

A photograph of a man in a plank position on a gym floor. He is shirtless, wearing dark shorts and sneakers. The background shows gym equipment like cones and kettlebells. The image has a blue and purple color overlay.

Your Personal Training UPT

당신의 운동 자세를 분석해드립니다.

33번 프로젝트팀, 글루글루 박지현, 박성철, 윤경지, 전주현

계속되는 코로나 시대.

사람들은 '홈트' (홈 트레이닝, 집에서 하는 운동의 약자) 에 열광 중이다.



하지만 그 방법은 고작 **동영상을 따라 하는 것.**

운동에서 가장 중요한 것은 **'자세'** 인 만큼 동영상을 따라하는 것만으로는 올바른 자세를 취하며 운동을 하기 힘들다.

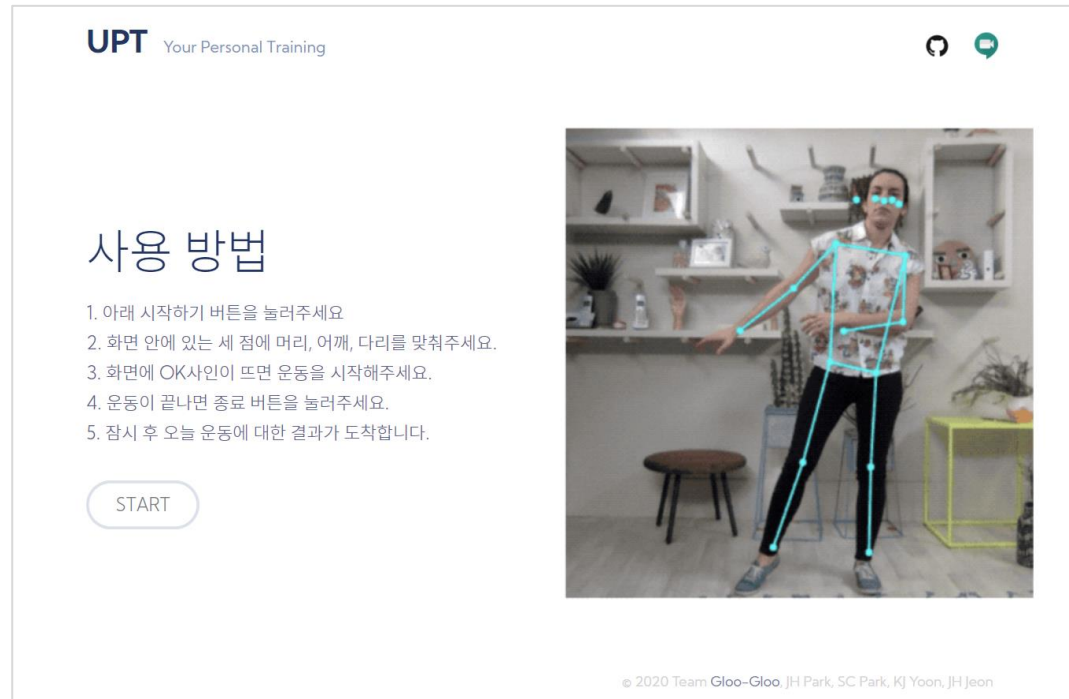
이를 도와줄 AI 기반 서비스.

Your Personal Training, UPT

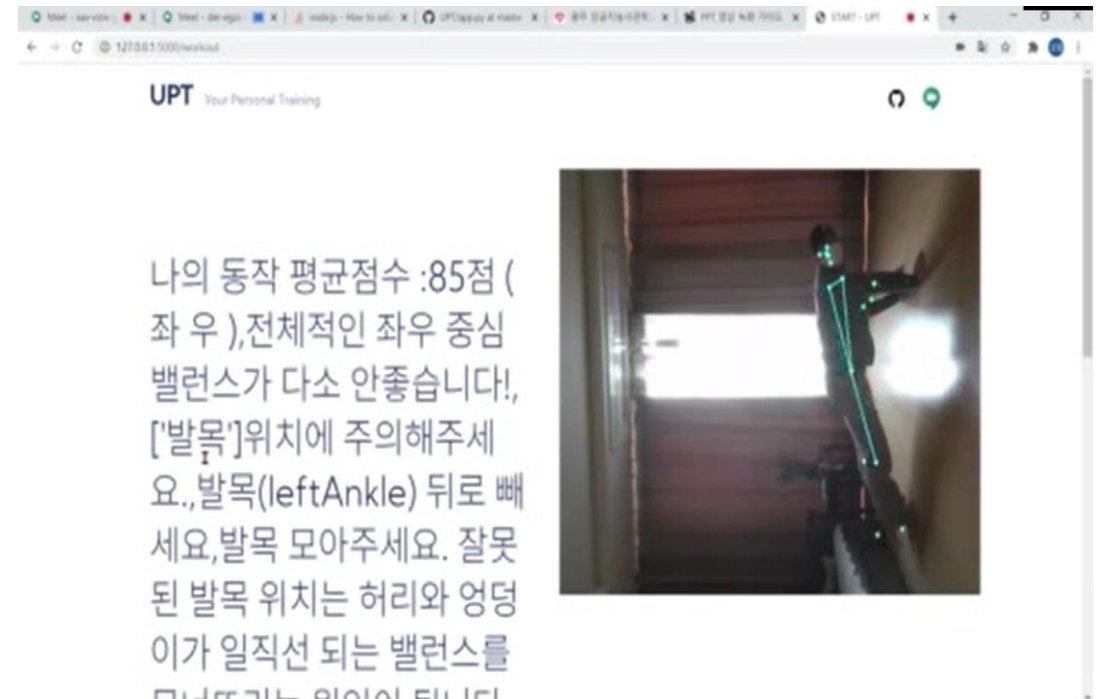
홈트를 할 때 자세를 자동으로 분석하여

좀 더 나은 운동을 할 수 있도록 운동 자세에 대한 보고서를 만들어 드립니다.

서비스 화면



사용 방법



결과 보고서

구현 과정 - (1) 머신 러닝을 위한 데이터 전처리

Google, Instagram, Bing, Shutterstock, 123RF 를 돌며 수집한
플랭크 자세 관련 사진 (약 2천개) 을 손수 라벨링



PIL 을 이용한 정방향 조정 (Posenet은 정방향 이미지만 넣을 수 있다.) ^{Google APIs} 및 Posenet 적용
모든 사진의 중심점을 맞추기 위한 알고리즘 개발

(Scale, Flip 등의 문제 일괄 해결)

Posenet을 통해 나온 모든 이미지의 좌표 값을
상대적인 0과 1사이의 좌표로 바꾸는 방식

구현 과정 - (2) **XGBoost** 모델 생성

이미지 전처리를 통해 나온 값들을 **XGBoost** 로 학습시켜준다.

	nose	leftEye	rightEye	leftEar	rightEar	leftShould	rightShould	leftElbow	rightElbow	leftWrist	rightWrist	leftHip	rightHip	leftKnee	rightKnee	leftAnkle	rightAnkle
x 좌표	0	0.007	0.008	0.033	0.056	0.146	0.16	0.191	0.509	0.174	0.965	0.265	0.458	0.181	0.158	0.153	1
y 좌표	0	0.048	0.174	0.492	0.334	0.195	0.211	0.824	0.425	1.467	0.943	0.354	0.886	0.987	0.97	1.344	1
정확도	0.941	0.271	0.966	0.104	0.964	0.469	0.968	0.928	0.855	0.774	0.82	0.057	0.08	0.125	0.133	0.067	0.096

GridSearchCV를 통해 하이퍼 파라미터를 최적화 & 모델 정확도 높이기.

max_depth, min_child_weight 튜닝

```
param_test1 = {
    'max_depth':range(3,10),
    'min_child_weight':range(1,10)
}
gsearch1 = GridSearchCV(estimator = XGBClassifier(learning_rate=0.1,
    n_estimators=100,
    max_depth=5,
    min_child_weight=1,
    gamma=0,
    subsample=0.8,
    colsample_bytree=0.8,
    objective='binary:logistic',
    nthread=-1,
    scale_pos_weight=1),
    param_grid = param_test1, scoring='accuracy', n_jobs=-1, iid=False, cv=5, verbose=10)
gsearch1.fit(train[predictors],train[target])
gsearch1.cv_results_, gsearch1.best_params_, gsearch1.best_score_
```

Gamma 튜닝

```
param_test2 = {
    'gamma':[i/10.0 for i in range(0,5)]
}
gsearch2 = GridSearchCV(estimator = XGBClassifier(learning_rate=0.1,
    n_estimators=100,
    max_depth=7,
    min_child_weight=1,
    gamma=0,
    subsample=0.8,
    colsample_bytree=0.8,
    objective='binary:logistic',
    nthread=-1,
    scale_pos_weight=1),
    param_grid = param_test2, scoring='accuracy', n_jobs=-1, iid=False, cv=5)
gsearch2.fit(train[predictors],train[target])
gsearch2.cv_results_, gsearch2.best_params_, gsearch2.best_score_
```

subsample and colsample_bytree 튜닝


```
param_test3 = {
    'subsample':[i/10.0 for i in range(6,10)],
    'colsample_bytree':[i/10.0 for i in range(6,10)]
}
gsearch3 = GridSearchCV(estimator = XGBClassifier(learning_rate=0.1,
    n_estimators=100,
    max_depth=7,
    min_child_weight=1,
    gamma=0,
    subsample=0.8,
    colsample_bytree=0.8,
    objective='binary:logistic',
    nthread=-1,
    scale_pos_weight=1),
    param_grid = param_test3, scoring='accuracy', n_jobs=-1, iid=False, cv=5,
    verbose=10)
gsearch3.fit(train[predictors],train[target])
gsearch3.cv_results_, gsearch3.best_params_, gsearch3.best_score_
```

Learning Rate 튜닝


```
param_test6 = {
    'learning_rate':[i/10.0 for i in range(0,10)]
}
gsearch6 = GridSearchCV(estimator = XGBClassifier(learning_rate=0.1,
    n_estimators=100,
    max_depth=7,
    min_child_weight=1,
    gamma=0,
    subsample=0.8,
    colsample_bytree=0.8,
    objective='binary:logistic',
    nthread=-1,
    scale_pos_weight=1),
    param_grid = param_test6, scoring='accuracy', n_jobs=-1, iid=False, cv=5, verbose=10)
gsearch6.fit(train[predictors],train[target])
gsearch6.cv_results_, gsearch6.best_params_, gsearch6.best_score_
```

pickle을 이용해 모델을 저장한다.

구현 과정 - (3) 모델을 바탕으로 실제 사용자 자세 예측하기

 **Flask** 를 이용해 사용자 웹을 구성하고, 사용자의 운동 영상을 가져와 Posenet을 이용하여 각 부위의 위치를 계산하고 이를 전처리 한 후 학습 시켜놓았던 모델(XGBoost)에 넣고 정확도를 예측한다.



전처리 된 Posenet 값과 모델의 출력값 (자세가 올바른지) 을 다시  **Flask** 로 보낸 후 운동 보고서 작성 알고리즘에 따라 운동 보고서를 작성한다.

발전해 나가야 할 점

1. 적은 데이터 (2천 개) 로 인한 낮은 정확도 (약 70%)

최대한 데이터를 수집하려 했지만 중복되는 데이터가 많아 많은 데이터를 수집하지 못한 점이 아쉽습니다. 데이터가 더 많았다면 모델의 정확도가 높아져 훨씬 완성도 높은 웹이 되었을 거라 생각합니다.

2. 현재 플랭크 자세만 서비스 가능한 한계

데이터를 얻는 문제부터 관련 알고리즘을 짜는 일까지, 최대한 많은 운동을 지원하고 싶었지만 2주라는 기간의 문제로 오직 '플랭크' 운동만을 지원하게 되었습니다. 추후 가능하다면 더 다양한 운동에 도전해보고 싶습니다.

팀 프로젝트 과정 - Git을 이용한 협업 <https://github.com/JooEHyeon/UPT>

The screenshot shows a GitHub repository interface. At the top, there are navigation tabs: Actions, Projects, Wiki, Security, Insights, and Settings. Below these, there's a header with 'master' branch selected, '5 branches', and '0 tags'. There are buttons for 'Go to file', 'Add file', and 'Code'. The main content area displays a list of files and folders with their commit history. The files include 'ijh', 'scp', 'toram', 'ykj', '.gitignore', '0909노가다문서_전반적활용.txt', '0909박준영강사님.txt', '0909최주원강사님.txt', '33번 프로젝트팀 기획서.docx', 'README.md', 'demo_완성본재현.pptx', and '데이터 라벨링 가이드.docx'. The right sidebar shows 'About' (Make online Personal Training using POSNET (with YouTube)), 'Readme', 'Releases' (No releases published), 'Packages' (No packages published), 'Contributors' (5 contributors), and 'Languages' (JavaScript 89.4%, Python 5.9%, Jupyter Notebook 2.3%, HTML 1.8%, Other 0.6%).

seongchoel Merge pull request #34 from JooEHyeon/psc 266cb9e 11 hours ago 86 commits

File	Commit Message	Time
ijh	use 'pip install -r requirements.txt' command	5 days ago
scp	0920_dataframe_2	11 hours ago
toram	전처리완료	8 days ago
ykj	add 0920file	14 hours ago
.gitignore	use 'pip install -r requirements.txt' command	5 days ago
0909노가다문서_전반적활용.txt	메일보시게될문서입니다.	12 days ago
0909박준영강사님.txt	오전 박준영강사님	12 days ago
0909최주원강사님.txt	오전최주원강사님	12 days ago
33번 프로젝트팀 기획서.docx	메일보시게될문서입니다.	12 days ago
README.md	Update README.md	14 days ago
demo_완성본재현.pptx	우리의 목표	11 days ago
데이터 라벨링 가이드.docx	PLZ see this before labelling	11 days ago

README.md

UPT (Your Personal Training) _ 200907 ver.

- 광주 인공지능 사관학교 워밍업 프로젝트
- 팀장 : 박지현
- 팀원 : 박성철 윤경지 전주현
- 팀플 규칙 : 자신의 이름(이니셜, 닉네임)로 branch명을 작성해주시고, 하나의 폴더를 업데이트하는 방식을 취해주세요. (단일 파일 X)

자신이 한 것을 commit하고
회의에서 발표하며
지속적으로 소통하는
팀플을 경험하였습니다.



2 주간 많이 배울 수 있었습니다.

감사합니다