

การพัฒนาโปรแกรมประยุกต์และปัญญาประดิษฐ์ เพื่อการมองเห็นของเครื่องจักร Computer Programing and Artificial Intelligence in Machine Vision

4/4 – Machine Learning + Case Study

- Artificial Intelligence, Machine Learning and Deep Learning
- การเรียนรู้ของเครื่องจักร (Machine Learning)
- 10-Basic Machine Learning Algorithm
- Case Study 1 -- Sudoku to Text by Tesseract
- Case Study 2 -- Gender and Age Detection
- Case Study 3 -- Object Detection and Tracking
- Case Study 4 -- Visual Inspection
- คำถามท้ายบทเพื่อทดสอบความเข้าใจ

4/8 -- Case Study 1 -- Sudoku to Text by Tesseract

<https://ichi.pro/th/ocr-phrxm-tesseract-opencv-laea-python-231743215466598>

<https://medium.com/@bact/ทดลอง-tesseract-4-oalpha-กับภาษาไทย-8248a73c5ae5>

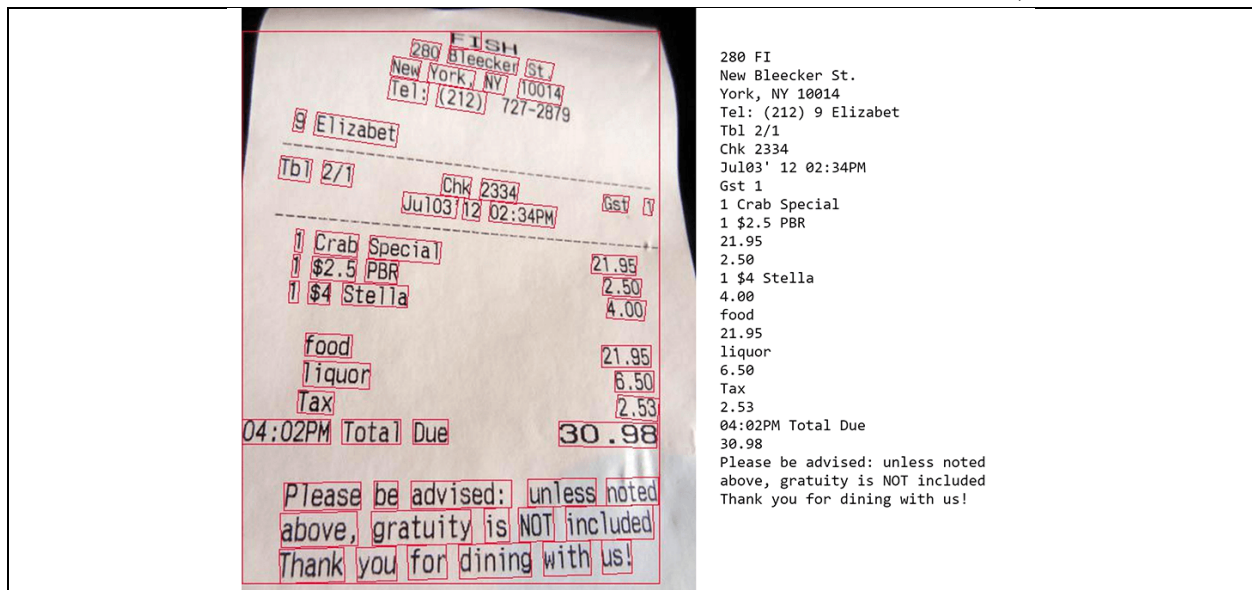
<https://bact.cc/2018/tesseract-thai/>

Tesseract-OCR

Tesseract เป็นซอฟต์แวร์และไลบรารีแปลงภาพข้อความ (ที่คนอ่านเข้าใจ) ให้เป็นข้อความ (ที่คอมพิวเตอร์อ่านเข้าใจ) หรือที่เรียกกันว่า OCR จะเรียกใช้จริงๆ ทาง command line ก็ได้ หรือจะเขียนโปรแกรมเชื่อมกับ API มันก็ได้

ภาพที่จะส่งมา Tesseract ต้องเป็นภาพที่ปรับแต่งมาให้เหมาะกับการอ่านข้อความแล้ว คือหมุนมาค่อนข้างตรง และปรับแสงและสีให้อ่านง่าย พื้นหลังสีขาวหรือสีอ่อน ตัวอักษรสีดำ ใน StackOverflow มีคนอธิบายการใช้ OpenCV ปรับภาพเพื่อ OCR เอาไว้

Tesseract รองรับภาษาไทย (น่าจะตั้งแต่รุ่น 3) ตอนนี้อยู่รุ่น 4 กำลังจะออก เพิ่มเอนจินที่ใช้โมเดล Deep Learning แบบ LSTM เข้ามา เท่าที่ทีมพัฒนาทดสอบกันเอง มีข้อผิดพลาดน้อยกว่าเอนจินของรุ่นก่อน



Lab404: Sudoku to Text by Tesseract

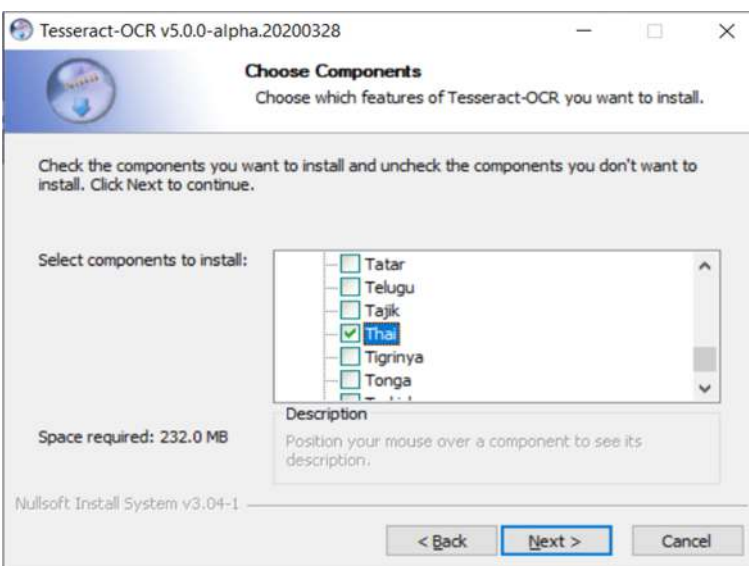
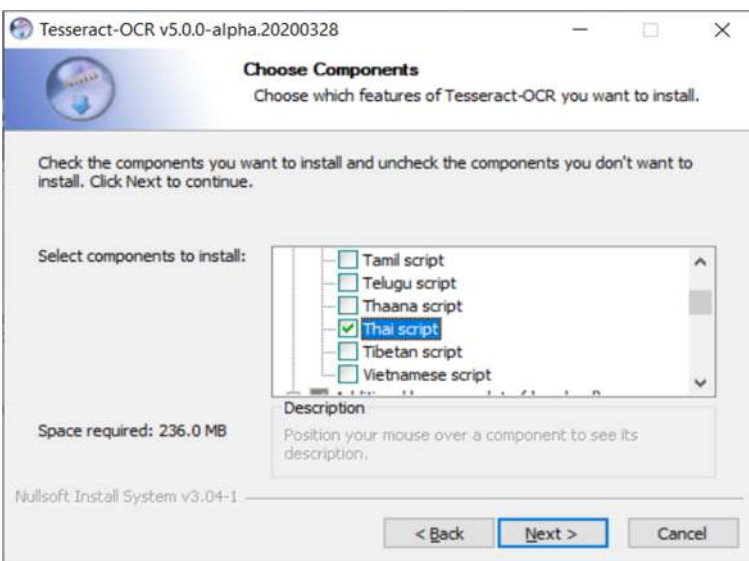
1. Read

- <https://medium.com/@navapat.tpb/python-3-ติดตั้งและใช้งาน-tesseract-ocr-สำหรับ-window-เพื่อสกัดข้อความจากภาพ-734dae2fb4d3>
- <https://github.com/UB-Mannheim/tesseract/wiki>

2. Download

- [tesseract-ocr-w32-setup-v5.0.0-alpha.20210506.exe](#) (32 bit) and
- [tesseract-ocr-w64-setup-v5.0.0-alpha.20210506.exe](#) (64 bit) resp.

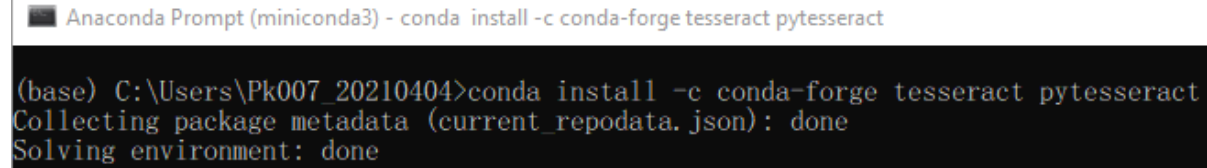
3. Install



4. Add to Anaconda

- Anaconda Command Prompt

conda install -c conda-forge tesseract pytesseract



```
Anaconda Prompt (miniconda3) - conda install -c conda-forge tesseract pytesseract
(base) C:\Users\Pk007_20210404>conda install -c conda-forge tesseract pytesseract
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

5. Test-A

```
import cv2
from PIL import Image
import pytesseract
pytesseract.pytesseract.tesseract_cmd = r'C:\\Program Files\\Tesseract-OCR\\tesseract.exe'

imageC = Image.open("./image/test_image.png")
imageC.show()

text_from_image = pytesseract.image_to_string(imageC)
print(text_from_image)
```



```
1
2 import cv2
3 from PIL import Image
4 import pytesseract
5 pytesseract.pytesseract.tesseract_cmd = r'C:\\Program Files\\Tesseract-OCR\\tesseract.exe'
6
7 imageC = Image.open("./image/test_image.png")
8 imageC.show()
9
10 text_from_image = pytesseract.image_to_string(imageC)
11 print(text_from_image)
12
13
```

This is a lot of 12 point text to test the ocr code and see if it works on all types of file format.

The quick brown dog jumped over the lazy fox. The quick brown dog jumped over the lazy fox. The quick brown dog jumped over the lazy fox. The quick brown dog jumped over the lazy fox.

6. Test-B

```
import cv2
from PIL import Image
import pytesseract
pytesseract.pytesseract.tesseract_cmd = r'C:\\Program Files\\Tesseract-OCR\\tesseract.exe'
```

```
imageC = cv2.imread("./image/Sudoku_01.jpg")
imageC = cv2.imread("./image/Sudoku_02.jpg")
imageC = cv2.imread("./image/Sudoku_02z.jpg")
#imageC = cv2.imread("./image/Sudoku_03.jpg")
```

```
ret,imageB = cv2.threshold(imageC,127,255,cv2.THRESH_BINARY)
imageX = cv2.cvtColor(imageB, cv2.COLOR_BGR2RGB)
imageP = Image.fromarray(imageX)
text_from_image = pytesseract.image_to_string(imageP)
print(text_from_image)
```

```
cv2.imshow('Test image',imageC)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

```
1 import cv2
2 from PIL import Image
3 import pytesseract
4 pytesseract.pytesseract.tesseract_cmd = r'C:\\Program Files\\Tesseract-OCR\\tesseract.exe'
5
6
7 imageC = cv2.imread("./image/Sudoku_01.jpg")
8 imageC = cv2.imread("./image/Sudoku_02.jpg")
9 imageC = cv2.imread("./image/Sudoku_02z.jpg")
10 #imageC = cv2.imread("./image/Sudoku_03.jpg")
11
12 ret,imageB = cv2.threshold(imageC,127,255,cv2.THRESH_BINARY)
13 imageX = cv2.cvtColor(imageB, cv2.COLOR_BGR2RGB)
14 imageP = Image.fromarray(imageX)
15 text_from_image = pytesseract.image_to_string(imageP)
16 print(text_from_image)
17
18 cv2.imshow('Test image',imageC)
19 cv2.waitKey(0)
20 cv2.destroyAllWindows()
21
22
```

```
123456789
456789123
789123456
```



7. Numeric Only

- <https://stackoverflow.com/questions/46574142/pytesseract-using-tesseract-4-0-numbers-only-not-working>

8. Step-by-Step Test → Sudoku01_SplitCell

- เปิดไฟล์
- ตีเส้น
- หาขอบ

9. Step-by-Step Test → Sudoku02_Tesseract Test

- Test Text Image
- Test Numeric Image

10. Step-by-Step Test → Sudoku03_Pic2Text

- Open and Split Cell
- Cell Image to Numeric

กิจกรรมที่ 3/6 – Sudoku to Text by Tesseract

- Capture ผลการทำงานที่ได้ลองปฏิบัติ
- ลองใช้ตารางชุดอื่น ในการทดสอบ
- อภิปรายผล
- คำถามที่อยากถาม
- บอกแนวการใช้งาน กับงานที่รับผิดชอบ