

Progress Report

24조 신중환, 이상호, 최병주

1. 문제 정의

현대인들의 PC와 스마트폰 사용 시간 증가는 거북목, 라운드 숄더 등의 만성적인 자세 불균형을 유발한다. 이것이 장기간 지속되는 경우 목/허리 디스크 등의 질환과 건강 문제로 이어질 수 있다. 많은 PC 이용자들이 해당 문제를 인지하고 있으나 습관에서 비롯된 문제이기 때문에 자발적인 해결에 어려움을 겪기 쉽다. 본 프로젝트는 웹캠을 통해 입력받은 사용자의 자세를 머신러닝 방법론을 적용하여 실시간으로 분석하고 잘못된 자세를 인지시켜서 자세 문제의 근본적인 해결을 목표로 한다.

2. 시도한 방법론

2-1. 데이터셋 구축

- 데이터 수집

모델이 특정 환경에만 치우치지 않고 다양한 상황에서 잘 작동하도록 여러 경로를 통해 데이터를 수집하는 혼합 방식을 사용하였다.

① Kaggle 공개 데이터셋 활용

공신력 있는 데이터를 확보하기 위해 Kaggle(www.kaggle.com)을 활용했다. 여기서 "posture" 키워드로 검색하여 공개된 이미지 데이터셋을 확보했다. 이는 우리 모델이 자세의 기본 특징을 학습하는 데 좋은 기반이 될 것이라 판단된다.

② Google 이미지 스크래핑

Kaggle 데이터만으로는 데이터가 충분하지 않고 프로젝트에서 판별하고자 하는 거북목이나 라운드 숄더 같은 특정 자세 이미지가 부족하다. 그래서 Google Colab 환경에서 `simple_image_download` 같은 스크래핑 라이브러리를 사용해 'forward head posture', 'text neck', 'slouching computer', 'good posture'의 키워드로 이미지를 추가 수집했다.

스크래핑한 이미지에는 학습에 부적합한 다이어그램, 만화 등 관련 없는 사진들이 많이 섞여있었다. 그래서 모든 이미지를 직접 눈으로 검토하며, 사람의 전신 또는 상반신이 명확하게 나온 실제 사진만 골라내는 수동 정제 과정을 거쳤다.

③ 데이터 라벨링 및 구성

수집한 모든 이미지는 좋은 자세와 나쁜 자세 두 가지 클래스로 분류했다. 이 과정을 효율적으로 관리하기 위해 'good' 폴더와 'bad' 폴더를 만들고, 수집한 이미지들을 기준에 맞게 나누어 저장했다.

모델 학습을 위한 라벨링은 이 폴더 구조를 기반으로 자동화했다. 전처리 스크립트가 'good' 폴더 및 그 하위 폴더들 안의 모든 이미지에는 좋은 자세를 뜻하는 라벨 1을, 'bad' 폴더 및 그 하위 폴더 안의 모든 이미지에는 나쁜 자세를 뜻하는 라벨 0을 자동으로 부여하도록 설정했다. 이 방식 덕분에 데이터 소스에 관계없이 일관된 라벨을 줄 수 있었고, 나중에 데이터를 더 추가할 때도 폴더에 넣기만 하면 되어 편리하게 관리할 수 있다.

- 데이터 전처리

① 특징 추출

Google에서 제공하는 MediaPipe Pose 라이브러리를 활용하여 각 이미지에서 33개의 주요 신체 관절 위치를 감지하고, 이들의 3D 좌표를 추출하였다.

전체 프로세스는 다음과 같다. 먼저 os.walk 기능으로 good, bad 폴더와 그 하위 폴더들을 순회하며 이미지 파일을 찾았다. 각 이미지 파일을 cv2.imread로 불러온 뒤 MediaPipe Pose 모델에 입력하여 랜드마크(관절 정보)를 감지했다. 마지막으로, 추출된 33개 랜드마크 각각의 x, y, z 좌표와 visibility 값을 뽑아 (33, 4) 형태의 NumPy 배열로 변환해 저장했다. 이때 visibility는 MediaPipe 모델이 감지한 랜드마크의 신뢰도를 의미하며, 1에 가까울수록 해당 랜드마크(관절)이 이미지에 존재할 확률이 높음을, 0에 가까울수록 이미지에 나타나지 않았음을 의미한다.

② 데이터 검증 및 오류 처리

처음 스크립트를 실행했을 때, 일부 이미지에서 랜드마크 추출이 아예 실패하거나(None 반환), 예상과 다른 형태의 데이터가 반환되어 ValueError가 발생하는 문제가 있었다.

이 문제를 해결하기 위해, 랜드마크를 데이터셋에 추가하기 전에 검증 단계를 추가했다. 랜드마크가 None이 아니어야 하고, NumPy 배열 형태여야 하며, 그 shape이 정확히 (33, 4)여야 한다는 세 가지 조건을 모두 만족하는 경우에만, 해당 랜드마크 배열은 data_X 리스트에, 라벨은 data_Y 리스트에 동시에 추가하도록 했다. 이렇게 함으로써 데이터와 라벨이 1:1로 정확하게 매칭되도록 보장할 수 있었다.

- 전처리 결과

① X_dataset.npy (특징 데이터)

이 파일은 (1223, 33, 4) Shape을 가진다. 이는 총 1223개의 성공적으로 처리된 이미지 샘플을 의미하며, 각 샘플은 33개의 랜드마크를, 각 랜드마크는 [x, y, z, visibility] 4개의 값을 가지고 있다는 뜻이다.

② Y_dataset.npy (라벨 데이터)

이 파일은 (1223,) Shape를 가지며, X_dataset.npy의 1223개 샘플 각각에 1:1로 매칭되는 라벨(0: 나쁜 자세, 1: 좋은 자세)이 저장되어 있다.

2-2. 이진 분류 모델

본 자세 판별 문제는 “좋은 자세”와 “나쁜 자세”를 분류하는 이진 분류 문제이다. 이러한 이진 분류 문제를 위한 다양한 모델이 존재하며, 본 보고서에서는 Logistic Regression 모델과 Multi-Layer Perceptron 모델 두 모델을 활용하였으며, 모델의 특징은 다음과 같다.

- Logistic Regression 모델

Logistic Regression 모델은 입력된 피쳐 공간에서 가장 잘 분리되는 평면을 찾기 위해 시도한다. 이 평면을 경계로 좋은 자세(1) 특징 데이터와 '나쁜 자세(0) 특징 데이터가 나뉘게 된다. 모델은 132개의 특징에 가중치를 곱하고 합산한 값(Z)을 시그모이드 함수에 넣어 0과 1 사이의 확률값을 출력한다. 이 확률이 0.5를 넘으면 '좋은 자세(1)'로, 0.5 이하면 '나쁜 자세(0)'로 최종 판별한다.

장점: 모델의 구조가 단순하여 학습 속도가 매우 빠르다. 또한, 학습된 가중치(계수)를 통해 입력 피쳐들 중 어떤 특징이 '좋은 자세' 판별에 긍정적 또는 부정적 영향을 미치는지 직관적으로 해석하기 용이하다.

한계: 본질적으로 선형 모델이므로, 특징들 간의 복잡한 상호작용이나 비선형 관계를 학습하는 데 명확한 한계가 있다.

• 구현 상세

주요 라이브러리	Scikit-learn (sklearn)
모델 구현	sklearn.linear_model.LogisticRegression
모델 설정	solver='liblinear' 옵티마이저 사용 (비교적 규모가 작은 데이터셋의 이진 분류 문제에서 안정적이고 우수한 성능을 보임)
학습	solver가 훈련 데이터셋 전체를 대상으로 설정된 최대 반복 횟수(기본값 100) 내에서 손실 함수가 수렴할 때까지 반복적으로 가중치를 업데이트

- Multi-Layer Perceptron(MLP) 모델

MLP (다층 퍼셉트론) 모델은 피쳐 공간에서 단순한 평면이 아닌, 더 복잡하고 유연한 비선형(non-linear) 경계면을 학습하려 시도한다. 이 모델은 132개의 특징을 여러 개의 은닉층에 단계적으로 통과시킨다. 각 은닉층은 입력된 특징들을 relu 와같은 활성화 함수를 이용해 새롭게 조합하며 학습한다. 이 고차원 특징들을 조합한 값을 시그모이드 함수에 넣어 0과 1 사이의 확률값을 출력한다. 이 확률이 0.5를 넘으면 '좋은 자세(1)'로, 0.5 이하면 '나쁜 자세(0)'로 최종 판별한다.

장점: 은닉층을 통과하는 과정에서 입력 피쳐들이 새롭게 조합되며, 모델이 고차원적이고 추상적인 특징을 스스로 학습할 수 있다. 이로 인해 로지스틱 회귀가 찾지 못하는 복잡한 비선형 결정 경계를 생성하여 더 높은 분류 성능을 달성할 잠재력을 가진다.

한계: 모델이 복잡하고 학습할 파라미터가 많아, 상대적으로 적은 데이터에서는 훈련 데이터에만

과도하게 최적화되는 과적합(Overfitting) 위험이 높다. (이를 완화하기 위해 Dropout 규제를 적용하였다.) 또한, 수많은 가중치가 복합적으로 작용하므로 로지스틱 회귀와 달리 결과를 직관적으로 해석하기 어렵다.

• 구현 상세

주요 라이브러리	TensorFlow 2.19 (tf.keras)
모델 구현	tf.keras.models.Sequential 레이어 구성: tf.keras.layers.Dense, tf.keras.layers.Dropout, tf.keras.layers.Input
모델 설정	입력층 (12개 뉴런) → 은닉층 1 (Dense, 64개 뉴런, relu 활성화) → 규제 1 (Dropout, 30%) → 은닉층 2 (Dense, 32개 뉴런, relu 활성화) → 규제 2 (Dropout, 30%) → 출력층 (Dense, 1개 뉴런, sigmoid 활성화)
학습	optimizer='adam': 경사 하강법 알고리즘 loss='binary_crossentropy': 이진 분류 문제의 확률 오차를 계산 epochs=50, batch_size=16, EarlyStopping patience = 5

즉, Logistic Regression 모델과 MLP 모델은 모델의 복잡성에 있어 양 극단에 존재하는 모델이라고 볼 수 있다. 본 보고서에서는 이 두 모델을 학습 데이터에 각각 반영하여 모델의 복잡도에 따른 분류 성능 차이를 확인하고자 한다.

2-3. 피쳐 엔지니어링

학습 데이터인 33x4 행렬에는 MediaPipe가 인식한 전신 관절들의 위치 정보와 가시성 점수가 담겨 있다. MLP과 같은 복잡한 모델은 충분한 데이터가 주어진다면 이 고차원 공간 내에 내재된 패턴을 스스로 학습하여 좋은 자세와 나쁜 자세 사이의 경향성을 파악할 잠재력을 가진다.

그러나 이러한 원본 좌표 값은 자세의 본질적인 특성 외에도 카메라의 원근, 피사체의 고유한 체형, 촬영 거리 등 외적 변수에 민감하게 의존한다. 이는 모델이 자세 판별의 핵심 규칙이 아닌, 데이터 수집 환경의 우연적 편향에 과적합될 위험을 높인다.

이에 본 연구에서는 도메인 지식에 기반한 특징 공학을 적용하여, 모델이 자세라는 현상에 더 직접적으로 집중할 수 있도록 특징 공간을 재설계하였다.

관절 각도는 피사체의 크기나 위치에 큰 연관을 가지지 않는 요소이며, 사람의 자세를 표현하는 결정적인 지표이다. 본 보고서에서는 '귀-어깨-골반'으로 이어지는 각도, 팔꿈치 각도, 어깨-팔 각도를 연산한 피쳐를 모델에 입력하였다.

이러한 가공된 특징들은 원본 데이터의 노이즈를 줄이고, 모델이 일반화(generalization) 성능이 높은 결정 경계를 더 효율적으로 학습할 수 있도록 유도한다. 즉, 모델이 132개의 개별 좌표 간의 복잡한 관계를 처음부터 탐색하도록 맡기는 대신, 자세 판별과 직접적으로 연관된 소수의 핵심 특징을 제공함으로써 더 견고하고 정확한 분류 성능을 기대할 수 있다.

또한 이러한 피쳐의 영향을 확인하기 위해, raw 데이터를 피쳐로 활용하여 학습한 모델과 관절 각도

를 피쳐로 활용한 모델 사이 직접적인 비교를 수행하였다. raw 데이터의 경우 자세 판별과 무관한 부위 데이터(하반신 등)가 포함되어있으며 신체 조건에 따른 여러 노이즈가 발생하기때문에, 추가 계산과 여과가 거처진 관절 피쳐 사용 모델의 성능이 더 우수할 것으로 예상된다.

3. 중간 결과 및 해석

3-1. 학습 및 검증 결과

앞서 언급한 대로 동일한 데이터셋에 대해 Logistic Regression 모델과 Multi Layer Perceptron 모델에 대해, 피쳐 엔지니어링 반영 / 미반영 두 가지 버전으로 학습 및 검증을 시행하였으며, 검증 데이터 비율은 20%로 설정하였다.

이에 대한 최종 검증 정확도는 다음과 같았다.

	Logistic Regression	Multi Layer Perceptron
Raw data(33 x 4)	0.9469	0.9388
Angle feature	0.6939	0.7918

검증 정확도 수치로만 보았을때, Logistic Regression에 Raw data로 학습시킨 모델의 정확성이 더 우수하다는 결과가 나왔다. 이는 기존 예상에 어긋나는 결과이기 때문에, 이에 대한 분석이 필요하다고 판단하여 추가 검증을 수행하였다.

3-2. 재검증

데이터셋이 대부분 측면 이미지로 구성되어 있기 때문에 추가 탐색과 촬영을 통해 good/ bad posture 이미지 각 4개를 확보하여 재검증을 수행하였다. 재검증 정답률은 다음과 같았다.

	Logistic Regression	Multi Layer Perceptron
Raw data(33 x 4)	0.5	0.5
Angle feature	0.75	0.75

적은 데이터로 이루어진 실험이었으나, Raw data를 입력한 두 모델 모두 새로 입력된 모든 데이터를 "bad posture"로 분류하였다. 이는 초기 학습 결과의 정확도 수치와 모순되며 Raw data로 학습된 모델이 자세 정보에 과적합 되어있다는 판단 증거가 된다. 추가로 분석한 한계와 문제점은 다음 문단에서 다룬다.

4. 개선이 필요한 부분 및 개선 계획

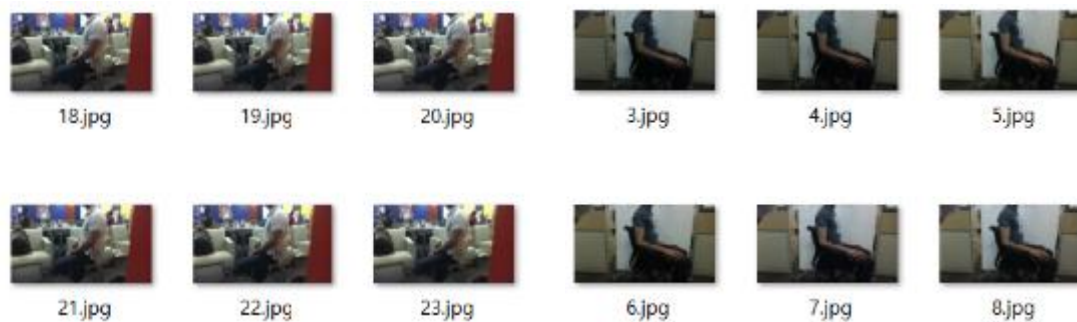
4-1. 데이터 수집의 한계점과 개선 방향

현재 학습 데이터셋을 위한 이미지를 수집하는 과정에서 문제가 존재했다. 프로젝트의 최종 목표는 웹캠을 통한 실시간 자세 판별로, 주로 사용자의 정면 모습을 기준으로 한다. 하지만 'posture'나 'forward head posture' 같은 키워드로 찾은 이미지들은 대부분 자세 교정 방법을 설명하기 위한 측면 사진들이었다.

이렇게 정면 데이터가 부족한 상태로 모델을 학습시키면, 실제 웹캠(정면) 환경에서 사용할 때 판별이 거의 불가능하다. 이 데이터 편향성 문제를 해결하기 위해, 우리는 두 가지 방향으로 데이터셋을 보강할 계획이다. 첫째는 정면을 명시하는 키워드로 이미지를 추가 스크래핑하는 것이고, 둘째는 팀원들이 직접 웹캠 앞에서 좋은 정면 자세와 나쁜 정면 자세를 촬영하여 실제 사용 환경에 가장 적합한 데이터를 확보하는 것이다.

또한 현재 수집한 측면 이미지에도 문제점을 발견하였다. 수집한 이미지의 대부분이 앉아있는 사람을 측면에서 촬영한 영상을 프레임 단위로 추출한 영상(하단 예시)이다. 이러한 이미지셋의 경우 거의 동일한 위치와 자세의 이미지가 수십여장 반복되어 해당 이미지의 좌표에 과적합될 위험이 높다. Raw data를 입력한 모델의 검증 정확도가 높은 반면 새로운 이미지에 대해 적절히 반응하지 못한 이유 역시 입력된 검증 데이터 중 일부 또는 다수가 학습 데이터와 유사하거나 거의 동일한 관절 좌표를 가지고 있었기 때문일 가능성이 높다.

언급한 바와 같이, 남은 기간동안 집중해야 할 개선 방향은 데이터셋 구축이다. 팀원이 모두 참여하여 스크래핑, 기존 데이터셋 탐색, 직접 촬영 등 다양한 방법으로 최대한 많고 다양한 이미지를 구할 계획이다. 또한 검증/테스트 데이터셋을 학습 데이터셋과 완전히 무관한 이미지로 새로 구축하여 검증 및 테스트의 신뢰성을 끌어올리고자 한다.



- Good posture 이미지 반복된 예시



- Bad posture 이미지 반복된 예시



- 다양한 이미지 예시

4-2. 모델 선정 및 피쳐 엔지니어링 개선 방향

분류를 위한 모델로 선정된 logistic regression 모델과 MLP 모델은 모델 복잡도 면에서 양극단에 있는 모델이다. 때문에 이 두 모델로 현재 문제에 적합한 복잡도의 모델을 선정하는 것은 비현실적이다. 향후 분류 성능을 높이기 위해 Random forest 등의 Decision tree 기반 모델이나 Support Vector Machine(SVM)과 같은 추가 분류 모델을 실험에 포함하고자 한다.

또한 수행 과정에서, 각도 등의 피쳐를 모델의 입력으로 활용하는 경우 raw 데이터를 피쳐로 활용하는 것보다 새로운 상황에 더 적절히 대처함을 확인했기 때문에, 더 다양한 피쳐를 적용 및 실험해볼 예정이다. 특히 다양한 각도의 이미지를 구별해낼 능력이 필요하기 때문에, 현재 카메라를 기준으로 사람의 몸이 어느쪽을 향하고 있는지와 관련된 피쳐를 구현하면 더 좋은 성능의 분류 모델을 구축할 수 있으리라 기대된다.