

광학 문자 인식(Optical Character Recognition)



일반적인 컴퓨터 비전 과제는 이미지에서 텍스트를 감지하고 해석하는 것이다. 이러한 종류의 처리를 흔히 광학 문자 인식(OCR)이라고 한다.

Computer Vision 서비스를 이용하여 이미지에 있는 텍스트 읽기

Computer Vision Cognitive 서비스는 다음과 같은 OCR 작업을 위한 기능을 제공해준다.

- **OCR API**를 이용하여 여러 언어로 된 텍스트를 읽는다. 이 API는 동기식으로 사용할 수 있으며, 이미지에 소량의 텍스트를 감지하고 읽어야 할 때 잘 작동한다.
- **Read API**는 대용량 문서에 최적화되어 있다. 이 API는 비동기식으로 사용되며 인쇄 텍스트와 필기 텍스트 모두에 사용할 수 있다

이 서비스는 **Computer Vision**나 **Cognitive Services** 리소스를 생성하여 사용할 수 있다.

만일 아직 이 서비스가 없다면 Azure 구독에서 **Cognitive Services** 리소스를 생성할 수 있다.

1. 브라우저의 새로운 탭을 열고, Azure portal(<https://portal.azure.com>)을 (<https://portal.azure.com>)을 입력하고 , Microsoft계정으로 로그인한다..
2. + 리소스 만들기 버튼을 클릭하고, *Cognitive Services* 서비스를 찾은 다음, **Cognitive Services** 리소스를 다음과 같은 내용으로 생성한다.
 - 이름: 유일한 이름을 입력한다(가능하면 영문과 숫자사용).
 - 구독: *Azure* 구독선택.
 - 위치: 가능한 위치(한국 중부 추천):
 - 가격책정계층: 표준 S0
 - 리소스 그룹: 원하는 유리한 이름(가능하면 영문과 숫자사용).
3. 배포가 완료될 때까지 기다린다. 그런 다음 Cognitive Services 리소스로 이동하여 *개요 페이지에서 링크를 클릭하여 서비스 키를 관리한다. 클라이언트 응용 프로그램에서 Cognitive Services 리소스에 연결하려면 엔드포인트와 키가 필요하다.

Cognitive Services 리소스에 있는 키와 엔드포인트 가져오기

Cognitive Services 리소스를 사용하기 위해서는, 클라이언트 응용프로그램에서는 엔드포인트와 인증 키가 필요하다.

1. Azure portal에서, Cognitive Services 리소스를 선택하고 **키 및 엔드포인트** 페이지를 선택한 다음 **키1** 을 복사하여 아래의 **YOUR_COG_KEY**.를 붙여 넣는다.
2. 리소스에 있는 **엔드포인트** 를 복사해서 아래의 **YOUR_COG_ENDPOINT**.에 붙여 넣는다.

이제 세팅한 키와 엔드포인트를 사용하여 코드를 실행하면, 컴퓨터 비전 서비스를 사용할 수 있다.

In []:

```
cog_key = 'YOUR_COG_KEY'
cog_endpoint = 'YOUR_COG_ENDPOINT'

print('Ready to use cognitive services at {} using key {}'.format(cog_endpoint, cog_key))
```

이제 키와 엔드포인트를 설정했으니 이미지로부터 텍스트를 추출하기 위해 Computer Vision 서비스 리소를 사용할 수 있다.

Python에서 이런 작업을 하기 위해서는 아래의 셀을 실행하여 the Azure Cognitive Services Computer Vision 패키지를 실행한다.

In []:

```
! pip install azure-cognitiveservices-vision-computervision
```

이제 Computer Vision 서비스를 이용하여 이미지에 있는 텍스트를 읽을 수 있을 준비가 되어 있다.

OCR API로 시작해보자. 이 API를 사용하면 이미지를 동기적으로 분석하고 포함된 텍스트를 읽을 수 있다. 이 경우, 가상의 Northwind Traders 소매 회사의 광고 이미지에 포함된 텍스트를 확인할 수 있다.

In []:

```
from azure.cognitiveservices.vision.computervision import ComputerVisionClient
from msrest.authentication import CognitiveServicesCredentials
import matplotlib.pyplot as plt
from PIL import Image, ImageDraw
import os
%matplotlib inline

# Computer Vision 서비스에 접근하기 위한 클라이언트를 가져온다.
computervision_client = ComputerVisionClient(cog_endpoint, CognitiveServicesCredentials(cog_key))

# 이미지 파일을 읽는다.
image_path = os.path.join('data', 'ocr', 'advert.jpg')
image_stream = open(image_path, "rb")

# Computer Vision 서비스를 이용하여 이미지에 포함된 텍스트를 가져온다.
read_results = computervision_client.recognize_printed_text_in_stream(image_stream)

# 텍스트를 한줄씩 처리한다
for region in read_results.regions:
    for line in region.lines:

        # 텍스트 줄에 있는 단어별로 가져온다.
        line_text = ''
        for word in line.words:
            line_text += word.text + ' '
        print(line_text.rstrip())

# 이미지를 화면에 나타낸다.
fig = plt.figure(figsize=(7, 7))
img = Image.open(image_path)
draw = ImageDraw.Draw(img)
plt.axis('off')
plt.imshow(img)
```

이미지에서 발견된 텍스트는 영역, 선 및 단어의 계층 구조로 구성되고 코드는 결과를 검색하기 위해 이를 읽는다.

그 결과 이미지 위에 있는 텍스트를 볼 수 있다.

경계 상자(Bounding Box) 나타내기

또한 결과에는 이미지에 있는 텍스트 행과 개별 단어에 대한 경계 상자의 좌표가 포함됩니다. 위에서 검색한 광고 이미지의 텍스트 행에 대한 경계 상자를 보려면 아래 셀을 실행하라.

In []:

```
# 이미지를 표시하기위에 연다.
fig = plt.figure(figsize=(7, 7))
img = Image.open(image_path)
draw = ImageDraw.Draw(img)

# 텍스트를 줄별로 처리한다.
for region in read_results.regions:
    for line in region.lines:

        # 각 줄에서 텍스트의 위치를 나타낸다.
        l,t,w,h = list(map(int, line.bounding_box.split(' ')))
        draw.rectangle(((l,t), (l+w, t+h)), outline='magenta', width=5)

        # 텍스트 줄에 있는 단어를 읽는다.
        line_text = ''
        for word in line.words:
            line_text += word.text + ' '
        print(line_text.rstrip())

# 텍스트의 위치를 강조한 이미지를 보여준다.
plt.axis('off')
plt.imshow(img)
```

그 결과 텍스트의 각 라인에 대한 경계상자가 이미지 위에 사각형으로 나타난다.

Read API 사용하기

이전에 사용한 OCR API는 적은 양의 텍스트가 있는 이미지에 적합하다. 스캔한 문서와 같이 더 많은 텍스트 본문을 읽어야 할 경우 **Read** API를 사용할 수 있다. 그 과정은 다음과 같이 여러 단계가 필요하다.:

1. 이미지를 Computer Vision 서비스로 보내어 읽고 비동기적으로 분석하도록 한다.
2. 분석작업이 완료될때까지 기다린다.
3. 분석결과 값을 받아서 확인한다.

아래의 셀을 실행함으로 Northwind Traders 가게의 매니저에게 보낼 스캔한 편지 속에 있는 텍스트를 읽는 작업을 수행한다.

In []:

```
from azure.cognitiveservices.vision.computervision import ComputerVisionClient
from azure.cognitiveservices.vision.computervision.models import OperationStatusCodes
from msrest.authentication import CognitiveServicesCredentials
import matplotlib.pyplot as plt
from PIL import Image
import time
import os
%matplotlib inline

# 이미지 파일을 읽는다.
image_path = os.path.join('data', 'ocr', 'letter.jpg')
image_stream = open(image_path, "rb")

# Computer Vision 서비스를 사용하기 위한 클라이언트를 가져온다.
computervision_client = ComputerVisionClient(cog_endpoint, CognitiveServicesCredentials(cog_key))

# 이미지에 프린트되어 있는 텍스트를 읽고 작업 ID를 가져오기 위한 요청을 보낸다.
read_operation = computervision_client.read_in_stream(image_stream,
                                                       raw=True)
operation_location = read_operation.headers["Operation-Location"]
operation_id = operation_location.split("/")[-1]

# 비동기적인 작업이 완료될때까지 기다린다.
while True:
    read_results = computervision_client.get_read_result(operation_id)
    if read_results.status not in [OperationStatusCodes.running]:
        break
    time.sleep(1)

# 만일 작업이 완성되었다면 텍스트를 줄단위로 처리한다.
if read_results.status == OperationStatusCodes.succeeded:
    for result in read_results.analyze_result.read_results:
        for line in result.lines:
            print(line.text)

# 이미지를 열고 나타낸다.
print('\n\n')
fig = plt.figure(figsize=(12,12))
img = Image.open(image_path)
plt.axis('off')
plt.imshow(img)
```

결과를 살펴보자. 편지의 전체 복사본이 있는데 이것은 손으로 쓴 사인과 함께 대부분은 인쇄된 텍스트로 구성되어 있다. 편지의 원래 이미지는 OCR결과 밑에 나타나게 된다(내용을 보기 위해서 스크롤 할 필요가 있다)

손으로 쓴 텍스트 읽기

이전 예제에서 이미지 분석 요청은 프린트된 텍스트에 대한 텍스트 인식 작업에 최적화 되어 있었다. 그럼에도 불구하고 손으로 쓴 서명이 읽혀졌다는 점에 유의하자.

손으로 쓴 글을 읽는 능력은 매우 유용하다. 예를 들어, 수기로 쓴 쇼핑 목록이 포함된 노트를 작성한 경우 스마트폰 앱을 사용하여 노트를 읽어 노트에 포함된 텍스트를 찾아내기를 원한다고 가정하자.

아래의 셀을 실행하여 수기로 작성한 쇼핑 목록에 대한 읽기 작업의 예를 확인하라.

In []:

```
from azure.cognitiveservices.vision.computervision import ComputerVisionClient
from azure.cognitiveservices.vision.computervision.models import OperationStatusCodes
from msrest.authentication import CognitiveServicesCredentials
import matplotlib.pyplot as plt
from PIL import Image
import time
import os
%matplotlib inline

# 이미지 파일을 읽는다.
image_path = os.path.join('data', 'ocr', 'note.jpg')
image_stream = open(image_path, "rb")

# 컴퓨터 비전 서비스를 위한 클라이언트를 가져온다.
computervision_client = ComputerVisionClient(cog_endpoint, CognitiveServicesCredentials(cog_key))

# 이미지에 포함되어 있는 프린트된 텍스트를 읽기 위한 요청을 보내고 operation ID를 가져온다
read_operation = computervision_client.read_in_stream(image_stream,
                                                       raw=True)
operation_location = read_operation.headers["Operation-Location"]
operation_id = operation_location.split("/")[-1]

# 비동기적인 작업이 완료될 때까지 기다린다.
while True:
    read_results = computervision_client.get_read_result(operation_id)
    if read_results.status not in [OperationStatusCodes.running]:
        break
    time.sleep(1)

# 만일 작업이 성공적으로 끝났다면 줄 단위로 텍스트를 읽는다.
if read_results.status == OperationStatusCodes.succeeded:
    for result in read_results.analyze_result.read_results:
        for line in result.lines:
            print(line.text)

# 이미지를 열어서 나타낸다.
print('\n\n')
fig = plt.figure(figsize=(12,12))
img = Image.open(image_path)
plt.axis('off')
plt.imshow(img)
```

더 많은 정보

OCR을 위한 Computer Vision 서비스의 사용에 대한 더 많은 정보를 여기를 확인하라 [Computer Vision 문서](https://docs.microsoft.com/ko-kr/azure/cognitive-services/computer-vision/concept-recognizing-text)
(<https://docs.microsoft.com/ko-kr/azure/cognitive-services/computer-vision/concept-recognizing-text>)