Finger Print Recognition 알고리즘 / CNN 모델링

Anaconda환경

* Jupyter Notebook ( ipynb 파일 실행 )
* Numpy : Python에서 수학 및 과학 연산을 하기위한 파이썬 패키지
* Keras : TensorFlow(텐서플로)의 딥러닝 모델 설계와 훈련을 위한 API
* Matplotlib : Python에서 자료(이 프로젝트에서는 opencv로 읽어들인 이미지 배열(numpy))를 차트나 플롯(plot)으로 시각화하는 패키지
* Sklearn : 머신러닝(ML)교육을 위한 파이썬 패키지(데이터셋 구성, 데이터 전처리, 모형 평가 및 선택) / Numpy나 Matplotlib 등을 기반으로 함
* Imgaug : Image Augmentation 딥러닝 모델에 필요한 다양한 데이터 확보를 위한 데이터 변조에 쓰이는 python 라이브러리
* OpenCV : 실시간 이미지 프로세싱에 중점을 둔 라이브러리. C/C++로 개발됐고, 파이썬, 자바 등으로 바인딩 되어있음
* Glob : 현재 디렉토리나 상위, 하위 등의 경로에 있는 파일을 읽어들이는 라이브러리
* os : 운영체제에서 제공되는 기본적인 기능들을 담은 라이브러리
* Pickle : 리스트나 클래스 같은 자료들을 파일로 저장하기 위한 모듈

DataSet

* SOCOFing (600명 10손가락의 정보)
* 1~100까지 삭제 후 팀원들의 지문정보를 1~100까지 라벨링한 후 입력해서 새로 DataSet 생성

모델링 알고리즘

CNN : Convolutional Neural Network, 딥러닝의 한 종류로 주로 영상이미지 처리하는데 사용. (전연결(Fully Connected) 신경망으로 모델링을 할 경우 3차원 데이터인 이미지를 1차원으로 평면화 해야하는데, 이 과정에서 정보가 손실될 수밖에 없음.) 이미지 데이터의 형태를 유지하면서 학습이 가능한 모델. 요약 : 신경망에 기존의 필터기술을 병합하여 신경망이 2차원(색 표현인 채널값까지 포함하여 3차원인) 이미지 및 영상을 잘 습득할 수 있도록 최적화시킨 알고리즘

이미지 전처리

가공되지 않은 지문 이미지들

* 이미지를 순번\_\_(남녀)\_(왼손오른손)\_(손가락)이라는 이름으로 저장 ex) 1\_\_M\_R\_index\_finger
* Real에 있는 이미지를 읽음 ( OpenCV – imread() )
* 일정한 사이즈로 재조정 ( OpenCV – resize() )
* 이미지의 이름에 따라 라벨링 순번/남or녀/왼or오/손가락
* 이미지들은 배열 하나로 만들어 x\_real.npz로 저장( Numpy – savez / 인자로 data=imgs )
* 라벨 역시 배열로 만들어 y\_real.npy로 저장( Numpy – save / 인자로 labels )

가공된 이미지들

* Real 디렉토리에 있던 이미지들을 흐릿하게 하거나 일정부분 잘라내거나 일부분 회전시켜서 저장 ( Imgaug – Sequential, GaussianBlur, Affine(회전, 잘라내기, 사이즈 조정 등) )
* Imgaug를 통해 어느 정도 변경된 이미지들을 Altered/Alterd-Easy, Altered/Alterd-Medium, Altered/Alterd-Hard에 각각 저장
* 저장된 이미지들은 Real이미지를 처리할 때와 같은 순서로 전처리 ( 이미지 읽어오기 – 사이즈 조정 – 라벨링 – 이미지는 npz, 라벨은 npy로 저장 )

학습

* Npz와 npy에 저장한 이미지와 라벨들을 모두 읽어옴
* 읽어온 npz중 Altered-Easy, Altered-Medium, Altered-Hard 이 세 디렉토리의 이미지들을 하나의 배열로 통합 ( Numpy – concatenate ) -> x\_data
* 라벨 데이터인 npy도 Altered-Easy, Altered-Medium, Altered-Hard 이 세 디렉토리의 라벨들을 하나의 배열로 통합 ( Numpy – concatenate ) -> label\_data
* x\_data 와 label\_data를 가지고 sklearn의 model 모듈을 통해 train\_set과 test\_set으로 나눔 ( sklearn.model\_selection – train\_test\_split )
* Real이미지에서 추출한 라벨 데이터인 y\_real.npy를 갖고 label에 대한 사전(dictionary)로 만듦
* Keras.utils.Sequence를 상속받는 DataGenerator라는 클래스를 생성
* train\_set과 test\_set을 기준으로 x\_real, y\_real, label\_dict 등을 인자로 받아 DataGenerator를 통해 train\_gen과 val\_gen이란 데이터셋을 생성
* 모델의 구조를 먼저 선언함 (Convolution, Pooling, Flatten 등을 세팅)
* 모델을 학습 시킴 ( keras – fit\_generator(), 인자로 train\_gen과 val\_gen을 입력 )

모델 저장

* Model을 json파일로 저장
* 모델에 쓰인 가중치를 h5파일로 저장(save\_weights)
* 모델과 함께 쓰일 데이터인 x\_real, y\_real, label\_dict를 Pickle로 저장

Evaluation

* 입력값(라즈베리파이의 지문센서에 입력할 지문 이미지)을 이미지 전처리와 비슷하게 약간의 가공
* 인증 요청을 통해 지문센서에 지문을 인식시키면 지문을 저장해 이미지로 전송.
* 이미지를 라벨링, 사이즈 조정, 이미지 회전 등을 거쳐 model모듈의 predict를 통해 DB에 저장된 이미지와 비교하여 일치율을 반환