

Final Report

24조 신종환, 이상호, 최병주

1. 문제 정의

현대인들의 PC와 스마트폰 사용 시간 증가는 거북목, 라운드 솔더 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. 적용한 방법론

2-1. 데이터셋 구축

- 데이터 수집

프로젝트 중간 보고서까지는 Kaggle이나 구글 이미지 스크래핑을 통한 데이터 수집을 고려했으나, 인터넷상의 데이터는 주로 자세 교정 설명을 위한 측면 사진 위주라는 한계가 있었다. 본 프로젝트의 목표인 웹캠 기반 실시간 자세 교정은 주로 사용자의 정면 또는 반측면을 비추기 때문에, 기존 데이터셋으로는 실제 환경에서의 정확도를 보장하기 어렵다고 판단했다.

이에 따라 외부 데이터를 배제하고, 팀원들이 직접 영상을 촬영하여 학습 데이터를 구축하는 방식을 선택했다.

① 촬영 및 프레임 추출

팀원들이 웹캠 환경에서 직접 바른 자세와 나쁜 자세를 촬영했다. 데이터의 다양성을 확보하기 위해 정면뿐만 아니라 다양한 각도에서 촬영을 진행했으며, 의도적인 거북목, 라운드 솔더 등의 나쁜 자세와 이를 교정한 바른 자세를 번갈아 촬영하여 실제 사용 환경에 가장 적합한 데이터를 확보했다.

촬영된 영상 데이터는 학습에 사용하기 위해 프레임 단위의 이미지로 변환했다. 영상에서 일정 간격으로 프레임을 추출함으로써, 연속적인 동작 속에서 다양한 자세 이미지를 확보할 수 있었다.

② 데이터 구성

추출된 이미지들은 다음과 같은 폴더 구조로 정리했다.

good 폴더: 바른 자세 영상에서 추출한 이미지들이 담긴 하위 폴더들로 구성된다. (예: good/1, good/2...)

bad 폴더: 나쁜 자세 영상에서 추출한 이미지들이 담긴 하위 폴더들로 구성된다. (예: bad/1, bad/2...)

각 하위 폴더는 하나의 영상 촬영 세션을 의미하며, 유사한 조명과 배경, 동일 인물의 이미지가 모여 있는 단위가 된다.

③ 데이터 라벨링

수집된 모든 이미지 데이터는 바른 자세와 나쁜 자세라는 두 가지 클래스로 분류했다. 이 과정을 효과적으로 관리하기 위해, 프로젝트 디렉토리 내에 good 폴더와 bad 폴더를 생성하여 촬영된 이미지들을 각각의 기준에 맞게 나누어 저장했다.

모델 학습을 위한 데이터 라벨링은 이 폴더 구조를 기반으로 자동화하여 진행했다. 전처리 스크립트가 good 폴더 및 그 하위 폴더 내에 있는 모든 이미지에 대해 바른 자세를 의미하는 라벨 1을 할당하고, bad 폴더 및 그 하위 폴더 내의 모든 이미지에 대해서는 나쁜 자세를 의미하는 라벨 0을 할당하도록 설정했다.

이러한 방식은 별도의 메타데이터 파일을 관리할 필요 없이, 폴더 구분만으로 직관적인 데이터 관리가 가능하다는 장점이 있다. 또한, 추후 데이터를 추가할 때도 해당 폴더에 이미지를 넣기만 하면 전처리 과정에서 자동으로 올바른 라벨이 부여되므로 데이터셋 확장 및 관리가 매우 효율적이었다.

- 데이터 전처리

① 특징 추출

특징 추출 단계에서는 Google에서 제공하는 MediaPipe Pose 라이브러리를 사용했다. 각 이미지 프레임에서 33개의 주요 신체 랜드마크(관절)를 감지하고, 이들의 3D 좌표(x, y, z)와 가시성 정보를 추출하여 모델의 입력 데이터로 변환했다.

② 데이터 분할 및 처리 (Train/Test/Val)

단순히 모든 이미지를 섞어서 나눌 경우, 동일한 영상에서 추출된 아주 유사한 프레임들이 학습셋과 테스트셋에 동시에 들어가는 데이터 누수 문제가 발생할 수 있다. 이를 방지하기 위해 glob 라이브러리를 사용하여 폴더 단위로 데이터를 처리하고 분할했다.

전처리 프로세스는 다음과 같이 진행했다.

1. glob 함수를 사용하여 good과 bad 폴더 내의 하위 폴더 목록을 가져온다.
2. 각 클래스(good/bad) 별로 하위 폴더들을 순회하면서 데이터를 처리한다.
3. 데이터 분할의 공정성을 위해, 특정 하위 폴더를 지정하여 검증용(Val)과 테스트용(Test) 데이터로 따로 분류했다. 나머지 폴더들은 학습용(Train) 데이터로 사용했다.
4. 각 이미지에서 추출된 (33, 4) 형태의 랜드마크 배열을 해당되는 데이터셋 리스트(Train, Val, Test)에 추가했다.

③ 데이터 검증

프레임 추출 과정에서 흔들리거나 신체가 잘린 이미지가 포함될 수 있어 검증 단계를 적용했다. MediaPipe가 랜드마크를 감지하지 못해 None을 반환하거나, 데이터 형식이 NumPy 배열이 아니거나, shape이 (33, 4)가 아닌 경우 해당 데이터는 제외했다. 이 과정을 통해 학습 데이터의 품질을 균일하게 유지했다.

④ 전처리 결과

전처리 과정을 거쳐 최종적으로 학습(Training), 검증(Validation), 테스트(Test)의 세 가지 데이터셋 그룹을 생성했다. 각 그룹은 특징 데이터(X)와 라벨 데이터(Y)로 구성된다.

- Webcam Connect for Colab 구현

Google Colab은 클라우드 서버 환경에서 실행되므로 로컬 하드웨어(웹캠)에 cv2.VideoCapture로 직접 접근하는 것이 불가능하다. 이를 해결하기 위해 JavaScript와 Python 간의 양방향 통신 파이프라인을 구축하여 실시간 스트리밍을 구현했다.

JavaScript 웹캠 연결 구현: IPython.display 모듈을 활용해 JavaScript 코드를 실행하고, 브라우저의 navigator.mediaDevices API를 호출하여 사용자 웹캠 권한을 획득 및 제어한다.

데이터 전송 및 변환: 웹캠에서 캡처된 프레임은 Base64 문자열로 인코딩되어 Python 환경으로 전송되며, 이를 OpenCV 이미지 포맷(Numpy array)으로 디코딩하여 모델의 입력 데이터로 사용한다.

실시간 오버레이 시각화: Python에서 MediaPipe 랜드마크 추출 및 모델 추론을 수행한 뒤, 결과값(Good/Bad)과 골격 시각화 요소를 원본 프레임에 직접 렌더링하여 다시 브라우저로 전송함으로써 자연 없는 실시간 피드백을 제공한다.

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 모델, Multi-Layer Perceptron 모델, Random Forest 모델, SVM 모델로 총 4가지 모델로 실험하였다.

- 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 나누게 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
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=500, batch_size=16, EarlyStopping patience = 20

- Random Forest (RF) 모델

Random Forest는 다수의 결정 트리(Decision Tree)를 개별적으로 학습시킨 뒤, 각 트리의 예측 결과를 취합(Voting)하여 최종 결과를 도출하는 앙상블(Ensemble) 모델이다. 단일 결정 트리는 훈련 데이터에 과적합되기 쉽다는 단점이 있는데, Random Forest는 배깅(Bagging) 방식을 통해 서로 다른 데이터 샘플로 여러 트리를 학습시킴으로써 이러한 과적합 문제를 효과적으로 줄이고 일반화 성능을 높인다.

장점: 여러 트리의 의견을 종합하므로 잡음(Noise)에 강하고 안정적인 성능을 보인다. 또한, 별도의

정규화 과정 없이도 비교적 좋은 성능을 내며, 어떤 특징(Feature)이 자세 분류에 중요한지 '특성 중요도(Feature Importance)'를 파악할 수 있다는 큰 장점이 있다.

한계: 수백 개의 트리를 구축하고 예측해야 하므로, 단일 모델(로지스틱 회귀 등)에 비해 학습 및 예측 속도가 느릴 수 있고 모델의 메모리 사용량이 크다.

- 구현 상세

주요 라이브러리	Scikit-learn (sklearn)
모델 구현	sklearn.ensemble.RandomForestClassifier
모델 설정	n_estimators=100(100개의 트리 사용), n_jobs=-1(병렬 처리)
학습	학습 데이터셋에 대해 부트스트랩 샘플링을 적용하여 각 트리를 독립적으로 학습시키고, 다수결 투표를 통해 최종 클래스를 예측

- Support Vector Machine (SVM) 모델

SVM은 데이터가 존재하는 공간에서 두 클래스(좋은 자세 vs 나쁜 자세)를 나누는 가장 넓은 도로(Margin)를 가진 결정 경계(Hyperplane)를 찾는 모델이다. 단순히 데이터를 나누는 것을 넘어, 데이터와 경계 사이의 거리를 최대화함으로써 새로운 데이터가 들어왔을 때도 안정적으로 분류할 수 있도록 한다. 특히 본 프로젝트처럼 특징 차원이 높은 경우(132차원 등)에도 효과적으로 작동하며, 커널 기법(Kernel Trick)을 활용해 비선형 데이터 분포도 유연하게 분리할 수 있다.

장점: 마진을 최대화하는 원리 덕분에 적은 양의 데이터로도 일반화 능력이 뛰어나며, 고차원 데이터 처리에 매우 효율적이다. 특히 'Angle + Plane'과 같은 기하학적 특징이 잘 정의된 경우 매우 높은 정확도를 기대할 수 있다.

한계: 데이터 전처리(스케일링)에 매우 민감하여 반드시 정규화 과정이 선행되어야 한다. 또한, 데이터 샘플의 개수가 매우 많아질 경우 학습 속도가 현저히 느려질 수 있다.

- 구현 상세

주요 라이브러리	Scikit-learn (sklearn)
모델 구현	sklearn.svm.SVC
모델 설정	kernel='rbf'(비선형 분리를 위한 방사 기저 함수 커널 사용), probability=True(확률값 출력 활성화), StandardScaler 적용 필수
학습	StandardScaler로 데이터의 특징 스케일을 맞춘 후, 클래스 간 마진을 최대화 하는 최적의 초평면을 탐색

2-3. 피처 엔지니어링

학습 데이터인 33x4 행렬에는 MediaPipe가 인식한 전신 관절들의 위치 정보와 가시성 점수가 담겨

있다. 이러한 원본 좌표 값은 자세의 본질적인 특성 외에도 카메라의 원근, 피사체의 고유한 체형, 촬영 거리 등 외적 변수에 민감하게 의존한다. 이는 모델이 자세 판별 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. 이전 progress report에서 얻은 결론을 기반으로, 다양한 피쳐들에 대한 구현과 실험을 수행하였다. 각 피쳐들은 가시성 점수를 기반으로 연산한 피쳐 신뢰도값을 가진다.

feature 1. Normalized Landmark Coordinate 기존 Landmark raw coordinate를 활용했을 때 외적 변수에 의한 과적합이 발생하였다. 이를 해결하기 위해 공간에 대한 정규화를 수행하였다. 좌우 어깨의 중점을 원점(0,0,0)으로 설정하여, 피사체가 화면 어디에 위치하든 동일한 기준점을 갖도록 신체 좌표 전체를 이동시켰다. (Centering) 어깨 너비를 기준으로 모든 랜드마크 좌표를 나누어 scale을 정규화하였다. (Scale normalization) 이를 통해 피사체의 체격이나 카메라 줌인/줌아웃 상황과 무관하게 신체의 구조적 형태(Pose Structure)만을 학습하도록 설계하였다.

feature 2. Joint Angles 관절 각도는 피사체의 크기나 위치에 큰 연관을 가지지 않는 요소이며, 사람의 자세를 표현하는 결정적인 지표이다. progress report 때와 동일하게 '귀-어깨-골반'으로 이어지는 각도, 팔꿈치 각도, 어깨-팔 각도를 연산한 피쳐를 설계하였다.

feature 3. Plane 단순한 각도 정보만으로는 감지하기 어려운 신체의 3차원 회전과 방향성을 포착하기 위해, 주요 신체 부위가 형성하는 평면의 법선 벡터(Normal Vector)를 산출하였다. 양쪽 눈과 코를 잇는 평면의 수직 벡터를 통해 고개가 정면을 향하는지, 혹은 아래나 옆으로 회전했는지를 입체적으로 분석하며, 양쪽 어깨와 코를 잇는 평면의 벡터를 통해 상체의 굽힘 정도나 비틀림을 측정한다. 자세의 안정성을 평가하는 보조 지표로 사용한다.

세 피쳐들로 총 4가지 피쳐 조합(Landmark only, Plane only, Angle+Plane, Angle+Plane+Landmark)을 구성했으며 해당 조합들을 각각 모델에 적용하여 피쳐의 성능을 확인하였다.

2-4. 최종 모델 선정

각 모델들과 feature set의 조합을 구성하여, 정량/ 정성적 검증 및 평가를 수행해 최종 모델을 선정 한다. 선정 과정 및 결과는 chapter 3에 서술되었다.

3. 최종 결과 및 해석

3-1. 정량 평가

앞서 언급된 피쳐들과 모델 간의 조합을 기반으로 최종 모델 선정을 위한 정량적 평가를 수행하였다. 정량 평가는 두 단계로 진행된다. 먼저 val dataset을 통한 정량평가로 가장 우수한 feature set + model 조합 후보군을 구성한 후, test dataset을 통한 평가와 실제 실행을 통해 판단한 정성적 평가를 통해 최종 모델을 선정한다.

Feature Set	MLP	Random Forest	SVM	Logistic Regression
Landmark	0.9528	0.9906	0.8774	0.8113

Angle + Plane	0.9575	0.9623	0.9953	0.8774
Plane Only	0.7925	0.8962	0.8255	0.5708
All Features	0.9717	0.9906	0.8726	0.9481

1) 특징(Feature)에 따른 성능 변화

단순 좌표(Landmark) 정보보다, 관절의 각도와 평면 벡터를 결합한 'Angle + Plane' 특징을 사용했을 때 SVM 모델에서 0.9953(99.53%)이라는 전체 실험 중 가장 높은 정확도를 기록하였다. 이는 SVM이 기하학적 경계를 나누는 데 있어 각도와 평면 정보가 매우 효과적임을 보여준다.

평면 정보(Plane Only)만 단독으로 사용했을 때는 모든 모델에서 성능이 하락했으며, 특히 Logistic Regression은 0.5708로 급격히 성능이 저하되었다. 이는 평면 벡터만으로는 자세를 분류하기에 정보량이 충분하지 않음을 보여준다.

모든 특징을 결합했을 때(All Features), MLP(0.9717)와 Logistic Regression(0.9481)은 단일 특징을 사용할 때보다 전반적으로 향상된 성능을 보이며 정보 결합의 효과를 입증했다.

2) 모델별 성능 비교

Random Forest는 'Plane Only'를 제외한 모든 특징 조합(Landmark, Angle+Plane, All Features)에서 0.96 이상의 높은 정확도를 유지하며, 특징의 종류에 크게 구애받지 않고 안정적인 성능을 보여주었다.

SVM은 'Landmark'나 'All Features'에서는 87%대의 성능을 보였으나, 'Angle + Plane' 특징과 결합했을 때 비약적인 성능 향상(99.53%)을 보여 피쳐 엔지니어링이 중요한 모델임을 확인하였다.

딥러닝 모델인 MLP는 정보량이 많아질수록(All Features) 성능이 0.9717까지 상승하여, 복합적인 특징 학습에 유리한 면모를 보였다.

최종적으로 가장 높은 정확도를 기록한 SVM (Angle + Plane) 모델, 다양한 특징 조합에서도 일관되게 높은 성능을 보인 Random Forest(All features) 모델, 이어서 높은 성능을 보인 MLP (All features) 모델을 최종 모델 선정을 위한 후보군으로 선정하였다. 후보군의 test accuracy는 다음과 같다.

MLP(All features)	Random Forest(All features)	SVM(Angle + Plane)
0.9200	0.9040	0.4720

높은 val accuracy를 보였던 SVM 모델이 test dataset에서 일관성을 잃은 모습을 보였다.

3-2. 정성 평가

후보군 모델을 실제 demo에 적용하여 실험해 사용성을 평가하였다. Random forest는 bad posture에 대해서는 민감한 판별력을 보였으나 good posture에 대해 보수적인 평가(대부분의 good posture에 50~60%)를 출력해 사용자에게 명확한 피드백을 제공하지 못한다는 아쉬움이 있었다. 반면 SVM과 MLP 모델의 경우 자세에 따라 0%~100% 사이의 명확한 피드백을 제공한다는 강점을 보였으나, SVM 모델의 경우 목만 구부러진 bad posture에 대해 판별하지 못함을 확인하였다. MLP 모델의 경우 추론 속도가 느려 영상에 지연이 발생하였다.

고무적인 사실은 세 후보군 모두 신체 각도에 무관하게 일관적인 출력을 보였다는 점이다. 그러나 피쳐 판정 기준에 어깨 좌표가 큰 비중을 차지하기 때문에 캠에 어깨가 담기지 않는 경우 판정이 불안

정한 모습을 보였다.

3-3. 최종 모델 선정

정량적 정확도와 정성적 사용성을 모두 고려하였을 때, 최종 demo에는 MLP(All features)가 적용되었다. MLP 모델의 경우 다양한 자세에 대해 0~100% 사이의 풍부하고 명확한 피드백을 제공하였으며, 촬영 각도에도 강건성을 가짐을 확인하였다.

후보군이었던 SVM 모델의 경우 확실한 bad posture에 대해서 높은 판정 능력을 보였으나, 모호한 자세에 대한 판단 능력이 떨어진다. 반면 Random Forest 모델의 경우 bad posture에 대한 확실한 피드백을 제공하는 반면 good posture 사이의 피드백에 약하며 bad posture에 대해 과하게 민감하게 반응하여 사용자에게 불편함을 줄 수 있다. 이는 임계값 보정 등의 후처리를 통한 해결 방안을 기대해볼 수 있다.

4. 향후 연구/ 개발 가능성

본 프로젝트를 통해 머신러닝 접근으로 사람의 자세의 올바름을 판단하는 분류 모델을 개발하였다. 최종 모델로 선정된 MLP 모델과 여러 피쳐들의 조합은 입력된 자세 피쳐와 라벨 사이의 강건한 매핑에 성공하였으며, 데이터셋 뿐만 아니라, 실제 사용시에도 우수한 자세 판별 능력을 보였다.

모델 성능 외에 눈에 띠는 성취는 데이터셋 효율이다. 본 모델을 학습하기 위해 팀원들이 촬영한 good/bad posture 영상은 약 30분 분량으로, 매우 적은 양의 데이터셋으로도 높은 판정 능력을 기록하였다. 이를 통해 다양한 각도 촬영과 피쳐 개발 등 데이터셋 과적합을 막기 위한 노력이 성과를 보였음을 입증하였다.

선정한 MLP 모델의 한계인 상대적으로 긴 추론시간은 테스트 환경과 모델의 연산 복잡도가 복합적으로 작용한 결과이다. 향후 상용화 단계에서는 로컬 환경에서 웹캠을 실행하게 하여 통신 딜레이를 제거하고 MLP 모델의 경량화를 거쳐 실시간성을 개선시킬 수 있다.

"바른 자세"라는 개념은 PC 사용에서만 적용되는 개념이 아니다. 자세는 일상생활 뿐 아니라 운동, 악기 연주 등 신체를 활용하는 활동에서 중요한 요소로 작용하여, 올바른 자세에 대한 판단과 나쁜 자세의 교정은 필수적이다. 본 프로젝트의 접근법을 헬스 트레이닝, 악기 레슨, 재활 교정 등 다양한 분야로 확장시켜 전문가의 지도 없이도 사용자가 스스로 자세를 교정하도록 활용할 수 있다. 결론적으로 본 프로젝트는 단일 카메라만으로도 고도화된 자세 분석이 가능함을 보이며, 디지털 헬스케어 분야에서 기계학습 접근법의 유용성을 확인했다는 의의를 가진다.