



2024-1학기 창의학기제 주간학습보고서 (5주차)

창의과제	세종대학교 집현캠퍼스를 개선시킨 웹서비스 개발				
이름	이지민	학습기간	5월 1일 ~ 5월 9일		
학번	23012127	학습주차	5	학습시간	3
학과(전공)	인공지능	과목명	자기주도 창의전공1	수강학점	3
※ 수강학점에 따른 회차별 학습시간 및 10주차 이상 학습 준수					
금주 학습목표	앞으로 웹사이트에 모델을 넣을 것을 대비하고 현재 모델이 실전에서 잘 활용할 수 있는지 확인하기 위해 웹캠으로 이미지를 받아서 학습된 모델을 통해 얼굴 인식을 실행해본다.				
학습내용	<p>먼저 프로그램에서 웹캠을 가져와 이미지를 얻기 위해 파이썬의 cv2 라이브러리에서 VideoCapture라는 메서드를 사용하였다.</p> <p><웹캠으로 이미지를 가져오는 코드></p> <pre>import cv2 import time webcam = cv2.VideoCapture(0) if not webcam.isOpened(): print("웹캠을 열 수 없습니다.") exit() time.sleep(1) #카메라를 열고 iso, 심도 등을 맞추는 시간을 주기 위해 정지 status, frame = webcam.read() # frame 변수에 이미지를 받음 givenImage_path = './givenImage' filename = './givenImage/givenImage.jpg' if status: cv2.imwrite(filename, frame) # 이미지를 filename에 저장 else: print("프레임을 캡처할 수 없습니다.") webcam.release() cv2.destroyAllWindows()</pre> <p>웹캠에서 얻은 이미지를 이전에 만들어놓은 hercascale 알고리즘을 이용한 얼굴만 가져오는 코드를 이용해 얼굴만 가져온다.</p> <pre>import face_detaction as fd folder_path = fd.save_detect_image(givenImage_path)</pre> <p>웹캠으로 얻은 이미지와 사용자의 이미지를 모델 입력에 맞도록 리사이즈해주고 토치텐서로 변환한다.</p> <pre>anchorImage = Image.open('./Dog_v_face/WIN_20240312_16_17_38_Pro.jpg') givenImage = Image.open('./givenImage_face/givenImage.jpg') transform = transforms.Compose([transforms.Resize((120, 120)),</pre>				



	<pre> transforms.ToTensor(),]) anchorImage = transform(anchorImage).unsqueeze(0) givenImage = transform(givenImage).unsqueeze(0) 저장학 학습모델을 불러오기 위해 이전에 만들었던 siameseNetwork를 모듈로 만들어 불러오고 학습된 모델도 불러온다. from siamese_network import SiameseNetwork model = torch.load('shinchan.pt') 코드의 재활용을 위해 siamese network 모델의 연산과 거리차를 구해 theshold로 사용자의 얼굴인지를 판단하는 코드를 모듈화시켰다. <judge_function.py> import torch from tqdm import tqdm_notebook as tqdm import torch.nn.functional as F def judge_model(model, anchorImage, givenImage, device): anchor = anchorImage.to(device) compare = givenImage.to(device) output1, output2 = model(anchor, compare) distance = F.pairwise_distance(output1, output2) if distance < 0.2: print("pass,%.3f"%(distance)) else: print("fail,%.3f"%(distance)) 위의 모듈을 불러와 최종적으로 얼굴을 판단하도록 한다. <main.py에서 사용자의 얼굴인지 판단하는 코드> from judge_function import judge_model device = "cuda:0" if torch.cuda.is_available() "mps" elif torch.backends.mps.is_available() else "cpu") judge_model(model, anchorImage, givenImage, device) </pre>
학습방법	<p>pytorch에서 제공하는 튜토리얼에서 모델을 저장하고 불러오는 방법을 익혀 활용하였다. 코드 구현중 발생하는 사소한 오류는 오류를 읽고 해결하였다.</p> <p>웹캠으로 이미지를 가져오는 방법은 인터넷 자료와 chatgpt의 예시코드를 활용하였다.</p>
학습성과 및 목표달성도	<p>완성한 코드를 실행시켜 라이브로 얼굴인식을 시도해 보았다. 여러번 코드를 실행해본 결과 사용자의 얼굴인데도 불구하고 사용자의 얼굴이 아니라고 인식하는 경우가 대부분이었다. 저번 주차에서 예상한대로 모델이 단순한 학습 이미지에 오버피팅되었다고 생각하게 되었다.</p>
참고자료 및 문헌	<p>https://pytorch.org/tutorials/beginner/basics/saveloadrun_tutorial.html</p> <p>https://pytorch.org/docs/stable/generated/torch.load.html</p> <p>https://velog.io/@bangsy/Python-OpenCV3</p>
내주 계획	<p>현재 가지고 있는 데이터와 시간을 가지고 오버피팅문제를 해결하고 적정 수준의 얼굴인식 모델을 만들 방법을 모색한다.</p>



세종대학교

지도교수

(인)