

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

**Senzorski sustav za praćenje ispravnosti
izvođenja sportskih vježbi pomoću rubnog AI
modela i ESP32 mikrokontrolera**

Lovro Šantek

Zagreb, kolovoz 2025.

Ovaj rad izrađen je na Fakultetu elektrotehnike i računarstva pod vodstvom prof. dr. sc. Hrvoja Džape i predan je na natječaj za dodjelu Rektorove nagrade u akademskoj godini 2024./2025.

Sadržaj

1.	Uvod	1
2.	Cilj rada i arhitektura predloženog rješenja.....	3
3.	Tehnologije i alati.....	6
3.1.	Strojno učenje	6
3.1.1.	EdgeAI i TinyML	7
3.1.2.	Klasifikacijski model.....	7
3.1.3.	Neuronska mreža	8
3.1.4.	Ulazni parametri neuronske mreže	10
3.2.	TensorFlow	11
3.3.	MQTT	12
4.	Sklopovlje.....	14
4.1.	Mikrokontroler ESP32.....	14
4.2.	Inercijski senzor.....	15
4.3.	Baterija i upravljački sustav baterije	17
4.4.	Kućište	18
4.5.	Povezivanje elektroničkih komponenti u sustav	20
5.	Razvijeno rješenje.....	22
5.1.	Nosivi uređaj	22
5.2.	Prikupljanje skupa podataka	23
5.3.	Programska podrška	24
5.3.1.	Treniranje modela strojnog učenja	24
5.3.2.	Programska podrška za ESP32 mikrokontroler.....	26
5.3.3.	Web sučelje za prikaz rezultata	30
6.	Rezultati.....	32

6.1.	Vježba A – Čučanj.....	32
6.2.	Vježba B – Iskorak	34
6.3.	Vježba C – Sklek	36
6.4.	Veličina i vrijeme izvođenja modela strojnog učenja i aplikacije.....	38
7.	Zaključak	40
	Zahvale	42
	Literatura	43
	Sažetak.....	46
	Summary.....	47

1. Uvod

Tjelesna aktivnost često se promiče kao ključ zdravog načina života i procjenjuje se da redovna tjelesna aktivnost može produljiti životni vijek do 6 godina [1]. Jednako važno kao sama tjelovježba je to da se ona izvodi ispravno. Neodgovarajuća tehnika vježbanja poznati je problem koji pogađa osobe svih razina tjelesne spremne [2]. Za početnike to često znači slabije rezultate, dok kod iskusnijih pojedinaca može dovesti do opterećenja zglobova, mišićne neravnoteže ili čak dugotrajnih ozljeda. U rehabilitacijskim uvjetima pacijenti često imaju poteškoće s ispravnim izvođenjem vježbi zbog nedostatka stručnog vodstva. Ovisnost o drugoj osobi ne samo da povećava troškove, već oduzima vrijeme svim uključenim stranama. Na taj način ograničava se potencijal za samostalnu tjelovježbu.

U ovom radu opisano je rješenje u obliku nosivog senzorskog uređaja koji koristi model strojnog učenja (engl. *Machine Learning*) ograničene veličine. Model strojnog učenja na temelju podataka s inercijske mjerne jedinice (engl. *Inertial Measurement Unit*, IMU) u stvarnom vremenu određuje izvodi li se odabrana vježba ispravno. Procjena (ne)ispravnosti izvedene vježbe prikazuje se u stvarnom vremenu na web aplikaciji koja je povezana s nosivim uređajem. Temelj ovog rješenja su model strojnog učenja i ugradbeni računalni sustav. Vrlo niska energetska potrošnja sustava i mogućnost da se vrlo mali model strojnog učenja izvodi direktno na mikrokontroleru čini ovo rješenje posebnim. Većina dosadašnjih znanstvenih istraživanja u području primjene nosivih senzorskih sustava fokusira se na klasifikaciju različitih vježbi, a rijetko na određivanje ispravnosti izvođenja vježbe, što dodatno ukazuje na inovativnost ovakvog pristupa [3][4][5].

U nastavku rada opisan je prototip nosivog uređaja koji na mikrokontroleru s ograničenim resursima određuje ispravnost izvođenja odabranih vježbi. Cijeli uređaj smješten je u posebno dizajnirano kućište prikladno za nošenje za vrijeme tjelovježbe. Točnost sustava testirana je za odabrane vježbe (čućanj, iskorak, sklek). Iako je funkcionalnost sustava demonstrirana na tri vježbe, sustav se može prilagoditi na bilo koju drugu odabranu vježbu treniranjem odgovarajućeg modela. Ovakva fleksibilnost sustava koji uz vrlo male resurse i jednostavnu arhitekturu razlikuje ispravno i neispravno izvođenje vježbe predstavlja odličan primjer efikasnog korištenja ograničenih resursa. U radu je opisana i arhitektura

sustava koja je primjenjiva na bilo koji sličan nosivi uređaj koji na temelju mjerenja u stvarnom vremenu obavlja klasifikaciju. Ovaj može poslužiti kao predložak i vodič kroz proces integracije modela strojnog učenja na ugradbena računala, čime se otvara cijeli spektar novih, inovativnih i donedavno nemogućih rješenja.

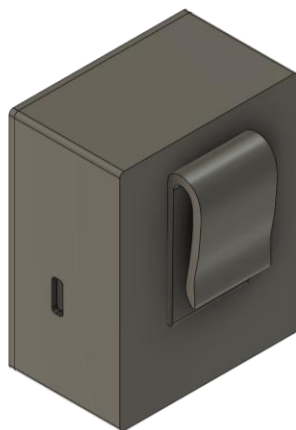
2. Cilj rada i arhitektura predloženog rješenja

Cilj ovog rada je projektirati sustav i izraditi prototip nosivog uređaja namijenjenog pružanju povratnih informacija u stvarnom vremenu o ispravnosti izvođenja sportskih vježbi. Sustav se temelji na modelu strojnog učenja koji, na osnovi podataka o pokretima tijela prikupljenim IMU senzorom, prepoznaje odstupanja od ispravne tehnike. Svaka vježba definirana je karakterističnim obrascem gibanja, a model procjenjuje ispravnost izvedbe usporedbom stvarnog pokreta s očekivanim uzorkom.

Položaj uređaja na tijelu ovisi o vrsti vježbe i odabire se tako da osigura relevantne podatke za pouzdanu procjenu izvedbe. Budući da se uređaj koristi u izravnom kontaktu s korisnikom tijekom vježbanja, prvi cilj bio je razviti bežično, baterijski napajano rješenje. Drugi cilj odnosio se na primjenu komponenti vrlo niske potrošnje energije, čime se omogućuje rad sustava kroz duži vremenski period, korištenjem što manje baterije. Treći cilj proizlazi iz ergonomske zahtjeva, odnosno iz činjenice da će uređaj biti na osobi za vrijeme izvođenja vježbe. To znači da je cilj prototipa maksimalno smanjiti fizičke dimenzije uređaja kako bi se osigurala nesmetana izvedba vježbi. Odluke o dizajnu sustava kao što su frekvencija uzorkovanja, položaj senzora na tijelu, digitalna obrada signala i broj inercijskih mjernih jedinica utemeljene su na zaključcima iz [6].

Konačan cilj sustava je jednostavno korištenje na način da osoba prikvači uređaj za odjevni predmet (primjerice hlače u visini pasa). Zatim se uređaj bežično povezuje s osobnim računalom na kojem se prikazuju rezultati u stvarnom vremenu. Kada osoba izvede vježbu na monitoru računala prikazuje se informacija o ispravnosti izvedene vježbe.

Problem određivanja ispravnosti izvođenja vježbe moguće je riješiti na različite načine. U radu [7] predloženo je rješenje u kojem se položaj ljudskog tijela u sve 3 prostorne dimenzije odrađuje pomoću kamera postavljenih oko osobe koja izvodi vježbu. Neki od tih sustava dodatno zahtijevaju da se na specifične točke na tijelu postave infracrveni markeri pomoću kojih kamere prate položaj ključnih točaka na ljudskom tijelu [7]. Nedostaci ovog pristupa su potreba za kalibracijom, niska otpornost na elektromagnetske smetnje te nepraktičnost upotrebe [8]. Takvi sustavi temelje se na obradi slika koje, iako sadrže mnogo korisnih informacija, zahtijevaju veću procesorsku snagu ili dodatno sklopovlje zbog velike količine podataka.



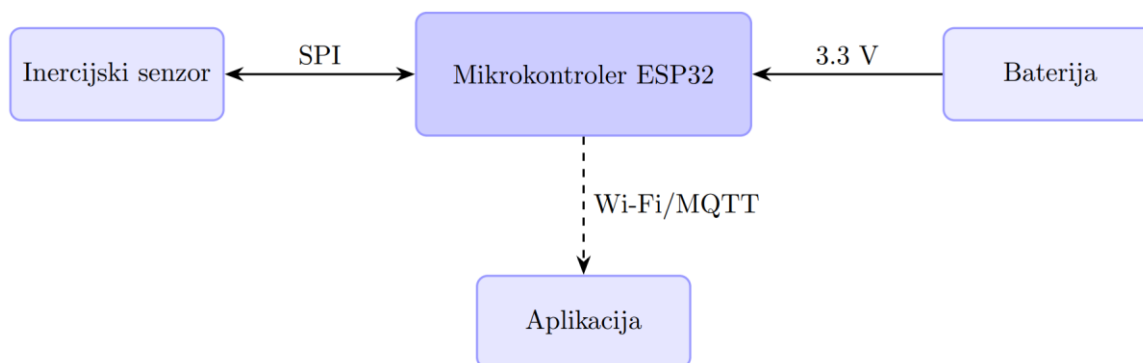
Slika 2.1 Prijedlog dizajna nosivog prototipa.

Općenito, ugradbeni sustavi koji rade sa slikama kao ulaznim podacima povećavaju minimalne resurse koji su potrebni za normalan rad. Fotografija niske kvalitete, rezolucije 320×240 piksela, komprimirana u JPEG formatu, obično zauzima nekoliko desetaka kilobajta memorije.

Pristup odabran u ovom radu temelji se na inercijskim senzorima (kao što su akcelerometar i žiroskop) prvenstveno zbog njihove niske potrošnje energije i jednostavnosti uporabe. Dodatna prednost proizlazi iz činjenice da je količina izlaznih podataka inercijske mjerne jedinice reda veličine 10 bajtova. Uspoređujući ovaj rezultat s fotografijom niske kvalitete, rezolucije 320×240 piksela, komprimirane u JPEG formatu, uočava se da inercijski senzori proizvode barem tri reda veličine manje podataka od fotografije, čime se značajno smanjuju potrebni procesorski resursi za njihovu obradu.

Većina radova koji obrađuju teme kao što su određivanje položaja ljudskog tijela pomoću inercijskih mjernih jedinica ili određivanje ispravnosti izvođenja vježbi temelje se na algoritmima koji ne uključuju modele strojnog učenja. Razvijena su i rješenja koja kombiniraju inercijske mjerne jedinice i markere (opto-inercijski sustavi), a jedno tako rješenje opisano je u [10]. Takav pristup zbog svoje složenosti nije prikladan za ovaj rad.

Primjer iz literature [9] pokazuje da se ispravnost izvođenja vježbi može pouzdano odrediti korištenjem samo jedne inercijske mjerne jedinice, što predstavlja i glavni cilj ovog rada. Premda se ispravnost izvođenja određenih vježbi može procijeniti korištenjem samo jedne senzorske jedinice, ovakav sustav ima inherentna ograničenja. Njegova funkcionalnost ograničena je na otkrivanje neispravnosti u izvođenju koje uzrokuju promjene gibanja u području struka, gdje je uređaj smješten, pa stoga nije moguće prepoznati pogrešne položaje ruku ili glave.



Slika 2.2 Arhitektura sustava.

Arhitektura opisana blok dijagramom prikazuje glavne funkcionalne komponente koje su potrebne za ispravan rad sustava. U centru sustava nalazi se mikrokontroler koji obavlja funkcije kao što su prikupljanje podataka s inercijske mjerne jedinice, digitalna obrada signala, zaključivanje modela strojnog učenja te u konačnici bežična komunikacija s aplikacijom za prikaz rezultata. Mikrokontroler je u ovom kontekstu ugradbeno računalo i služi svim komponentama i resursima kao zajednička točka.

Za mjerenje akceleracije i kutne brzine u sve tri prostorne koordinate koristi se inercijska mjerna jedinica. Podaci o akceleraciji i kutnoj brzini predstavljaju temelj sustava, budući da su to veličine na temelju kojih se procjenjuje ispravnost izvođenja vježbi. Također, inercijska mjerna jedinica jedini je način na koji sustav prikuplja informacije o načinu izvođenja vježbe. Za komunikaciju između mikrokontrolera i inercijske mjerne jedinice koristi se SPI serijski protokol opisan u poglavlju 4.2.

Sljedeća funkcionalna komponenta je baterija sa sklopovljem za upravljanje napajanjem mikrokontrolera. Puni se standardnim USB-C punjačem na 5 V, a omogućuje da uređaj bude potpuno bežičan, što je ključni zahtjev nosivih sustava.

Dio sustava koji ne postoji u obliku fizičke komponente jest web aplikacija namijenjena vizualizaciji rezultata. Po završetku svake procjene modela strojnog učenja, mikrokontroler bežično prenosi podatke o ispravnosti izvedene vježbe računalu, a oni se potom u stvarnom vremenu prikazuju na monitoru uz dodatna objašnjenja.

3. Tehnologije i alati

U ovom su poglavlju detaljno opisane temeljne tehnologije korištene u razvoju sustava, uz obrazloženje njihove prikladnosti u kontekstu praćenja ispravnosti izvođenja sportskih vježbi i ljudskog pokreta.

3.1. Strojno učenje

Problem procjene ispravnosti izvođenja sportskih vježbi u svojoj je osnovi klasifikacijski zadatak. Zadaća uređaja je na temelju akceleracije i kutne brzine inercijske mjerne jedinice pričvršćene na tijelo korisnika klasificirati vježbu kao ispravno ili neispravno izvedenu. Osim određivanja ispravnosti izvođenja vježbe, važno je i određivanje na koji način je vježba neispravno izvedena. Tako korisnik dobiva povratnu informaciju o tome što mora popraviti kod izvođenja vježbe. To znači da sustav mora biti u stanju izvedenu vježbu svrstati u jednu od unaprijed definiranih klasa: „*Ispravno izvedena vježba*“, „*Neispravno izvedena vježba zbog razloga A*“, „*Neispravno izvedena vježba zbog razloga B*“... Ovisno o broju načina na koji se vježba može napraviti neispravno, sustav na izlazu razlikuje odgovarajući broj rezultata klasifikacije.

Klasifikacija na temelju podataka s inercijske mjerne jedinice može se izvesti kao što je to napravljeno u [11]. U ovom radu je za rješavanje problema klasifikacije odabran pristup pomoću neuronske mreže. U zadnjih deset godina neuronske mreže i strojno učenje postaju sve prisutniji u znanstvenoj zajednici, a metode strojnog učenja pronašle su primjenu u klasifikaciji ljudskih pokreta [16]. Postoje brojni radovi koji koriste različite modele strojnog učenja za klasifikaciju ljudskog pokreta i u tim radovima nalazi se puno korisnih smjernica jer se radi o sličnom problemu kao u ovom radu [12][13]. U ovom radu naglasak je stavljen na određivanje ispravnosti izvođenja pokreta, što klasifikacijski problem čini složenijim od same klasifikacije potpuno različitih vježbi, ponajprije zbog suptilnih razlika između ispravno i neispravno izvedenih pokreta.

3.1.1. EdgeAI i TinyML

Pojam *Edge AI* označava primjenu algoritama umjetne inteligencije izravno na uređajima na rubu mreže, bez potrebe za obradom u oblaku. Takvi uređaji uključuju ugradbene sustave s ograničenim računalnim resursima, poput mikrokontrolera.

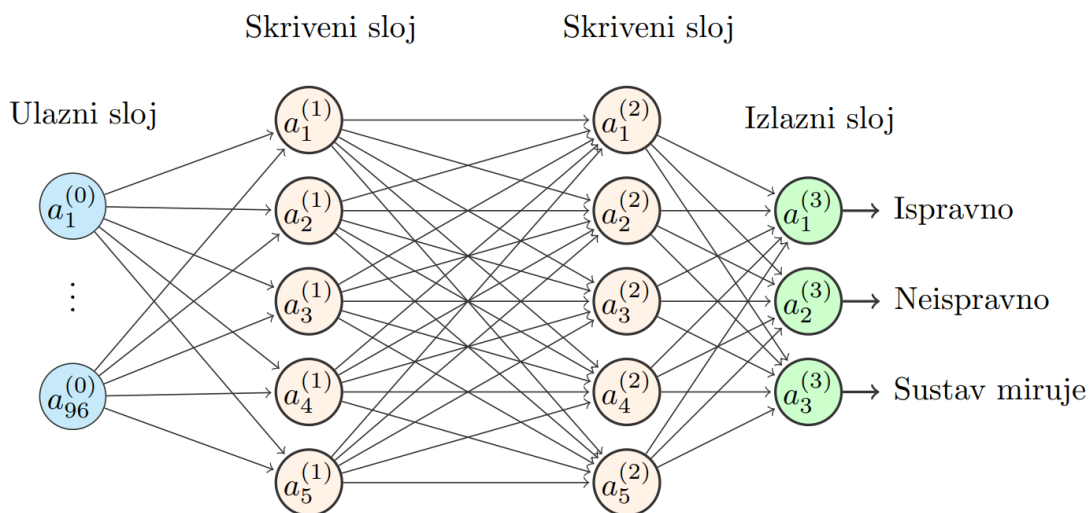
Zahvaljujući napretku u algoritmima strojnog učenja i sve većim mogućnostima ugradbenog sklopovlja, moguće je lokalno izvoditi jednostavnije modele, čime se omogućuje obrada podataka i donošenje odluka u stvarnom vremenu. Prednosti *Edge AI* pristupa uključuju manju latenciju, veću privatnost i sigurnost, te neovisnost o mrežnoj povezanosti. Svi podaci sa senzora ostaju unutar lokalnog sustava, što eliminira potrebu za slanjem informacija u oblak. Također, ovakav pristup povećava pouzdanost sustava, jer omogućuje rad i tijekom prekida veze s internetom [14].

TinyML definira se kao podskup *Edge AI*-a usmjeren na izvođenje modela strojnog učenja na uređajima s izrazito ograničenom memorijom i niskom potrošnjom energije. To je posebno važno u primjenama gdje je sustav baterijski napajan, kao što je to slučaj u ovom radu. Unatoč svim ograničenjima, uz ispravan odabir modela, moguće je izvesti korisne zadatke strojnog učenja direktno na mikrokontroleru. Uređaji namijenjeni *TinyML* primjenama obično posjeduju do nekoliko stotina kB radne memorije (SRAM) i do nekoliko MB flash memorije. Potrošnja energije im je niska, najčešće ispod 0,5 W. Cijena takvih mikrokontrolera kreće se oko nekoliko eura, što ih čini prikladnima za široku upotrebu [15].

U ovom radu primijenjen je koncept *Edge AI*-a, pri čemu su odabrani mikrokontroler i model strojnog učenja koji pripadaju kategoriji *TinyML*-a.

3.1.2. Klasifikacijski model

Klasifikacijski model temelji se na neuronskoj mreži. Na ulaz neuronske mreže dovodi se vektor čiji su elementi ulazni parametri. Neuronska mreža na svom izlazu daje procjenu vjerojatnosti pripadanja nekoj od unaprijed definiranih klasa. Model se sastoji od niza slojeva, organiziranih u arhitekturu specifičnu za dani zadatak.



Slika 3.1 Pojednostavljeni prikaz neuronske mreže korištene u ovom radu.

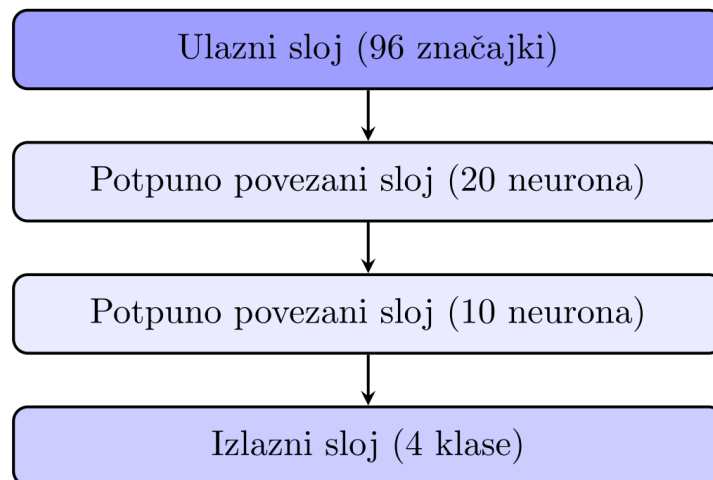
Uobičajeni slijed slojeva uključuje ulazni sloj, jedan ili više skrivenih slojeva te završni sloj. Posljednji sloj u klasifikacijskom zadatku obično je *softmax* sloj, koji za svaki uzorak izračunava raspodjelu vjerojatnosti po klasama. Arhitektura modela može se prilagoditi ovisno o složenosti podataka i ciljevima modeliranja. Za određivanje ispravnosti izvođenja sportskih vježbi korištena je neuronska mreža koju prikazuje Slika 3.1.

Tijekom treniranja, parametri mreže (težine i pristranosti) optimiziraju se metodom propagacije pogreške unatrag (engl. *backpropagation*) [19]. *Backpropagation* algoritam za svaki ulazni vektor računa vjerojatnost pripadnosti unaprijed definiranim klasama, uspoređuje je s referentnom klasom (koja je poznata unaprijed) te na temelju pogreške prilagođava parametre. Taj se proces ponavlja kroz više epoha, čime se model postupno prilagođava strukturi podataka.

3.1.3. Neuronska mreža

Slika 3.1 prikazuje pojednostavljenu neuronsku mrežu koja je u ovom radu korištena za klasifikaciju. Iako se radi o pojednostavljenom prikazu, stvarna neuronska mreža ne razlikuje se puno. Ulazni sloj sastoji se od 96 neurona, a svaki od njih prima jedan ulazni parametar tijekom procesa treniranja i klasifikacije. Nakon ulaznog sloja slijede dva potpuno povezana sloja s 20 neurona u prvom i 10 neurona u drugom sloju. Konačno, izlazni sloj neuronske mreže pretvara izlazne vrijednosti iz drugog potpuno povezanog sloja u vjerojatnosti pripadanju pojedinoj klasi („Sustav miruje“, „Ispravno izvedena vježba“, „Neispravno izvedena vježba“). Broj različitih klasa na izlazu je proizvoljan, a

definira se prije treniranja neuronske mreže. Za demonstracijske svrhe odabrane su klase koje prikazuju minimalnu funkcionalnost uređaja.



Slika 3.2 Blokovski prikaz slojeva neuronske mreže korištene u ovom radu.

Aktivacijska funkcija omogućuje ulaznom signalu poprimanje proizvoljnih vrijednosti, dok izlazne vrijednosti ostaju unutar strogo ograničenog raspona. Bez njih, neuronske mreže svodile bi se na jednostavne linearne modele. U skrivenim slojevima neuronske mreže korištena je ReLU (engl. *Rectified Linear Unit*) aktivacijska funkcija. ReLU je funkcija koja propušta samo pozitivne vrijednosti što unosi nelinearnost u sustav.

U izlaznom sloju (engl. *output layer*) korištena je *softmax* aktivacijska funkcija, koja pretvara izlazne vrijednosti mreže u distribuirane vjerojatnosti po klasama. Iako su ReLU i *softmax* funkcije opće prihvaćene funkcije u neuronskim mrežama, odabir je potkrijepljen i rezultatima iz [18].

Broj epoha za treniranje modela postavljen je na 40, što se pokazalo kao dovoljno za postizanje stabilne točnosti bez znakova prenaučivosti (engl. *overfitting*). Zadana vrijednost stope učenja (engl. *learning rate*) iznosi 0.001, što je uobičajena vrijednost za korišteni optimizator i pokazala se stabilnom tijekom procesa treniranja.

Priprema ulaznih parametara opisana je u poglavlju 3.1.4, a detaljna analiza treniranja i kvantizacije u poglavlju 3.2.

Razvijeni model strojnog učenja može se integrirati u sustav za praćenje ispravnosti izvođenja sportskih vježbi. Cijeli model strojnog učenja, koji uključuje obradu signala i klasifikator, zauzima 20-40 kB nakon kvantizacije, što ga čini prikladnim za ESP32

mikrokontroler s 4 MB *flash* memorije. Proces prikupljanja i obrade ulaznih podataka te treniranja i kvantizacije neuronske mreže proveden je na web aplikaciji *Edge Impulse* [17].

3.1.4. Ulazni parametri neuronske mreže

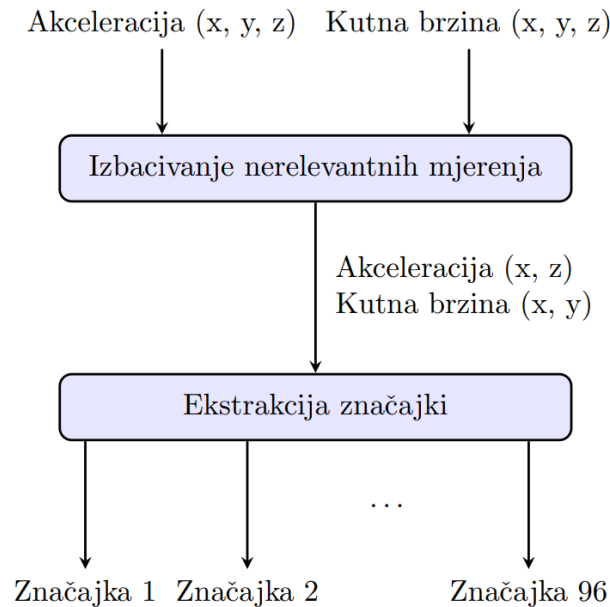
Kao što prikazuje Slika 3.1, na ulaz neuronske mreže dovode se ulazni parametri. Potrebno je naglasiti da ulazni parametri nisu isto što i izlazni podaci iz inercijske mjerne jedinice opisane u poglavlju 4.2.

Pri frekvenciji uzorkovanja od 100 Hz, inercijska mjerna jedinica očitava akceleraciju i kutnu brzinu u tri prostorne osi, što rezultira sa 600 mjernih vrijednosti u svakoj sekundi rada. Uređaj za praćenje ispravnosti izvođenja sportskih vježbi prikuplja mjerenja u razdoblju od dvije sekunde. Odnosno, pri svakoj klasifikaciji sustav koristi skup od 1200 vrijednosti prikupljenih iz inercijske mjerne jedinice. Iako je moguće konstruirati neuronsku mrežu s 1200 ulaznih neurona, to nije optimalno za primjenu u baterijski napajanom ugradbenom sustavu ograničene veličine. Rješenje odabrano u ovoj primjeni uključuje obradu signala i izvlačenje značajki iz sirovih podataka prije dovođenja na ulazni sloj neuronske mreže. U nastavku je opisano kako je inicijalnih 1200 vrijednosti reducirano na 96 ulaznih parametara.

Slika 3.3 prikazuje blok dijagram procesa ekstrakcije značajki. Na vrhu dijagrama nalazi se svih 1200 mjerenja akceleracije i kutne brzine s inercijske mjerne jedinice. Za vrijeme razvoja primijećeno je da ovisno o izvedenoj vježbi pojedina mjerenja ne doprinose točnosti klasifikacije, već samo unose dodatan šum u sustav. Kontraintuitivno, ali izbacivanjem pojedinih ulaznih signala točnost konačnog sustava može se povećati. Nakon ovog koraka broj uzoraka reduciran je za 200 po varijabli koja je izbačena. Slika 3.3 prikazuje izbacivanje mjerenja akceleracije u y-osi i kutne brzine oko z-osi (kao u poglavlju 6.1), što znači da na blok za digitalnu obradu signala dolazi 800 ulaznih vrijednosti, umjesto inicijalnih 1200.

Blok za digitalnu obradu signala propušta signale kroz niskopropusni filter šestog reda s gornjom graničnom frekvencijom 3 Hz. Vrsta filtra bira se prema tipu vježbe koja se klasificira. Ako se radi o vježbi koja nema naglih pokreta prikladan je niskopropusni filter, a u slučaju da se radi o vježbi s dinamičnim elementima prikladniji je visokopropusni filter. Nakon prolaska kroz filter signali se obrađuju brzom Fourierovom transformacijom

(FFT) [20]. Nakon obrade svih signala sustav izdvaja 96 značajki koje dolaze na ulaz neuronske mreže, a one su: srednja vrijednost (RMS) signala, spektralna snaga u raznim rasponima frekvencija, asimetrija, spljoštenost, spektralna asimetrija.



Slika 3.3 Blok dijagram procesa ekstrakcije značajki.

3.2. TensorFlow

Za treniranje i evaluaciju klasifikacijskog modela korišten je *TensorFlow*, popularni okvir otvorenog koda za izgradnju i implementaciju modela strojnog učenja razvijen od strane tvrtke *Google* [21]. *TensorFlow* omogućuje jednostavnu konstrukciju neuronskih mreža, efikasno treniranje modela te njihovu optimizaciju i implementaciju na uređaje ograničenih resursa, kao što su mikrokontroleri. Podržava različite razine apstrakcije, a u ovom radu model je razvijen pomoću Keras API-ja (engl. *Application Programming Interface*).

Treniranje modela provedeno je nad vlastitim skupom podataka koji je prikupljen specifično za zadatak prepoznavanja ispravnosti izvođenja sportskih vježbi. Podaci su prikupljeni mjerenjem na nekoliko različitih osoba, ovisno o vježbi. Skup podataka podijeljen je u omjeru 75:25, pri čemu se 75 % podataka koristi za učenje parametara modela, dok se preostalih 25 % koristi za evaluaciju generalizacijske sposobnosti modela. Detaljan opis procesa prikupljanja skupa podataka za treniranje i validaciju modela nalazi se u poglavlju 5.2.

Za optimizaciju modela korišten je Adam optimizacijski algoritam koji kombinira prednosti algoritama *AdaGrad* i *RMSProp*. Adam koristi prilagodljive stope učenja za svaki parametar, što doprinosi bržem i stabilnijem konvergiranju tijekom treniranja, osobito kod kompleksnijih modela i heterogenih podataka.

Po završetku treniranja, neuronska mreža optimizirana je metodom kvantizacije (engl. *quantization*) kako bi se smanjila njegova veličina u memoriji i povećala učinkovitost izvođenja na mikrokontroleru. Kvantizacija predstavlja tehniku smanjenja memorijskog otiska i računalne složenosti modela zamjenom 32-bitnih brojeva s pomičnim zarezom (*float32*) s 8-bitnim cjelobrojnim vrijednostima (*int8*). Time se postiže značajno smanjenje veličine modela i ubrzanje izvođenja klasifikacije na mikrokontrolerima, uz minimalan utjecaj na točnost. U ovom radu korišten je *int8* kvantizacijski pristup, koji je kompatibilan s većinom ugradbenih sustava i podržan unutar *TensorFlow Lite* biblioteke [21].

Konačni kvantizirani model, koji uključuje i digitalnu obradu signala te klasifikacijski sloj, zauzima 20-40 kB memorije, ovisno o vrsti vježbe, što ga čini prikladnim za implementaciju na mikrokontroleru ESP32. Ova optimizacija omogućuje izvođenje klasifikacije u stvarnom vremenu na uređaju s ograničenim resursima, bez potrebe za vanjskim poslužiteljima ili cloud infrastrukturom, što je i bio cilj pri projektiranju sustava.

3.3. MQTT

MQTT (engl. *Message Queuing Telemetry Transport*) je protokol za razmjenu poruka temeljen na načelu objave i pretplate (engl. *publish/subscribe*). Izvorno je razvijen za uređaje s ograničenim resursima i nepouzdanim mrežnim vezama, MQTT se danas široko koristi u sustavima Interneta stvari (IoT), gdje su niska latencija, pouzdanost i niska potrošnja energije ključni zahtjevi [22].

MQTT protokol organiziran je oko posrednika poruka (engl. *broker*), koji upravlja svim komunikacijama između izdavatelja poruka (engl. *publisher*) i pretplatnika (engl. *subscriber*). Uređaji se ne povezuju izravno jedan na drugi, već svi podaci prolaze kroz MQTT *broker*, što pojednostavljuje upravljanje povezanošću i povećava robusnost sustava. MQTT koristi TCP/IP sloj te podržava razine kvalitete usluge koje omogućuju kontrolu nad pouzdanošću isporuke poruka. Zbog jednostavnosti, male količine potrebnih podataka

i niskih zahtjeva za propusnošću, MQTT je pogodan za ugradbene sustave i komunikaciju u stvarnom vremenu.

U ovom radu MQTT se koristi kao glavni komunikacijski protokol između mikrokontrolera ESP32 i lokalnog poslužitelja. Sustav za praćenje ispravnosti izvođenja sportskih vježbi obrađuje podatke i provodi klasifikaciju direktno na uređaju. Nakon što neuronska mreža producira rezultat, rezultat klasifikacije („ispravno“ ili „neispravno“) šalje se putem MQTT protokola prema poslužitelju. Poruke se šalju unutar pretplaćenog MQTT kanala naziva `motion_tracker/exercise_result`.

Za lokalno upravljanje porukama korišten je *Mosquitto* MQTT broker. *Mosquitto* je posrednik koji omogućuje pokretanje MQTT brokera na lokalnom računalu. U konfiguracijskoj datoteci `mosquitto.conf` omogućeno je slušanje na standardnom MQTT portu 1883 i dopušten je pristup bez autentifikacije što je prikladno za lokalne testne svrhe.

Sadržaj objavljen putem MQTT-a zatim prima lokalni *Node.js* poslužitelj. Taj poslužitelj koristi MQTT biblioteku za *Node.js* kako bi se pretplatio na navedeni kanal, te po primitku poruke isti rezultat prosljeđuje putem *WebSocket* protokola prema korisničkom sučelju. Na taj način omogućeno je prikazivanje rezultata klasifikacije u stvarnom vremenu putem web aplikacije.

Komunikacijski tok unutar sustava može se sažeti na sljedeći način. Mikrokontroler ESP32 objavljuje rezultat klasifikacije putem MQTT protokola. *Mosquitto* broker, koji djeluje kao posrednik, prima te poruke i prosljeđuje ih dalje pretplatnicima. *Node.js* poslužitelj pretplaćen je na odgovarajući MQTT kanal i po primitku poruke šalje njezin sadržaj *web* klijentima putem *WebSocket* protokola. *Web* sučelje u pregledniku zatim prikazuje rezultat klasifikacije u stvarnom vremenu, čime se omogućuje povratna informacija korisniku u stvarnom vremenu.

U okviru ovog sustava MQTT se pokazao optimalnim rješenjem zbog niske potrošnje resursa, jednostavnosti uporabe i pouzdane podrške za rad u lokalnim mrežama.

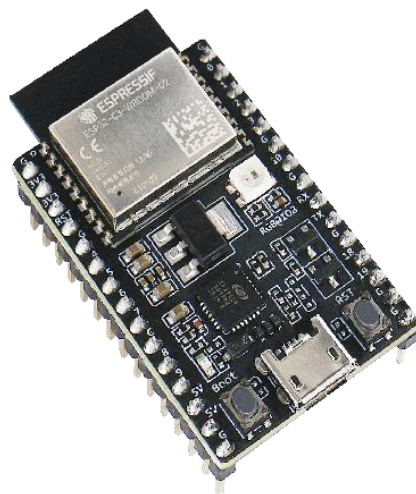
4. Sklopovlje

4.1. Mikrokontroler ESP32

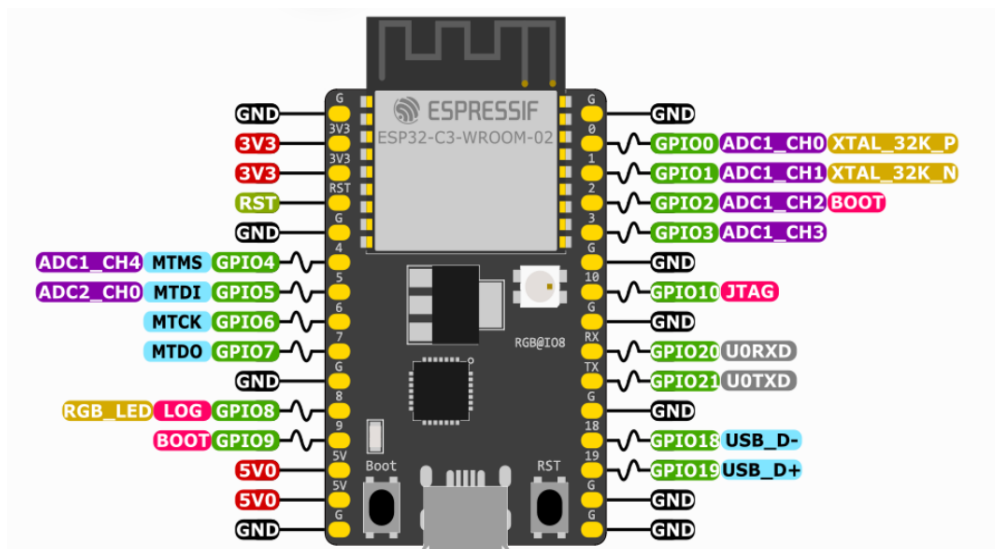
Za realizaciju sustava korišten je mikrokontroler ESP32, odnosno ESP32-C3-DevKitC-02 razvojna pločica koju prikazuje Slika 4.1. ESP32 je 32-bitni mikrokontroler razvijen od strane tvrtke *Espressif Systems* [23]. Zbog mogućnosti korištenja bežičnih tehnologija (*Wi-Fi* i *Bluetooth*) ESP32 je vrlo često korišten za projekte koji zahtijevaju nisku potrošnju i bežičnu povezanost. S obzirom na zahtjeve rada, ESP32 razvojna pločica pokazala se kao prikladan odabir za realizaciju prototipa opisanog sustava.

Tehničke karakteristike mikrokontrolera relevantne za razvoj ovog sustava uključuju 32-bitni RISC-V procesor s jednom jezgrom i frekvencijom takta od 80 MHz, 400 kB SRAM memorije, te 4 MB ugrađene *flash* memorije povezanom putem *Quad* SPI sučelja. Mikrokontroler podržava bežičnu komunikaciju sukladno IEEE 802.11b/g/n standardima, a integrirana je i PCB antena za prijenos podataka putem Wi-Fi mreže. Radni napon kreće se u rasponu od 3,0 V do 3,6 V, dok struja u aktivnom Wi-Fi načinu pri 2,4 GHz iznosi maksimalno 280 mA.

ESP32 komunicira s osobnim računalom putem MQTT protokola, a za komunikaciju s inercijskom mjernom jedinicom koristi SPI protokol. Obje komunikacijske metode podržane su bez potrebe za dodatnim elektroničkim komponentama.



Slika 4.1 ESP32-C3-DevKitC-02 razvojna pločica. Preuzeto s [23].



Slika 4.2 Raspored priključaka razvojne pločice ESP32-C3-DevKitC-02. Preuzeto s [23].

U ovom koraku potrebno je provjeriti je li veličina *flash* memorije na mikrokontroleru dovoljna da pohrani cijelu neuronsku mrežu ostavljajući dovoljno slobodnog memorijskog prostora za ostale dijelove programa. Prema podacima iz Poglavlja 3.1.3, veličina neuronske mreže iznosi par desetaka *kB*. To znači da će *flash* memorija ESP32 mikrokontrolera od 4 MB biti više nego dovoljna za ispravan rad sustava.

4.2. Inercijski senzor

Brojna istraživanja pokazuju da se na temelju podataka prikupljenih inercijskom mjernom jedinicom, poput akceleracije i kutne brzine, može klasificirati vrsta izvođene vježbe [4][5][9]. U ovom radu pristup je dodatno proširen te će se isti podaci koristiti u procjeni ispravnosti izvođenja samih vježbi.

Za potrebe ovog rada korišten je IMU senzor ICM-20948 proizvođača TDK, integriran na razvojnoj pločici *9DOF2 Click* tvrtke Mikroelektronika [25]. Inercijska mjerna jedinica sadrži tri senzora: akcelerometar, žiroskop i magnetometar. U okviru ovog rada korišteni su akcelerometar i žiroskop.

Akcelerometar mjeri ukupnu akceleraciju u tri ortogonalne osi (X, Y, Z), što uključuje komponentu gravitacijske sile i inercijsku komponentu uzrokovanu ubrzanjem sustava. U sustavu za praćenje ispravnosti izvođenja sportskih vježbi inercijska komponenta akceleracije nastaje pri pokretu za vrijeme izvođenja vježbe. Akcelerometar je izveden pomoću MEMS tehnologije i ima digitalni izlaz za svaku os. Moguće je mjeriti ubrzanja do ± 16 g, što je primjereno za sustav u kojem se nalazi [25].

Žiroskop mjeri kutne brzine rotacije oko pojedinih osi. Temelji se na Coriolisovu efektu te koristi MEMS tehnologiju. Raspon mjerenja kutne brzine može se programirati u granicama ± 250 dps, ± 500 dps, ± 1000 dps i ± 2000 dps. Svaki kanal ima integriran 16-bitni analogno-digitalni pretvornik (ADP) [25].

Razvojna pločica *9DOF2 Click* odabrana je zbog integriranog senzora ICM-20948 i podrške za komunikaciju putem SPI protokola. Pločica je kompatibilna s mikrokontrolerom ESP32 i zahtjeva napajanje od 3,3 V. Dimenzije pločice su 28,6 mm $\times$ 25,4 mm $\times$ 12 mm, a masa iznosi 18 g. Njezine dimenzije i masa čine je pogodnom za ugradnju u sustav za praćenje ispravnosti izvođenja sportskih vježbi.

Struja napajanja pločice, prema tehničkim specifikacijama, iznosi 3,11 mA [25].

Slika 4.3 i Slika 4.4 prikazuju vanjske priključke pločice *9DOF2 Click*. Korištena su 4 priključka za SPI komunikaciju (CS, SCK, SDO, SDI) te 2 priključka za napajanje pločice (3,3 V, GND). Dodatni priključci SYN i INT ne koriste se.



Slika 4.3 Pločica *9DOF2 Click*, prednja i stražnja strana. Preuzeto s [25].

Notes	Pin	mikroBUS				Pin	Notes
	NC	1	AN	PWM	16	SYN	External sync
	NC	2	RST	INT	15	INT	Interrupt
SPI Chip Select	CS	3	CS	RX	14	NC	
SPI Clock	SCK	4	SCK	TX	13	NC	
SPI Data OUT	SDO	5	MISO	SCL	12	NC	
SPI Data IN	SDI	6	MOSI	SDA	11	NC	
Power Supply	3.3V	7	3.3V	5V	10	NC	
Ground	GND	8	GND	GND	9	GND	Ground

Slika 4.4 Vanjski priključci pločice *9DOF2 Click*. Preuzeto s [25]. Crvenim pravokutnikom označeni su priključci korišteni za povezivanje pločice s ESP32 mikrokontrolerom.

4.3. Baterija i upravljački sustav baterije

S obzirom na prijenosnu prirodu sustava za praćenje ispravnosti izvođenja sportskih vježbi, ključno je da uređaj bude napajan baterijski te da ga je moguće jednostavno i sigurno napuniti nakon pražnjenja. To omogućuje korištenje uređaja bez ovisnosti o vanjskom napajanju, što je osobito važno za primjene u stvarnim uvjetima, poput vježbanja u sportskim ili rehabilitacijskim okruženjima.

Za napajanje sustava odabrana je litij-ionska baterija kapaciteta 900 mAh i nazivnog napona 3,7 V. Prema podacima iz poglavlja 4.1 i 4.2, potrošnja struje sustava iznosi maksimalno 300 mA. Na temelju kapaciteta baterije i struje koja je potrebna za ispravan rad računa se vrijeme koje sustav može kontinuirano raditi bez punjenja:

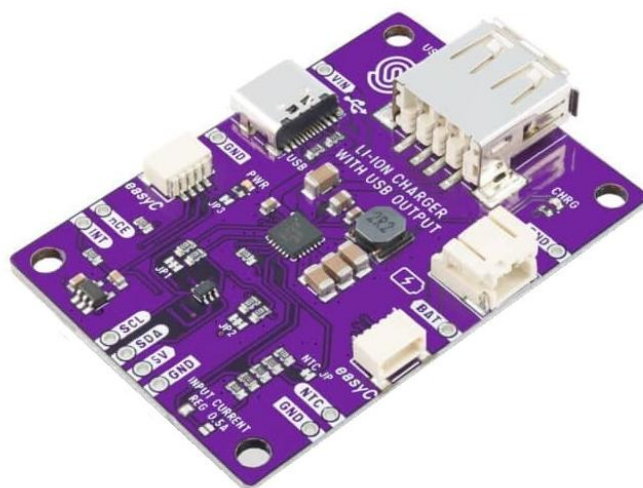
$$t_{\text{op}} = \frac{C_{\text{bat}}}{I_{\text{ESP32}}} = \frac{900 \text{ mAh}}{300 \text{ mA}} = 3 \text{ h}$$

Očekivano vrijeme rada sustava s punom baterijom iznosi tri sata, što je dovoljno za provođenje više mjernih sesija bez potrebe za ponovnim punjenjem.

Dimenzije baterije (75×48×26 mm) omogućuju jednostavnu integraciju u kućište prototipa bez značajnog povećanja ukupnog volumena uređaja.

Za punjenje baterije i upravljanje napajanjem koristi se upravljačka pločica temeljena na integriranom krugu BQ24295 (Slika 4.5). Pločica je razvijena u tvrtki Soldered Electronics [26]. Ova upravljačka jedinica omogućuje punjenje baterije putem standardnog USB-C priključka, što dodatno povećava praktičnost korištenja. Na kućištu uređaja nalazi se otvor za punjenje, namijenjen spajanju na uobičajeni USB-C punjač za mobilne uređaje.

BQ24295 omogućuje punjenje strujom do 3 A, preciznu regulaciju napona i struje, te podršku za različite tipove USB punjača. Osim toga, podržava rad i bez baterije (NVDC način), ima ugrađenu zaštitu od prenapona i pregrijavanja, nisku potrošnju struje u mirovanju (20 µA) te mogućnost komunikacije putem I2C sučelja za detaljan nadzor statusa punjenja. Ugrađene sigurnosne značajke, poput temperaturnog nadzora i zaštite od prekomjerne struje, osiguravaju sigurno korištenje sustava.



Slika 4.5 Kontroler za litij-ionsku bateriju temeljen na BQ24295. Preuzeto s [26].

4.4. Kućište

Za potrebe sustava razvijenog u ovom radu bilo je potrebno osmisliti i izraditi fizičko kućište prilagođeno specifičnostima sustava i korisničke upotrebe. Kućište je dizajnirano u alatu *Autodesk Fusion 360*. Cilj dizajna kućišta bio je osigurati siguran smještaj svih elektroničkih komponenti, stabilnost uređaja tijekom izvođenja vježbe te jednostavnu i intuitivnu upotrebu. Osim jednostavnosti korištenja, koja je ključna iz perspektive krajnjeg korisnika, glavni fokus pri razvoju kućišta bio je na funkcionalnosti u smislu zadatka klasifikacije ispravnosti izvedenih vježbi.

Kućíšte je oblikovano tako da omogućuje postavljanje na dio tijela koji ima ključnu ulogu u određivanju ispravnosti izvođenja vježbe. Budući da se u ovom radu analiziraju vježbe koje dominantno uključuju pokrete donjih ekstremiteta, najprikladnije mjesto za postavljanje uređaja je područje pojasa, odnosno hlaća u visini bokova. S obzirom na to, kućište je konstruirano s integriranom kvačicom koja omogućuje brzo i stabilno pričvršćivanje na odjeću, bez dodatnih remena ili vezivanja.

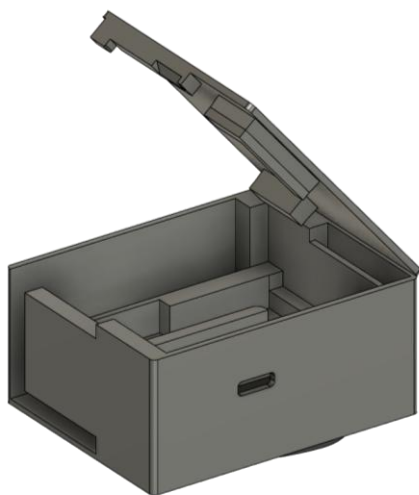
Slika 4.7 prikazuje kvačicu koja je dio kućišta te omogućuje jednostavno postavljanje i uklanjanje uređaja tijekom svakodnevne upotrebe, bez narušavanja udobnosti korisnika niti pouzdanosti mjerenja.

Dizajn kućišta u potpunosti je prilagođen dimenzijama korištene ESP32 pločice te ostalih komponenti navedenih u poglavlju 4. Na kućištu su predviđeni precizno oblikovani otvori, uključujući USB-C utor za punjenje baterije, kao i otvor za izravan pristup priključku na

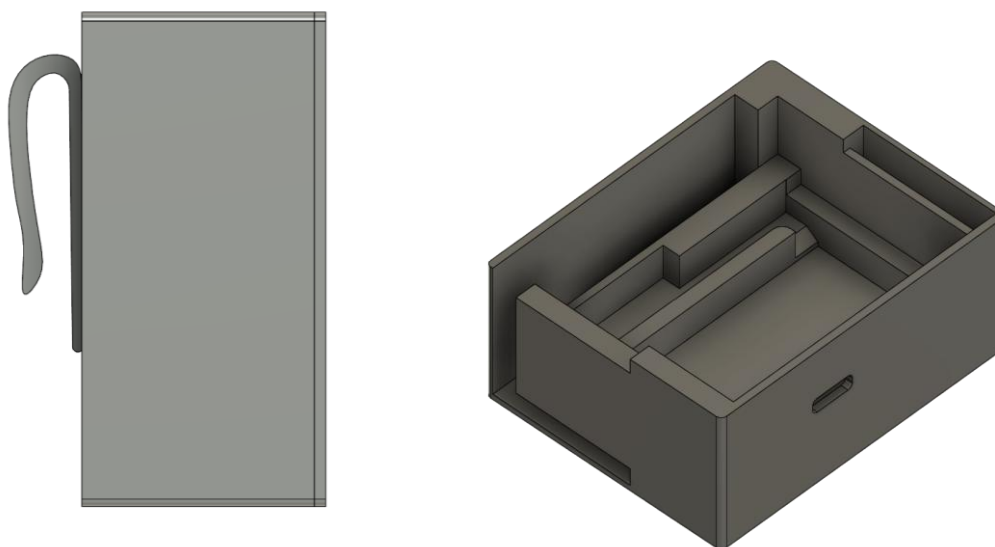
ESP32 mikrokontroleru. Time je omogućeno punjenje uređaja bez potrebe za vađenjem komponenti iz kućišta, čime se smanjuje mogućnost neispravnog rukovanja i pojednostavljuje svakodnevna uporaba.

Kućiste je ispisano metodom FDM (engl. *Fused Deposition Modeling*) korištenjem PLA (polilaktična kiselina) plastike. PLA je najpoznatiji materijal za FDM 3D ispis poznat po lakoći ispisa, dobroj dimenzijskoj stabilnosti i dovoljnoj mehaničkoj otpornosti za primjenu u ovom radu. Ispis je proveden na 3D printeru BambuLab P1S, pri čemu su korištene postavke sloja visine 0,2 mm, ispunjenosti 20 %, te brzinom ispisa od 200 mm/s.

Slika 4.6 i Slika 4.7 prikazuju konačni dizajn kućišta, uključujući vidljive otvore za priključke i način smještanja uređaja unutar kućišta. Oblik i dimenzije kućišta prilagođeni su prema ergonomskim zahtjevima krajnje primjene, s ciljem osiguravanja stabilnog položaja uređaja tijekom mjerenja i bez smetnji u izvođenju vježbi.



Slika 4.6 Kućište s otvorenim poklopcem.



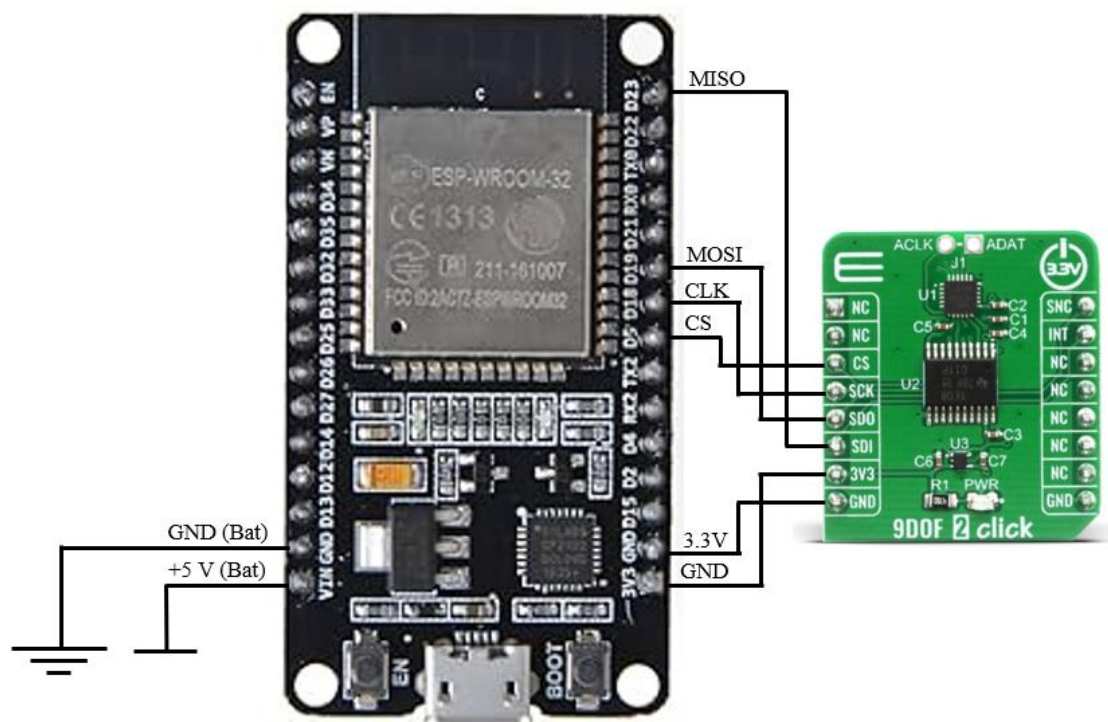
Slika 4.7 Bočni prikaz kućišta i kvačice (lijevo), kućište bez poklopca (desno).

4.5. Povezivanje elektroničkih komponenti u sustav

Za ispravan rad sustava za praćenje ispravnosti izvođenja sportskih vježbi, potrebno je ostvariti pouzdanu i funkcionalnu vezu između svih sklopovskih komponenti. To uključuje povezivanje senzorske mjerne jedinice, mikrokontrolera, napajanja te upravljačkih sklopova za punjenje i zaštitu baterije. U ovom poglavlju opisuje se način povezivanja navedenih komponenti, te raspored priključaka korištenih na prototipskoj pločici. Posebna pažnja posvećena je osiguravanju stabilnosti napajanja, ispravnoj komunikaciji između modula te minimizaciji smetnji koje bi mogle utjecati na točnost mjerenja.

Slika 4.8 jasno prikazuje način međusobnog povezivanja elektroničkih komponenti. Na priključke *VIN* i *GND* ESP32 razvojne pločice spojeni su odgovarajući priključci kontrolera baterije. Na taj način osigurano je sigurno napajanje sustava iz baterije.

Pločica *9DOF2 Click* na kojoj se nalaze akcelerometar i žiroskop napajana je direktno priključcima *3V3* i *GND* iz razvojne pločice mikrokontrolera, a ostala četiri priključka spojena su prema standardu SPI komunikacijskog protokola. Slika 4.2 jasno definira koji priključci razvojne pločice su namijenjeni za SPI serijsku komunikaciju.



Slika 4.8 Shema povezivanja ESP32 razvojne pločice i senzora.

5. Razvijeno rješenje

5.1. Nosivi uređaj

Rezultat spajanja svih komponenti je uređaj s integriranim sustavom za praćenje ispravnosti izvođenja sportskih vježbi. Slika 5.1 prikazuje unutrašnjost kućišta koje sadrži komponente povezane kao što je opisano u poglavlju 4.5. Dimenzije uređaja su 52 x 65 x 33 mm, a masa iznosi 80 grama. Iako se radi o prototipu, uređaj je prikladnih dimenzija i mase da se bez teškoća postavlja na osobu i koristi za vrijeme izvođenja vježbe. Uređaj radi bežično, a nakon praznjenja bateriju je moguće napuniti standardnim USB-C kabelom.

Budući da je uređaj korišten u istraživačke svrhe, sa stražnje strane ostavljena je mogućnost direktnog spajanja na ESP32 mikrokontroler radi lakšeg testiranja. U slučaju korištenja žičanog priključka moguće je koristiti serijsku komunikaciju za otklanjanje grešaka i brže prikupljanje podataka. Slika 5.2 prikazuje uređaj koji je spreman za korištenje i postavljen na hlače korisnika.



Slika 5.1 Unutarnji raspored komponenti u kućištu uređaja bez poklopca. Baterija se nalazi ispod svih komponenti i nije vidljiva na slici.



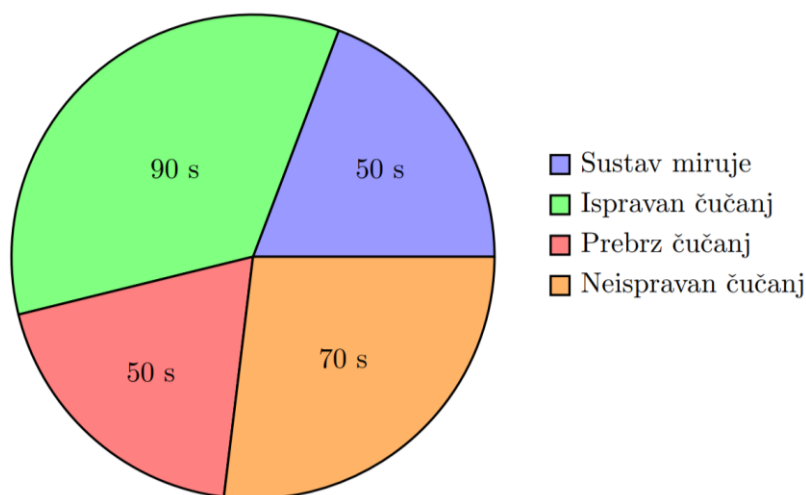
Slika 5.2 Uređaj postavljen na hlače korisnika.

5.2. Prikupljanje skupa podataka

Tek nakon što je prototip uređaja potpuno razvijen i spreman za upotrebu, moguće je započeti prikupljanje skupa podataka za treniranje i validaciju neuronske mreže. Proces prikupljanja podataka obuhvatio je izvođenje odabranih vježbi od strane više ispitanika, pri čemu je uređaj bio postavljen na struk (Slika 5.2) kako bi se osiguralo konzistentno mjerenje. Definicije ispravno izvedenih vježbi promatranih u ovom radu preuzete su iz radova [27][29][30]. Skupovi podataka prikupljeni su na 2-5 različitih osoba, ovisno o vrsti vježbe.

Za svaku vježbu zabilježeni su primjeri ispravnog i neispravnog izvođenja, čime je omogućeno stvaranje raznovrsnog i uravnoteženog skupa podataka. Svaka osoba koja je sudjelovala u prikupljanju morala je generirati podatke za sve definirane klase: „*Ispravno izvedena vježba*“, „*Neispravno izvedena vježba zbog razloga A*“, „*Neispravno izvedena vježba zbog razloga B*“... Redoslijed izvođenja bio je unaprijed definiran.

Ispitanik je najprije morao izvesti vježbu ispravno, zatim ju ponoviti prema uputi s pogreškom tipa A, a potom s pogreškom tipa B i tako sve dok nisu prikupljeni podaci za svaku definiranu klasu. Na taj način osigurano je da svaka osoba doprinese ravnomjernom broju primjera za sve klase. Tijekom snimanja vodilo se računa o uključivanju različitih brzina i intenziteta pokreta kako bi model mogao generalizirati stvarne uvjete korištenja. Nakon završetka snimanja, isti proces sustavno je ponovljen i za validacijski skup podataka. Prikupljeni zapisi zatim su označeni, organizirani u klase te podijeljeni na skup za treniranje i validaciju u omjeru 75/25. Takav pristup omogućio je uravnoteženost podataka i dovoljan broj primjera za učenje neuronske mreže, kao i provjeru njezine točnosti. Na primjeru čučnja [28], prikupljeni skup podataka ilustrira Slika 5.3.



Slika 5.3 Podjela skupa podataka prema klasama za treniranje.

5.3. Programska podrška

Cjelokupni izvorni kod, zajedno s uputama za korištenje, dostupan je na mrežnom repozitoriju [31].

5.3.1. Treniranje modela strojnog učenja

Za implementaciju i treniranje neuronske mreže korišten je programski jezik *Python* te biblioteka *TensorFlow* [21] s integriranim *Keras* API-jem. Korišten je *Sequential* model, koji omogućuje jednostavno definiranje arhitekture mreže kao niza slojeva. Od *Keras*ovih slojeva primijenjen je *Dense layer* (potpuno povezani sloj), dok je za optimizaciju parametara mreže korišten optimizator *Adam* iz *TensorFlow*-ovog modula *legacy*. Nakon

postavljanja hiperparametara modela i pripreme podataka programski je definirana arhitektura modela strojnog učenja opisana u poglavlju 3.1.3.

```
model = Sequential()

model.add(Dense(20, activation='relu',
               activity_regularizer=tf.keras.regularizers.l1(0.00001)))

model.add(Dense(10, activation='relu',
               activity_regularizer=tf.keras.regularizers.l1(0.00001)))

model.add(Dense(classes, name='y_pred',
               activation='softmax'))
```

Odsječak koda 1

Nakon definiranja arhitekture neuronske mreže pokrenut je optimizator, a odmah nakon i samo treniranje modela. Kod koji pokreće optimizator i treniranje modela prikazan je u nastavku.

```
opt = Adam(learning_rate=LEARNING_RATE, beta_1=0.9,
           beta_2=0.999)

callbacks.append(BatchLoggerCallback(BATCH_SIZE,
                                     train_sample_count, epochs=EPOCHS,
                                     ensure_determinism=ENSURE_DETERMINISM))

model.compile(loss='categorical_crossentropy', optimizer=opt,
              metrics=['accuracy'])

model.fit(train_dataset, epochs=EPOCHS,
        validation_data=validation_dataset, verbose=2,
        callbacks=callbacks)
```

Odsječak koda 2

Važno je napomenuti kako je model strojnog učenja koji je treniran na ovaj način funkcionalan točno za jednu vježbu. Za svaku novu vježbu koja se želi klasificirati potrebno je skupiti nova mjerenja i istrenirati model na podacima karakterističnim za vježbu.

Predložak *Python* programskog koda navedenog u ovom poglavlju preuzet je sa [17]. Rezultat treniranja modela strojnog učenja je C++ biblioteka koja u sebi sadrži istrenirani model strojnog učenja, kod za obradu podataka i sve funkcije za klasifikaciju podataka.

Takva biblioteka se jednostavno može uključiti u standardni ESP32 projekt. Struktura biblioteke prikazana je sljedećim dijagramom.

```
motion-quality-tracker-library/  
├── model-parameters  
├── tflite-model  
└── edge-impulse-sdk
```

5.3.2. Programska podrška za ESP32 mikrokontroler

U ovom poglavlju opisana je programska podrška razvijena za mikrokontroler ESP32 u sklopu sustava za praćenje ispravnosti izvođenja sportskih vježbi. Projekt je organiziran u strukturu direktorija kako bi se omogućilo jednostavno održavanje, proširivanje i ponovna uporaba koda. Glavna programska logika implementirana je u jeziku C/C++ koristeći razvojni okvir ESP-IDF.

Struktura projekta uključuje odvojene direktorije za glavnu aplikaciju, pomoćne module, model parametara, trenirani model neuronske mreže te vanjsku biblioteku opisanu u prethodnom poglavlju. Takva organizacija omogućuje modularan pristup razvoju, pri čemu su funkcionalnosti poput upravljanja IMU senzorom i komunikacije putem MQTT protokola izdvojene u zasebne datoteke i direktorije.

```
motion-quality-tracker/  
├── main/  
│   ├── main.cpp  
│   ├── CMakeLists.txt  
│   ├── inc/  
│   │   ├── IMU_driver.h  
│   │   └── mqtt_module.h  
│   └── src/  
│       ├── IMU_driver.c  
│       └── mqtt_module.c  
├── model-parameters/  
├── tflite-model/  
└── edge-impulse-sdk/
```

Datoteke `IMU_driver.c` i `IMU_driver.h` zajedno čine modul zadužen za upravljanje inercijskom mjernom jedinicom putem SPI serijskog protokola na mikrokontroleru ESP32. Datoteka `IMU_driver.h` sadrži deklaracije funkcija, definicije registara i

konfiguracijske makroe potrebne za inicijalizaciju i rad senzora, kao i definiciju SPI priključaka koji se koriste za komunikaciju. U njoj su specificirane adrese registara za očitavanje podataka akcelerometra i žiroskopa, konstante osjetljivosti senzora te tip enumeracije `IMU_status_t` koji standardizira povratne vrijednosti funkcija modula.

S druge strane, `IMU_driver.c` implementira funkcionalnosti deklarirane u zaglavlju. To uključuje inicijalizaciju SPI sabirnice i samog IMU senzora, čitanje i pisanje registara te dohvat mjernih podataka akcelerometra i žiroskopa u fizičkim jedinicama (g i °/s). Modul omogućuje apstrakciju komunikacije sa senzorom, čime se glavna aplikacija rasterećuje od detalja implementacije SPI protokola te se osigurava modularnost i lakša nadogradnja sustava.

Najvažnije funkcije `IMU_driver` modula navedene su u nastavku.

- `IMU_status_t IMU_init()` - inicijalizira IMU senzor i SPI komunikaciju. Funkcija konfigurira SPI sabirnicu, dodaje uređaj i budi senzor zapisom u odgovarajući registar za upravljanje napajanjem. Ova funkcija je ključna jer bez ispravne inicijalizacije podaci senzora ne bi bili dostupni.
- `uint8_t read_register(uint8_t address)` - omogućuje čitanje vrijednosti iz pojedinog registra senzora putem SPI protokola. Korištenjem ove funkcije modul može dohvatiti trenutno stanje senzora ili konfiguracijske podatke. Sve funkcije za dohvaćanje vrijednosti sa senzora interno koriste ovu funkciju.
- `float IMU_read_accel(uint8_t accel_out_address)` - dohvaća trenutnu akceleraciju u g jedinicama s odgovarajućeg registra na senzoru.
- `float IMU_read_gyro(uint8_t gyro_out_address)` - dohvaća brzinu rotacije u stupnjevima u sekundi (°/s) s odgovarajućeg registra na senzoru.

Implementacija ovih funkcija čini `IMU_driver` modul jednostavnim za korištenje jer glavna aplikacija može pristupiti svim potrebnim podacima senzora bez informacija o detaljima SPI komunikacije ili registrima inercijske mjerne jedinice, čime se osigurava modularnost i lakša nadogradnja te održavanje sustava.

Modul `mqtt_module` zadužen je za komunikaciju mikrokontrolera ESP32 s MQTT brokerom, što omogućuje prijenos podataka o izvršenim vježbama u stvarnom vremenu. Zaglavlje `mqtt_module.h` deklarira funkcije potrebne za inicijalizaciju Wi-Fi veze,

pokretanje MQTT klijenta te slanje rezultata vježbi na definirane MQTT teme. Takva apstrakcija omogućuje glavnoj aplikaciji jednostavan pristup mrežnoj komunikaciji bez detalja implementacije protokola.

Najvažnije funkcije modula opisane su u nastavku.

- `void wifi_init_and_start_mqtt(const char *ssid, const char *password, const char *broker_uri)` - inicijalizira Wi-Fi i povezuje ESP32 na zadanu mrežu, te pokreće MQTT klijenta prema specificiranom brokeru. Funkcija je ključna jer omogućuje sustavu prijenos podataka u realnom vremenu i komunikaciju s drugim uređajima ili serverima.
- `esp_err_t mqtt_publish_exercise_result(const char* category)` - objavljuje rezultat izvođenja vježbe putem MQTT protokola. Parametar `category` označava status vježbe, primjerice *"Ispravan čučanj"* ili *"Sustav miruje"*. Funkcija vraća `ESP_OK` ako je objava uspješna, odnosno `ESP_FAIL` u slučaju pogreške.

Implementacija modula osigurava pouzdanu mrežnu povezanost i standardizirani način slanja podataka, čime se glavna aplikacija oslobađa detalja Wi-Fi i MQTT protokola.

Glavni program nalazi se u `main.cpp` datoteci, dok funkcija `app_main()` predstavlja ulaznu točku aplikacije na ESP32 mikrokontroleru. Funkcija `app_main()` izvršava korake opisane u nastavku.

1. Inicijalizacija senzora i komunikacije - poziva se funkcija `IMU_init()` koja inicijalizira IMU senzor. Nakon toga program poziva `wifi_init_and_start_mqtt()` kako bi uspostavio Wi-Fi vezu i pokrenuo MQTT klijenta prema definiranoj adresi brokera.
2. Priprema za klasifikaciju - deklariraju se varijable koje omogućuju klasifikaciju i pohraniti rezultate izvođenja modela strojnog učenja. U ovom koraku na serijskom monitoru ispisuje se poruka da je aplikacija uspješno pokrenuta i inicijalizirana, čime se olakšava praćenje statusa rada uređaja, `ei_printf("Application started!\n")`. Nakon inicijalizacije definira se i polje u koje se spremaju mjerenja s inercijske mjerne jedinice koja su ulazne vrijednosti bloka za obradu podataka, `signal_t features_signal`, a veličina tog bloka postavlja se na vrijednost `EI_CLASSIFIER_DSP_INPUT_FRAME_SIZE`.

3. Glavna petlja - program ulazi u beskonačnu petlju u kojoj se kontinuirano prikupljaju i obrađuju podaci s inercijske mjerne jedinice.

```
features_signal.get_data = &raw_feature_get_data;

EI_IMPULSE_ERROR res = run_classifier(&features_signal,
&result, false /* debug */);
if (res != EI_IMPULSE_OK) {
    ei_printf("ERR: Failed to run classifier (%d)\n", res);
    return res;
}
print_inference_result(result);
```

U ovom koraku poziva se funkcija `raw_feature_get_data(offset, length, *out_ptr)` koja služi za prikupljanje mjerenja s inercijske mjerne jedinice u zadanom vremenskom okviru, a implementirana je na sljedeći način:

```
for(int i = 0; i < EI_CLASSIFIER_RAW_SAMPLE_COUNT; i++) {
    float ax = IMU_read_accel(ACCEL_XOUT_H);
    float ay = IMU_read_accel(ACCEL_YOUT_H);
    float az = IMU_read_accel(ACCEL_ZOUT_H);

    float gx = IMU_read_gyro(GYRO_XOUT_H);
    float gy = IMU_read_gyro(GYRO_YOUT_H);
    float gz = IMU_read_gyro(GYRO_ZOUT_H);

    out_ptr[EI_CLASSIFIER_RAW_SAMPLES_PER_FRAME * i + 0] = ax;
    out_ptr[EI_CLASSIFIER_RAW_SAMPLES_PER_FRAME * i + 1] = az;
    out_ptr[EI_CLASSIFIER_RAW_SAMPLES_PER_FRAME * i + 2] = gx;
    out_ptr[EI_CLASSIFIER_RAW_SAMPLES_PER_FRAME * i + 3] = gy;

    ei_sleep((int)(1000 / EI_CLASSIFIER_FREQUENCY));
}
```

Odsječak koda 3

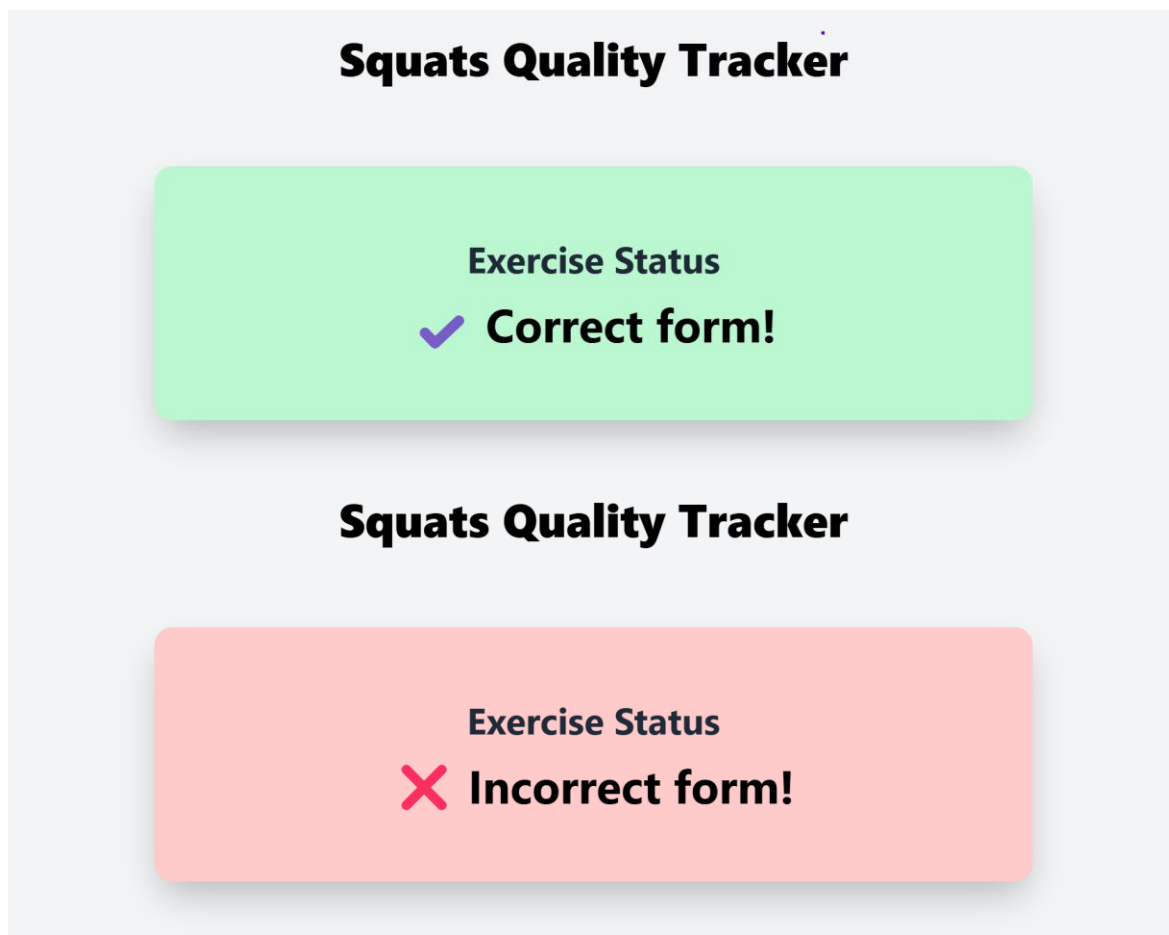
Tok programa omogućuje kontinuirano praćenje i evaluaciju ispravnosti izvođenja sportskih vježbi, dok je većina funkcionalnosti implementirana kroz IMU i MQTT module. Na ovaj način, svaki unaprijed definirani vremenski period od nekoliko sekundi dolazi novi rezultat klasifikacije izvedene vježbe.

5.3.3. Web sučelje za prikaz rezultata

Kako bi se omogućila vizualizacija rezultata u stvarnom vremenu, razvijeno je web sučelje koje prikazuje podatke dobivene s ESP32 mikrokontrolera. ESP32 se spaja na bežičnu mrežu i rezultate klasifikacije izvedene vježbe šalje putem MQTT protokola. MQTT broker je povezan s lokalnim serverom, postavljenim na računalu korisnika, koji podatke prosljeđuje web aplikaciji koristeći WebSocket komunikaciju. Tako se ostvaruje dvosmjerna i niskolatentna komunikacija između senzorskog sustava i korisničkog sučelja.

Web stranica služi kao centralno mjesto za pregled i analizu rezultata izvođenja vježbi. Podaci o svakoj klasifikaciji („*Ispravno izvedena vježba*“, „*Neispravno izvedena vježba*“...) prikazuju se u preglednom obliku, čime se korisniku omogućuje brza procjena ispravnosti izvedene vježbe. Sučelje je implementirano tako da podržava prikaz rezultata u stvarnom vremenu, bez potrebe za ručnim osvježavanjem stranice.

Ovakav pristup omogućuje korisniku intuitivno i interaktivno praćenje stanja sustava, dok istovremeno olakšava daljnju analizu i arhiviranje rezultata. Osim toga, modularna struktura web aplikacije omogućuje jednostavno proširenje funkcionalnosti, primjerice dodavanje novih vizualizacija, podrške za različite vrste vježbi ili integraciju s udaljenim poslužiteljima u oblaku. Slika 5.4 prikazuje jednostavno internetsko sučelje koje pomoću teksta i prikladnih boja prikazuje rezultate klasifikacije izvedene vježbe.



Slika 5.4 Web sučelje dojavljuje ispravno (gore) ili neispravno (dolje) izvođenje vježbe.

6. Rezultati

U ovom poglavlju prikazani su rezultati rada razvijenog sustava na odabranim vježbama. Cilj evaluacije je analizirati točnost klasifikacije i procijeniti pouzdanost sustava u prepoznavanju ispravnog i neispravnog izvođenja odabranih vježbi.

Ispravno izvođenje svake vježbe unaprijed je definirano, uz jasno navedene neispravnosti koje sustav može otkriti. Sustav je testiran na tri različite vježbe koje obuhvaćaju donji dio tijela i mišiće nogu. Za svaku vježbu pružen je opis, te informacije o prikupljenom skupu podataka (ukupan broj uzoraka te broj različitih osoba koje su sudjelovale u prikupljanju podataka).

Nadalje, opisane su ulazne značajke korištene u neuronskoj mreži, odnosno signali akceleracije i kutne brzine prikupljeni inercijskom mjernom jedinicom. Rezultati klasifikacije prikazani su pomoću matrice konfuzije. Dodatno, klasifikacijski rezultati za svaku vježbu ilustrirani su dvodimenzionalnim grafom koji prikazuje raspored točaka u prostoru značajki te njihovo grupiranje u klastere.

Poglavlje završava prikazom tehničkih parametara treniranog modela, uključujući njegovu memorijsku veličinu i prosječno vrijeme izvođenja klasifikacije na odabranom mikrokontroleru. Takva analiza omogućuje cjelovit uvid u učinkovitost i praktičnu primjenjivost razvijenog sustava.

6.1. Vježba A – Čučanj

Prva vježba koja je odabrana za analizu je klasični čučanj. Budući da je razvijeni uređaj za vrijeme korištenja postavljen na struk osobe, očekuje se da će uređaj ispravno klasificirati ispravnost izvođenja čučnja.

U prikupljanju podataka za treniranje modela strojnog učenja sudjelovale su dvije muške i dvije ženske osobe. Ukupno su prikupljene 332 sekunde mjerenja čučnjeva. 75 % od svih mjerenja korišteno je za treniranje modela, a 25 % za validaciju. Znanstveni rad [27] služi kao definicija ispravnog čučnja. Nakon prikupljanja podataka i treniranja modela strojnog učenja s različitom kombinacijom ulaznih parametara primijećeno je da zanemarivanjem akceleracije Y osi i kutne brzine Z osi ne dolazi do smanjenja točnosti konačnog modela.

Ova pojava u skladu je s očekivanjima jer su to dvije veličine koje se za vrijeme izvođenja čučnja ne mijenjaju pa njihovo korištenje u procesu klasifikacije negativno utječe na brzinu izvođenja programa i efektivno samo unosi šum.

Odluka o tome koje veličine nisu relevantne pri treniranju modela strojnog učenja i klasifikaciji potrebno je odrediti za svaku vježbu zasebno.

Dakle, na ulaz bloka za obradu podataka za čučanj dovedeni su sljedeći signali: akceleracija X osi, akceleracija Z osi, kutna brzina X osi i kutna brzina Y osi. Frekvencija uzorkovanja iznosi 100 Hz, a svako mjerenje traje dvije sekunde.

Blok za obradu signala sve signale skalira faktorom 0,05. Nakon skaliranja signal je propušten kroz niskopropusni filter šestog reda i gornje granične frekvencije 3 Hz. Nakon prolaska kroz filter nad signalom se vrši brza Fourierova transformacija (FFT) u 64 točke. Rezultati Fourierove transformacije koriste se kao ulazni parametri neuronske mreže.

Model strojnog učenja na izlazu razlikuje četiri klase:

1. Sustav miruje – osoba se ne kreće ili uređaj uopće nije postavljen na osobu.
2. Ispravan čučanj – čučanj izveden prema pravilima iz [27].
3. Neispravan čučanj – čučanj koji je nastao kršenjem nekih od pravila iz [27].
4. Prebrzo izveden čučanj – čučanj koji je izveden brže od najveće predviđene brzine.

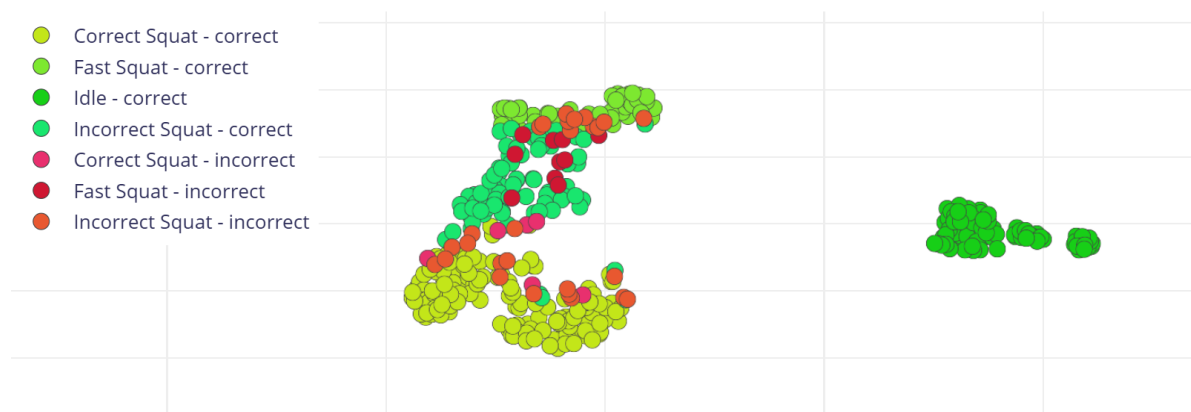
Tablica 1 Matrica konfuzije klasifikacije čučnja.

Stvarna klasa	Sustav miruje	Ispravno	Neispravno	Prebrzo	
	Sustav miruje	100.0%	0.0%	0.0%	0.0%
	Ispravno	0.0%	97.0%	3.0%	0.0%
	Neispravno	0.0%	22.7%	72.7%	4.5%
	Prebrzo	0.0%	0.0%	20.0%	80.0%
		Predviđena klasa			
		Sustav miruje	Ispravno	Neispravno	Prebrzo

Važno je istaknuti da sustav može otkriti samo neispravnosti koje rezultiraju pomakom u razini struka gdje je uređaj postavljen. To znači da nije moguće odrediti neispravan položaj ruku ili glave. Baš iz toga razloga odabran je čučanj koji primarno uključuje mišiće nogu i donji dio trupa. Sustav je razvijen tako da prepoznaje neispravnosti pri izvođenju čučnja kao što su: odvajanje pete od poda i neispravan kut koljena pri čučnju. Radi se o najčešćim neispravnostima koje ljudi rade pri pokušaju izvođenja ispravnog čučnja.

Tablica 1 prikazuje točnost modela pri klasifikaciji čučnjeva. Na prikupljenom setu podataka prosječna točnost modela pri klasifikaciji čučnja iznosi 87,5 %. Važno je napomenuti da se u 20 % slučajeva prebrzi čučanj klasificira kao neispravan, što je u praksi točan rezultat klasifikacije, ali u tablici zabune narušava točnost.

Slika 6.1 sadrži najviše informacija za intuitivno shvaćanje koliko je sustav uspješan u klasifikaciji izvođenja vježbi. Crvene i narančaste točke prikazuju neispravne rezultate klasifikacije, a točke u nijansama zelene boje prikazuju gdje se u prostoru značajki nalaze mjerenja koja pripadaju istoj klasi. Slika 6.1 dokaz je da je moguće pomoću inercijske mjerne jedinice i neuronske mreže klasificirati ispravnost izvođenja čučnja što je i bio cilj znanstvenog rada.



Slika 6.1 Raspored mjerenja u prostoru značajki za čučanj, preuzeto s [28].

6.2. Vježba B – Iskorak

Sljedeća vježba koja je odabrana za analizu je iskorak. Budući da je razvijeni uređaj za vrijeme korištenja postavljen na struk osobe, očekuje se da će uređaj ispravno klasificirati ispravnost izvođenja iskoraka, kao što je to bio slučaj i za čučanj u prethodnom poglavlju.

U prikupljanju podataka za treniranje modela strojnog učenja sudjelovale su dvije muške osobe. Ukupno je prikupljeno 270 sekundi mjerenja čučnjeva. 78 % od svih mjerenja

korišteno je za treniranje modela, a 22 % za validaciju. Znanstveni rad [29] služi kao smjernica za definiciju ispravnog iskoraka. Na ulaz bloka za obradu podataka za čučanj dovedeni su sljedeći signali: akceleracija X, Y i Z osi, te kutna brzina X, Y i Z osi, dakle svi signali koji su prikupljeni inercijskom mjernom jedinicom. Frekvencija uzorkovanja iznosi 100 Hz, a svako mjerenje traje dvije sekunde.

Blok za obradu signala sve signale skalira faktorom 0,5. Nakon skaliranja signal je propušten kroz niskopropusni filter šestog reda i gornje granične frekvencije 14 Hz. Nakon prolaska kroz filter nad signalom se vrši brza Fourierova transformacija (FFT) u 32 točke. Rezultati Fourierove transformacije koriste se kao ulazni parametri neuronske mreže.

Model strojnog učenja na izlazu razlikuje tri klase:

1. Sustav miruje – osoba se ne kreće ili uređaj uopće nije postavljen na osobu.
2. Ispravan iskorak – iskorak izveden prema pravilima iz [29].
3. Neispravan iskorak – iskorak koji je nastao kršenjem nekih od pravila iz [29].

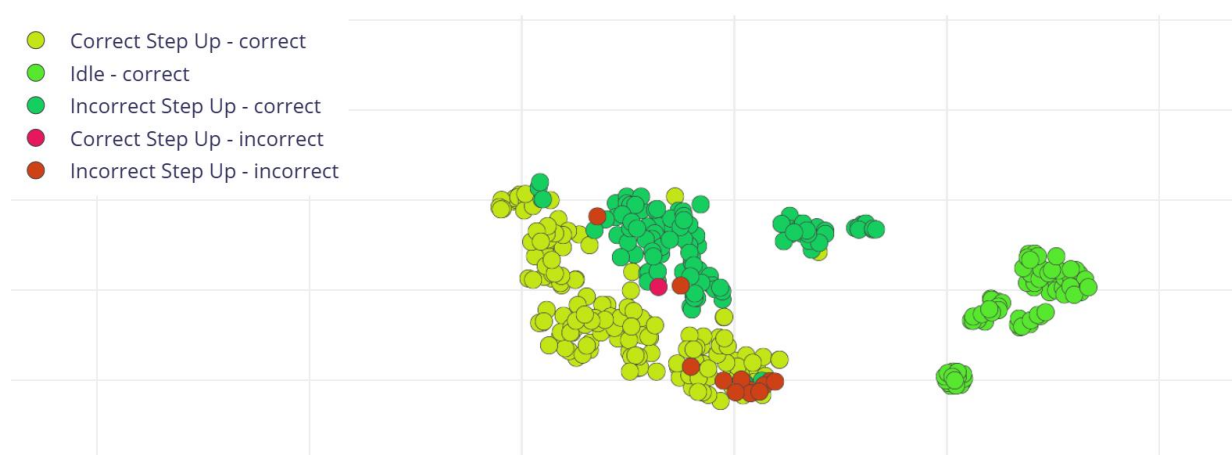
Važno je istaknuti da sustav može otkriti samo neispravnosti koje rezultiraju pomakom u razini struka gdje je uređaj postavljen. To znači da nije moguće odrediti neispravan položaj ruku ili glave. Baš iz toga razloga odabran je iskorak koji primarno uključuje mišiće nogu i donji dio trupa. Sustav je razvijen tako da prepoznaje neispravnosti pri izvođenju iskoraka kao što su: neispravan kut koljena pri iskoraku i neispravan međusobni položaj nogu u trenutku iskoraka. Radi se o najčešćim neispravnostima koje ljudi rade pri pokušaju ispravnog iskoraka.

Tablica 2 Matrica konfuzije klasifikacije iskoraka.

Stvarna klasa	Predviđena klasa		
	Sustav miruje	Pravilno	Nepravilno
Sustav miruje	100.0%	0.0%	0.0%
Pravilno	0.0%	96.7%	3.3%
Nepravilno	0.0%	20.0%	80.0%

Tablica 2 prikazuje točnost modela pri klasifikaciji iskoraka. Na prikupljenom setu podataka prosječna točnost modela pri klasifikaciji iskoraka iznosi 90,1 %.

Slika 6.2 sadrži najviše informacija za intuitivno shvaćanje koliko je sustav uspješan u klasifikaciji izvođenja vježbi. Crvene točke prikazuju neispravne rezultate klasifikacije, a točke u nijansama zelene boje prikazuju gdje se u prostoru značajki nalaze mjerenja koja pripadaju istoj klasi. Slika 6.2 još je jedan dokaz da je moguće pomoću inercijske mjerne jedinice i neuronske mreže klasificirati ispravnost izvođenja vježbe što je i cilj znanstvenog rada. Iskorak je druga vježba na kojoj je pokazano da je moguće raditi klasifikaciju točnosti izvođenja vježbe opisanim sustavom.



Slika 6.2 Raspored mjerenja u prostoru značajki za iskorak, preuzeto s [28].

6.3. Vježba C – Sklek

U poglavljima 6.1 i 6.2 pokazano je da sustav radi za vježbe koje uključuju mišiće nogu i značajna gibanja donjeg dijela trupa. U ovom poglavlju cilj je testirati granice sustava i istražiti je li moguće izvući korisne informacije o ispravnosti vježbe koja ne uključuje mišiće nogu. Preciznije, je li moguće otkriti neispravnost u izvođenju vježbe kao što je sklek. Ispravnost skleka znatno ovisi o položaju svih dijelova tijela (glava, ruke, leđa, noge) [30]. Baš iz tog razloga očekuje se da će sustav imati poteškoća s određivanjem ispravnosti izvođenja skleka.

U prikupljanju podataka za treniranje modela strojnog učenja sudjelovale su dvije muške osobe. Ukupno je prikupljeno 290 sekundi mjerenja sklekova. 75 % od svih mjerenja korišteno je za treniranje modela, a 25 % za validaciju. Znanstveni rad [30] služi kao smjernica za definiciju ispravnog skleka. Na ulaz bloka za obradu podataka za čučanj

dovedeni su sljedeći signali: akceleracija X, Y i Z osi, te kutna brzina X, Y i Z osi, dakle svi signali koji su prikupljeni inercijskom mjernom jedinicom. Frekvencija uzorkovanja iznosi 100 Hz, a svako mjerenje traje dvije sekunde.

Blok za obradu signala sva signale skalira faktorom 0,5. Nakon skaliranja signal je propušten kroz niskopropusni filter šestog reda i gornje granične frekvencije 8 Hz. Nakon prolaska kroz filter nad signalom se vrši brza Fourierova transformacija (FFT) u 32 točke. Rezultati Fourierove transformacije koriste se kao ulazni parametri neuronske mreže.

Model strojnog učenja na izlazu razlikuje tri klase:

1. Sustav miruje – osoba se ne kreće ili uređaj uopće nije postavljen na osobu.
2. Ispravan sklek – sklek izveden prema pravilima iz [30].
3. Neispravan sklek – sklek koji je izveden kršenjem nekih od pravila iz [30] ili sklek koji je izveden tako da tijelo nije spuštено u krajnji donji položaj ili nije podignuto u krajnji gornji položaj za vrijeme trajanja jednog skleka.

Tablica 2 prikazuje točnost modela pri klasifikaciji skleka. Na prikupljenom setu podataka prosječna točnost modela pri klasifikaciji čučnja iznosi 92,0 %. Iako je za inicijalna pretpostavka bila da sustav neće s visokom razinom točnosti odrediti ispravnost izvođenja skleka, rezultati govore da je to moguće. Slično kao u prethodna dva poglavlja, i ovaj sustav ima određena ograničenja te ne može otkriti neispravnosti poput neispravnog položaja glave.

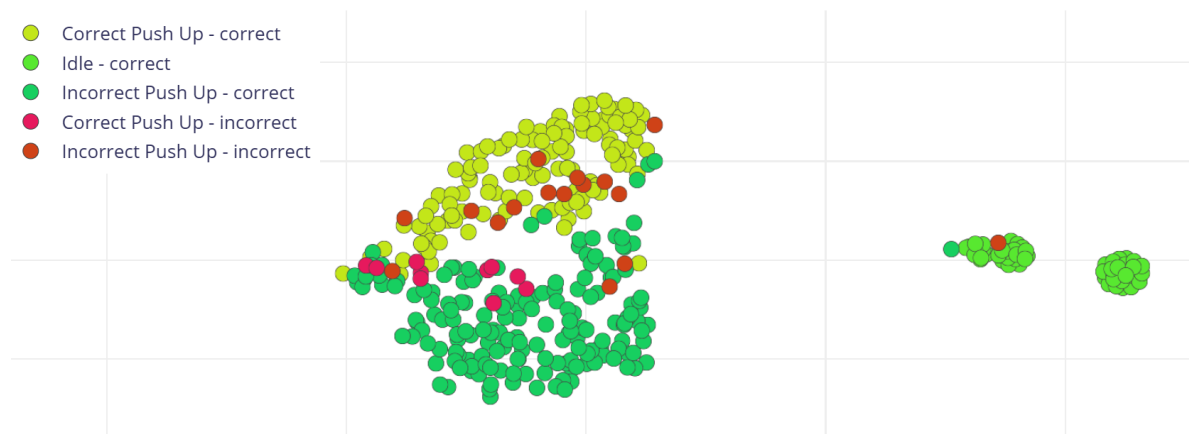
Tablica 3 Matrica konfuzije klasifikacije skleka.

Stvarena klasa	Sustav miruje	100.0%	0.0%	0.0%
	Pravilno	0.0%	92.9%	7.1%
	Nepravilno	0.0%	13.2%	86.8%
		Sustav miruje	Pravilno	Nepravilno
		Predviđena klasa		

Slika 6.3 sadrži najviše informacija za intuitivno shvaćanje koliko je sustav uspješan u klasifikaciji izvođenja vježbi. Crvene točke prikazuju neispravne rezultate klasifikacije, a točke u nijansama zelene boje prikazuju gdje se u prostoru značajki nalaze mjerenja koja pripadaju istoj klasi. Slika 6.3 još je jedan dokaz da je moguće pomoću inercijske mjerne jedinice i neuronske mreže klasificirati ispravnost izvođenja vježbe što je i cilj znanstvenog rada.

Iako bi se moglo pretpostaviti da zbog toga što sklek primarno cilja mišić ruku i gornjeg dijela trupa, sustav neće moći otkriti neispravnosti pri izvođenju vježbe, rezultati govore da to nije slučaj.

Sklek je treća vježba na kojoj je pokazano da je moguće raditi klasifikaciju točnosti izvođenja vježbe, što pokazuje spektar mogućnosti ovakvog sustava.



Slika 6.3 Raspored mjerenja u prostoru značajki za sklek, preuzeto s [28].

6.4. Veličina i vrijeme izvođenja modela strojnog učenja i aplikacije

Nakon treniranja modela strojnog učenja iz poglavlja 6.1, 6.2 i 6.3 svaki model prolazi kroz proces kvantizacije. Kvantizacija uključuje konverziju parametara neuronske mreže iz tipa podataka s pomičnom točkom *float* u cjelobrojni tip podatka *uint8*. Kvantizacija je nužna jer mikrokontroler postiže veće brzine pri računanju s cjelobrojnim tipom podataka. Veličina neuronske mreže opisane u ovom radu iznosi 20-40 kB (ovisno u kojoj se vježbi radi) nakon kvantizacije što je prikladno za mikrokontrolere s ograničenom memorijom.

Predobrada podataka (računanje ulaznih parametara neuronske mreže) traje između 8 i 16 ms, ovisno o modelu, a samo zaključivanje neuronske mreže traje 1 ms. Vremena

izvođenja do par desetaka ms omogućuju primjenu sustava u stvarnom vremenu za potrebe odrađivanja ispravnosti izvođenja sportskih vježbi.

Kada je riječ u cijeloj aplikaciji razvijenoj za mikrokontroler, ukupna veličina generirane slike programa iznosi ~1 MB, pri čemu konačna `.bin` datoteka može sadržavati dodatno poravnanje zbog organizacije *flash* memorije. Odabrana razvojna pločica s ESP32 mikrokontrolerom opisana u poglavlju 4.1 posjeduje 4 MB *flash* memorije.

Nakon kompilacije i povezivanja, alat ESP-IDF generirao je izvještaj o raspodjeli memorije ugradbenog sustava. Rezultati pokazuju da statička potrošnja DRAM-a iznosi 40 kB. Od toga 20 kB otpada na inicijalizirane globalne i statičke varijable (`.data` segment), a preostalih 20 kB zauzimaju neinicijalizirane varijable (`.bss` segment).

Potrošnja IRAM-a, namijenjenog izvršavanju koda s niskom latencijom, iznosi 108 kB. Najveći dio IRAM-a zauzima izvršni kod (`.text` segment, 107 kB), dok vektorska tablica prekida zauzima 1 kB.

7. Zaključak

U ovom radu razvijen je i demonstriran senzorski sustav za praćenje ispravnosti izvođenja sportskih vježbi temeljen na inercijskoj mjernoj jedinici i rubnom modelu umjetne inteligencije male veličine. Pokazano je da je moguće ugraditi neuronsku mrežu veličine svega nekoliko desetaka kB u mikrokontroler ograničenih resursa te u stvarnom vremenu klasificirati ispravnost izvođenja vježbi. Sustav je validiran na primjerima čučnjeva, iskoraka i sklekova, a rezultati pokazuju točnost klasifikacije višu od 90 %.

Na temelju inicijalno postavljenih ciljeva izrađen je funkcionalni prototip nosivog uređaja. Sustav je baterijski napajan, odlikuje se niskom potrošnjom energije te jednostavnim punjenjem putem standardnog USB-C priključka, što ga čini praktičnim i jednostavnim za upotrebu. Kućište uređaja izrađeno je na temelju vlastitog dizajna primjenom metode 3D ispisa. Svi podaci potrebni za treniranje neuronske mreže prikupljeni su samostalno.

Za razliku od većine dosadašnjih istraživanja u ovom području, koja su bila usmjerena isključivo na klasifikaciju različitih vježbi, ovaj rad napravio je iskorak dokazujući da je moguće klasificirati i ispravnost izvođenja vježbi. Štoviše, pokazano je da se samo jednom inercijskom mjernom jedinicom može ne samo razlikovati ispravno od neispravno izvedenih vježbi, već u pojedinim slučajevima identificirati način na koji je vježba izvedena neispravno. Rezultati rada dokazuju da se ograničeni računalni resursi mikrokontrolera mogu učinkovito iskoristiti za zadatke klasifikacije u području praćenja i evaluacije sportskih vježbi.

Iako prototip pokazuje zadovoljavajuće rezultate, postoje određena ograničenja. Sustav se temelji na jednoj inercijskoj mjernoj jedinici i može otkriti neispravnosti koje uzrokuju promjene gibanja u razini struka, dok položaj gornjeg dijela tijela i glave ostaje izvan dosega mjerenja. Također, za svaku novu vježbu potrebno je prikupiti vlastiti skup podataka i ponovno istrenirati neuronsku mrežu, što je inherentno svojstvo stojnog učenja. Unatoč navedenim ograničenjima, ovakav sustav pokazuje značajan potencijal primjene u rekreativnom vježbanju, rehabilitaciji, sportskom treningu te u razvoju personaliziranih nosivih uređaja. Budući rad mogao bi se usmjeriti na proširenje sustava s više senzora,

povećanje broja podržanih vježbi te integraciju naprednijih modela strojnog učenja, uz zadržavanje male potrošnje energije i jednostavne primjene.

Zahvale

Zahvaljujem roditeljima, sestri i djevojci na pomoći u prikupljanju skupa podataka za treniranje modela strojnog učenja.

Zahvaljujem mentoru, prof. dr. sc. Hrvoju Džapi, na podršci tijekom izrade rada te na usmjeravanju u istraživačko područje praćenja ljudskog pokreta inercijskim senzorima.

Također zahvaljujem Chrisu Lušetiću na vrijednom savjetu pri odabiru alata *Edge Impulse* korištenog u realizaciji rada.

Literatura

- [1] Reimers, C. D., Knapp, G., Reimers, A. K. Does physical activity increase life expectancy? A review of the literature. *J Aging Res.* 2012;2012:243958. doi:10.1155/2012/243958.
- [2] Gray, S. E., Finch, C. F. The causes of injuries sustained at fitness facilities presenting to Victorian emergency departments - identifying the main culprits. *Inj Epidemiol.* 2015;2(1):6. doi:10.1186/s40621-015-0037
- [3] Celic, L., & Magjarevic, R. (2019). Seamless connectivity architecture and methods for IoT and wearable devices. *Automatika*, 61(1), 21–34. <https://doi.org/10.1080/00051144.2019.1660036>.
- [4] Džaja, D., Čibarić, M., Šeketa, G., & Magjarević, R. (2022). Accelerometer-based algorithm for the segmentation and classification of repetitive human movements during workouts. *Automatika*, 64(2), 211–224. <https://doi.org/10.1080/00051144.2022.2121247>
- [5] Begovac J, Šeketa G, Celić L, Lacković I, Magjarević R. Quality of Human Activities Measurement from Accelerometer Data of a Smartphone. U: Lacković I, Vasić D, urednici. 6th European Conference of the International Federation for Medical and Biological Engineering. Cham: Springer; 2015. Dostupno na: https://doi.org/10.1007/978-3-319-11128-5_67.
- [6] Phan, V., Song, K., Silva, R. S., Silbernagel, K. G., Baxter, J. R., Halilaj, E. Seven things to know about exercise classification with inertial sensing wearables. *IEEE J Biomed Health Inform.* 2024;28(6):3411–3421. doi:10.1109/JBHI.2024.3368042.
- [7] Dill, S., Ahmadi, A., Grimmer, M., Haufe, D., Rohr, M., Zhao, Y., et al. Accuracy Evaluation of 3D Pose Reconstruction Algorithms Through Stereo Camera Information Fusion for Physical Exercises with MediaPipe Pose. *Sensors.* 2024;24:7772. doi:10.3390/s24237772.
- [8] Obukhov, A., Volkov, A., Pchelintsev, A., Nazarova, A., Teselkin, D., Surkova, E., et al. Examination of the accuracy of movement tracking systems for monitoring exercise for musculoskeletal rehabilitation. *Sensors.* 2023;23:8058. doi:10.3390/s23198058.
- [9] Giggins, O., Kelly, D., Caulfield, B. Evaluating rehabilitation exercise performance using a single inertial measurement unit. In: 2013 7th International Conference on Pervasive Computing Technologies for Healthcare and Workshops; 2013 May; Venice, Italy. p. 49–56. doi:10.4108/icst.pervasivehealth.2013.252061.
- [10] M. P. Wilk, M. Walsh and B. O'Flynn, "Low Cost Embedded Multimodal Opto-Inertial Human Motion Tracking System," 2020 31st Irish Signals and Systems Conference (ISSC), Letterkenny, Ireland, 2020, pp. 1-6, doi: 10.1109/ISSC49989.2020.9180190.
- [11] Ashapkina, M. S., Alpatov, A. V., Sablina, V. A., Chekushin, A. A. Quantitative parameters developing for estimation of rehabilitation exercising on the basis of IMU

- sensor device. In: 2018 7th Mediterranean Conference on Embedded Computing (MECO); 2018 Jun; Budva, Montenegro. p. 1–4. doi:10.1109/MECO.2018.8405987.
- [12] Panwar, M. Rehab-Net: Deep learning framework for arm movement classification using wearable sensors for stroke rehabilitation. *IEEE Trans Biomed Eng.* 2019;66(11):3026–3037.
 - [13] Hnoohom, N., Chotivatunyu, P., Mekruksavanich, S., Jitpattanakul, A. Recognition of shoulder exercise activity based on EfficientNet using smartwatch inertial sensors. In: 2022 3rd International Conference on Big Data Analytics and Practices (IBDAP); 2022 Jul; Bangkok, Thailand. p. 6–10. doi:10.1109/IBDAP55587.2022.9907217.
 - [14] A. Garcia-Perez, R. Miñón, A. I. Torre-Bastida, and E. Zulueta-Guerrero, "Analysing Edge Computing Devices for the Deployment of Embedded AI," *Sensors*, vol. 23, no. 23, p. 9495, Nov. 2023. DOI: 10.3390/s23239495.
 - [15] Banbury, C., Zhou, C., Fedorov, I., Matas Navarro, R., Thakker, U., Gope, D., et al. MicroNets: Neural network architectures for deploying TinyML applications on commodity microcontrollers. *arXiv [Preprint]*. 2020. Dostupno na: <https://doi.org/10.48550/arXiv.2010.11267>
 - [16] Azad A, Banu A. Publication trends in artificial intelligence conferences: the rise of super prolific authors. *arXiv preprint arXiv:2412.07793v1 [cs.DL]*. 2024 Nov 28 [pristupljeno 27. srpnja 2025.]. Dostupno na: <https://arxiv.org/abs/2412.07793>
 - [17] Edge Impulse Dokumentacija. Edge Impulse Inc.; [Internet] [pristupljeno 27. srpnja 2025.]. Dostupno na: <https://docs.edgeimpulse.com/docs>
 - [18] Kaloev M, Krastev G. Comparative analysis of activation functions used in the hidden layers of deep neural networks. In: 2021 3rd International Congress on Human-Computer Interaction, 2021 May; Ankara. IEEE; 2021. p. 1–5. doi:10.1109/HORA52670.2021.9461312.
 - [19] Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. *Nature*. 1986;323(6088):533–6. doi:10.1038/323533a0.
 - [20] Lee M, Kim D, Kim W. Fault Type Classification of Rotating Machinery based on Machine Learning with Fourier Transform Feature Extraction. In: 2023 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia); 2023 Nov; Busan, Korea, Republic of. IEEE; 2023. p. 1–6. doi:10.1109/ICCE-Asia59966.2023.10326396.
 - [21] TensorFlow. TensorFlow: An end-to-end open source machine learning platform [Internet]. Google; [pristupljeno 29. srpnja 2025.]. Dostupno na: <https://www.tensorflow.org/>
 - [22] Andy Stanford-Clark and Arlen Nipper, MQTT Version 3.1.1 Protocol Specification, OASIS Standard, 2014. Dostupno na: <https://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html>
 - [23] Espressif Systems. ESP-IDF Programming Guide for ESP32. Espressif Systems; [pristupljeno 1. kolovoza 2025.]. Dostupno na: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/index.html>
 - [24] Eclipse Foundation. Eclipse Mosquitto - An open source MQTT broker [Internet]. [pristupljeno 2. kolovoza 2025.]. Dostupno na: <https://mosquitto.org/>

- [25] Mikroelektronika, *Click Board 9DOF 2*. Poveznica: <https://www.mikroe.com/9dof-2-click>; travanj 2024.
- [26] Soldered Electronics. Li-ion charger BQ24295 with 5 V USB output board [Internet]. Soldered; [pristupljeno 3. kolovoza 2025.]. Dostupno na: <https://soldered.com/product/li-ion-charger-bq24295-with-5v-usb-output-board/>
- [27] Straub RK, Powers CM. A biomechanical review of the squat exercise: implications for clinical practice. *Int J Sports Phys Ther*. 2024 Apr 1;19(4):490-501. doi:10.26603/001c.94600. PMID: 38576836; PMCID: PMC10987311.
- [28] Edge Impulse. Motion Quality Tracker projekt [Internet]. Edge Impulse Studio; [pristupljeno 17. kolovoza 2025.]. Dostupno na: <https://studio.edgeimpulse.com/studio/684917>
- [29] Escamilla RF, Thompson IS, Asuncion R, Bravo J, Chang T, Fournier T, Garcia H, Hockenbery E, Nagasawa K, Ozor J, Snoeberger H, Wilk KE, Bizzini M. Effects of step length and stride variation during forward lunges on lower-extremity muscle activity. *J Funct Morphol Kinesiol*. 2025 Jan 22;10(1):42. doi:10.3390/jfmk10010042. PMID: 39982282; PMCID: PMC11843896.
- [30] Hewit J, Jaffe D, Bedard A. Teaching the Proper Push-up Position. *J Phys Educ Recreat Dance*. 2018;89(7):50-2. doi:10.1080/07303084.2018.1491765.
- [31] Online repozitorij. Motion Quality Tracker [Internet]. GitHub; [pristupljeno 25. kolovoza 2025.]. Dostupno na: <https://github.com/LovroSantek/Motion-Quality-Tracker/>

Sažetak

Autor: Lovro Šantek

Naslov: Senzorski sustav za praćenje ispravnosti izvođenja sportskih vježbi pomoću rubnog AI modela i ESP32 mikrokontrolera

U ovom radu razvijen je nosivi senzorski sustav koji u stvarnom vremenu korisniku pruža povratnu informaciju o ispravnosti izvođenja sportskih vježbi. Sustav podatke prikuplja s jedne inercijske senzorske mjerne jedinice (akcelerometar i žiroskop) i izvodi klasifikaciju u stvarnom vremenu na samom uređaju s pomoću modela strojnog učenja manjeg od 40 kB. Model je treniran korištenjem vlastito prikupljenog skupa podataka i okvira *TensorFlow*. Sustav se sastoji od bežičnog baterijski napajanog nosivog uređaja temeljenog na mikrokontroleru iz porodice ESP32, s ugrađenim inercijskim senzorom i vlastito dizajniranim ergonomskim 3D ispisanim kućištem, te web aplikacije za prikaz rezultata. Funkcionalnost sustava validirana je i demonstrirana na vježbama čučnja, iskoraka i skleka. Na vlastitom skupu podataka postignuta je točnost klasifikacije ispravnosti izvođenja sportskih vježbi veća od 90 %, uz potrošnju sklopovlja manju od 1 W i vrijeme izvođenja algoritma ispod 20 ms izravno na mikrokontroleru. Rezultati potvrđuju učinkovitost inovativnog rješenja te ukazuju na potencijal za širu primjenu u sustavima za praćenje i analizu vježbanja na ugradbenim računalima niske potrošnje.

Ključne riječi: ugradbeni računalni sustavi, nosivi uređaji za praćenje tjelesne aktivnosti, umjetna inteligencija na rubnim uređajima, inercijski senzori, strojno učenje, praćenje ljudskog pokreta, ESP32, bežični senzori niske potrošnje

Summary

Author: Lovro Šantek

Title: Sensor system for monitoring the correctness of sports exercises using an edge AI model and an ESP32 microcontroller

In this work, a wearable sensor system was developed to provide users with real-time feedback on the correctness of performing sports exercises. The system collects data from a single inertial measurement unit (accelerometer and gyroscope) and performs real-time classification directly on the device using a machine learning model smaller than 40 kB. The model was trained on a custom-collected dataset using the TensorFlow framework. The system consists of a wireless, battery-powered wearable device based on an ESP32 family microcontroller with an integrated inertial sensor and a custom-designed ergonomic 3D-printed enclosure, as well as a web application for result visualization. The functionality of the system was validated and demonstrated on squat, lunge, and push-up exercises. On the custom dataset, the system achieved an accuracy of over 90 % in classifying exercise correctness, with hardware power consumption below 1 W and an algorithm execution time under 20 ms directly on the microcontroller. The results confirm the efficiency of the proposed innovative solution and highlight its potential for broader application in exercise monitoring and analysis systems on low-power embedded devices.

Keywords: embedded systems, physical activity monitoring wearables, edge AI, inertial sensors, machine learning, human motion tracking, ESP32, low-power wireless sensors