





## 2024-1학기 창의학기제 주간학습보고서 (2주차)

창의과제	세종대학교 집현캠퍼스를 개선시킨 웹서비스 개발				
이름	이지민	학습기간	3월4일 ~ 3월 15일		
학번	23012127	학습주차	2	학습시간	3
학과(전공)	인공지능	과목명	자기주도 창의전공1	수강학점	3
※ 수강학점에 따른 회차별 학습시간 및 10주차 이상 학습 준수					
금주 학습목표	사진에서 얼굴만을 따주는 객체인식 모델을 찾고 학습시키기				
학습내용	<p>저번주에는 노트북으로 찍은 이미지를 가지고 pytorch에서 pre-train된 MobileNetV3를 가져와 분류 모델을 만들어 학습시켜 보았다. 이번 주에는 배경에 의한 불필요한 데이터의 발생을 줄이기 위해 이미지에서 얼굴 부분만 가져와 학습시켜 보기로 하였다.</p> <p>얼굴 데이터를 가진 input data를 만들기 위해 조원의 사진을 가져와 opencv의 cascade classifier를 활용하여 얼굴 부분만 자른 후 pytorch의 imageTransform 클래스를 활용해 픽셀의 크기를 일정하게 맞춰 주었다.</p> <p>&lt;image detection(cascade classifier) code&gt;</p> <pre>image = cv2.imread(image_path) if image is None:     continue # opencv cascade 라이브러리를 활용해 왼쪽 상단 모서리 좌표(x,y)와 감지된 # 얼굴 사각형의 너비와 높이(w,h)를 얻음 gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY) faces = face_cascade.detectMultiScale(gray, 1.1, 4) for (x, y, w, h) in faces:     # 얻어낸 x,y,w,h를 가지고 그 부분만 자름     padding_w = int(w * padding_factor)     padding_h = int(h * padding_factor)     x_pad = max(x - padding_w, 0)     y_pad = max(y - padding_h, 0)     w_pad = min(w + 2 * padding_w, image.shape[1] - x_pad)     h_pad = min(h + 2 * padding_h, image.shape[0] - y_pad)     #컬러 저장     #face_img_with_padding = image[y_pad:y_pad+h_pad,     #                               x_pad:x_pad+w_pad]     #흑백 저장     face_img_with_padding = gray[y_pad:y_pad+h_pad, x_pad:x_pad+w_pad]     save_path = os.path.join(save_folder_path, filename)     cv2.imwrite(save_path, face_img_with_padding)</pre>				



	<p>cascade classifier 코드는 향후 재사용할 것을 대비해 간단한 모듈화를 진행하였다.</p> <div style="display: flex; justify-content: space-around; align-items: center;">   </div> <div style="display: flex; justify-content: space-around; align-items: center;"> <span>&lt;cascade classify 전&gt;</span> <span>&lt;cascade classify 후&gt;</span> </div> <p>이와 같이 전처리한 이미지를 가지고 이전에 만들었던 딥러닝 모델을 학습시켜 보았다.</p> <p>향후 모델 성능 향상을 위해 이미지를 feature scaling하는 코드를 추가하였다.</p> <p>&lt;feature scaling code&gt;</p> <pre>file_path = os.path.join(folder_path, filename) image = cv2.imread(file_path) image = image.astype(np.float32) / 255.0 mean = np.mean(image) std = np.std(image) standardized_image = (image - mean) / std standardized_image = np.clip(standardized_image * 255.0, 0, 255).astype(np.uint8) cv2.imwrite(file_path, standardized_image)</pre>
학습방법	opencv document를 통해 cascade classifier에 대해 알아보고 인터넷에서 간단 예제를 찾아 활용해 간단 모듈로 만들어 활용했다.
학습성과 및 목표달성도	수치상으로는 모델의 정확도가 좀 더 상승하였다. 하지만 아직 label에 존재하지 않는 data를 넣었을 때 여러 label에 대한 softmax probabilities가 너무 높았다.
참고자료 및 문헌	<p>cascade classifier reference:  <a href="https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html">https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html</a>            cascade classifier 예시 코드: <a href="https://deep-learning-study.tistory.com/244">https://deep-learning-study.tistory.com/244</a></p>
내주 계획	<p>현재 모델을 얼굴의 특징을 학습시킨 모델을 사용하는 것이 아니므로 정확도를 더 높이기 위해서는 얼굴을 특징을 학습한 모델을 사용하거나 더 많은 데이터를 모아 얼굴의 특징을 학습시키는 방법을 강구한다.</p> <p>혹은 해당 모델이 분류모델인 이상 label에 존재하지 않는 데이터를 넣었을 때 높은 softmax probabilities가 나올 수 밖에 없다. 이는 얼굴인식 프로그램에 큰 문제를 야기한다. 이를 해결하기 위한 방법도 찾아본다.</p>

년 월 일

지도교수

(인)