

Analiza vježbanja putem snimanja pokreta bez markera

Leko, Ruža

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:498715>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-02-20**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA OSIJEK**

Sveučilišni studij

**ANALIZA VJEŽBANJA PUTEM SNIMANJA POKRETA
BEZ MARKERA**

Završni rad

Ruža Leko

Osijek, 2023.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju**

Osijek, 14.09.2023.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

| | |
|---|---|
| Ime i prezime Pristupnika: | Ruža Leko |
| Studij, smjer: | Programsko inženjerstvo |
| Mat. br. Pristupnika, godina upisa: | R 4380, 21.10.2021. |
| OIB Pristupnika: | 90340884816 |
| Mentor: | izv. prof. dr. sc. Časlav Livada |
| Sumentor: | , |
| Sumentor iz tvrtke: | |
| Naslov završnog rada: | Analiza vježbanja putem snimanja pokreta bez markera |
| Znanstvena grana rada: | Obradba informacija (zn. polje računarstvo) |
| Zadatak završnog rad: | U radu je potrebno napraviti aplikaciju koja će biti analizator virtualnih treninga (analiza položaja tijela te uspješnost odrađivanja vježbi u usporedbi s referentnim virtualnim "vodičem"). Potrebno je koristiti sljedeću biblioteku: https://freemocap.org/ |
| Prijedlog ocjene završnog rada: | Izvrstan (5) |
| Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova: | Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina |
| Datum prijedloga ocjene od strane mentora: | 14.09.2023. |
| Datum potvrde ocjene od strane Odbora: | 24.09.2023. |
| Potvrda mentora o predaji konačne verzije rada: | <i>Mentor elektronički potpisao predaju konačne verzije.</i> |
| | Datum: |

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**IZJAVA O ORIGINALNOSTI RADA**

Osijek, 17.02.2024.

Ime i prezime studenta:

Ruža Leko

Studij:

Programsko inženjerstvo

Mat. br. studenta, godina upisa:

R 4380, 21.10.2021.

Turnitin podudaranje [%]:

7

Ovom izjavom izjavljujem da je rad pod nazivom: **Analiza vježbanja putem snimanja pokreta bez markera**

izrađen pod vodstvom mentora izv. prof. dr. sc. Časlav Livada

i sumentora ,

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija. Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

| | |
|--|-----------|
| 1. UVOD | 1 |
| 1.1. Zadatak završnog rada | 1 |
| 2. PREGLED PODRUČJA TEME | 2 |
| 2.1. Snimanje pokreta korištenjem markera | 2 |
| 2.1.1. Akustični sustavi | 2 |
| 2.1.2. Mehanički sustavi | 2 |
| 2.1.3. Magnetski sustavi | 3 |
| 2.1.4. Optički sustavi | 4 |
| 2.2. Snimanje pokreta bez markera | 5 |
| 3. KORIŠTENE TEHNOLOGIJE | 6 |
| 3.1. <i>MediaPipe</i> | 6 |
| 3.2. <i>Scikit-learn</i> | 6 |
| 3.3. <i>OpenCV</i> | 7 |
| 4. IMPLEMENTACIJA RJEŠENJA | 8 |
| 4.1. Postavljanje kamere | 8 |
| 4.2. Bilježenje podataka | 8 |
| 4.3. Učenje modela | 9 |
| 4.4. Ocjena ispravnosti izvođenja vježbe | 11 |
| 4.4.1. Iskorak | 12 |
| 4.4.2. Bočno podizanje ruku | 13 |
| 4.4.3. <i>Plank</i> | 14 |
| ZAKLJUČAK | 16 |
| LITERATURA | 17 |
| POPIS SLIKA | 18 |
| SAŽETAK | 19 |
| ABSTRACT | 20 |
| ŽIVOTOPIS | 21 |

1. UVOD

U ovom završnom radu razmotrit će se analiza uspješnosti odrađivanja vježbe snimanjem pokreta primjenom tehnike bez upotrebe markera. Ovaj pristup omogućuje praktično prikupljanje podataka u analizi pokreta što ga čini pogodnim za širok spektar primjene. U sklopu drugog poglavlja, detaljno će se objasniti problematika snimanja pokreta, kao i njegova primjena. Analizirat će se kako pristupi koji ovise o korištenju markera imaju ograničenu efikasnost i praktičnost prikupljanja podataka. Suprotno tome, snimanje pokreta bez markera bit će istaknuto kao rješenje koje omogućuje preciznu analizu pokreta uz minimalne zahtjeve u pripremi i postavljanju opreme. Nadalje, u trećem poglavlju bit će predstavljene korištene tehnologije, s naglaskom na biblioteke *MediaPipe*, *Scikit-learn* i *OpenCV*. Ove biblioteke su ključne za realizaciju programskog rješenja koje će biti opisano u kasnijim dijelovima ovog rada. Četvrto poglavlje pružit će detaljan uvid u razvoj programskog rješenja. Proces postavljanja kamere, prikupljanje podataka te implementacija algoritma strojnog učenja bit će opisani korak po korak. Poseban naglasak bit će stavljen na ocjenu ispravnosti izvođenja vježbi, koja će biti ključni element analize uspješnosti odrađivanja vježbe u stvarnom vremenu.

1.1. Zadatak završnog rada

U završnom radu potrebno je napraviti aplikaciju koja će biti analizator virtualnih treninga putem snimanja pokreta bez markera, koristeći biblioteku *MediaPipe*. Aplikacija treba omogućiti analizu položaja tijela te uspješnost odrađivanja vježbi u usporedbi s ispravnim načinom izvođenja vježbe predstavljenim referentnim virtualnim vodičem. Za realizaciju aplikacije koristit će se programski jezik *Python* te *MediaPipe* biblioteka.

2. PREGLED PODRUČJA TEME

Tehnika se sastoji od snimanja pokreta (engl. *Motion Capture*) uživo i transformiranja u matematičke izraze prateći ključne točke u prostoru tijekom vremena. Omogućuje transformaciju pokreta uživo u digitalnu reprezentaciju tog pokreta [1]. Današnje najnaprednije tehnologije snimanja pokreta su *Ascension's Motion Star* [2] sustav koji pokušava prevladati nedostatke magnetskih sustava snimanja pokreta, Sveučilište *Stanford* je razvilo sustav za snimanje pokreta bez markera koji precizno računa kutove i centre zglobova [3]. Sveučilište *Carnegie Mellon* također je razvilo sličan sustav koji pruža modele snimanjem pokreta tijela i zglobova [4]. *Finally Image Metrics* pruža sustav snimanja pokreta bez markera, temeljen na sferičnim rasvjetnim sustavom kamera visoke rezolucije [5]. Glavna tržišta koja koriste snimanje pokreta jesu medicina, sport, industrija zabave, pravo i automobilska industrija. Tehnologija ima širok spektar primjene pa velik broj industrija u svijetu ima pogodnosti od njenog razvoja [6].

2.1. Snimanje pokreta korištenjem markera

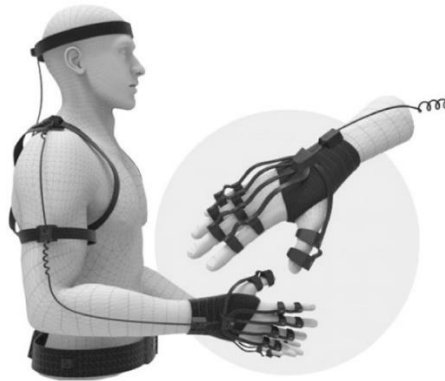
2.1.1. Akustični sustavi

U ovom tipu sustava skup zvučnih predajnika (engl. *Sound transmitters*) je smješten na izvođačeve glavne zglobove, dok su tri receptora postavljena na mjestu snimanja. Zatim se predajnici aktiviraju sekvencijalno proizvodeći karakteristične frekvencije koje receptori bilježe i koriste za izračun pozicije predajnika u trodimenzionalnom prostoru. Računanje položaja svakog predajnika odvija se koristeći kao podatke vremenski interval između emitiranja šuma od strane predajnika, primanja tog šuma od strane receptora i brzinu zvuka u okolini, čime je moguće izračunati udaljenost koju je šum prešao. Za određivanje trodimenzionalnog položaja svakog predajnika, izračunava se trilateracija udaljenosti između predajnika i svakog od receptora. Neki od problema koji se mogu javiti u ovim sustavima su: teškoća u dobivanju točnog opisa podataka u trenutku, ograničena sloboda kretanja izvođača zbog kablova te ograničen broj predajnika koji se mogu koristiti. Akustični sustavi nemaju problema s ometanjem zbog prekrivanja ili metalnih objekata, koji su karakteristični za optičke sustave i magnetske sustave, ali su osjetljivi na buku koja je često prisutna [7].

2.1.2. Mehanički sustavi

Mehanički sustavi su napravljeni od potencijometara i kliznika postavljenih na zglobove koji omogućuju reprezentaciju njihovih položaja, kao što je prikazano na slici 2.1. Iako su nedovoljno

razvijeni, mehanički sustavi imaju određene prednosti. Posjeduju sučelje koje je slično *stop-motion* sustavima koji su popularni i korišteni u filmskoj industriji, što omogućuje jednostavan prijelaz između ovih tehnologija. Nisu pod utjecajem magnetskih polja i neželjenih refleksija što ih čini lakim za korištenje.



Slika 2.1. Oprema za mehanički sustav snimanja pokreta[8].

2.1.3. Magnetski sustavi

Koristeći niz receptora koji su postavljeni na izvođačeve zglobove, moguće je mjeriti poziciju i orijentaciju zglobova u odnosu na antenu. Magnetski sustavi su jeftiniji od ostalih sustava snimanja pokreta i imaju visoku preciznost podataka. Prigodni su za snimanje jednostavnih pokreta. Koriste velik broj žica povezanih na antenu što ograničava izvođačev raspon kretanja. Moguće smetnje u magnetskom polju koje uzrokuju metalni objekti i strukture, postavljaju ograničenja na materijal u okolini, što može biti ozbiljan nedostatak. Na slici 2.2. prikazana je oprema korištena u magnetskim sustavima snimanja pokreta [6].



Slika 2.2. Oprema za magnetski sustav snimanja pokreta[9].

2.1.4. Optički sustavi

U optičkim sustavima izvođači nose odijelo s reflektorima koji su smješteni na glavne zglobove. Kamere visoke rezolucije su smještene tako da prate spomenute reflektore tijekom izvođačevog kretanja. Svaka kamera generira dvodimenzionalne koordinate za svaki reflektor putem koraka segmentacije. Koristi se prigodni *software* za analizu prikupljenih podataka za računanje trodimenzionalnih koordinata reflektora. Optički su sustavi najskuplji sustavi snimanja pokreta. Imaju visoku stopu uzrokovanja koja omogućuje snimanje brzih pokreta. Nude slobodu kretanja izvođaču jer za razliku od prethodno navedenih sustava ne koriste žice te nemaju ograničenja radnog okruženja. Budući da reflektori ne pružaju otpor, gotovo nema ograničenja u broju reflektora koji se koriste u procesu snimanja. Nedostatak kod ovih sustava može biti prekrivanje nekih predajnika, pogotovo kod malih objekata ili vrlo blizu postavljenih objekata. Također, nedostatak interaktivnosti zato što prikupljeni podaci moraju biti obrađeni prije korištenja. Navedeni nedostaci mogu dovesti do visokih troškova. Snimanje pokreta s aktivnim markerima koristi LED svjetla koje umjesto reflektiranja emitiranog svjetla, emitiraju vlastito koje se napaja malom baterijom. To može povećati korisni volumen snimanja. Za razliku od aktivnih markera pasivni markeri prekriveni su retroreflektivnim materijalom koji reflektira svjetlo prema kamerama koje moraju prethodno biti kalibrirane. Razlika aktivnih i pasivnih optičkih sustava prikazana je na slici 2.3 [10].



Slika 2.3. Aktivni i pasivni optički sustavi snimanja pokreta[11].

2.2. Snimanje pokreta bez markera

Snimanje pokreta bez korištenja markera ne zahtjeva dodatnu opremu za praćenje izvođačevog pokreta. Snima se kretanje u više video tokova te algoritmi računalnog vida analiziraju video kako bi identificirali oblik te ga razložili na izolirane, pojedinačne dijelove koji se koriste za praćenje. Proces snimanja pokreta ostvaren je u potpunosti, korištenjem *softwarea*. Ovaj proces nema fizičkih ograničenja, ali se pojavljuju računalna. Primjer sustava snimanja pokreta bez korištenja markera namijenjen za široke mase je *Microsoft Kinect*. Proces generiranja podataka korištenjem snimanja pokreta bez korištenja markera uključuje postavljanje i osiguravanje ispravnosti niza parametara kako bi bilo uspješno. Područje snimanja mora biti pokriveno s više kamera postavljenim na ključna mjesta kako bi snimili podatke iz više različitih kutova i perspektiva. Nadalje, osim kamera, česta je primjena infracrvenih senzora ili svjetlosnog trodimenzionalnog skenera. Navedeni uređaji pružaju informacije o dubini i time omogućuju stvaranje trodimenzionalne reprezentacije scene snimanja. Snimljeni se podaci obrađuju korištenjem računalnih algoritama. Algoritmi analiziraju vizualne informacije kako bi pratili ključne točke promatranog objekta. Kod čovjeka, ključne točke su najčešće zglobovi. Na slici 2.4. prikazano je da sustav procjenjuje poziciju i orijentaciju izvođača u vremenu te rekonstruira virtualni objekt koji predstavlja virtualnu strukturu tijela i pokrete zglobova. Podaci se potom spremaju te obrađuju. Nakon obrade, podaci se mogu koristiti u razne svrhe [12].

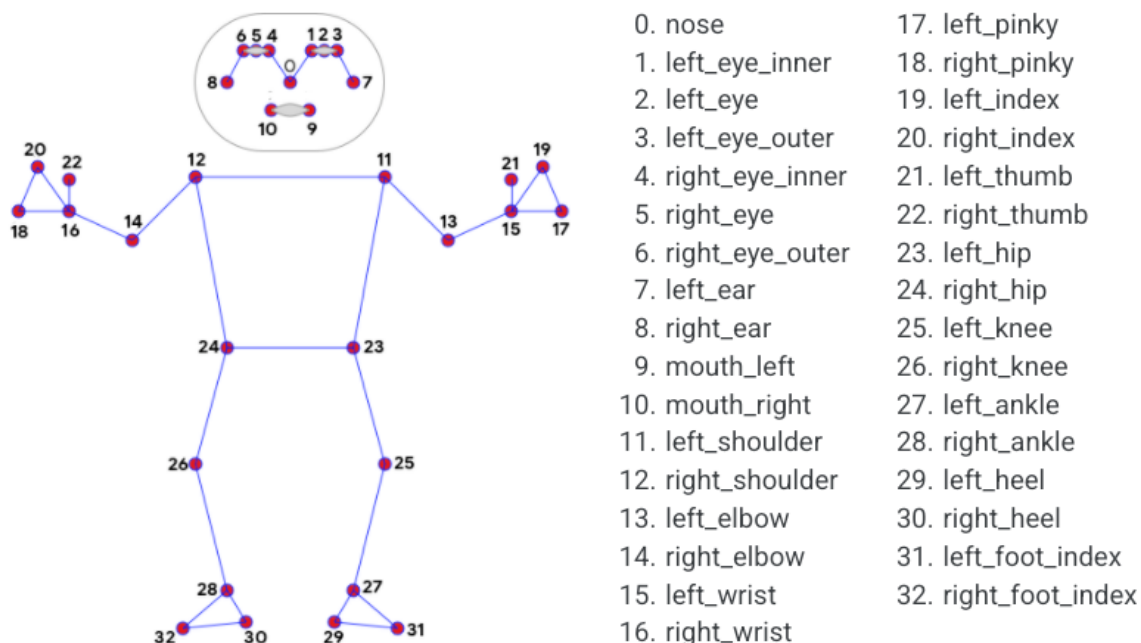


Slika 2.4. Sustav snimanje pokreta bez markera[13].

3. KORIŠTENE TEHNOLOGIJE

3.1. *MediaPipe*

MediaPipe Solutions daje širok raspon biblioteka i alata za jednostavnu primjenu tehnika umjetne inteligencije i strojnog učenja. Omogućuje izradu multimedijских aplikacija u stvarnom vremenu. Pruža pogodnosti podrške na više platformi, modularni dizajn, gotove komponente, integraciju strojnog učenja i drugo. Unutar ovog projekta fokus je na modulu za prepoznavanje položaja tijela. Modul omogućuje detekciju zglobova tijela na mediju slike ili videa, a osim za primjene detekcije može se koristiti za kategorizaciju pokreta te analizu držanja tijela. Rezultat djelovanja modula jesu oznake koordinate zglobova vidljive na slici 3.1. Modul je sastavljen od dva dijela, prvi koji služi detekciji pokreta te drugi koji dodaje obilježja koordinata [14].



Slika 3.1. Obilježja koja *MediaPipe* detektira [15].

3.2. *Scikit-learn*

Scikit-learn biblioteka je za *Python* programski jezik koja odrađuje poslove strojnog učenja. Za svoju potpunu funkcionalnost koristi *NumPy*, *SciPy* te *Matplotlib* biblioteke. *Scikit-learn* pruža metode strojnog učenja za nadzirano i nenadzirano učenje, odabir modela te vizualizaciju. Pogodnosti koje nudi: jednostavnu integraciju raznih funkcionalnosti, razne algoritme, modele procjene i selekcije te skalabilnost i učinkovitost [16].

3.3. *OpenCV*

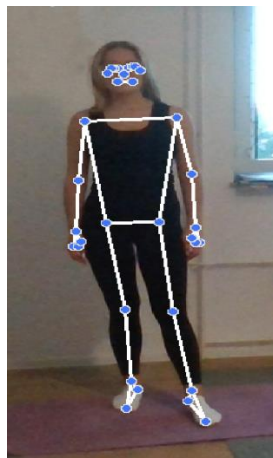
OpenCV (*Open Source Computer Vision Library*) je biblioteka za računalni vid i strojno učenje. *OpenCV* razvijen je kako bi pružio zajedničku infrastrukturu za aplikacije računalnog vida te za ubrzavanje uporabe strojne percepcije unutar komercijalnih proizvoda. Biblioteka sadrži više od 2500 optimiziranih algoritama pogodnih za računala opremljena hardwareima od amaterske do profesionalne namjene. Ovi algoritmi koriste se za obradu videozapisa i slika te omogućuju prepoznavanje bilo kojeg obilježja ili radnje na istima. Podržava rad unutar C++, *Python*, *Java* te *MATLAB* programskih jezika na operacijskim sustavima poput *Windowsa*, *Linuxa* te *Androida*. *OpenCV* se primarno koristi za aplikacije stvarnog vremena [17].

4. IMPLEMENTACIJA RJEŠENJA

Aplikacija je osmišljena tako da daje ocjenu ispravnosti izvođenja vježbe. Izvođenje vježbe može biti ispravno ili neispravno. Osoba mora biti pozicionirana ispred kamere tako da joj se vidi cijelo tijelo kako bi se omogućila detekcija ključnih zglobova u stvarnom vremenu.

4.1. Postavljanje kamere

Prvo je postavljena kamera koja obavlja video prijenos. U ovom programskom rješenju je korištena *web* kamera. Inicijalizira se objekt za snimanje videa koristeći *OpenCV* te se stvara instanca klase *VideoCapture* kojoj je kroz konstruktor predana *web* kamera. Očitava se video prijenos koji je predajemo metodi *imshow* iz *OpenCV*, čime je omogućeno prikazivanje prijenosa kamere. Nakon toga stvorena je *Mediapipe* instanca s modelom za procjenu položaja tijela. Preko stvorene instance ostvarena je detekcija pokreta. Predavanjem detektiranih koordinata zglobova i njihovih poveznica metodi *draw_landmarks* iz modula *drawing_utils*, prikazana je detekcija pokreta na ekranu ocrtavanjem zglobova i linija koje ih povezuju, Na slici 4.1. vidljive su ocrtane ostvarene detekcije zglobova.



Slika 4.1. Prikaz oznaka zglobova.

4.2. Bilježenje podataka

Nakon prethodno postavljane kamere potrebno je zabilježiti potrebne podatke za učenje modela. Na slici 4.2. vidljivo je stvaranje nove datoteke s nazivom *coord.csv* koja je otvorena za čitanje i pisanje. Stvorena je instanca klase *csv_writer* za koji su predani odgovarajući parametri. Na *csv_writeru* pozva se funkcija *writerow* kojoj se predaje lista x, y i z koordinata detektiranih pokreta u stvarnom vremenu. Funkcija bilježi predane koordinate u *csv* datoteku s oznakom klase koja je navedena. Oznake koje su korištene u ovom radu predstavljaju 3 različite vježbe: *Lunge*,

Lateral i *Plank*. Navedeni postupak bilježenja koordinata ponovljen je na isti način za svaku vrstu vježbe čime su skupljeni označeni podaci za postupak učenja modela.

```
# Export coordinates
try:
    # Extract Pose landmarks
    pose = results.pose_landmarks.landmark
    pose_row = list(np.array([[landmark.x, landmark.y, landmark.z, landmark.visibility] for landmark in pose]).flatten())

    # Append class name
    pose_row.insert(0, class_name)

    #Export to CSV
    with open('coords.csv', mode='a', newline='') as f:
        csv_writer = csv.writer(f, delimiter=',', quotechar='"', quoting=csv.QUOTE_MINIMAL)
        csv_writer.writerow(pose_row)
except:
    pass
```

Slika 4.2. Postupak bilježenja koordinata u csv datoteku.

4.3. Učenje modela

Cilj treniranja modela pronaći je model koji na temelju dostupnih x, y i z koordinata koje predstavljaju značajke, klasificira detektirane pokrete u jednu od 3 moguće klase: *Lunge*, *Lateral* i *Plank*.

```
df = pd.read_csv('coords.csv')

X = df.drop('class', axis=1) # features
y = df['class'] # target value

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1234)
```

Slika 4.3. Obrada učitanih podataka.

Slika 4.3. prikazuje učitavanje podataka spremljenih u csv datoteku s dodanim stupcem klase koji je izlazna veličina modela. Također, vidljive su varijabla X u kojoj su spremljeni podatkovni primjeri za ulazne veličine i varijabla y u kojoj su spremljene pripadne vrijednosti izlazne veličine. Učitani podaci podijeljeni su na skup podataka za učenje i skup podataka za testiranje u omjeru 70% naprema 30% pomoću funkcije *train_test_split* iz *Scikitlearn* biblioteke. *X_train* i *y_train* sadrže podatke za učenje modela, a *X_test* i *y_test* sadrže podatke za testiranje dobivenog modela.

```
pipelines = {
    'lr':make_pipeline(StandardScaler(), LogisticRegression()),
    'rc':make_pipeline(StandardScaler(), RidgeClassifier()),
    'rf':make_pipeline(StandardScaler(), RandomForestClassifier()),
}
```

Slika 4.4. Pipelines rječnik.

Nakon podjele podataka definiran je rječnik *pipelines* prema slici 4.4, u kojemu je svaki ključ predstavljen imenom algoritma strojnog učenja za klasifikaciju, a svaka vrijednost predstavlja *pipeline* strojnog učenja za pripadni klasifikator. Svaki *pipeline* koristi *StandardScaler()* i klasifikacijski algoritam. *StandardScaler()* vrši standardizaciju značajki tako da im je srednja vrijednost jednaka nuli, a varijanca jednaka jedinici. Standardizacija vrijednost računa se po formuli:

$$\tilde{x}_j^{(i)} = \frac{x_j^{(i)} - \bar{x}_j}{\sigma_j} \quad (4-1)$$

gdje je: $\tilde{x}_j^{(i)}$ ($i=1,2,\dots,n$ $j=1,2,\dots,m$) – standardizirana vrijednost j -te značajke za i -ti podatkovni primjer nakon primjene standardnog skaliranja, dobivena kao kvocijent razlike originalne vrijednosti j -te značajke za i -ti podatkovni primjer $x_j^{(i)}$ i srednje vrijednosti j -te značajke \bar{x}_j te standardne devijacije te značajke σ_j . Broj značajki jednak je m , a n predstavlja broj podatkovnih primjera u skupu podataka.

Navedeni algoritmi strojnog učenja za klasifikaciju su logistička regresija, *Ridge* klasifikacijski model i *RandomForest* klasifikator.

```
fit_models = {}
for algo, pipeline in pipelines.items():
    model = pipeline.fit(X_train, y_train)
    fit_models[algo] = model

for algo, model in fit_models.items():
    yhat = model.predict(X_test)
    print(algo, accuracy_score(y_test, yhat))

with open('pose_det.pkl', 'wb') as f:
    pickle.dump(fit_models['rf'], f)
```

Slika 4.5. *Treniranje, vrednovanje i spremanje modela.*

Na slici 4.5 je kod kojim se prolazi kroz svaki *pipeline* u rječniku te se na svakome poziva *fit* metoda koja trenira pojedini model, a predani su joj podatci za treniranje. Istrenirani modeli su spremljeni u rječnik. Nakon učenja na svakome modelu pozva se metoda *predict* kojoj su predani podaci za testiranje X_{test} . Predviđene su vrijednosti izlazne veličine te se one uspoređuju sa stvarnim vrijednostima izlazne veličine y_{test} . Usporedbom predviđanja i stvarnih vrijednosti funkcijom *accuracy_score* dobivena je točnost svakog modela. Točnost je metrika koja računa omjer točnih predikcija (ispravno klasificiranih instanci) u odnosu na ukupan broj instanci u skupu podataka. Najveću točnost imao je *RandomForest* klasifikator koji je klasificirao sve primjere

točno. *RandomForest* klasifikator je algoritam koji prilagođava niz klasifikatora stabla odluke na poduzorcima skupa podataka i koristi prosjek kako bi poboljšao točnost i pretjerano usklađivanje na podatke za učenje. *RandomForest* klasifikacijski model je spremljen te je korišten za raspoznavanje vježbe koja se izvodi u stvarnom vremenu.

4.4. Ocjena ispravnosti izvođenja vježbe

```
# Load the pose detection model
with open('pose_det.pkl', 'rb') as f:
    model = pickle.load(f)
```

Slika 4.6. Otvaranje modela.

Nakon postavljene kamere za snimanje pokreta te provedenog učitavanja podataka i učenja modela. Izrađeni je model otvoren te učitani u programsko rješenje, što se može vidjeti na slici 4.6.

```
# Make Detections
X = pd.DataFrame([pose_row])
body_language_class = model.predict(X)[0]
body_language_prob = model.predict_proba(X)[0]
```

Slika 4.7. Predviđanje vježbe.

Na učitanoj prethodno istreniranom modelu pozivana je metoda *predict* za predviđanje vježbe koja se izvodi. Metodi se predaju koordinate dobivene snimanjem pokreta u stvarnom vremenu. Također pozvana je metoda *predict_proba* za izračunavanje vjerojatnosti klase na temelju ulaznih vrijednosti ulaznih veličina, što je prikazano na slici 4.7. Vrijednosti ulaznih veličina koordinate su zglobova snimljene u stvarnom vremenu. Ako je vjerojatnost klasifikacije vježbe veća od 70% ispituje se koja je vježba detektirana uz pripadnu ocjenu ispravnosti izvođenja koja ovisi o referentnim kutovima koji se računaju korištenjem funkcije *calculate_angles* koja se može vidjeti na slici 4.8.

```
def calculate_angle(a, b, c):
    angle_rad = math.atan2(c[1] - b[1], c[0] - b[0]) - math.atan2(a[1] - b[1], a[0] - b[0])
    angle_deg = math.degrees(angle_rad)
    return angle_deg
```

Slika 4.8. Funkcija *calculate_angle*.

Funkcija *calculate_angle* vraća kut u stupnjevima između početne, srednje i krajnje točke koje se predaju kao parametri a, b i c. U ovom radu su funkciji predane x i y koordinate zglobova između kojih se računa kut. Kut u radijanima je u ovoj funkciji izračunat tako što je od dobivenog kuta vektora BC($\text{math.atan2}(c[1] - b[1], c[0] - b[0])$) oduzet kut vektora AC($\text{math.atan2}(a[1] - b[1], a[0] - b[0])$). Izračunati kut u radijanima je pomoću *math.degrees* pretvoren u stupnjeve.

4.4.1. Iskorak

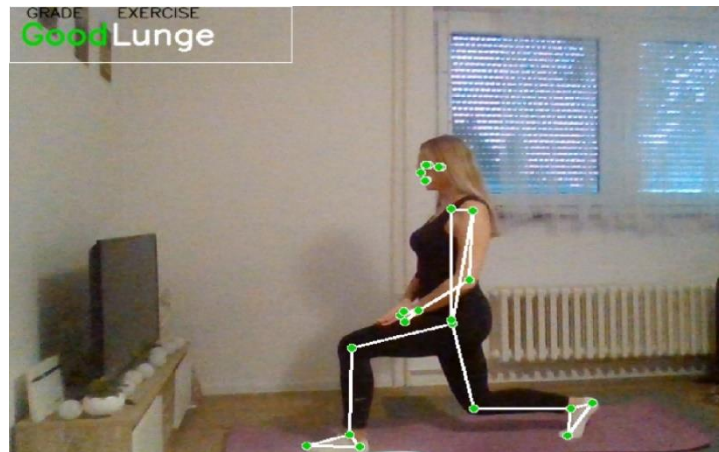
```
if str(body_language_class) == 'Lunge':  
    if len(results.pose_landmarks.landmark) >= 33:  
        left_hip = (results.pose_landmarks.landmark[mp_pose.PoseLandmark.LEFT_HIP].x,  
                    results.pose_landmarks.landmark[mp_pose.PoseLandmark.LEFT_HIP].y)  
        left_knee = (results.pose_landmarks.landmark[mp_pose.PoseLandmark.LEFT_KNEE].x,  
                    results.pose_landmarks.landmark[mp_pose.PoseLandmark.LEFT_KNEE].y)  
        left_ankle = (results.pose_landmarks.landmark[mp_pose.PoseLandmark.LEFT_ANKLE].x,  
                     results.pose_landmarks.landmark[mp_pose.PoseLandmark.LEFT_ANKLE].y)  
        right_hip = (results.pose_landmarks.landmark[mp_pose.PoseLandmark.RIGHT_HIP].x,  
                    results.pose_landmarks.landmark[mp_pose.PoseLandmark.RIGHT_HIP].y)  
        right_knee = (results.pose_landmarks.landmark[mp_pose.PoseLandmark.RIGHT_KNEE].x,  
                     results.pose_landmarks.landmark[mp_pose.PoseLandmark.RIGHT_KNEE].y)  
        right_ankle = (results.pose_landmarks.landmark[mp_pose.PoseLandmark.RIGHT_ANKLE].x,  
                      results.pose_landmarks.landmark[mp_pose.PoseLandmark.RIGHT_ANKLE].y)
```

Slika 4.9. Izdvajanje određenih koordinata.

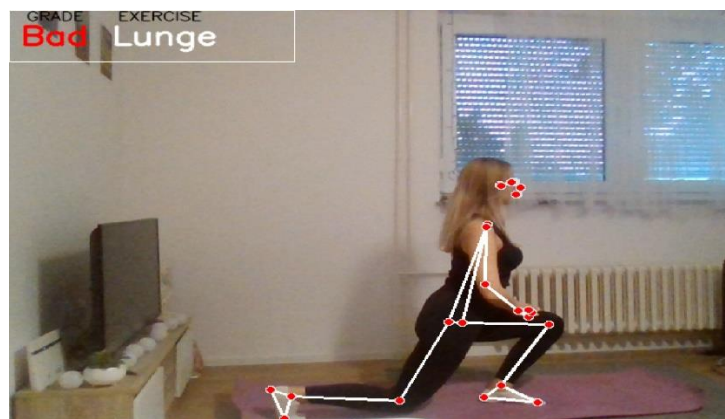
Na slici 4.9. se vidi ako je detektirana vježba iskorak(engl. *Lunge*) te ako su očitane koordinate svakog od 33 zglobova, izdvojene su x i y koordinate kukova, koljena te gležnjeva. Izdvojene koordinate predaju se funkciji *grade_lunge*, prikazanoj na slici 4.10. Unutar funkcije *grade_lunge* izračunat je lijevi i desni kut između kukova, koljena i gležnjeva pozivom funkcije *calculate_angle* kojoj su predane koordinate prethodno izdvojenih zglobova. Ako su kutovi veći od referentnih 85 stupnjeva vježba je ocjenjena kao ispravna te funkcija vraća ocjenu *Good*, u suprotnom vraća *Bad* što znači da se vježba izvodi neispravno. Vježba *Lunge* te ocjena izvođenja prikazuju se na ekranu kao na slici 4.11. i slici 4.12.

```
def grade_lunge(left_hip, left_knee, left_ankle, right_hip, right_knee, right_ankle):  
    # Calculate the angles at the knee joint  
    left_angle = calculate_angle(left_hip, left_knee, left_ankle)  
    right_angle = calculate_angle(right_hip, right_knee, right_ankle)  
    if left_angle >= 85 and right_angle >= 85:  
        return "Good"  
    else:  
        return "Bad"
```

Slika 4.10. Funkcija *grade_lunge*.



Slika 4.11. Ispravan iskorak.



Slika 4.12. Neispravan iskorak.

4.4.2. Bočno podizanje ruku

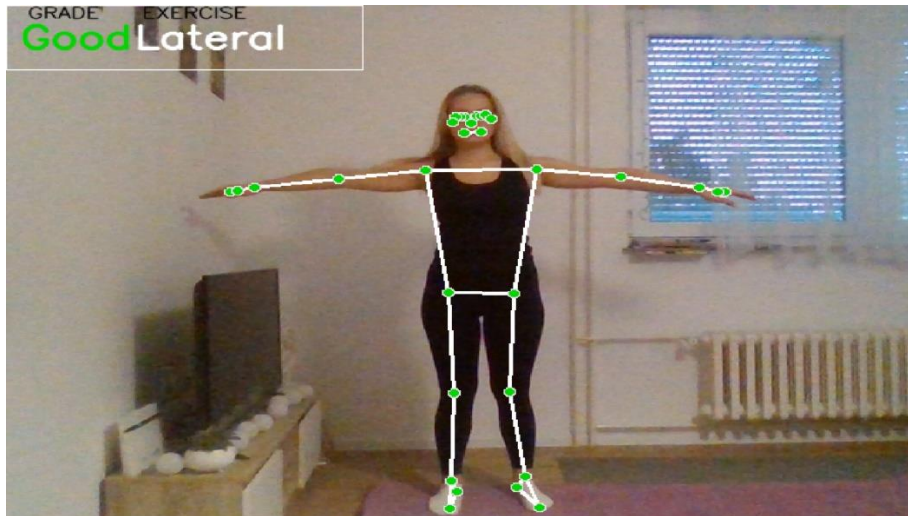
U slučaju kada je detektirana vježba bočno podizanje ruku (engl. *Lateral raise*) i ako su očitane koordinate više od 13 zglobova, ponavlja se sličan postupak kao za prethodnu vježbu. Izdvajaju se x i y koordinate lijevog i desnog zgloba zapešća, ramena i kuka. Poziva se funkcija *grade_lateral* koja vraća ocjenu *Good* ako su kutovi između izdvojenih zglobova manji od referentnih 120 stupnjeva, prema slici 4.13. Kutovi su izračunati pozivom funkcije *calculate_angle*. Veći kutovi od 120 stupnjeva ukazuju na neispravno izvođenje vježbe, stoga funkcija tada vraća ocjenu *Bad*.

```
def grade_lateral_raise(left_shoulder, left_wrist, right_shoulder, right_wrist, left_hip, right_hip):
    # Calculate the angle between the shoulders and wrists
    left_angle = calculate_angle(left_wrist, left_shoulder, left_hip)
    right_angle = calculate_angle(right_wrist, right_shoulder, right_hip)

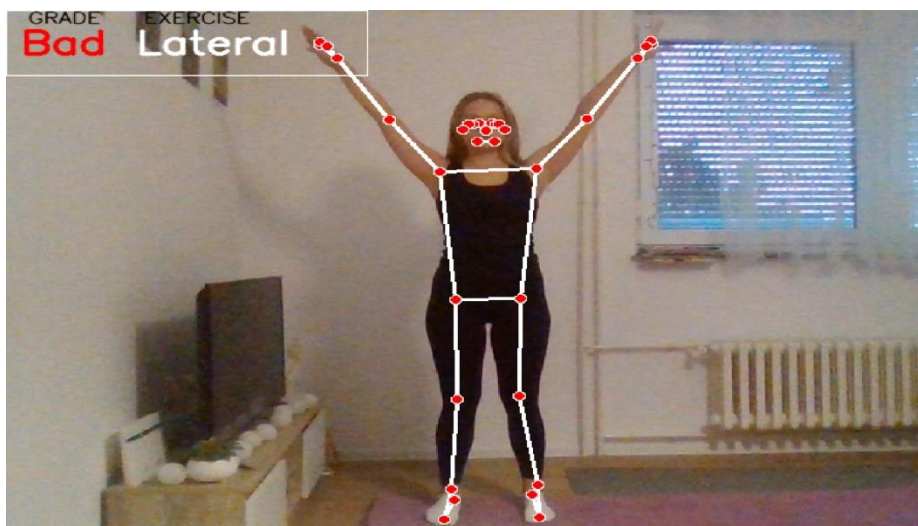
    if (left_angle <= 120 and right_angle <= 120):
        return "Good"
    else:
        return "Bad"
```

Slika 4.13. Funkcija *grade_lateral_raise*.

Slika 4.14. i slika 4.15. prikazuju ocjenu ispravnog i neispravnog izvođenja bočnog podizanja ruku.



Slika 4.14. Ispravno bočno podizanje ruku.



Slika 4.15. Neispravno bočno podizanje ruku.

4.4.3. Plank

Treća vježba koju aplikacija prepoznaje je *plank*. Ocjenjuje se izvođenje vježbe ako je očitano više od 13 zglobova. Ocjenu daje funkcija *grade_plank* koja provjerava kutove izračunate *calculate_angle* funkcijom između izdvojenih relevantnih koordinata gležnjeva, kukova i ramena, kao na slici 4.16. Slika 4.17. prikazuje ispravno izvođenje vježbe. Ispravnim izvođenjem vježbe smatra se položaj tijela u kojemu su referentni kutovi između 170 i 190 stupnjeva. Sve druge vrijednosti kutova ocjenjene su kao neispravne, što je prikazano na slici 4.18.

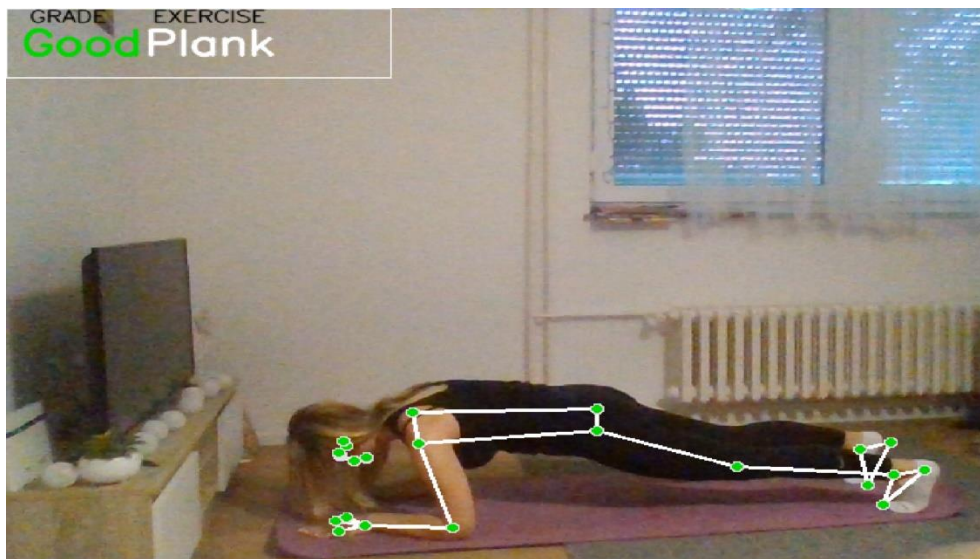
```

def grade_plank(left_shoulder, left_ankle, right_shoulder, right_ankle, left_hip, right_hip):
# Calculate the angle between the shoulders and wrists
    left_angle = calculate_angle(left_ankle, left_hip, left_shoulder)
    right_angle = calculate_angle(right_ankle, right_hip, right_shoulder)

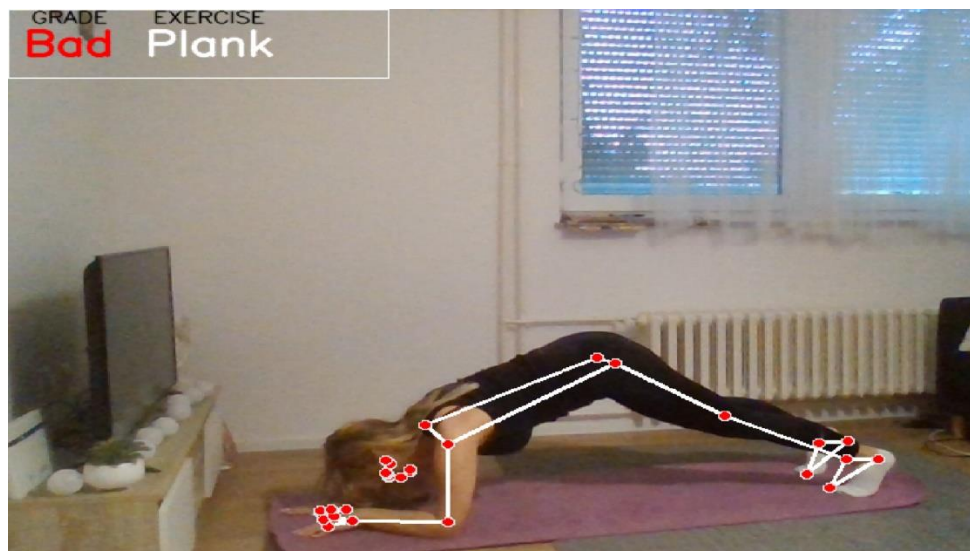
    if (left_angle <= 190 and left_angle >= 160) or (right_angle <= 180 and right_angle >= 160):
        return "Good"
    else:
        return "Bad"

```

Slika 4.16. Funkcija grade_plank.



Slika 4.17. Ispravan plank.



Slika 4.18. Neispravan plank.

ZAKLJUČAK

Ovaj rad analizira vježbanje putem snimanja pokreta bez markera. U prvom dijelu su predstavljene aktualne metode rješavanja takvog problema te su opisane tehnike snimanja pokreta korištenjem markera i bez korištenja markera. Nadalje, objašnjene su korištene biblioteke za ostvarivanje rješenja, a to su: *MediaPipe*, *Scikit-learn* i *OpenCV*. Postavljena je kamera koja snima pokret te detektira koordinate zglobova korištenjem biblioteka *MediaPipe* i *OpenCV*. Opisano je odrađeno prikupljanje podataka s pripadnim oznakama izlazne veličine. Prikazan je postupak učenja modela kojemu je cilj raspoznati koja vježba se izvodi u stvarnom vremenu na temelju koordinata zglobova. Testirana su tri algoritma za učenje od kojih je odabran *RandomForest* klasifikator zbog najboljih performansi. Odabrani model korišten je kako bi detektirao izvodi li se vježba *Lunge*, *Lateral raise* ili *Plank*. Za ostvarenje konačnog rješenja definirane su funkcije koje pomoću izdvojenih koordinata zglobova i vrijednosti kutova specifičnih za određenu vježbu, uspoređuju izračunate kutove s referentnim ispravnim vrijednostima. Na temelju odnosa izračunatih kutova prikazana je ocjena uspješnosti odrađivanja vježbe. Rad programa testiran je za svaku vježbu i ocjenu izvođenja. Prilikom testiranja, problematičnim se pokazao slučaj kada korisnik nije u cijelosti unutar okvira kamere. Programsko rješenje moguće je proširiti dodavanjem novih vježbi te unaprijeđivanjem kriterija ocjenjivanja uspješnosti.

LITERATURA

- [1] A. Menache, Understanding Motion Capture for Computer Animation, Elsevier Science, 2011.
- [2] A. Technology, MotionStar (Tethered Model), 2011.
- [3] S. Corazza, L. Mundermann, A. S. Chaudari, T. Demattio, C. Cobelli i T. P. Andriacchi, A Markerless Motion Capture System to Study Musculoskeletal Biomechanics: Visual Hull and Simulated Annealing Approach, 5 Svibanj 2006. Dostupno na: <https://link.springer.com/article/10.1007/s10439-006-9122-8> [4. lipnja 2023.]
- [4] G. K. Cheung, S. Baker i T. Kanade, Shape-From-Silhouette of Articulated Objects and its Use for Human Body Kinematics Estimation and Motion Capture, 20 Lipanj 2003. Dostupno na: <https://ieeexplore.ieee.org/abstract/document/1211340>. [4. lipnja 2023.]
- [5] E. O'Brien, Image-Metrics Facil Animation, 2008.
- [6] K. Midori i W. Brian, MoCap for Artists, Elsevier/Focal Press, 2008.
- [7] E. Volná, M. Kotyrba i R. Jarušek, Acoustic signal processing via neural network towards motion capture systems, Dostupno na: <chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/http://worldcomp-proceedings.com/proc/p2013/IPC7173.pdf> [4. lipnja 2023.]
- [8] L. Han, A Table Tennis Motion Correction System Based on Human Motion Feature Recognition, Security and Communication Network, 13. Rujan 2022. Dostupno na: https://www.researchgate.net/figure/A-classic-mechanical-motion-capture-system_fig1_363524612. [13. rujna 2023.]
- [9] B. R. Mursyid, Motion Capture II, 20. Siječanj 2015. Dostupno na: <https://baswaramursyid.wordpress.com/2015/01/20/motion-capture-ii/>. [13 rujna 2023]
- [10] G. B. Guerra-Filho, Optical Motion Capture: Theory and Implementation, 2005. Dostupno na: https://www.researchgate.net/publication/220162283_Optical_Motion_Capture_Theory_and_Implementation [4. lipnja 2023.]
- [11] L. Niklaus, Suit Up! The Art And Magic of MoCap Suit, Mountcg.com, Dostupno na: <https://mountcg.com/how-do-mocap-suits-work/>. [13. rujna 2023.]
- [12] R. Bodo, M. Dimitris i K. Reinhard, Human Motion, Springer, 2008.
- [13] Simi Reality Motion Systems, Next Level Markerless Motion Capture, 2019. Dostupno na: <https://simishape.com/> [13. rujna 2023.]
- [14] MediaPipe, Google, Dostupno na: <https://developers.google.com/mediapipe>. [3. lipnja 2023.]
- [15] MediaPipe, GitHub repository, 2023. Dostupno na: <https://github.com/google/mediapipe/blob/master/docs/solutions/pose.md>. [3. lipnja 2023.]
- [16] Scikit-learn, Scikit-learn, 2023. Dostupno na: <https://scikit-learn.org/stable/>. [5. lipnja 2023.]
- [17] OpenCV, OpenCV, 2023. Dostupno na: <https://opencv.org/about/>. [3. lipnja 2023.]

POPIS SLIKA

| | |
|---|----|
| Slika 2.1. <i>Oprema za mehanički sustav snimanja pokreta</i> | 3 |
| Slika 2.2. <i>Oprema za magnetski sustav snimanja pokreta</i> | 3 |
| Slika 2.3. <i>Aktivni i pasivni optički sustavi snimanja pokreta</i> | 4 |
| Slika 2.4. <i>Sustav snimanje pokreta bez markera</i> | 5 |
| Slika 3.1. <i>Obilježja koja MediaPipe detektira [13]</i> | 6 |
| Slika 4.1. <i>Prikaz oznaka zglobova</i> | 8 |
| Slika 4.2. <i>Postupak bilježenja koordinata u csv datoteku</i> | 9 |
| Slika 4.3. <i>Obrada učitanih podataka</i> | 9 |
| Slika 4.4. <i>Pipelines rječnik</i> | 9 |
| Slika 4.5. <i>Treniranje, vrednovanje i spremanje modela</i> | 10 |
| Slika 4.6. <i>Otvaranje modela</i> | 11 |
| Slika 4.7. <i>Predviđanje vježbe</i> | 11 |
| Slika 4.8. <i>Funkcija calculate_angle</i> | 11 |
| Slika 4.9. <i>Izdvajanje određenih koordinata</i> | 12 |
| Slika 4.10. <i>Funkcija grade_lunge</i> | 12 |
| Slika 4.11. <i>Ispravan iskorak</i> | 13 |
| Slika 4.12. <i>Neispravan iskorak</i> | 13 |
| Slika 4.13. <i>Funkcija grade_lateral_raise</i> | 13 |
| Slika 4.14. <i>Ispravno bočno podizanje ruku</i> | 14 |
| Slika 4.15. <i>Neispravno bočno podizanje ruku</i> | 14 |
| Slika 4.16. <i>Funkcija grade_plank</i> | 15 |
| Slika 4.17. <i>Ispravan plank</i> | 15 |
| Slika 4.18. <i>Neispravan plank</i> | 15 |

SAŽETAK

Zadatak ovog završnog rada je analizirati uspješnosti odrađivanja vježbe snimanjem pokreta bez markera u programskom jeziku *Python*. Za realizaciju programskog rješenja korištene su biblioteke *MediaPipe*, *Scikit-learn* i *OpenCV*. Pomoću *MediaPipe* biblioteke prikupljena su potrebna obilježja snimanja pokreta bez markera koja su korištena za treniranje modela za prepoznavanje vježbe algoritmom logističke regresije koji pruža *Scikit-learn*. Ostale funkcije uspoređuju izvođenje vježbe u stvarnom vremenu. Korištenjem spomenutih metoda ostvarena je aplikacija koja daje ocjenu ispravnosti izvođenja vježbe.

Ključne riječi: *Mediapipe*, *Python*, *Scikit-learn* , Snimanje pokreta

ABSTRACT

EXERCISE ANALYSIS USING MARKERLESS MOTION CAPTURE

The task of this paper is to analyze performance using markerless motion capture in the Python programming language. To solve this problem, the MediaPipe, Scikit-learn, and OpenCV libraries, along with their functions, were used to improve the algorithm's performance. Using the MediaPipe library, the necessary landmarks were collected from markerless motion capture, which were then used to train a pose recognition model using the logistic regression algorithm provided by Scikit-learn. Other functions compare real-time exercise performance with the correct form of exercise. By employing these methods, an application was developed that grades the accuracy of exercises.

Key words: Mediapipe, Python, Scikit-learn , Motion capture

ŽIVOTOPIS

Ruža Leko rođena je 19.05.2000. u Slavonskom Brodu. Pohađala je osnovnu školu „Dragutin Tadijanović“ te prirodoslovno- matematičku gimnaziju „Matija Mesić“ u Slavonskom Brodu. Tijekom pisanja ovog rada pohađa 3.godinu studija računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek.

Potpis autora