



2024-1학기 창의학기제 주간학습보고서 (4주차)

창의과제	세종대학교 집현캠퍼스를 개선시킨 웹서비스 개발				
이름	이지민	학습기간	4월13일 ~ 4월 30일		
학번	23012127	학습주차	4	학습시간	3
학과(전공)	인공지능	과목명	자기주도 창의전공1	수강학점	3
※ 수강학점에 따른 회차별 학습시간 및 10주차 이상 학습 준수					
금주 학습목표	인터넷에 오픈된 얼굴인식 인공지능이 어떤식으로 작동하는지 알아보기				
학습내용	<p>먼저 지난 주차에 이어 contrastive loss와 triplet loss 함수를 사용했을 때 좋은 학습결과가 나올 수 있는지에 대해 알아보았다. 얼굴 인식을 위한 인공지능은 아무리 비슷해 보이는 얼굴이라도 분류해내야 하는 능력을 가져야 하기 때문에 모델을 학습시킬 때 랜덤으로 얼굴을 학습시키기 보다는 데이터가 적더라도 비슷한 얼굴을 매칭시켜 학습시키는 것이 더 중요하다는 것을 알 수 있다.</p> <p>인터넷을 통해 dlib라는 얼굴인식을 위한 인공지능에 대해 알아보게 되었다. dlib의 face_recognition이 어떻게 학습된 것인지는 자세하게 나와있는 페이지를 찾지 못했다. 그래서 한번 모델을 불러와 결과값을 확인해 보았다.</p> <p>다음은 통하여 dlib의 face_recognition 패키지를 가져올 수 있다, pip install numpy pandas opencv-python face-recognition git clone https://github.com/ageitgey/face_recognition.git</p> <p>패키지를 가져왔다면 다음 코드로 import할 수 있다. import face_recognition</p> <p>이때 face_recognition.face_locations() 함수를 통해 얼굴의 위치를 face_recognition.face_encodings() 함수를 통해 사전에 학습된 모델을 통해 128차원의 특징벡터로 바꿔준다.</p> <p>lfw라는 얼굴 데이터셋을 발견하여 dlib의 face_recognition을 통해 추출된 lfw의 특징벡터와 내 얼굴 특징벡터의 거리값을 비교해 보았다.</p> <pre># 내 첫 번째 이미지와 lfw의 거리 계산 np_dist_user_db = np.linalg.norm(np_embs_user[0] - np_embs, axis=1) # 내 첫 번째 이미지와 내이미지 전체의 거리 계산 np_dist_user_user = np.linalg.norm(np_embs_user[0] - np_embs_user, axis=1)</pre> <div><pre>10 print(np_dist_user_user.mean()) 1 print(np_dist_user_db.mean()) 0.1988684414014364 1.3513789389585384</pre></div> <p>출력값을 확인해 보면 내 사진 사이의 거리의 평균보다 lfw 사진과의 거리의 평균이 더 큼을 알 수 있다.</p>				



	<p>이를 잘 분류하는 모델을 만들 수 있도록 특징 벡터를 사용해 svm을 학습시켜보았다.</p> <pre># sklearn을 활용한 svm학습 import numpy as np from sklearn.model_selection import train_test_split from sklearn.svm import SVC from sklearn.metrics import accuracy_score clf = SVC(gamma='auto', probability=True) clf.fit(X_train, y_train) y_pred = clf.predict(X_test) print("Accuracy:", accuracy_score(y_test, y_pred))</pre> <p>Accuracy: 0.9891345073061072</p> <p>요기서 주의해야 할 점은 정확도는 높게 나왔지만 Ifw의 데이터가 훨씬 많기 때문에 믿을 만한 수치는 아니다.</p> <p>학습에 활용되지 않은 내 얼굴 이미지를 추가하여 한번 더 예측을 시행해보았다.</p> <pre>total = len(np_embs_user) rights = 0 for i in np_embs_user: prediction = clf.predict([i]) if prediction[0] == 0: # 예측값이 0일 때 맞은 값으로 인식 rights += 1 print(rights/total*100)</pre> <p>100.0</p> <p>이를 통해 모델이 overfitting됐을 가능성을 확인해볼 수 있었으며 이 원인으로 데이터의 단순성을 꼽았다.</p>
학습방법	<p>인터넷 자료를 활용하여 dlib모델을 활용한 얼굴인식 기능을 사용해봄</p> <p>특징 벡터를 통해 얼굴을 인식하기 위한 SVM활용을 위해 sklearn.svc의 사용법에 대해 알아봄</p>
학습성과 및 목표달성도	<p>본래는 다른 얼굴인식 모델이 어떻게 학습되었는지 알고 싶었는데 이를 알 수 없었다. 하지만 이미 완성된 모델에 데이터를 넣어 믿을 만한 특징 벡터를 가지고 SVM 모델을 학습시킨 결과 현재 데이터가 너무 단순할 수 있다는 가능성을 알게 되었다.</p>
참고자료 및 문헌	<p>https://github.com/ageitgey/face_recognition</p> <p>https://github.com/davisking/dlib-models#dlib_face_recognition_resnet_model_v1datbz2</p> <p>https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html</p>
내주 계획	<p>데이터의 단순성을 줄여 overfitting을 해결할 수 있는 방법을 모색한다.</p>

년 월 일

지도교수

(인)