That's Fit to Print"

The AI Newsletter

AI Newsletter No.10B

AI LAB, MONDAY, JUL 22, 2024

Al Tech



Tips



디지털전환과 AI 시대에 인터넷과 언론에서 자주 들을 수 있는 AI 및 코딩관련 온라인 플랫 폼들을 소개해 드리겠습니다. 소프트웨어 개발 자들에게는 필수적인 깃허브나 전세계 빅데이 터 분석가들이 데이터를 얻고 분석 실력을 연마 하는 케글 등이 아래 설명을 읽어 보시는 순간 부터 더 이상 낮설게 느껴지시지는 않으실 겁니다. ③ 케글(Kaggle)

① 깃(Git)

소스코드의 백업(Backup), 버전관리(Version Control), 협업(Collaboration)을 위한 도구로 명 령어 인터페이스 기반으로 개인 또는 소규모 팀 을 위한 로컬 또는 자체 호스팅 시스템

② 깃허브(GitHub)

Git과 유사하나 보다 풍성한 기능을 제공하는 클라우드 기반의 웹 플랫폼으로 개인 및 대규모 팀, 상업용 프로젝트까지 지원하는 강력한 소프 트웨어 개발 협업 플랫폼

<주요기능>

1. 저장소 호스팅: Git 저장소를 온라인상에 호스팅

- 코드 리뷰 등
- 3. 프로젝트 관리: 프로젝트 보드, 마일스톤, 위키 등 등과 같은 다양한 사전 훈련된 모델을 제공
- 4. 소셜 코딩: 다른 개발자의 프로젝트를 포크 하고 기여할 수 있음
- 5. CI/CD 통합: GitHub Actions를 통한 자동화 된 빌드 및 배포

Google이 소유한 데이터 과학자와 머신러닝 전문가를 위한 온라인 커뮤니티 플랫폼으로 데 이터 분석 경진대회를 호스팅하고, 데이터셋을 공유하며, 코드를 작성하고 실행할 수 있는 환 경을 제공

④ 스택 오버플로우(Stack overflow)

프로그래밍 관련 질문과 답변을 공유하는 커 뮤니티로 거의 모든 프로그래밍 문제에 대한 해 답을 찾을 수 있음

⑤ 허깅 페이스(Hugging face)

기계 학습 모델을 구축, 배포 및 교육하기 위 한 도구와 리소스를 개발하는 프랑스계 미국 회

2. 협업 도구: 이슈(버그, 요청사항) 트래킹, 풀 사이자 오픈 소스 커뮤니티로 개발자들이 모델과 리퀘스트(변경사항을 제안하고 리뷰를 요청), 데이터셋을 공유하고 협업할 수 있는 플랫폼. NLP를 위해 구축된 Transformers 라이브러리











Hugging Face

고차원 함수로 Python 코드를 간결하게!

고차원 함수(Higher-order Functgion)란 다른 함수를 인자로 받거나 함 수를 반환하는 함수를 말합니다. Python에서 고차원 함수는 함수형 프로 그래밍 스타일을 지원하며 코드를 간결하고 재사용 가능하게 만드는 데 도움을 줍니다. 이번 호에서는 몇 가지 대표적인 고차원 함수들과 간단 한 예제를 소개하겠습니다.

DUICKTIPS) 파이썬 고차원함수 3좆세트

map(square, numbers) # numbers 리스트의 각 요소에 square 함수를 적용하여 제곱한 값을 반환합니다. list()로 감싸서 결과를 리스트로 변환합니다.

filter(is_even, numbers) #numbers 리스트에서 is_even 함수가 True를 반환하는 짝수만 남기고 리스트로 반환합니다

reduce(multiply, numbers) # numbers 리스트의 요소들을 왼쪽에서 오른쪽으로 순차적으로 multiply 함수에 적용하여 모든 요소의 곱을 반환합니다.

① map() 함수

map() 함수는 iterable(예: 리스트, 튜플 등)의 각 요소에 함수를 적용하 여 새로운 iterable을 반환합니다.

def square(x): return x ** 2 numbers = [1, 2, 3, 4, 5]# map()함수로 리스트내 각 요소를 함수에 적용 squared numbers = list(map(square, numbers)) print(squared numbers)

[1, 4, 9, 16, 25] # 결과

② filter() 함수

filter() 함수는 iterable의 각 요소를 함수에 전달하고, 함수가 True를 반환하는 요소만 남겨서 반환합니다.

함수를 정의 ven(x): # 나머지가 0이면 True 반환 return x % 2 == 0 numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] #리스트 정의 # filter()함수로 리스트내 각 요소를 함수에 적용 even_numbers = list(filter(is_even, numbers)) print(even_numbers)

[2, 4, 6, 8, 10] #결과

③ reduce() 함수

reduce() 함수는 iterable의 요소를 순서대로 함수에 적용하여 단일 값 으로 축소합니다. Python 3.0부터는 functools 모듈에서 제공됩니다.

from functools import reduce #함수를 정의 oly (x, y): # 두 요소를 곱한 값 반환 # 리스트 정의 # multiply() 함수로 리스트내 각 요소를 함수에 적용해 모든 요소를 곱한다. product = reduce(multiply, numbers) orint(product)

120 #결과

이러한 고차원 함수들은 코드를 간결하게 유지하고, 함수를 재사용 가능 하게 하며, 함수형 프로그래밍의 핵심 기능을 지원합니다. Python에서는 이 외에도 sorted(), any(), all() 등의 다양한 고차원 함수를 제공하여 작업 을 더욱 효율적으로 할 수 있습니다.



Hands-on AI Project



Al Tech

파이썬으로 얼굴인식과 감정을 분석하는 알고리즘을 구현하자!

이번 Hands-On에서는 구글 코랩(Google Colab)에서 이미지를 업로드하고 얼굴을 인식한 후, 감정 분석을 수행하는 방법을 다룹니다. 이 프로젝트는 사전 학습된 모델을 활용하여 컴퓨터 비전의 기본 개념을 익히는 데 유용하며, 실무에서 활용할 수 있는 기술을 배우게 될 것입니다. 향후 다양한 이미지로 테스트를 해보시고, 실시간 웹캠 스트림에서 얼굴 인식 및 감정 분석을 구현해보는 것도 좋은 도전이 될 것입니다.

필요 라이브러리 설치

- !pip install opencv-python-headless #이미지 및 비디오 처를 위한 하이브러리
- !pip install keras #고수준의 신경망 api로, TensorFlow 위에서 실행
- !pip install tensorflow #딥러닝 모델을 만들고 학습시키기 위한 오픈소스
- !pip install mtcnn #얼굴 인식을 위한 멀티 태스크 캐스케이드 합성곱 신경망
- ※ 위 4개의 라이브러리 설치 및 전체 Source 코드 및 결과는 분량상 생략, 전체 코드는 아래 링크를 참고해주세요.

구글 코랩에서 열기(인터넷망)

① 관련 라이브러리 참조

import cv2 # OpenCV 라이브러리 import, 이미지 및 비디오 처리에 사용 import numpy as np # NumPy 라이브러리 import, 수치 연산에 사용 # Keras에서 모델 로드 함수 import, 훈련된 모델을 불러오기 위해 사용

from keras.models import load_model # MTCNN 라이브러리에서 MTCNN 클래스 import, 얼굴 인식에 사용

from mtcnn.mtcnn import MTCNN

from google.colab import files #Google Colab에서 파일 업로드/다운로드 기능 import import matplotlib.pyplot as plt #Matplotlib 라이브러리 import, 이미지 표시에 사용 import requests #requests 라이브러리 import, URL에서 데이터 가져오기 위해 사용

② 감정 분석 모델 로드

사전에 학습된 모델을 사용 (예: FER2013 데이터셋으로 학습된 모델 model_path =

'https://github.com/oarriaga/face_classification/raw/master/trained_models/emotion_models/fer2013_mini_XCEPTION.102-0.66.hdf5'

response = requests.get(model_path) # URL에서 모델 파일을 가져옴

#모델을 로컬 파일로 저장

with open('fer2013_mini_XCEPTION.102-0.66.hdf5', 'wb') as f:

f.write(response.content)

로컬 파일에서 모델을 로드

model = load_model('fer2013_mini_XCEPTION.102-0.66.hdf5')

③ 이미지 업로드

사용자가 이미지를 업로드하고, OpenCV를 사용하여 이미지를 로드합니다.

이미지 업로드

uploaded = files.upload()

업로드한 이미지 로드

image path = list(uploaded.keys())[0]

image = cv2.imread(image_path)

 $image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)$

④ 얼굴 인식

MTCNN(Multi-task Cascaded Convolutional Networks)을 사용하여 이미지에서 얼굴을 인식합니다.

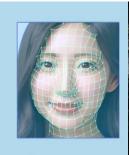
얼굴 인식

detector = MTCNN()

faces = detector.detect_faces(image_rgb)

select ident, name, municipality from airports where iso_region= 'US-CA' and type= 'large_airport'







⑤ 감정 분석

인식된 얼굴 영역을 모델에 입력하여 감정을 분석하고, 결과를 이미지에 표시합니다.

감정 분석

emotion_labels = ['Angry', 'Disgust', 'Fear', 'Happy', 'Sad', 'Surprise', 'Neutral']
for face in faces:

x, y, width, height = face['box']

face_region = image_rgb[y:y+height, x:x+width] # 64x64로 크기 조정하고 그레이스케일로 변환

face_region = cv2.resize(face_region, (64, 64))

#그레이스케일로 변환

face_region = cv2.cvtColor(face_region, cv2.COLOR_RGB2GRAY)

face_region = np.expand_dims(face_region, axis=0)

face_region = np.expand_dims(face_region, axis=-1) # 채널 차원 추가

face_region = face_region / 255.0

prediction = model.predict(face_region)
emotion_label = emotion_labels[np.argmax(prediction)]

얼굴 영역과 감정 라벨 그리기

cv2.rectangle(image_rgb, (x, y), (x+width, y+height), (255, 0, 0), 2)

cv2.putText(image_rgb,emotion_label,(x, y-10),cv2.FONT_HERSHEY_SIMPLEX,0.9,(255, 0,0),2)

⑥ 결과 이미지 출력

plt.figure(figsize=(10, 10))
plt.imshow(image_rgb)
plt.axis('off')
plt.show()

WARNING:tensorflow:Error in loading the saved optimizer state. As a result, your model is starting with a freshly initialized optimizer.

파일선택 선택된 파일 없음

Cancel upload









※ 주의

발연기는 못찾아냅니다. ㅠㅠ 언젠간 속마음까지 들킬지도... 참고로 뇌파 읽는 AI가 있기는 합니다. 뉴스레터 7호(인터넷망)

