

# 利用 Robotic Process Automation + CRNN 進行 證碼辨識並自動登入流程

Dr. Ming-Chuan Chiu

Presenter: 馮品叡



# Outline



- 背景介紹
- 問題定義
- 模型建構
- 結論與未來展望

# Outline



- 背景介紹
- 問題定義
- 模型建構
- 結論與未來展望

## 背景介紹



PwC 2018年報告指出，RPA指的是針對那些**花人力時間處理、跨系統作業、重複性的工作**，運用機器流程自動化來解決這些問題

The text '24/7' is rendered in large, blue, 3D block letters, indicating 24 hours a day, 7 days a week availability.

RPA具有以下優勢：**降低成本提升品質、非侵入式軟件(可與舊系統相容)、全天性工作、穩定度高、效率好等優勢**



組織將**機器人技術**和先進的數字技術結合在一起，平衡了人**和機器人的勞動力**，**並實現了智能自動化**。

## 背景介紹



在Grand View Research進行的一項研究中“RPA的市場目前價值11億美元，預計從2020年到2027年的複合年增長率為**33.6%**。”



2025年，自動化市場將被易於部署的RPA模型所充斥，這些RPA模型具有預先構建的功能，可在企業中實現一系列文書工作的自動化。



Forrester Institute估計到2021年底，將有**400萬台機器人**在生產(2017)

# Outline



- 背景介紹
- 問題定義
- 模型建構
- 結論與未來展望

# 問題定義

## **WHAT**

RPA為代替人工執行的**重複性和無聊**的文書工作

## **WHO**

所有使用者

## **WHEN**

基於規則的流程  
成熟穩定的流程

## **WHERE**

進行重複性操作、檢驗手續等工作，皆可使用RPA來代替員工

## **WHY**

解放人力，消除人為錯誤，降低成本提升品質。

## **HOW**

建立明確的流程架構

## 問題定義

研究目的：

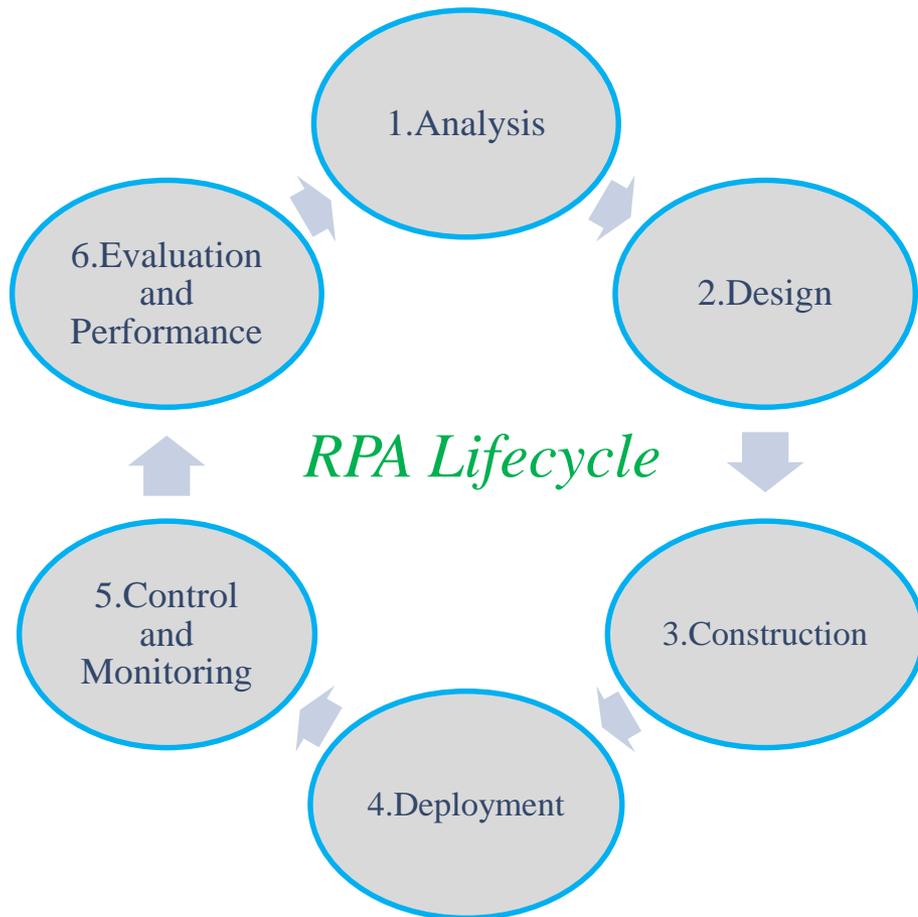
- 過去RPA專門處理rule-based的流程問題，此CASE由於驗證碼的動態變動特徵，導致RPA基於規則的流程遭到破壞。
- 本研究融合RPA+AI以期解決此一動態問題。

# Outline



- 背景介紹
- 問題定義
- 模型建構
- 結論與未來展望

# 模型建構



Step1 考慮並分析自動登入流程可行性及特性

Step2 詳細說明RPA流程中必須執行的操作動作和執行活動等。

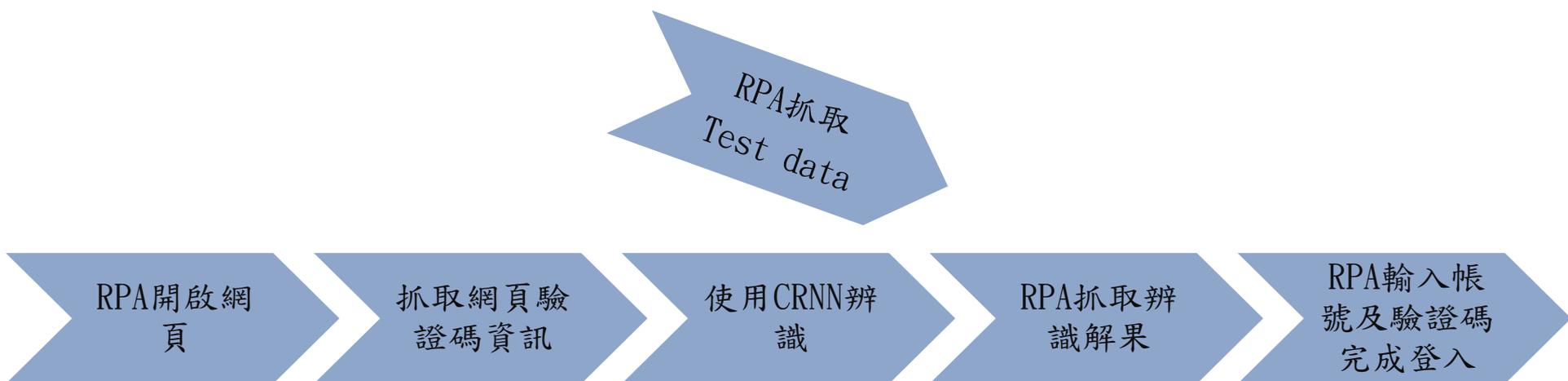
Step3 建構流程構成要素

Step4 給robot一個環境執行工作

Step5 控制和監視每個機器人的性能。

Step6 評估機器人的表現

# 模型建構



# 模型建構

RPA開啟網頁

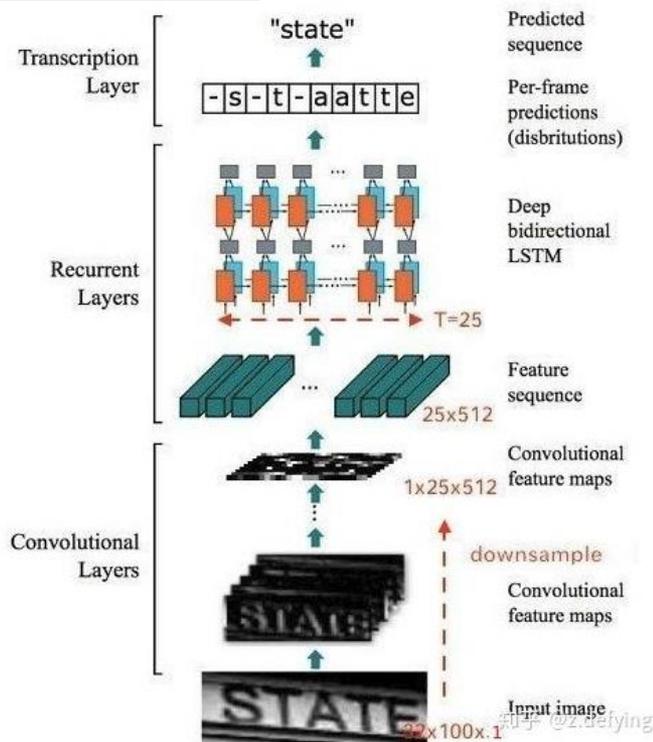
抓取網頁驗證碼資訊

The screenshot displays an RPA workflow interface with the following steps:

- Browser URL:** A text box containing the URL "http://insdb.tii.org.tw/pivot/".
- Do:** A container block containing:
  - Take Screenshot 'IMG':** A sub-step showing a screenshot of a login page. The CAPTCHA code "3165" is highlighted with a green box. Below the screenshot, the "Output to:" options are set to "Image". A "Saved image" field contains the value "VAL".
  - Save Image:** A sub-step that takes the "VAL" from the previous step and saves it to a file named "驗證碼.png".

# 模型建構

## 使用CRNN辨識



图来自文章：一文读懂CRNN+CTC文字识别

<https://zhuanlan.zhihu.com/p/71506131>

Model: "CRNN\_Model\_with\_CTC\_LOSS"

Layer (type)	Output Shape	Param #	Connected to
Input (InputLayer)	[(None, 104, 32, 1)]	0	
Conv_1 (Conv2D)	(None, 104, 32, 64)	640	Input[0][0]
MaxPooling_1 (MaxPooling2D)	(None, 52, 16, 64)	0	Conv_1[0][0]
Conv_2 (Conv2D)	(None, 52, 16, 128)	73856	MaxPooling_1[0][0]
MaxPooling_2 (MaxPooling2D)	(None, 26, 8, 128)	0	Conv_2[0][0]
Conv_3 (Conv2D)	(None, 26, 8, 256)	295168	MaxPooling_2[0][0]
BN_1 (BatchNormalization)	(None, 26, 8, 256)	1024	Conv_3[0][0]
Conv_4 (Conv2D)	(None, 26, 8, 512)	1180160	BN_1[0][0]
MaxPooling_3 (MaxPooling2D)	(None, 13, 4, 512)	0	Conv_4[0][0]
Conv_5 (Conv2D)	(None, 12, 3, 512)	1049088	MaxPooling_3[0][0]
Reshape (Reshape)	(None, 12, 1536)	0	Conv_5[0][0]
Dense (Dense)	(None, 12, 64)	98368	Reshape[0][0]
Dropout (Dropout)	(None, 12, 64)	0	Dense[0][0]
LSTM_1 (Bidirectional)	(None, 12, 512)	657408	Dropout[0][0]
LSTM_2 (Bidirectional)	(None, 12, 512)	1574912	LSTM_1[0][0]
Label (InputLayer)	[(None, None)]	0	
Softmax (Dense)	(None, 12, 11)	5643	LSTM_2[0][0]
CTC_Loss (CTCLayer)	(None, 12, 11)	0	Label[0][0] Softmax[0][0]

Total params: 4,936,267  
Trainable params: 4,935,755  
Non-trainable params: 512

# 模型建構

資料來源:GitHub

[https://github.com/YenLinWu/RPA\\_UiPath/tree/master/OpenCV%20x%20Tesseract%20x%20UiPath](https://github.com/YenLinWu/RPA_UiPath/tree/master/OpenCV%20x%20Tesseract%20x%20UiPath)

## Training and Validation Sets 訓練集與驗證集

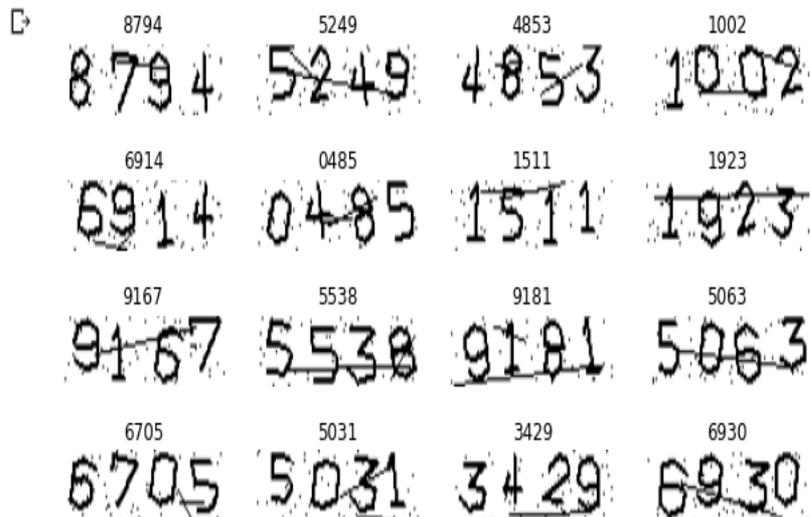
```

▶ # Splitting dataset into training and validation sets
def split_data( Images, Labels, train_size=0.9, shuffle=True ):
    # 1. Get the total amount of the dataset
    size = len( Images )
    # 2. Make an indices array and shuffle it, if required
    indices = np.arange( size )
    if shuffle:
        np.random.seed(42)
        np.random.shuffle( indices )
    # 3. Get the size of training samples
    train_samples = int( size*train_size )
    # 4. Split data into training and validation sets
    x_train, y_train = Images[indices[:train_samples]], Labels[indices[:train_samples]]
    x_valid, y_valid = Images[indices[train_samples:]], Labels[indices[train_samples:]]
    return x_train, x_valid, y_train, y_valid

x_train, x_valid, y_train, y_valid = split_data( np.array(Images), np.array(Labels) )

print( 'Number of training set:', len(x_train) )
print( 'Number of validation set:', len(x_valid) )
print(x_train)
print(y_train)

```



▶ Number of training set: 1620  
 Number of validation set: 180  
 [' /content/drive/My Drive/IIE FINAL/CaptchaImages/8794.png',  
 ' /content/drive/My Drive/IIE FINAL/CaptchaImages/5249.png',  
 ' /content/drive/My Drive/IIE FINAL/CaptchaImages/4853.png' ...  
 ' /content/drive/My Drive/IIE FINAL/CaptchaImages/3004.png',  
 ' /content/drive/My Drive/IIE FINAL/CaptchaImages/5782.png',  
 ' /content/drive/My Drive/IIE FINAL/CaptchaImages/3139.png']  
 ['8794' '5249' '4853' ... '3004' '5782' '3139']

Train data:1620  
 Validation data:180

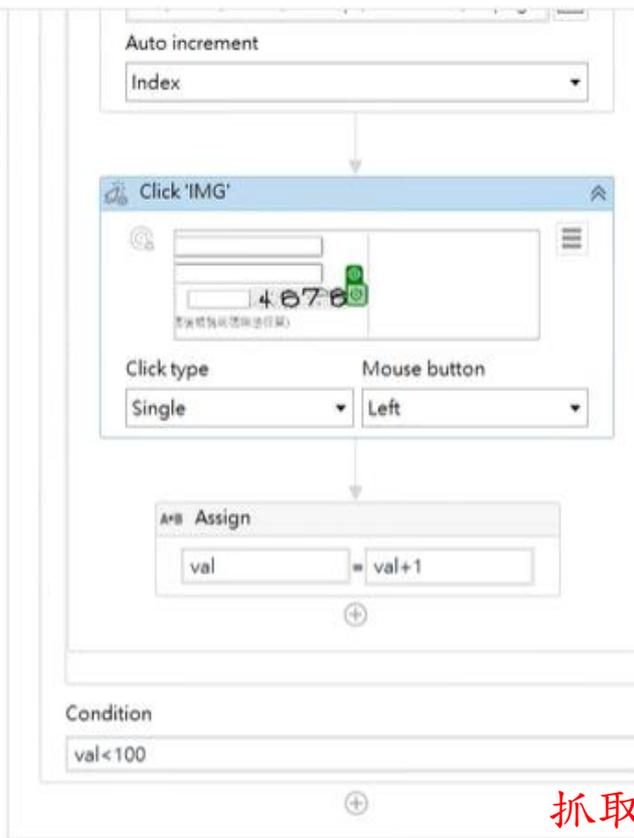
# 模型建構

RPA抓取  
Test data

The screenshot displays the RPA software interface. At the top, there is a 'Browser URL' field containing the text "http://insdb.tii.org.tw/pivot/". Below this, a 'Do' loop is visible, containing a 'Do While' loop. Inside the 'Do While' loop, there is a 'Sequence' block, which contains a 'Take Screenshot' action. The 'Take Screenshot' action has a preview window showing a screenshot of a webpage with a green box highlighting a specific element. Below the preview, the 'Output to' options are set to 'File'. The 'File name' field is set to "C:\Users\PinRui\Desktop\新增資料夾\val.png". The 'Auto increment' dropdown is set to 'Index'.

# 模型建構

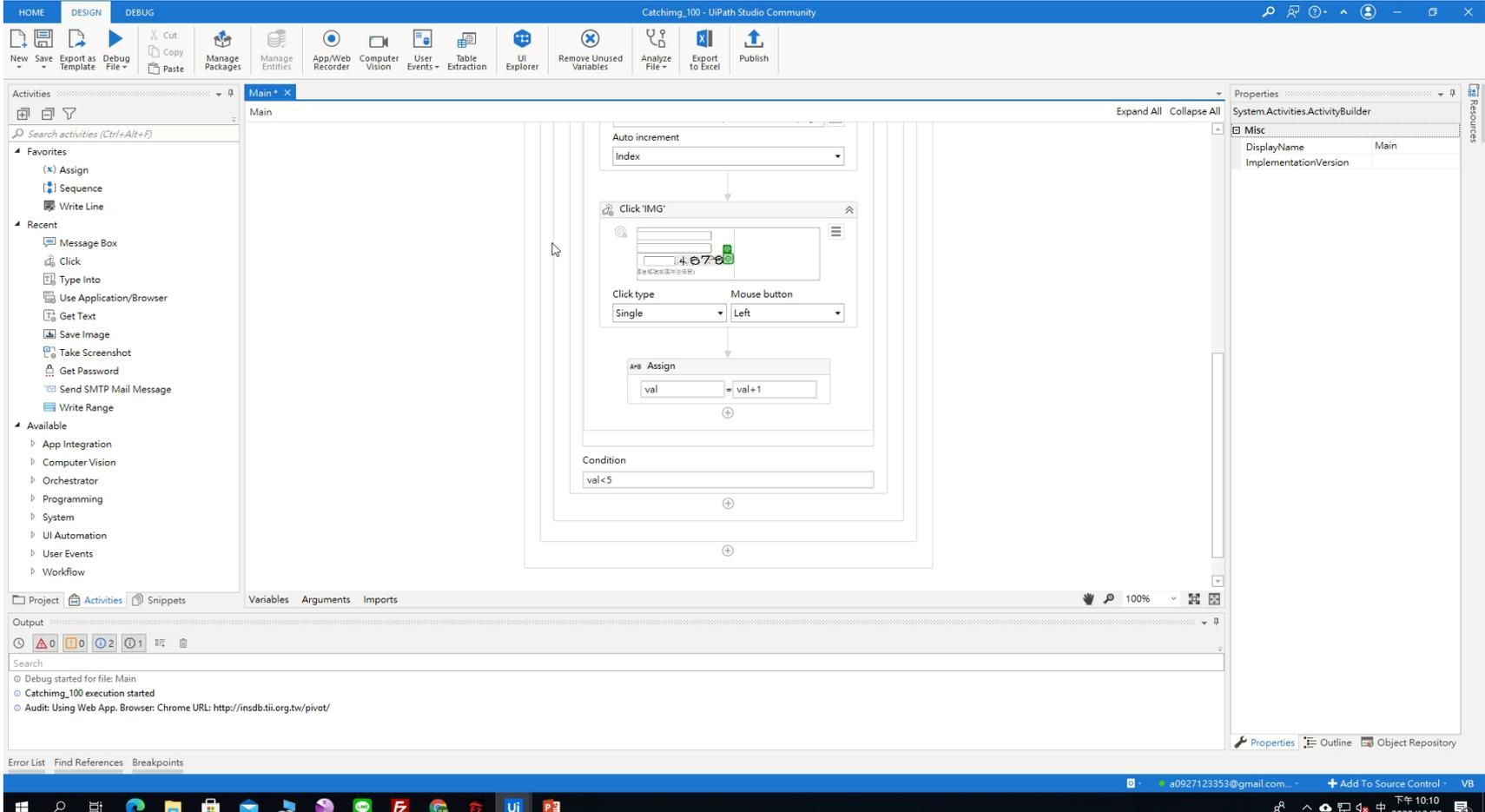
RPA抓取  
Test data



抓取100張驗證碼圖片只需3-4分鐘

Name	Variable type	Scope	Default
img	Image	Sequence	Enter a VB expression
val	Int32	Sequence	0

# 模型建構



The screenshot displays the UIPath Studio Community interface. The main workspace shows a workflow diagram for a 'Click' activity. The workflow includes the following steps:

- Auto increment**: An activity with a dropdown menu set to 'Index'.
- Click 'IMG'**: A 'Click' activity with a 'Click type' set to 'Single' and a 'Mouse button' set to 'Left'. The target is a UI element with a value of '4.678'.
- Assign**: An activity with the expression `val = val + 1`.
- Condition**: A condition activity with the expression `val < 5`.

The interface also shows a 'Properties' window on the right, a 'Recent' list on the left, and a 'Search' bar at the bottom. The status bar at the bottom indicates the current user is 'a0927123353@gmail.com' and the date is '2020/12/29'.

# 模型建構

0115

0015

0367

0367

1022

1022

1806

1806

2261

2261

3629

3629

4686

4686

4997

4997

5590

5590

6061

6061

6665

6665

7217

7217

7689

7689

8183

8183

8689

8689

9165

9165

0505

0505

0180

0180

0404

0404

1185

1185

1829

1829

2459

2459

3787

3787

4753

4753

5401

5401

5669

5669

6153

6153

6697

6697

7222

7222

7737

7737

8203

8203

8717

8717

9234

9234

0505

0505

0197

0197

0592

0592

1443

1443

1962

1962

2963

2963

3804

3804

4812

4812

5442

5442

5820

5820

6499

6499

6721

6721

7529

7529

8003

8003

8341

8341

8760

8760

9329

9329

0505

0505

0215

0215

0655

0655

1697

1697

2010

2010

3144

3144

3962

3962

4896

4896

5459

5459

5950

5950

6607

6607

6757

6757

7530

7530

8134

8134

8384

8384

8838

8838

9355

9355

0505

0505

0366

0366

0980

0980

1724

1724

2210

2210

3483

3483

4330

4330

4979

4979

5471

5471

6034

6034

6646

6646

6812

6812

7657

7657

8172

8172

8515

8515

8979

8979

9463

9463

0505

0505

# 模型建構

Optimizer	Accuracy	Epochs
Adam	96.78%	19 
Adagrad	95.57%	41
Nadam	95.56%	29
SGD	95.50%	11

準確率些微差異對此CASE結果影響不大

$$1-(1-0.9678)^5=0.9999999654$$

$$1-(1-0.9550)^5=0.9999998155$$

Note: 假設RPA中Do while重複登入5次

# 模型建構

RPA抓取辨  
識解果

The screenshot displays the configuration for a 'Use Application' task in an RPA tool. The main window is titled '填入資訊' (Enter Information). Inside, there is a sub-window 'Use Application: info - 記事本' (Use Application: info - Notepad). The configuration includes:

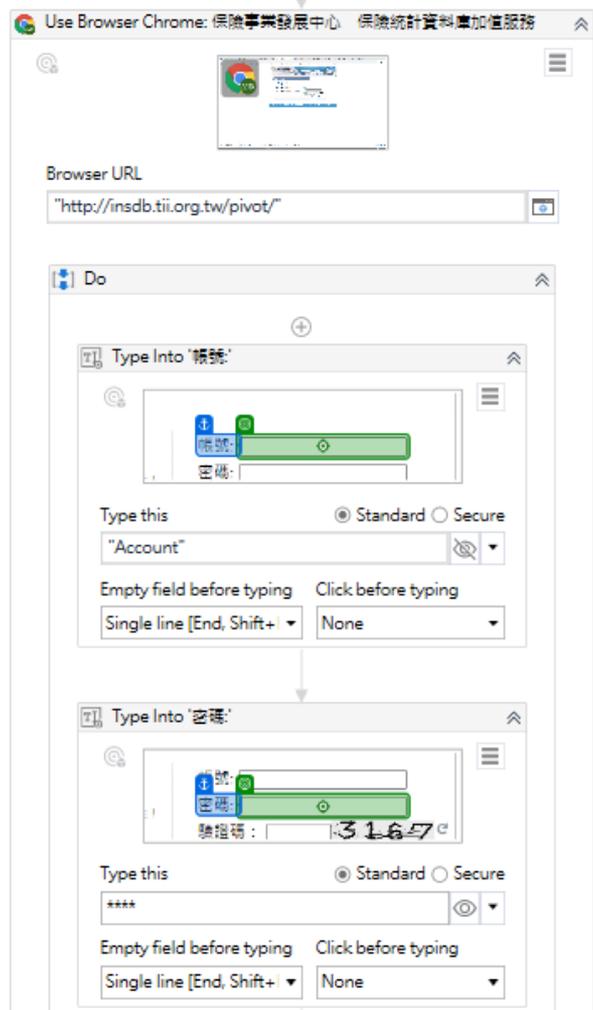
- Application path:** "C:\Windows\System32\notepad.exe"
- Application arguments:** *Text must be quoted*
- Match exact title: info - 記事本

Below this, there is a 'Do' section containing a 'Get Text' task:

- Get Text '3167':** A task with a green selection box and a 'Save to' field containing 'Val'.

# 模型建構

RPA輸入帳號及驗證碼完成登入

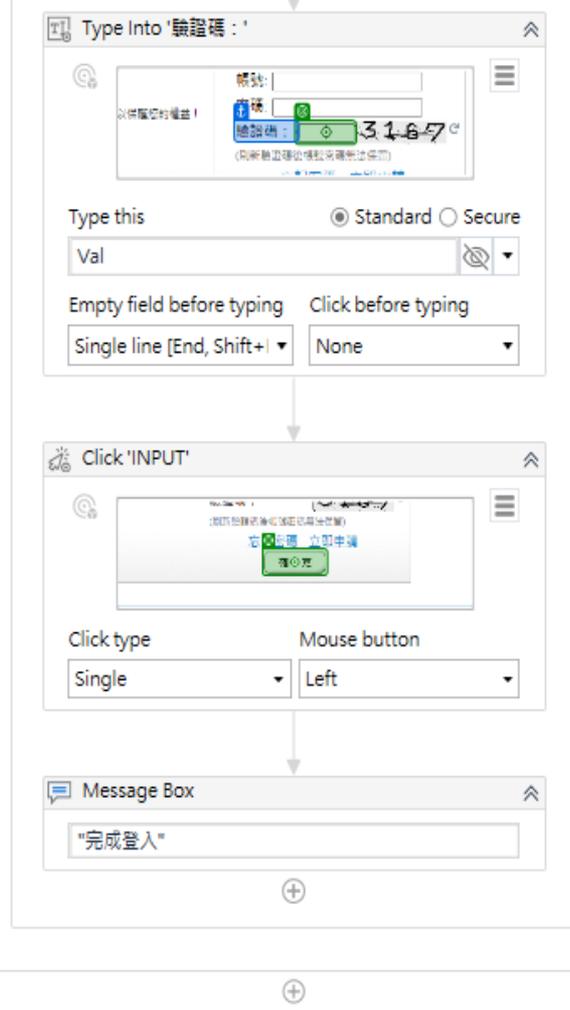


Use Browser Chrome: 保險專業發展中心 保險統計資料查詢服務

Browser URL  
"http://insdb.tii.org.tw/pivot/"

Do

- Type Into '帳號':
  - Type this: "Account"
  - Empty field before typing: Single line [End, Shift+]
  - Click before typing: None
- Type Into '密碼':
  - Type this: \*\*\*\*
  - Empty field before typing: Single line [End, Shift+]
  - Click before typing: None



- Type Into '驗證碼':
  - Type this: Val
  - Empty field before typing: Single line [End, Shift+]
  - Click before typing: None
- Click 'INPUT':
  - Click type: Single
  - Mouse button: Left
- Message Box:
  - Text: "完成登入"

# Outline



- 背景介紹
- 問題定義
- 模型建構
- 結論與未來展望

## 結論

融合RPA+AI有以下兩點效益：

RPA作為AI模型前的資料蒐集與處理達到非常高的效率，也替使用者大幅節省時間

*RPA+AI*

此專案目的為達成帳號登入，RPA在執行重複性動作更是專精，因此可解決AI模型追求高準確率的問題，下述算式表示準確率在0.85與0.99在執行五次後完成登入，其成功機率差異並不大。

$$1-(1-0.99)^5=0.9999999999$$

$$1-(1-0.85)^5=0.9999240625$$

## 未來展望



組織將機器人技術和先進的數字技術結合在一起，平衡了人力和機器人的勞動力，並實現了智能自動化。



結合AI人工智能，以便RPA機器人能夠了解問題何時發生並做出決定以解決問題。



Analytics（分析）是一種被人類使用的工具，但最終將被機器人等所取代。

## 其他

本研究使用UiPath Community 20.10.2及Colab

本研究在UiPath中使用以下Activities:

Use Application/Browser、Take screenshot、Save Image、  
Click、Get Text、Type Into、Do while、Message box...等。

**Thanks for your listening**