in data]
    im_show = draw_ocr(visual, rec_boxes, rec_texts, probability, font_path='font/Yozai-Regular.ttf')
    im_show = Image.fromarray(im_show)
    im_show.save('static/result' + str(counter_number) + '.jpg', quality=100)

    process_finish = datetime.datetime.now() # process finishing time

    print('Page ' + str(counter_number) + ' OCR Process Time =', (process_finish -
        process_start).seconds)

    return data
```

✧ export.py 關鍵部分

下圖四包含兩個函式，用以實際產出 app.py 中敘述之打包文件。本系統選擇兩個較為常見的檔案格式，一為 TXT，二為 JSON。作為一個目標為輸出文字的工具，TXT 可以十分直觀的對輸出文字進行複製或擷取；而 JSON 作為 Web API 之間傳遞最為通用的格式，預留

作未來的功能串接（如將 OCR 辨識出的文字通過其他 AI 進行潤色等），保留系統功能擴充之彈性。

```
def txt_export_web(datafile):
    # process_start = datetime.datetime.now() # process starting time

    storage_path = "download/output.txt"

    text_file = open(storage_path, 'w', encoding='UTF-8')

    # with open(output_folder + "SunHan" + ".txt", 'w', encoding='UTF-8') as textfile:
    #     textfile.write(str(datafile))

    text_file.write(datafile)
    text_file.close()

    # process_finish = datetime.datetime.now() # process finishing time

def json_export_web(datafile):
    # process_start = datetime.datetime.now() # process starting time

    storage_path = "download/output.json"

    text_file = open(storage_path, 'w', encoding='UTF-8')

    # with open(output_folder + "SunHan" + ".txt", 'w', encoding='UTF-8') as textfile:
    #     textfile.write(str(datafile))

    text_file.write(datafile)
    text_file.close()

    # process_finish = datetime.datetime.now() # process finishing time
```

✧ function.py 關鍵部分

下圖五乃針對特殊符號進行去除，用以減少檔案格式相關錯誤。

```
def remove_special_characters(text):
    # punctuation marks
    text = re.sub('[\uFF5E\uFF03-\uFF06\uFF08\uFF09\uFF1C-\uFF1E]', r'', text)

    text = re.sub('[~#%&()<=>\"]', r'', text)

    return text
```

專案代碼（前端）

✧ index.html 關鍵部分

```
<div>
  <form class="form-inline pt-3 text-center" action="uploadimg" method="post" enctype="multipart/form-
data">
    <input class="form-control" type="file" multiple="" name="file1[]">
    <br/>
    <input class="form-control btn btn-primary" type="submit" value="IMG Upload">
  </form>
</div>

<div class="pt-3"></div>

<div>
  <form class="form-inline pt-3 text-center" action="uploadpdf" method="post" enctype="multipart/form-
data">
    <input class="form-control" type="file" name="file2[]">
    <br/>
    <input class="form-control btn btn-primary" type="submit" value="PDF Upload">
  </form>
</div>
```

上傳功能介面

✧ result.html 關鍵部分

```
<div id="resultcarousel" class="carousel slide" data-bs-ride="carousel">
  <p class="text-center pt-3">Optical Character Recognition Visualization(s)</p>

  <div class="carousel-inner" style="text-align: center">
    {% for i in range(1, carousel_index + 1) %}
    <div class="carousel-item {% if i == 1 %} active {% endif %}">
      <a data-fancybox="gallery" href="{{url_for('static', filename='result' + i|string +
'.jpg')}}">
        
      </a>
    </div>
    {% endfor %}
  </div>

  <button class="carousel-control-prev" type="button" data-bs-target="#resultcarousel" data-bs-
slide="prev"
    style="background: black; border-radius: 50%; opacity: 0.3; height: 50px; width: 50px; top:
50%">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Previous</span>
  </button>

  <button class="carousel-control-next" type="button" data-bs-target="#resultcarousel" data-bs-
slide="next"
    style="background: black; border-radius: 50%; opacity: 0.3; height: 50px; width: 50px; top:
50%">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Next</span>
  </button>
</div>
```

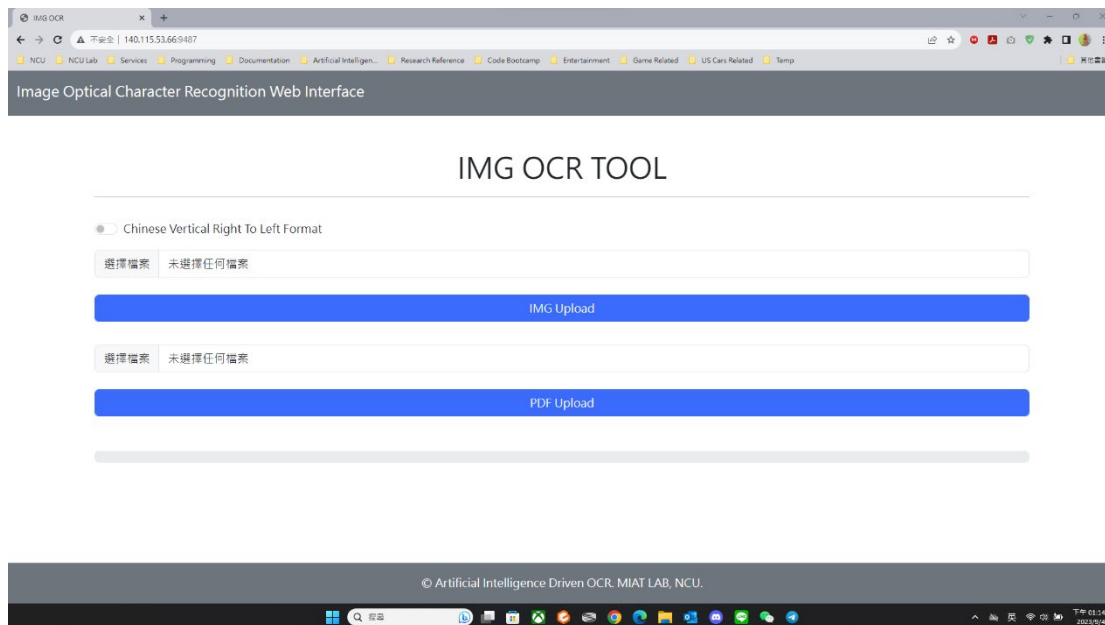
辨識結果視覺化燈箱

```
<div class="text-center pt-3 pb-3">
  <a href="{{url_for('download_txt')}}" type="button" class="btn btn-success text-center">Download
  TXT</a>
  <a href="{{url_for('download_json')}}" type="button" class="btn btn-primary text-center">Download
  JSON</a>
  <a href="{{url_for('index')}}" type="button" class="btn btn-danger text-center">New Process</a>
</div>
```

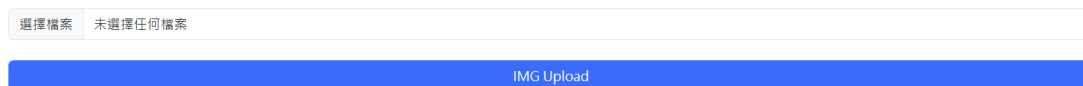
各項功能跳轉

網頁功能入口

✧ 功能入口



✧ 上半部選擇框：支持 JPG、JPEG、PNG



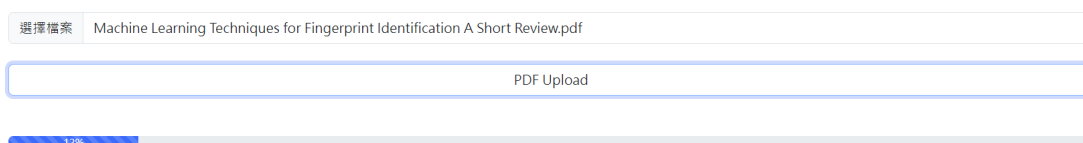
✧ 下半部選擇框：支持 PDF



✧ 圖像特殊排序：中文直式由右至左（預設標準為由上至下）

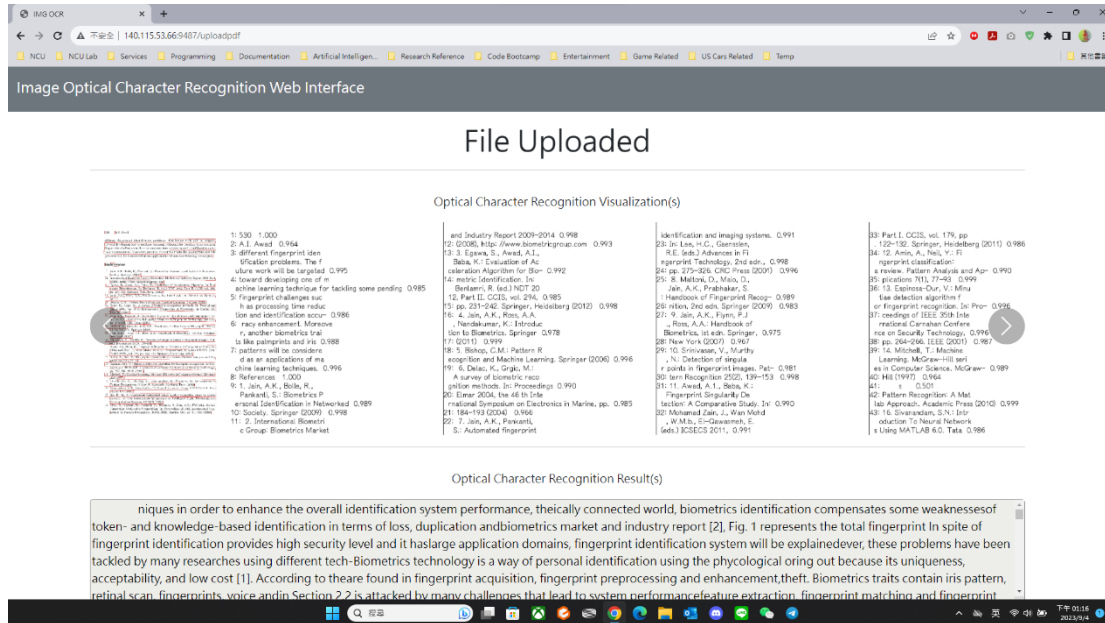


✧ 成功讀取資料後，可於下方看見處理進度條

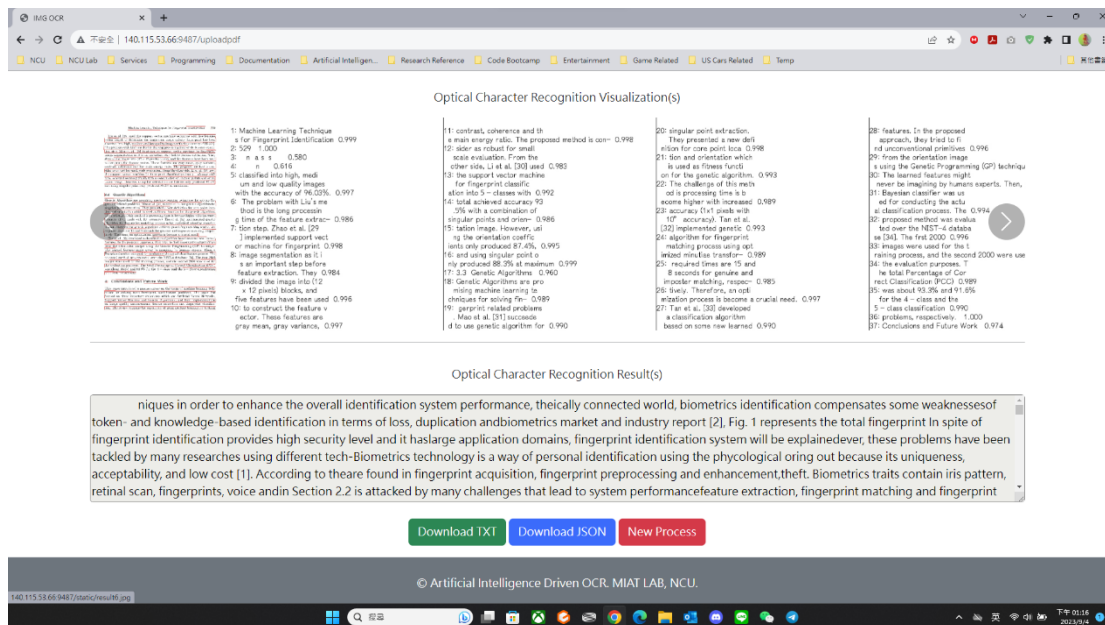


網頁結果輸出

◇ 結果輸出頁面：上半部



◇ 結果輸出頁面：下半部



◇ 輸出視覺化：包含辨識框及輸出結果展示，左右按鍵可變換當前展示頁面

Optical Character Recognition Visualization(s)

1: 530 1.000	and Industry Report 2009-2014. 0.998	identification and imaging systems. 0.991	33: Part I, CCIS, vol. 179, pp.
2: A.I. Awad, 0.964	12: 2008, http://www.biometricsgroup.com . 0.993	23: In: Lee, K.C., Ganesan,	122-132. Springer, Heidelberg (2011) 0.986
3: different fingerprint iden-	13: 3. Egeci, S., Awad, A.I.,	R.E. (eds.) Advances in Fi-	34: 12, Amin, A., Neil, Y.: Fi-
ification problem. The f	Baba, K.: Evaluation of Ac-	ngerprint Technology, 2nd edn., 0.998	ngerprint classification:
ature work will be targeted. 0.995	eration Algorithm for Bio-	84: pp. 275-326. CRC Press (2001) 0.996	a review. Pattern Analysis and App-
5: fingerprint challenges suc-	14: metric identification. In:	25: 8. Meloni, D., Mao, D.,	35: pications 7(1), 77-93 0.999
cessing time reduc-	Bertalmio, R. (ed.) NDT 20	Jain, A.K., Prabhakar, S.	36: 13. Espinosa-Dur, V.: Minu
tion and identification accu-	12, Part II, CCIS, vol. 234, 0.985	Handbook of Fingerprint Recog-	face detection algorithm f
6: race enhancement, Moreno	15: pp. 231-242. Springer, Heidelberg (2012) 0.988	26: nition, 2nd edn. Springer (2009) 0.983	or fingerprint recognition. In: Pro-
7: patterns will be consid-	16: 4. Jain, A.K., Ross, A.A.	27: 9. Jain, A.K., Flynn, P.J.	ceedings of IEEE 35th Inte-
ered as an applications of ma-	Handaunur, K.: Introduc-	10, A.A.: Handbook of	national Canadian Confere-
chine learning techniques. 0.996	tion to Biometrics, Springer 0.978	Biometrics, 1st edn. Springer, 0.975	ence on Security Technology, 0.996
8: References 1.000	17: 2010. 0.999	28: New York (2007) 0.967	36: pp. 264-286. IEEE (2001) 0.987
9: 1. Jain, A.K., Bolla, R.,	18: 5. Bishop, C.M.: Pattern R-	29: 10. Srinivasan, V., Murthy	39: 14. Mitchell, T.: Machine
Parkavi, S.: Biometrics P-	ecognition and Machine Learning. Springer (2006) 0.996	1, N.: Detection of singula-	Learning. McGraw-Hill seri-
ersonal Identification in Networked 0.989	19: 6. Delac, K., Grig, M.	20: r points in fingerprint images. Pat-	es in Computer Science, McGraw- 0.989
11: 2. International Biometr-	A survey of biometric reco-	30: tern Recognition (202), 139-153 0.998	40: H (1997) 0.964
c Group Biometrics Market	gnition methods. In: Proceedings. 0.990	31: 11. Awad, A.I., Baba, K.: Fingerprint Singularity De-	41: s 0.501
	20: Eswar (2004), the 46th In-	tection: A Comparative Study. In: 0.990	42: Pattern Recognition: A Mat-
	ternational Symposium on Electronics in Marine, pp. 0.985	32: Mohamed Zain, J., Wan Mohd	lab Approach. Academic Press (2010) 0.999
	21: 184-192 (2004) 0.966	W.M.B., El-Dawamneh, E. (eds.) ICSECS 2011, 0.991	43: 16. Svanenstam, S.N.: Intr-
	22: 7. Jain, A.K., Parkavi, S.: Automated fingerprint		duction To Neural Network
			s Using MATLAB 6.0. Tata 0.986

✧ 直接點擊圖片進入相簿模式，可以對圖片進行放大

4/8 Page Optical Character Recognition Web Interface

File Uploaded

Optical Character Recognition Visualization(s)

<p>1: Machine Learning Techniques for Fingerprint Identification. 0.987</p> <p>2: 227 0.999</p> <p>3: Feature. 0.999</p> <p>4: Fingerprint. 0.999</p> <p>5: Predefined. 0.904</p> <p>6: Classification. 0.993</p> <p>7: Acquisition. 0.995</p> <p>8: Extraction. 1.000</p> <p>9: Indano. 0.784</p> <p>10: Matching. 0.999</p> <p>11: Database. 1.000</p> <p>12: Identification. 0.994</p> <p>13: Fingerprint. 0.994</p> <p>14: Features. 0.999</p> <p>15: PreProc. 0.992</p> <p>16: Classification. 0.990</p> <p>17: Acquisition. 0.979</p> <p>18: Extraction. 0.992</p> <p>19: Fig. 2. A flowchart of fingerprint identification system (basic component). 0.977</p> <p>20: Training techniques can be good contributors for enhancing the system performance. 0.986</p> <p>21: minuscule as reported in Section 3. 0.993</p> <p>22: Machine Learning Techniques. 0.994</p> <p>23: Machine learning systems are concerned with building flexible algorithms or techniques. 0.998</p> <p>24: ing [14]. Machine learning system is first trained with source data, and following, 0.980</p> <p>25: it is used to perform required operations according to its acquired experience. 0.993</p> <p>26: The problem of machine learning techniques is related to their sensitivity to 0.992</p> <p>27: the training data and the training parameters as they may produce different. 0.994</p> <p>28: results by changing the training data. However machine learning includes many 0.989</p> <p>29: techniques such as Artificial Neural Networks, Support Vector Machine, Genetic 0.999</p> <p>30: Algorithms, Bayesian Train</p>	<p>ing and Probabilistic Models [15], we will stress only 0.988</p> <p>31: on the implementation of the first three techniques on fingerprint identification. 0.977</p> <p>32: 3.1 Artificial Neural Networks 0.973</p> <p>33: Artificial Neural Network is the most widely used algorithm of the machine. 0.987</p> <p>34: learning system [16]. The quality assurance of the acquired fingerprint image is 0.985</p> <p>35: an important process before the feature extraction. Xie and G [17] designed a 0.987</p> <p>36: supervised back propagation neural network that uses the gray scale fingerprint. 0.994</p> <p>37: image for continuous image quality estimation. The problem of this method 0.991</p> <p>38: is the lack of evaluation as it has been evaluated for small fingerprint images. 0.991</p> <p>39: from Fingerprint Verifica-</p>	<p>on Completion 2002 (VC2 032) [18]. Moreover, the 0.985</p> <p>40: fingerprint image needs to be divided into blocks which is computationally expensive. 0.993</p> <p>41: genuine process before running the proposed method. Zhu et al. [19] used the 0.995</p> <p>42: edge orientation. The correct ridge orientation is estimated using the trained 0.992</p> <p>43: neural network. Labati et al. [20] proposed the usage of neural network for in- 0.985</p> <p>44: age quality measurement in contactless fingerprint acquisition. They discovered 0.994</p>
---	--	--

Optical Character Recognition Results(s)

techniques in order to enhance the overall identification system performance. In a connected world, biometrics identification compensates some weaknesses of token- and knowledge-based identification in terms of loss, duplication and biometrics market and industry report [2]. Fig. 1 represents the total fingerprint identification provides high security level and it has large application domains. fingerprint identification system will be explained. These problems have been tackled by many researchers using different techniques. Biometrics technology is a way of personal identification using the physiological or behavioral uniqueness, acceptability, and low cost [1]. According to the findings in fingerprint processing and enhancement techniques, Biometrics traits contain its pattern information. Fingerprints, as we explain in Section 2.2, are affected by system performance, feature extraction, fingerprint matching and fingerprint

1/8 Page Optical Character Recognition Web Interface

File Uploaded

Optical Character Recognition Visualization(s)

<p>1: Machine Learning Techniques for Fingerprint 0.999</p> <p>2: Identification: A Short Review 0.990</p> <p>3: Ali Ismail Awad, 1.2 0.952</p> <p>4: 1 Electrical Engineering Department, Faculty of Engineering 0.986</p> <p>5: Al Azhar University, Qena, Egypt 0.976</p> <p>6: 2 Member of Scientific Research Group in Egypt (SRGE) 0.987</p> <p>7: aawad@ieee.org 0.979</p> <p>8: Abstract. Fingerprint is considered as a dominant biometric trait due 0.984</p> <p>9: to its acceptability, reliability, high security level and low cost. Due 0.995</p> <p>10: to the high demand on fingerprint identification system deployments, 0.991</p> <p>11: a lot of challenges are kept arising in each system's phase including fin- 0.987</p> <p>12: gerprint image enhancement, feature extraction, features matching and 0.991</p> <p>13: fingerprint classification. Machine learning techniques introduce non tra- 0.995</p> <p>14: ditional solutions to the fingerprint identification challenges. This paper 0.985</p>	<p>15: presents a short survey that emphasizes the implementations of basic machine learning 0.984</p> <p>16: survey contributes as a ground truth for developing machine learning 0.996</p> <p>17: based algorithms for fingerprint identification in the near future. 0.989</p> <p>18: Keywords: Biometrics, Fingerprints, Machine Learning Techniques. 0.993</p> <p>19: Introduction 0.999</p> <p>20: Biometrics technology is a way of personal identification using the physiological or 0.976</p> <p>21: the behavioural characteristics. Driven from the security needs for the electronic- 0.997</p> <p>22: ically connected world, biometrics identification compensates some weaknesses 0.992</p> <p>23: of token- and knowledge-based identification in terms of loss, duplication and 0.996</p> <p>24: theft. Biometrics traits contain iris pattern, retina scan, fingerprints, voice and 0.996</p>	<p>25: signature. Fingerprint is one of the dominant biometrics traits that keeps spread 0.984</p> <p>26: out because of its uniqueness, acceptability, and low cost [1]. According to the 0.990</p> <p>27: biometrics market and industry report [2], Fig. 1 represents the total fingerprint 0.986</p> <p>28: revenue which is around 66 % compared to the other biometrics 0.985</p> <p>29: In spite of fingerprint identification provides high security level and it has 0.999</p> <p>30: large application domains, fingerprint identification system (will be explained) 0.995</p> <p>31: in Section 2.2) is attacked by many challenges that lead to system performance 1.0</p> <p>32: are found in fingerprint acquisition, fingerprint preprocessing and enhancement, 1.0</p>
--	--	---

Optical Character Recognition Results(s)

According to the findings in fingerprint processing and enhancement techniques, Biometrics traits contain its pattern information. Fingerprints, as we explain in Section 2.2, are affected by system performance, feature extraction, fingerprint matching and fingerprint

✧ 完整文字輸出：包含所有頁面辨識出之完整文章

Optical Character Recognition Result(s)

niques in order to enhance the overall identification system performance, theically connected world, biometrics identification compensates some weaknesses of token- and knowledge-based identification in terms of loss, duplication and biometrics market and industry report [2], Fig. 1 represents the total fingerprint. In spite of fingerprint identification provides high security level and it has large application domains, fingerprint identification system will be explained ever, these problems have been tackled by many researches using different tech- Biometrics technology is a way of personal identification using the phycological oring out because its uniqueness, acceptability, and low cost [1]. According to theare found in fingerprint acquisition, fingerprint preprocessing and enhancement, theft. Biometrics traits contain iris pattern, retinal scan, fingerprints, voice and in Section 2.2 is attacked by many challenges that lead to system performance feature extraction, fingerprint matching and fingerprint

✧ 文章打包功能：包含 TXT 文字檔下載、API 串接用 JSON 檔案下載，以及重新開始新一輪識別

Download TXT

Download JSON

New Process