



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

DEPARTMENT OF INTELLIGENT SYSTEMS

INTELIGENTNÍ FITNESS ASISTENT

SMART FITNESS ASSISTANT

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MARTIN MATĚJKA

VEDOUcí PRÁCE

SUPERVISOR

Ing. TOMÁŠ GOLDMANN

BRNO 2024

Zadání diplomové práce



156439

Ústav: Ústav inteligentních systémů (UITS)
Student: **Matějka Martin, Bc.**
Program: Informační technologie a umělá inteligence
Specializace: Vestavěné systémy
Název: **Inteligentní fitness asistent**
Kategorie: Umělá inteligence
Akademický rok: 2023/24

Zadání:

1. Seznamte se s možnostmi pro vizuální sledování a vyhodnocování pohybů osob při cvičení na fitness strojích.
2. Seznamte se s možnostmi pro senzorické měření počtu opakování cviků přímo na fitness strojích.
3. Navrhněte řešení pro počítání opakování fitness cviků s využitím vizuálních a senzorických dat. Řešení se bude skládat z kamerového modulu a vlastních senzorických modulů, které budou bezdrátově komunikovat se serverem.
4. Implementujte firmware pro senzorické moduly a software pro serverovou část řešení, který bude provádět počítání opakování fitness cviků a vytváření statistik.
5. Aplikaci otestujte v reálném prostředí. Především se zaměřte na vyhodnocení přesnosti počítání opakování cviků. Navrhněte další rozšíření.

Literatura:

- WANG, Jinbao, et al. Deep 3D human pose estimation: A review. *Computer Vision and Image Understanding*, 2021, 210: 103225.
- ANDRILUKA, Mykhaylo, et al. 2d human pose estimation: New benchmark and state of the art analysis. In: *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*. 2014. p. 3686-3693.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Goldmann Tomáš, Ing.**
Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.
Datum zadání: 1.11.2023
Termín pro odevzdání: 17.5.2024
Datum schválení: 21.4.2024

Abstrakt

Práca sa zaoberá problematikou určovania opakovaní a meraní výkonnosti športovcov vo verejných fitness centrách. Navrhuje kombináciu softvéru a hardvéru, ktorá je schopná poskytnúť spätnú väzbu užívateľom fitness centier. Opísaný je postup vytvárania softvéru a hardvéru na realizáciu merania a vyhodnocovania dát. Tieto výsledky sú spracovávané takmer ihneď po dokončení cvičenia a je možné ich ihneď prezentovať užívateľovi. Predstavená kombinácia technológií ponúka nový spôsob monitorovania a zlepšovania výkonnosti vo fitness prostredí zo strany fitness centra.

Abstract

The thesis addresses the issue of determining repetitions and measuring the performance of athletes in public fitness centers. It proposes a combination of software and hardware capable of providing feedback to fitness center users. The process of developing software and hardware for data measurement and evaluation is outlined. These results are processed almost immediately after the completion of the exercise and can be presented to the user immediately. The presented combination of technologies offers a new way of monitoring and improving performance in the fitness environment from the perspective of the fitness center.

Klíčové slová

počítanie opakovaní, fitness stroje, návrh senzorov, spätná väzba pri cvičení, výkonnostné metriky, meranie fitness cvikov

Keywords

counting repetitions, fitness machines, sensor design, exercise feedback, performance metrics, measuring fitness exercises

Citácia

MATĚJKA, Martin. *Inteligentní fitness asistent*. Brno, 2024. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Tomáš Goldmann

Inteligentní fitness asistent

Prehlásenie

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením pána Ing. Tomáša Goldmanna. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....
Martin Matějka
16. mája 2024

Podakovanie

Rád by som sa poďakoval vedúcemu práce, ktorý mi počas jej realizácie poskytoval cenné rady, materiály a predovšetkým svoj čas. Taktiež by som sa rád poďakoval svojej rodine, ktorá ma podporovala počas celého štúdia.

Obsah

1	Úvod	5
2	Fitness a cvičenie	6
2.1	Typy cvičení	6
2.2	Zmysel merania cvikov	6
2.3	Automatizácia zapisovania výkonu	7
2.4	Fitness stroje a cviky	8
3	Analýza merania cvikov	14
3.1	Extrakcia dát	14
3.2	Metódy pre rozpoznávanie opakovaní a pohybu	16
3.3	Existujúce riešenia	19
3.4	Sledovanie cvikov podľa použitých strojov	20
4	Návrh senzorov	22
4.1	Hardvérové senzory	22
4.2	Virtuálne kamerové senzory	27
4.3	Návrh senzora C3	28
4.4	Návrh vývojového setu S3	32
5	Implementácia softvéru	34
5.1	Firmvér hardvérového senzora	34
5.2	Softvér virtuálneho senzora	36
5.3	Server pre spracovanie dát	37
5.4	Určovanie opakovaní z akcelerometrických dát	39
5.5	Určovanie opakovaní z vizuálnych dát	43
6	Zobrazovanie štatistík a testovanie	48
6.1	Štatistiky a prehľad senzorov	48
6.2	Testovanie zamerané na presnosť detekcie počtu opakovaní	51
7	Záver	54
	Literatúra	55
A	Obsah pamäťového média	59
B	Schéma zapojenia obvodu senzora C3	60

Zoznam obrázkov

2.1	DB Benchpress [22].	9
2.2	DB Shoulder Press [22].	9
2.3	Lateral Raise [22].	10
2.4	BB Biceps Curls [22].	10
2.5	Mŕtvy ťah [22].	11
2.6	Leg Press [22].	11
2.7	Hack Squat [22].	12
2.8	Rope Biceps Curl [22].	13
2.9	Rope Biceps Curl [22].	13
3.1	Zobrazenie rozpoznávaných bodov.	15
3.2	Architektúra RepNetu [14].	18
3.3	Matice podobnosti pre rôzne pohyby [14].	18
3.4	Architektúra modelu YoNet [8].	20
4.1	Formát WebSocket rámca podľa RFC6455.	24
4.2	Bloková schéma LIS3DH [30].	28
4.3	DPS a 3D model dvojvrstvovej dosky.	30
4.4	DPS a 3D model štvorvrstvovej dosky.	31
4.5	Puzdro pre dvojvrstvovú verziu DPS.	31
4.6	Držiak pre ToF modul.	32
4.7	DPS a 3D model verzie S3.	33
5.1	Formát správy prijatej cez WebSocket.	38
5.2	Štruktúra tabuliek v databáze.	39
5.3	Tri sety desiatich opakovaní cviku Lateral Row.	40
5.4	Aplikovanie prahového algoritmu.	41
5.5	Cvik so zadržiavaním váhy.	42
5.6	Filtrovanie signálu.	42
5.7	Nájdené vrcholy a inverzné vrcholy.	43
5.8	Nespracované dáta.	44
5.9	Aplikovaný filter klzavého priemeru.	45
5.10	Algoritmus pre hľadanie intervalu.	45
5.11	Šum zachytený algoritmom.	47
5.12	Dáta opakovaní spracované algoritmom.	47
6.1	Úvodné zobrazenie.	49
6.2	Prehľad senzorov.	50
6.3	Prehľad opakovaní.	50

6.4	Krabicový graf z meraní opakovaní hardvérových senzorov.	52
6.5	Histogram rozdielov výsledkov všetkých setov.	52
6.6	Výsledky analýzy opakovaní zo softvérových senzorov.	53
6.7	Histogram rozdielov výsledkov všetkých setov.	53

Kapitola 1

Úvod

Možnosti získavania objektívnych dát a spätnej väzby v športových aktivitách sú v modernom svete oveľa dostupnejšie, ako kedykoľvek predtým. Častou voľbou užívateľov sú práve inteligentné náramky a hodinky, ktoré vďaka pokroku v oblasti biometrických senzorov dokážu sledovať tep, pohyb a ďalšie metriky. Dôvod potreby užívateľov merať dáta zo svojich športových výkonov môže byť prisudzovaný ich túžbe kvantifikovať svoj výkon v oblasti, ktorá ich baví. Potreba kvantifikovať tieto údaje môže vyplývať aj z dnešnej doby, v ktorej sa úspech meria množstvom údajov – napríklad najťažší zdvih pri cvičení, počet sledovateľov na sociálnych sieťach, množstvom zobrazení a podobne [7].

Všetky takéto zariadenia sú zamerané priamo na koncového užívateľa a je nutné, aby si ich zakúpil. Táto práca sa zameriava na vývoj systému, ktorý je vhodný pre fitness centrá, ktoré svojim užívateľom chcú poskytnúť spätnú väzbu. Užívateľia tak nebudú nútení kupovať si vlastné zariadenia. Tento systém tiež nevyžaduje od užívateľa prakticky žiadny ďalší vstup okrem toho, ktorý bežne vo fitness centre robí – cvičenie. Výhodou tak nie je len získanie individuálnej spätnej väzby cvičenia užívateľa, ale aj získanie prehľadu fitness centra o výkonoch svojich klientov, či prípadne porovnanie užívateľov s ostatnými výkonmi.

Systém získava dáta (nielen) z fitness strojov prostredníctvom navrhnutého hardvéru a spracováva ich pomocou softvérového riešenia. Snaží sa užívateľom sprostredkovať objektívny prehľad o kvalite vykonávania cvikov, počte vykonaných opakovaní a ďalších metrikách vhodných pre zlepšenie svojho výkonu.

V kapitole 2 sú čitateľovi priblížené rôzne typy fitness strojov, dôležitosť merania cvikov a zmysel tejto práce, ktorá môže byť prvým krokom ku komplexnému riešeniu pre poskytovanie spätnej väzby užívateľom zo strany fitness centra. Ďalej opisuje bežné cviky, ktorých mechanika a možnosti merania budú priblížené v kapitole 3.

Nadväzujúca kapitola 4 popisuje problematiku návrhu takéhoto systému. Hovorí o kompromisoch, ktoré je nutné pri návrhu zvážiť. Tie zahŕňujú aj výslednú obstarávaciu cenu, ktorá by mala byť finančne dostupná aj pre menšie fitness centrá. Opisuje sa v nej akými spôsobmi je možné pomocou hardvéru merať opakovania cvikov vykonávaných na strojoch. Kapitola 5 opisuje implementáciu softvéru pre navrhnuté senzory a server, ktorý vykonáva analýzu senzorických dát. V nej sú priblížené problémy správneho určovania opakovaní z prijímaných dát. V tejto kapitole sú predstavené riešenia na niektoré z identifikovaných problémov.

Navrhnutý systém ukladá analyzované dáta do databázy, ku ktorej poskytuje webové aplikačné rozhranie, cez ktoré dokáže dáta posielať koncovým aplikáciám. Tie tak môžu tvoriť rôzne štatistiky z poskytnutých dát. Spolu s prehľadom dosiahnutej úspešnosti sú tieto možnosti opísané v kapitole 6.

Kapitola 2

Fitness a cvičenie

Fitness je koncept, ktorý je možno popísať rôznymi spôsobmi. V kontexte zdravia je fitness vnímaný ako stav dobrej fyzickej kondície. Dá sa zároveň tvrdiť, že fitness cvičenie a posilňovanie má rastúci trend popularity, čo naznačuje aj zdvojnásobenie počtu tréningových centier vo svete od roku 2009 [5]. Moderné fitness stroje a vybavenie využívajú priestor fitness centra veľmi efektívne, to po stránke ich konštrukcie ako aj ich použitia. Nie je preto prekvapením snaha neustáleho zlepšovania ich funkčnosti a efektívnosti. Táto kapitola vysvetľuje základné pojmy v oblasti cvičenia využívané v tejto práci, spolu so samotnými prínosmi zaznamenávania cvičenia.

2.1 Typy cvičení

V dnešnej dobe existujú rôzne druhy cvičení, rozdeľujú sa predovšetkým v cieľoch, ktoré chcú jednotlivci dosiahnuť. Medzi ne patria napríklad kardiovaskulárne cvičenia, rôzne skupinové triedy cvičení a touto prácou sledované silové cvičenia.

Kardiovaskulárne cvičenia predstavujú cviky zamerané na zvýšený srdečný tep ako napríklad beh, cyklistika či plávanie. Cieľom týchto cvičení je zlepšenie kondície – to najmä v zmysle výdrže – a spaľovania kalórií. Tieto typy cvičení sú všeobecne jednoducho merateľné elektronickými zariadeniami, keďže ich ciele sú často stanovené ako kombinácia vzdialenosti (resp. sledu úkonov) v závislosti od času. Tomu pomáha aj fakt, že existuje niekoľko typov senzorov umožňujúcich merať srdečnú frekvenciu v podobe tepu, ktoré dokážu merateľne preukázať efektívnosť cvičenia a kondíciu užívateľa.

Skupinové triedy cvičení v sebe zahŕňajú široké spektrum cvikov pre ľudí rôznych vekových kategórií. Ich cieľom môže byť zvýšenie flexibility, znižovanie bolesti pohybového aparátu, či prosté udržiavanie zdravého množstva pohybu. Sú to cvičenia ako jóga, aerobic či pilates.

Silové cvičenie, často nazývané aj posilňovanie, je forma cvičenia zameraná predovšetkým na zvýšenie sily a svalovej hmoty. Tieto typy cvičenia majú spoločný prvok vo forme odporu resp. záťaže. V tejto kategórii je cvičenie so strojmi, vlastnou váhou alebo napríklad vzpieranie a silový trojboj.

2.2 Zmysel merania cvikov

Táto práca sa zameriava špecificky meraním **silových cvičení**, ktoré sú cielejšie na **svalovú hypertrofiu**. Tento pojem označuje zväčšovanie veľkosti svalových vlákien – speci-

ficky ich priečny priemer. Tento štýl cvičenia využívajú najmä kulturisti, ktorých cieľom je mimo iného práve maximalizácia vlastnej svalovej hmoty. Samotná efektívnosť cvičenia je teda meraná skôr ako výsledok po určitej dobe času, než ako samotný priebeh. Existujú však princípy, ktorých sa cvičenci držia, pre dosiahnutie najefektívnejšieho výsledku. Takýmto princípom je **progresívne preťažovanie**. Progresívne preťažovanie je princíp postupného zvyšovania záťaže svalovej partie medzi jednotlivými tréningovými dňami. Využívanie tohoto princípu výrazne ovplyvňuje efektívnosť rastu svalovej hmoty a je považovaný za štandard v modernej kulturistike [29].

Tréningový plán jednotlivca zameraného na rast svalov je typicky rozdelený na tréningové dni. Každý takýto tréningový deň obsahuje sadu cvikov, ktoré by mal daný človek postupne odcvičiť. Jeden cvik je v tréningovom pláne tvorený určitým počtom sérií a počtom opakovaní v každej takejto sérii. Pod pojmom séria rozumieme neprerušované vykonávanie cviku podľa určitého počtu opakovaní pri konzistentnom tempe. V nadväznosti na princíp progresívneho preťažovania je možné zvýšiť záťaž tromi merateľnými spôsobmi (ak zanedbáme frekvenciu cvičenia):

1. Zvýšenie veľkosti záťaže.
2. Zvýšenie počtu opakovaní v jednej sérii.
3. Zvýšenie počtu sérií v rámci jedného cviku.

Najefektívnejšie spôsoby pre docelenie rastu svalovej hmoty na základe cvičenia sú neustále predmetom rôznych štúdií, avšak vo väčšine štúdií vládne konsenzus v tom, že dodržovanie správnej stravy a zvyšovanie záťaže kladenej na svaly vedie k efektívnej svalovej hypertrofii [15, 29]. Metodickým zapisovaním a navyšovaním týchto parametrov dokáže jedinec efektívne dosahovať lepších výsledkov, ako bez akejkoľvek reflexie svojho cvičenia. Na týchto tvrdeniach sa zhoduje niekoľko štúdií, a to najmä v spojení s redukovaním telesnej hmotnosti [4].

2.3 Automatizácia zapisovania výkonu

Rovnako ako pri iných druhoch cvičení, ako sú beh alebo plávanie, technológia automatizovaného merania a zaznamenávania výkonu, ktorá dokáže poskytnúť konkrétnu spätnú väzbu výrazne prispieva k zlepšeniu výkonu a sledovaniu individuálneho pokroku [3]. Poskytnutie detailnej a objektívnej spätnej väzby umožňuje jednotlivcom lepšie pochopiť svoje fyzické schopnosti a stanoviť si realistické ciele. Moderné technológie, ako sú fitness náramky, inteligentné hodinky a mobilné aplikácie zaznamenávajú rozličné aspekty ľudského výkonu, a to vrátane už zmienenej srdcovej frekvencie, prejdenej vzdialenosti a dokonca vypočítanej kvality spánku, úrovne stresu alebo počtu spálených kalórií. Tieto dáta poskytujú užitočné informácie nielen pre amatérskych športovcov, ale aj pre profesionálov, ktorí si vďaka nim môžu prispôbiť svoj tréningový plán.

Nedostatok spätnej väzby je často považovaný za hlavný dôvod poklesu motivácie, čo môže viesť k strate záujmu o fitness [17]. V dnešnej dobe plnej zaneprázdnenosti a neustále meniaceho sa životného štýlu potrebujú jednotlivci okamžité a merateľné výsledky svojich snáh, aby zostali motivovaní. Automatizované sledovanie pokroku teda nielenže podporuje udržiavanie zdravého životného štýlu, ale tiež pomáha vytvárať pocit dosiahnutia a uspokojenia z vlastného pokroku a úspechov. Navyše, integrovanie sociálnych prvkov, ako je zdieľanie pokroku na sociálnych sieťach alebo súťaženie s priateľmi, môže ešte viac zvýšiť motiváciu a udržať dlhodobý záujem o fyzickú aktivitu [19].

2.4 Fitness stroje a cviky

Základ tejto práce tvorí meranie samotných fitness strojov, je pre to nutnosťou aspoň v krátkosti predstaviť najčastejšie typy strojov a mechaniku ich využitia. Pre jednoduchosť budeme stroje rozlišovať podľa možného spôsobu merania pohybu závažia do troch kategórií. Pod každou takouto kategóriou sú opísané príklady cvikov, ktoré je možné na nich vykonávať. Opis cvikov nie je exaktný, avšak v spojení s obrázkom má za úlohu dať čitateľovi aspoň hrubú predstavu, ako takýto pohyb vyzerá. Vďaka tomu je možné vytvoriť si hrubú predstavu o tom, ktoré aspekty je nutné zohľadniť pri počítaní opakovaní takéhoto cviku.

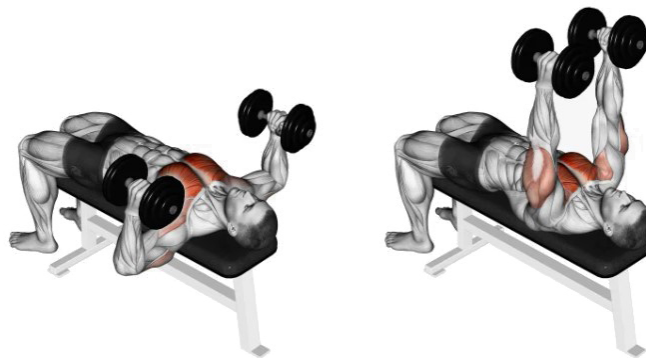
2.4.1 Pomocné konštrukcie

Do tejto kategórie patria všetky pomocné konštrukcie a lavičky, na ktorých je možné vykonávať cviky pomocou závažia v podobe jednoručných či vzpieračských čínok, alebo vlastnou váhou. Tieto lavičky, ktoré sú často polohovateľné a sú navrhnuté tak, aby poskytovali podporu a umožnili presné cielenie na rôzne svalové skupiny. Polohovateľnosť lavičiek prispieva k ich všestrannosti, keďže umožňujú meniť uhol a intenzitu cvičenia, čo je dôležité pre zasiahnutie rôznych svalových partií a zvýšenie efektivity tréningu. Mäkké polstrovanie a nastaviteľné opierky znižujú riziko zranení a zabezpečujú správnu podporu pre chrbát a iné časti tela počas cvičenia. Multifunkčnosť týchto lavičiek umožňuje vykonávanie širokého spektra cvičení, od výtlakov na prsia až po bicepsový zdvih. To však znamená, že meranie výkonu na takýchto konštrukciách bude nutné robiť pomocou vizuálnych metód spracovania obrazu.

Pomocné konštrukcie môžu byť napríklad aj kliečky, v ktorých sa vykonávajú cviky ako drep so vzpieračskou činkou, alebo napríklad špeciálne lavičky na vykonávanie cvikov ako sú sed-lahy (*angl. sit up*).

DB Bench Press

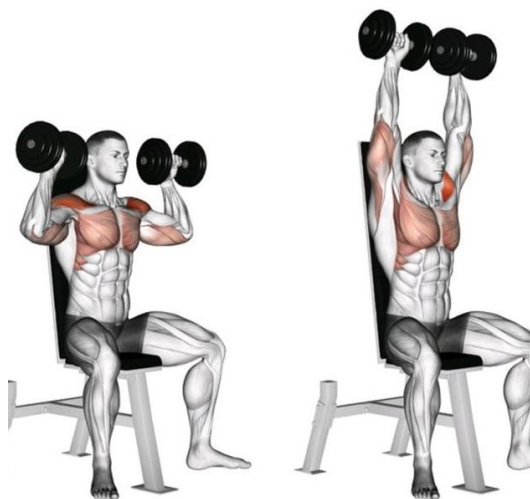
DB (dumbbell) Bench Press, teda tlaky na lavičke s jednoručnými činkami, je základným cvikom na rozvoj svalov prsnej oblasti. Cvik sa vykonáva tak, že cvičenec leží na chrbte na lavičke a činky zdvíha vertikálne hore od hrudníka a späť pričom využíva ohyb laktových kĺbov. Opakovania je možné sledovať iba v pohybe rúk.



Obr. 2.1: DB Benchpress [22].

DB Shoulder Press

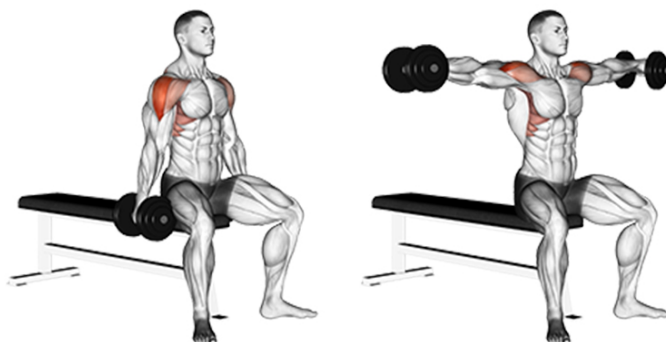
DB Shoulder Press, alebo tlaky jednoručných čínok nad hlavou je efektívny cvik pre posilnenie ramien a horného dielu chrbta. Cvičenec sedí na lavičke, drží činky v úrovni ramien a zdvíha ich vertikálne nad hlavu. Sledovanie opakovaní teda znova závisí iba na pohybe rúk.



Obr. 2.2: DB Shoulder Press [22].

Lateral raise

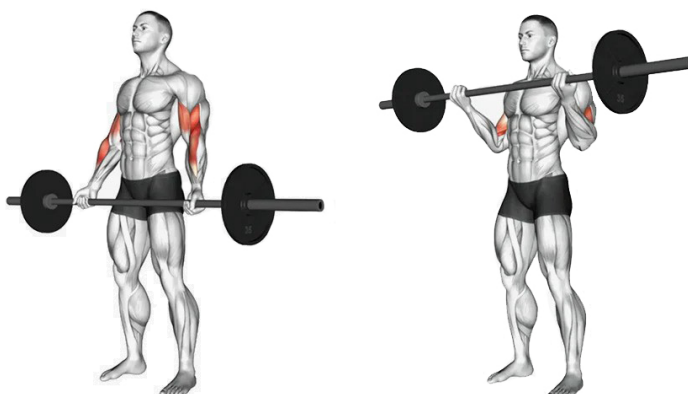
Lateral Raise sú bočné zdvihy s jednoručnými činkami, sú zamerané na posilnenie bočných a zadných častí ramien. Cvičenec stojí alebo sedí so vzpriamenou resp. jemne predklonenou postavou, činky drží pozdĺž tela a zdvíha ich bočne nahor až do úrovne ramien, pričom laktové kĺby sú mierne ohnuté. Tento cvik sa môže vykonávať aj v postoji, v oboch prípadoch sú v zmysle počítania opakovaní opäť dôležité iba ruky.



Obr. 2.3: Lateral Raise [22].

BB Biceps Curls

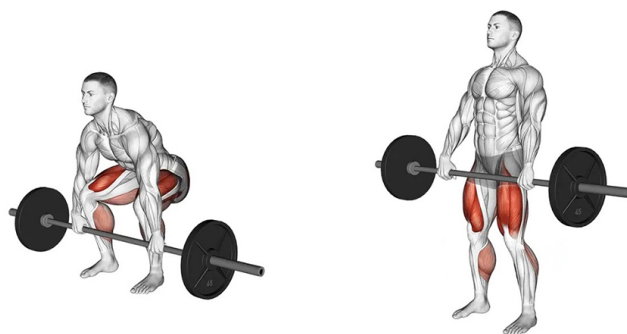
BB (Barbell) Biceps Curls – základné cvičenie pre posilnenie bicepsu – sa vykonáva so vzpieračskou činkou. Cvičenec drží činku pred sebou, lakte sú fixne pritlačené k telu. Pomalým zdvíhaním činky k hrudi a následným kontrolovaným spúšťaním sa aktivuje biceps.



Obr. 2.4: BB Biceps Curls [22].

Deadlift

Deadlift, alebo mŕtvy ťah, je komplexné cvičenie, ktoré zasahuje viaceré svalové skupiny, vrátane chrbta, nôh a rúk. Cvik posilňuje hlavne dolnú časť chrbta, zadok a stehná. Cvičenec stojí s nohami v šírke ramien, nad vzpieračskou činkou. Pri ohýbaní sa v kolenách a bedrách sa činka zdvíha zo zeme, pričom chrbát by mal zostať rovný. V koncovnej polohe je cvičenec vyrovnaný s činkou zarovno pásu. Opakovania sa teda môžu oproti ostatným cvikom sledovať ako pohyb celého tela, nielen rúk.



Obr. 2.5: Mŕtvy ťah [22].

2.4.2 Stroje s kotúčovým závažím

Fitness stroje, na ktoré sa pridáva váha pomocou kotúčov fungujú na princípe mechanického odporu. Tento odpor je vytváraný pohybom váh, ktoré sú umiestnené na stroji.

Rám stroja poskytuje pevnú a stabilnú štruktúru, ktorá podporuje závažia a ďalšie mechanické časti. Váhové kotúče sú základným a najbežnejším prostriedkom pre vytváranie odporu. Užívateľ nastavuje požadovanú váhu umiestnením kotúčov na určené miesto na stroji. Vodiace tyče alebo kladky slúžia na usmernenie a stabilizáciu pohybu váhových kotúčov. Kladky môžu tiež zmeniť smer pohybu alebo umožniť viacero cvičebných pozícií.

Sedačky a opierky poskytujú pohodlie a správnu podporu pri cvičení. Sú často nastaviteľné, aby vyhovovali rôznym veľkostiam a typom cvičení. Rukoväte alebo držadlá umožňujú užívateľovi vykonávať rôzne druhy cvikov. Ich tvar a umiestnenie sú špecifické pre každý typ stroja.

Leg Press

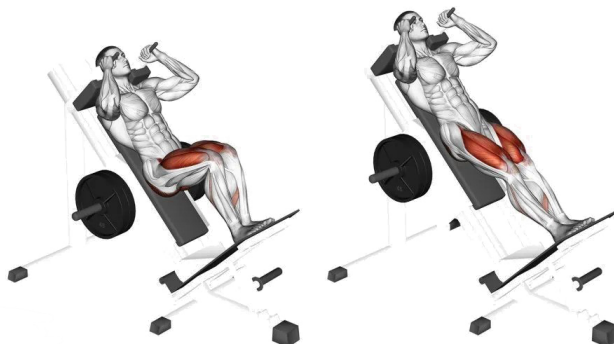
Leg Press je cvičenie zamerané na posilnenie dolných končatín, najmä stehien a zadku. Pri tomto cviku sa cvičenec umiestni do stroja na leg press, kde leží alebo sedí s nohami na platforme. Cvik sa vykonáva tlačením platformy od seba pomocou nôh.



Obr. 2.6: Leg Press [22].

Hack Squat

Hack Squat je variáciou drepov, ktorý sa vykonáva na špeciálnom stroji. Cvičeniec stojí s chrbtom opretým o podložku a má ramená umiestnené pod držiakmi. Nohy sú na platforme v šírke ramien. Cvik sa vykonáva ohýbaním kolien a spúšťaním tela dole, podobne ako pri klasických drepoch. Po dosiahnutí spodnej pozície sa cvičeniec zdvihne späť do východiskovej polohy. Tento stroj umožňuje ciele posilňovanie predných a zadných stehien a zadku, pričom zároveň minimalizuje záťaž na chrbticu. V prípade sledovania pohybu váhy je tento pohyb inverzný k pohybu predošlého cviku – Leg Pressu.



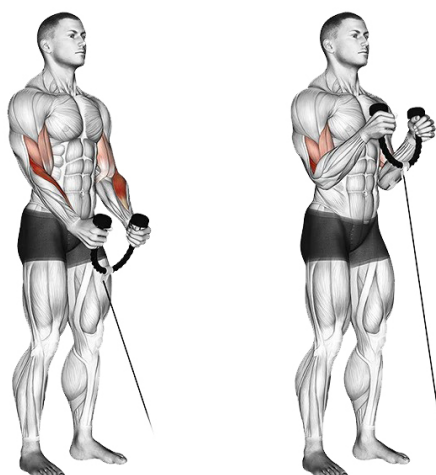
Obr. 2.7: Hack Squat [22].

2.4.3 Stroje s kladkou

Do tejto kategórie patria všetky stroje, ktoré majú zásobník váhy v podobe veže železných plátov, kde si cvičeniec vyberie váhu vkladáním kolíka medzi zliatinové kvádre. Vložením kolíka zafixuje od kolíka vrchnú časť kvádrov do kladkového systému, ktorú zdvíha ťahaním či tlačením častí stroja – adaptérov. Niektoré stroje sú multifunkčné a je možné na nich vykonávať rôzne cviky, pre testovanie počítania opakovaní takýchto cvikov na jednom stroji sú v množine zahrnuté cviky **Rope Biceps Curl** a **Rope Pulldown**, ktoré je možno vykonávať na tom istom stroji. Pohyb váhy je viacmenej uniformný pre všetky takéto stroje, a to nezávislo od cviku. To znamená, že ak je stroj navrhnutý len na jeden konkrétny cvik, je možné aplikovať rovnaké rozpoznávacie metódy na celú množinu týchto strojov.

Rope Biceps Curl

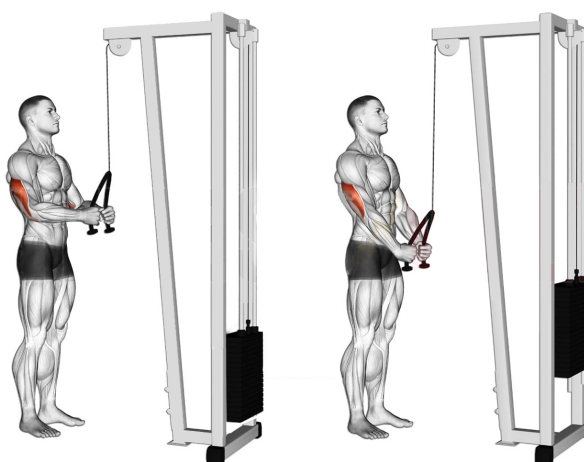
Rope Biceps Curl je varianta cvičenia na bicepsy, ktoré sa vykonáva s pomocou kladky a lana. Cvičeniec stojí pri kladkovom stroji a drží lano pripevnené k **spodnej** časti stroja. Cvik sa vykonáva zdvíhaním lanových koncov smerom nahor k hrudi, pričom sa zameriava na kontrakciu bicepsov.



Obr. 2.8: Rope Biceps Curl [22].

Triceps rope pushdown

Triceps Rope Pushdown je efektívne cvičenie na posilnenie tricepsov, vykonávané pomocou lana a kladky. Cvičenec stojí pred strojom, kde drží lano pripojené k **hornému** výstupu stroja a čelí k stroju. Cvik sa začína s lanom držaným v uhle 90 stupňov v lakťoch a rukami blízko tela. Cvičenec potom tlačí lano nadol, kde končí s plne rozťahnutými rukami v spodnej pozícii.



Obr. 2.9: Rope Biceps Curl [22].

Kapitola 3

Analýza merania cvikov

3.1 Extrakcia dát

Prvým krokom väčšiny metód rozpoznávania pohybu či opakovaní je získavanie dát, na základe ktorých je možné aplikovať konkrétnu rozpoznávaciu metódu. Medzi najpoužívanéjšie metódy extrakcie takýchto dát sú neurónové siete a rôzne hardvérové senzory.

3.1.1 Hardvérové senzory

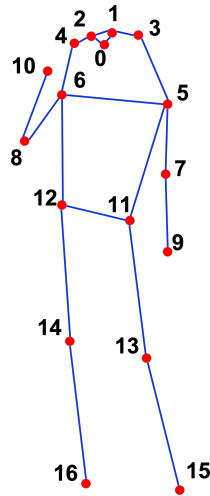
Pod pojmom hardvérové senzory sú myslené zariadenia, ktoré musia byť v kontakte so strojom alebo užívateľom, aby mohli produkovať dáta. Najpopulárnejšou voľbou fitness aplikácii pri klasifikácii a určovaní opakovaní cvikov je využitie už zmienených IMU jednotiek, ktoré sú umiestnené v smart hodinkách a telefónoch. Tie kombinujú akcelerometer, magnetometer a gyroskop. Pomocou vzorkovania týchto dát a následnej analýzy pomocou napr. Bayesovského klasifikátora dokážu určovať s veľkou úspešnosťou cviky a opakovania. Aplikácia s názvom myHealthAssistant deklaruje 93% úspešnosť v adaptívnom rozoznávaní pomocou takéhoto klasifikátora [31]. To všetko zvláda v kombinácii s veľkou úspešnosťou v počítaní samotných opakovaní. Je ale nutné podotknúť, že množina identifikovaných cvikov je zásadne obmedzená.

3.1.2 Neurónové siete na určovanie polohy tela

State-of-the-art metódy sú populárnym riešením čo sa týka rýchlostou určovania a potrebným výkonom. Momentálne existuje niekoľko modelov, ktoré efektívne určujú prítomnosť ľudského tela a polohu jednotlivých kľúčových bodov – kĺbov. Väčšina týchto modelov je pri určovaní kĺbov natrénovaná na dostupných datasetoch ako je COCO-Pose. Polohy kľúčových bodov sú zvyčajne reprezentované ako sada dvoch súradníc $[x, y]$. K týmto súradniciam sa zväčša pridá tretí bod – konštanta viditeľnosti $[x, y, \text{viditeľnosť}]$ súradníc. Výstupom je pevne definovaný počet bodov s dopredu známymi indexmi pre jednotlivé kĺby – tzn. že je možné sledovať špecifický kĺb a nie len iba “náhodný” kĺb [21], pričom v spomínanej COCO-Pose je zadefinovaných až 17 sledovaných bodov zobrazených na obrázku (nie všetky sú kĺby) 3.1.

Modely sa medzi sebou líšia v ich úspešnosti identifikácie, možným počtom ľudí (multi-person/single-person) a rýchlostou inferencie. Medzi dôležité rozdiely patrí aj fakt, či potrebujú viacero snímok s pohybom na identifikovanie bodov, alebo im stačí iba jeden snímok.

Pre jednoduchšie použitie sú porovnané modely, ktoré už z jedného snímku s viacerými ľuďmi dokážu produkovať výstup.



Obr. 3.1: Zobrazenie rozpoznávaných bodov.

Pri určovaní polohy kĺbov na obraze je možné použiť modely, ktoré odhadujú pozíciu kĺbu v 3D priestore. Tieto modely prichádzajú do popularity vďaka ich širokému spektru využitia. Okrem komplikácií, ktoré majú aj klasické 2D modely (komplikovaná rôznorodosť pozadia, rôzne oblečenie, rôzne polohy tela, viacerí ľudia na obraze, ...), tieto prinášajú napr. nutnosť určenia približnej veľkosti objektov na obraze, čo je potrebné k celkovému určeniu 3D polohy danej časti tela. Do toho vstupujú ostatné problémy, ktorými sú pohyb kamery a tieň, ktoré menia zdanie hĺbky obrazu a tvaru ľudí na ňom. Fúzia dát je nutná, ak je k určovaní pozície tela v priestore používaných viacerých snímkov vytvorených z rôznych uhlov. Takáto fúzia však nemá triviálne riešenie ani zanedbateľný čas výpočtu [35]. Z týchto dôvodov – aj napriek tomu, že by použitie takéhoto modelu značne zjednodušilo riešenie – nie je implementácia určovania opakovaní v reálnom čase pomocou takéhoto riešenia možná, a to kvôli dostupnosti nejakého rozumného výpočtového výkonu.

Yolov8

Yolo (You only look once) model je jeden z momentálne najpopulárnejších a aktívne vyvíjaných modelov. Najnovšia 8. verzia (schopná detekcie pózy) dokáže na COCO datasete správne určovať kĺby s priemernou presnosťou až 90 %. Ponúka prívetivú Python knižnicu s rôznymi možnosťami vykonávania inferencie na rôznych zariadeniach a platformách ako sú ONNX, TensorRT apod. V svojich meraniach uvádza výkon až 280 FPS (snímokov za sekundu) pri použití grafickej karty Nvidia A100 s nástrojmi TensorRT. Je potrebné dodať, že cena takejto karty je až príliš vysoká (približne pol milióna CZK) [34].

RTMPose

RTMPose (Real-Time Multi-Pose) je model, ktorý si dáva za cieľ zväčšiť rýchlosť inferencie a zároveň znížiť požiadavky na potrebný hardvér. Ponúka rôzne veľkosti modelov s rôznymi rýchlosťami, pričom najefektívnejší model M dosahuje 75 % priemernú úspešnosť na COCO

datasete. Tieto výsledky dokáže generovať rýchlosťou 430 FPS na relatívne dostupnej grafickej karte typu Nvidia GTX 1660 Ti. Zároveň je táto karta už staršia, preto predpokladaný výkon novej karty za podobnú cenu by mal byť teoreticky až o 30% vyšší. V tabuľke 3.1 sú zobrazené výkony jednotlivých modelov na tejto grafickej karte a na procesore Intel i7-11700. V týchto dátach je možno vidieť výhodu, že pri obetovaní pár percent presnosti, je možné výkon zrýchliť až o tretinu [16].

Tabuľka 3.1: Dosiahnuté výsledky modelu [16].

Model	AP(COCO)	CPU-FPS	GPU-FPS
t	68.5	300+	940+
s	72.2	200+	710+
m	75.8	90+	430+
l	76.5	50+	280+

Detectron2

Detectron je pokročilý softvérový rámec pre objektovú detekciu a segmentáciu obrazu vyvinutý výskumníkmi z Facebook AI Research (FAIR). Zahŕňa model s názvom DensePose postavený na modeli typu R-CNN. Vstupom sú snímky obrazu a výstupom môže byť napríklad 3D sieť, ktorá opisuje postavu detegovaného (nielen) človeka v 3D alebo 2D priestore. Tento nástroj je však v reálnom čase len ťažko použiteľný, kvôli jeho nároku na výkon a samotnej výpočtovej zložitosti spracovania výslednej 3D siete [28].

Iný nástroj, ktorý výskumníci z tejto skupiny predstavili je VideoPose3D. Tento model využíva body detegovanej pózy v 2D priestore, ktoré následne využíva na predikciu pózy v 3D priestore. Napriek tomu, že takéto rozdelenie úlohy značne zjednodušuje celkový problém, zo svojej podstaty zavádza nový problém, a síce viacero póz v 3D priestore sa môže mapovať na jednu pózu v 2D priestore [27].

3.2 Metódy pre rozpoznávanie opakovaní a pohybu

Rozpoznávanie pohybu je úzko prepojené so samotným počítaním opakovaní daného pohybu. V prípade použitia správnej metódy rozpoznania pohybu, je možné spočítať výskyt daného pohybu v zaznamenaných dátach. Po extrakcii vhodných dát z reálneho sveta sa tieto dáta následne vložia na vstup metód, ktoré ich dokážu ďalej analyzovať – či už samotný pohyb alebo opakovania.

3.2.1 Path signature

Matematickou metódou rozpoznávania pohybu je metóda počítania tzv. podpisu cesty pochádzajúceho z oblasti Stochastickej analýzy, a to konkrétne z Rough path theory. Táto metóda sa spolieha na vstup v podobe extrahovaných bodov – ako napríklad polohy kĺbov. Tento prístup spočíva v transformácii údajov o pozícií týchto bodov na trajektóriu v čase, z ktorej sa potom extrahujú charakteristické funkcie prostredníctvom metódy “path signature”. Tento proces zahŕňa rozklad trajektórie na jednoduchšie sub-trajektórie a ich transformáciu, čím sa získavajú robustné vlastnosti pre klasifikáciu akcií. Výsledkom je schopnosť identifikovať a klasifikovať rôzne ľudské akcie s vysokou presnosťou.

Jedna z výhod tohoto prístupu je, že súčet podpisov čiastočných pohybov je rovný podpisu pohybu zloženého z týchto čiastočných pohybov. To umožňuje odhadnutie cviku už vopred z dát nejakého čiastočného pohybu vrámci tohoto cviku. Nevýhodou je, že výpočet podpisu cesty môže byť výpočtovo náročný, najmä pri vyššom počte bodov a zložitejších trajektóriách. To je spôsobené samotným spôsobom výpočtu, ten je počítaný ako niekoľko-dimenzionálny opakovaný integrál a výpočet sa sťažuje aj podľa toho, či sa trajektória skladá z veľkého počtu bodov alebo je potrebné vysoké rozlíšenie. Tento prístup môže byť tiež náchylný na numerické chyby pri aproximácii týchto integrálov. Avšak, efektívnosť môže byť zlepšená použitím optimalizovaných algoritmov a výpočtových techník [38].

Po aplikovaní metódy na rôzne datasety, metóda výrazne prekonala iné metódy založené na skeletových údajoch vrátane tých, ktoré používajú rekurentné neurónové siete (RNN) alebo Long Short-Term Memory (LSTM) siete [38]. Tento prístup tiež umožnil lepšie porozumenie vzájomných interakcií jednotlivých telesných bodov.

3.2.2 Detekcia a počítanie pomocou Bayesovského klasifikátora a Markovového modelu

V štúdiu predstavenú v roku 2007 výskumníci použili dáta z akcelerometra umiestneného v rukavici a na opasku. Ich cieľom bolo identifikovanie cvikov a počtu opakovaní vykonávaných s voľnou váhou – to sú cviky, kde váha nie je na stroji, ako napr. cviky opísané v sekcii 2.4.1.

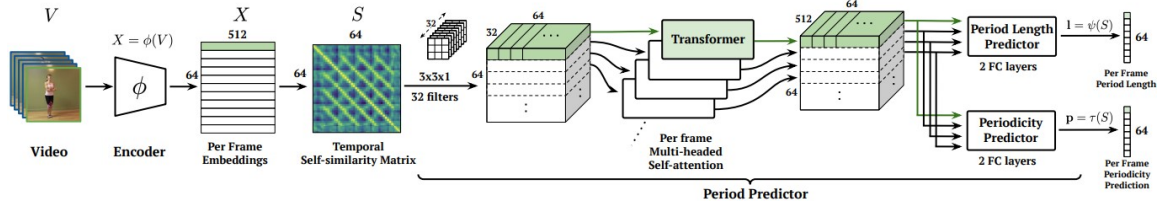
Najskôr bolo potrebné určiť najbežnejšie cviky a postoj, v ktorom sa vykonávajú. V štúdiu sa zamerali na 9 cvikov, medzi ktoré patrili aj v prvej kapitole opísané **DB Bench Press**, **Deadlift**, **DB Shoulder Press** a **Lateral raise**. Následne boli dáta z rukavice vyhodnocované. V práci bola spomenutá aj možnosť umiestnenia senzoru priamo na váhu, avšak hlavným argumentom bolo stratenie stupňov voľnosti v pohybe ruky. Senzor na opasku určoval postoj človeka – ležanie a státie.

Na detekciu cviku bol použitý naivný Bayesovský klasifikátor a skrytý Markovov model, a to na základe úspechu z predošlých štúdií. Na určovanie opakovaní bol vyvinutý algoritmus počítania vrcholov z akcelerometrických dát. Identifikácia cviku prebiehala posúvaním okna cez dáta, kde každé okno zahrnovalo 50 % dát zo susedného okna, následne bol z tohoto okna extrahovaný vektor rysov (*angl. feature vector*), ktorý bol vkladáný na vstup klasifikátora. Algoritmus potom zoskupil všetky okná s rovnakým typom cviku na ďalšiu analýzu. Čo sa týka určovania cviku pomocou Markovového modelu, pre každý cvik bolo nutné vytvoriť vlastný model. Algoritmus tak mohol extrahovať rysy a z tých vypočítal pravdepodobnosť daného cviku. Klasifikácia potom prebiehala vybraním cviku s najväčšou pravdepodobnosťou vypočítanou Markovovým modelom. Vo výsledku mal tento prístup úspešnosť 90 % v určovaní typu cviku a iba 5 % chybovosť v určovaní počtu opakovaní [12].

3.2.3 RepNet

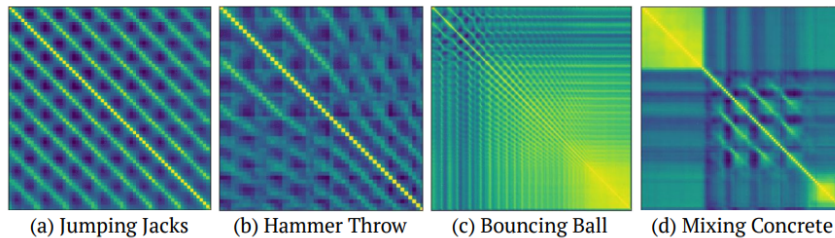
RepNet je neurónová sieť predstavená v roku 2020 výskumným tímom spoločnosti Google, ktorá dokáže určovať periódu opakujúceho sa pohybu z obrazu videa. Toto riešenie adresuje niekoľko problémov v spojení s počítaním opakovaní [14]. Práca ukazuje, že keď sa dizajnér neurónovej siete zameria na detailné pochopenie a modelovanie špecifickej úlohy – ako je analýza periodického pohybu – môže to výrazne zvýšiť efektívnosť a presnosť siete. Tým, že vývojári RepNetu neprístupovali k architektúre siete ako k čiernej skrinke, ale naopak, detailne skonštruovali jej architektúru s ohľadom na konkrétne výzvy a charakteristiky úlohy, dokázali dosiahnuť lepšie výsledky. Tento prístup umožňuje sieti optimálne využívať

dostupné dáta a presnejšie modelovať opakujúce sa vzory v pohyboch, čo vedie k lepšiemu rozpoznávaniu a kvantifikácii periodických aktivít z videa.



Obr. 3.2: Architektúra RepNetu [14].

Na obrázku 3.2 je zobrazená architektúra siete so vstupom o veľkosti 64 snímkov. Na extrakciu vlastností (*angl. features*) z obrazu je použitá 2D konvolučná sieť, ktorá je aplikovaná na každý snímok z videa. V tomto momente je pridaná informácia o okolitých snímkoch pomocou 3D konvolúcie. Tým vzniká vektor informácií pre každý snímok, ktorý obsahuje kombináciu svojich a okolitých vlastností. Nasleduje srdce celej architektúry – dočasná matica podobnosti. Táto matica je deterministická a je možné ju dostať vypočítaním konštanty podobnosti snímkov každého s každým. Na obrázku 3.3 sú zobrazené matice pre rôzne pohyby. Hlavná diagonálna čiara hovorí, že snímok je podobný sám sebe – čo je logické, ďalšie vzniknuté diagonály však ukazujú na prítomnosť rovnakých snímkov, ktoré sú vzdialené periodicky o niekoľko snímkov ďalej.



Obr. 3.3: Matice podobnosti pre rôzne pohyby [14].

V tomto bode je možnosť výpočtom súčtu cez všetky stĺpce dostať akýsi graf s vrcholmi, z ktorých by bolo možné heuristicky určiť počet a rýchlosť opakovaní. Autori však išli s riešením o krok ďalej, a síce v prípade videa (b) na obrázku je vidieť prípad, kedy sa opakovania hádzania kladiva zrýchľovali. Značí to priblíženie okolitých čiar k diagonále – podobné snímky sa v čase približujú. Rovnako tak v prípade (c), loptička pustená zo vzduchu stráca svoju energiu čím skáče na menšiu vzdialenosť a rýchlejšie. V prípade (d) v prvých snímkoch videa sa snímky opakujú a až následne po týchto snímkoch (v strede matice) vidno opakovania. Na tieto prípady slúži zvyšok reťazca architektúry (v obr. 3.2 označený ako “period predictor”), ktorý predstavuje natrénovanú časť tejto práce. Ďalším elementom, ktorý spôsobuje nepresnosť získaných dát môže byť pohyb kamery. Na ten sú použité afinné transformácie a vyhladzovanie obrazu.

Výstupom spracovania obrazu sú dva vektory – predikcia periódy na snímok a binárny vektor periodicity na snímok.

3.3 Existujúce riešenia

Momentálne existuje niekoľko štúdií a softvérov zameraných na detekciu cvikov resp. opakovaní. Väčšina týchto štúdií je zameraných na rozoznávanie a počítanie základných cvikov s vlastnou váhou ako sú kľuky, zhyby či drepy. V zmysle rozpoznávania pohybu existuje aj niekoľko prác, ktoré sa zaoberajú rozpoznávaním póz pri cvičení jógy.

3.3.1 GymCam

Jedným z úspešných riešení využívajúcich kamerové záznamy je práca a rovnomenný softvér **GymCam**. Celková deklarovaná úspešnosť v rozoznávaní medzi cvičením a ostatnými aktivitami je 84.6%, zatiaľ čo následné rozpoznávanie konkrétneho cviku má úspešnosť 93.6%. Počítanie opakovaní dosahuje presnosť $\pm 1,7$ opakovania. Práca pozostávala zo spracovávania obrazu z kamier, kde sa v prvom kroku algoritmicky určil optický tok, ktorý vytváral trajektórie všetkých zachytených pohybov. Zo signálu trajektórií bol vytvorený aj vektor vlastností zameraný najmä na frekvenciu zachytených udalostí na obraze. Patrili do neho hodnoty ako autokorelácia, dominantná frekvencia signálu, počet a rozptyl pretínania nulového bodu v obidvoch osách, počet vrcholov a podobne. Takýto vektor bol vstupom klasifikátora, ktorý rozoznáva medzi vykonávaním cviku a inej aktivity nesúvisiacej s cvičením. Z trajektórií sa vytvárali aj časopriestorové zhluky, ktoré sú dôležité pri počítaní samotných opakovaní. Zhlukom pomáhali manuálne naznačené zóny, kde by cviky mali prebiehať.

Pre presnú detekciu však nie je logicky možné tento vektor počítať zo všetkých dát. Trajektórie tak boli rozdelené do okien so šírkou intervalu päť sekúnd. Tie boli vložené na vstup zmienému klasifikátoru, v tomto prípade sieť typu MLP (Multi Level Perceptron). V prípade zachytenia viacerých po sebe nasledujúcich okien vyhodnotených ako cvičenie, sú tieto okná spojené a tvoria tak trajektóriu potencionalneho cviku. Cvik je klasifikovaný pomocou časopriestorových zhlukov, ktoré sa tvoria vykonávaním konkrétneho typu cviku. Opakovania sú určované pomocou MLP regresora na trajektórií vyhodnotenej ako cvik.

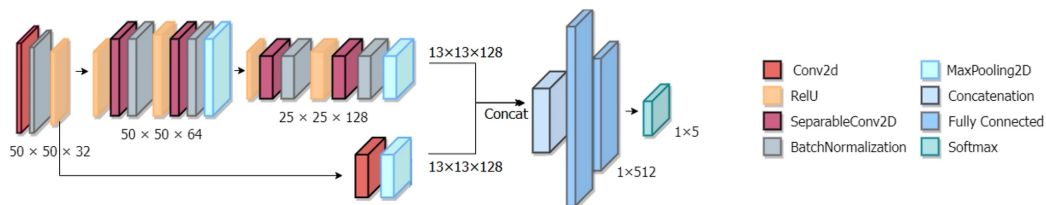
V práci sú taktiež spomenuté niektoré problémy tohoto prístupu. Jednou z nich je spracovávanie dát z prostredia, kde sú užívatelia pri cvičení príliš blízko seba. Pri vykonávaní podobného cviku pri rovnakej frekvencii tak môže dôjsť k splynutiu týchto dvoch cvikov do jedného. Tento problém vzniká najmä kvôli použitému prístupu určovania cvikov len na základe vytvorených zhlukov. Jedno zo spomenutých riešení bolo použitie detekcie hĺbky bodov alebo ľudských póz (ako napr. spomenutý model YOLOv8). Ďalším problémom bol presný opak tejto situácie, a to že jeden cvik mohol byť chybné rozdelený ako dva úplne iné cviky. Z práce taktiež vyplýva otázka integrácie v rôznom prostredí, aj keď samotná práca deklaruje pravdepodobnú invarianciu ku konkrétnemu prostrediu, ktorá však nebola formálne otestovaná. Opiera sa najmä o fakt, že využíva iba dáta získané z obrazu založené na čase a frekvencii – manuálne dokreslené zóny sú jednoduchý jednorázový implementačný krok [18].

3.3.2 YoNet

YoNet predstavuje relatívne nový model hlbokého učenia, navrhnutý na identifikáciu piatich rôznych jógových póz. Tento model podľa autorov prekonáva existujúce modely klasifikácie obrazu, ako sú ResNet, InceptionNet, InceptionResNet a Xception, dosahujúc vyššiu presnosť s menším množstvom trébovaných dát. Architektúra YoNet má výhodu v individuálnom spracovaní priestorových a hĺbkových vlastností obrazu pre klasifikáciu, čo významne prispieva k jeho vysokému výkonu napriek použitiu relatívne menšieho datasetu.

Štúdia dokazuje potenciál YoNet v rozpoznávaní jógových póz, zdôrazňujúc jeho efektívnosť a presnosť pri klasifikácii póz s obmedzenými dátami [8].

Na obrázku 3.4 je zobrazená architektúra modelu a jednotlivé konvolučné jadrá. Táto architektúra sa od ostatných spomínaných líši v extrakcii priestorových a hĺbkových znakov z obrázkov oddelene. Tento prístup dvojitej extrakcie znakov umožňuje sieti efektívnejšie spracovávať vlastnosti snímku počas procesu klasifikácie.



Obr. 3.4: Architektúra modelu YoNet [8].

3.4 Sledovanie cvikov podľa použitých strojov

Táto sekcia diskutuje možnosti merania opakovaní a typov cvikov podľa využitých strojov.

3.4.1 Lavička

Lavička umožňuje vykonávať veľké množstvo cvikov, a teda je náročné rozoznávať úplne všetky. Najčastejšie sa v kombinácii s lavičkou používajú jednoručné činky. S nimi cvičenci vykonávajú opakovania cvikov, ktoré sú v práci počítané, tie sú benchpress (môže byť vo verzii “incline” pri naklonenej lavičke), shoulder press a dumbbell row.

Lavička neobsahuje žiadne prvky, ktoré by sa hýbali, preto hardvérové senzory je možno vložiť iba do samotných činiek. Veľa cvikov je z pohľadu pohybu činky veľmi podobných, preto dáta získané z takýchto senzorov by boli len ťažko klasifikovateľné, avšak kompromisom je, že počítanie opakovaní z akcelerometrických dát na činke je veľmi presné. Nevýhodou umiestňovania senzorov na činky je ich počet.

Pre určovanie opakovaní na týchto konštrukciách bude oveľa presnejšie a efektívnejšie použitie jednej zo sietí na extrakciu polohy kĺbov v čase. Jednou z možností spracovania takýchto dát je vytvorenie klasifikátora detegujúceho istú polohu vrámci cviku. V prípade opakovania polôh by následne bolo možné určiť počet opakovaní. Síce sa samotné cviky dajú dobre rozlišovať pomocou pohybu a polohy tela, táto metóda by vniesla zbytočnú komplexnosť. Riešením bude preto algoritmicky spracovávať polohu kĺbov, čo bude pravdepodobne dosahovať väčšej rýchlosti, ako použitie neurónovej siete.

3.4.2 Bez konštrukcii a strojov

Cviky bez konštrukcii a strojov môžu byť vykonávané len pomocou jednoručných a obojručných činiek. Budeme uvažovať upažovanie s jednoručnými činkami (*angl. lateral raise*), predpažovanie (*angl. front raise*) a bicepsový zdvih s jednoručnými aj obojručnou činkou (*angl. biceps curl*).

Ak je cieľom užívateľa navrhovaného systému “ochrániť” pred upínaním nejakého druhu senzoru k telu, tak aj pri týchto cvikoch je nutné rozlišovať polohy kĺbov a tela. Tieto cviky s podobnými pohybmi končatín v priestore sa však veľmi líšia relatívne k pozícii tela. Majú však spoločný periodický pohyb rúk v ypsilónovej osi.

3.4.3 Cviky so strojmi

Cviky so strojmi možno merať senzormi umiestnenými na stroje. Pri strojoch, ktoré majú zásobník s váhou je možné sledovať pohyb váhy a jej veľkosť. Pri strojoch využívajúcich kotúče s váhou sa môžu senzory umiestniť pri rameno s kotúčmi, čím bude možné sledovať váhu – tu je však nutné poznamenať, že zatiaľ čo pri zásobníku s váhou sa váha vďaka kladkovému systému pohybuje len v jednej osi, pri kotúčovej váhe sa váha pohybuje vo viacerých osiach, často paralelne s pohybom, ktorý je aplikovaný na rúčku stroja.

Niektoré stroje sú multifunkčné a umožňujú vykonávanie viacerých typov cvičení. Tieto cvičenia môžu vytvárať signály, ktoré môžu byť trochu odlišné. Pri použití správnych metód na spracovanie signálu bude tento rozdiel minimálny. V konečnom dôsledku je teda z dôvodu menších výpočtových nárokov a presnosti merania vhodné použiť hardvérové senzory.

Kapitola 4

Návrh senzorov

System je potreba kategoricky rozdelit na dve casti, skupinu senzorov a serverovu cast, ktorá bude spracovávať dáta zo senzorov v reálnom čase, resp. po dokončení každého setu. Táto kapitola diskutuje jednotlivé kompromisy, ktoré boli uvažované pri návrhu častí senzorov.

4.1 Hardvérové senzory

V ideálnom prípade by mali byť senzory čo najlacnejšie, najmenšie, najrýchlejšie, najpresnejšie a mali by mať čo najmenšiu spotrebu a stabilné napájanie. Takýchto vlastností je pri návrhu v skutočnom svete len ťažko dosiahnuť. Pri návrhu je preto nutné dostať sa k akémusi ideálnemu stavu vyváženia všetkých týchto vlastností.

4.1.1 Komunikačný protokol

Jedným z dôležitých aspektov návrhu je spôsob, akým senzory budú odovzdávať dáta hlavnému serveru. V prípade rozhodnutia medzi drôtovou a bezdrôtovou komunikáciou je výber ľahký, a sice bezdrôtová komunikácia umožňuje jednoduchšie nasadenie v reálnom svete. Množstvo bezdrôtových komunikačných protokolov ponúka výhody v podobe zníženej spotreby energie ale za cenu zníženej rýchlosti prenosu. Do úvahy prichádzajú tri populárne možnosti – Bluetooth Low Energy, Zigbee a Wi-Fi. Z týchto možností bola vybraná technológia Wi-Fi, a to najmä z dôvodu charakteru prenášaných dát a ďalších výhod, ktoré z tejto voľby vyplývajú a sú ďalej opísané.

Bluetooth Low Energy (BLE)

Bluetooth Low Energy (BLE) prináša niekoľko výhod. Jeho energetická náročnosť je závislá od architektúry a integrácie v cieľovom MCU, no vo všeobecnosti je rádovo nižšia ako napríklad pri Wi-Fi protokole. To ho robí ideálnym pre aplikácie, kde je kritická nízka spotreba energie. Aj keď má BLE nižší dosah a priepustnosť v porovnaní s klasickým Bluetooth, jeho zariadenia môžu byť zoskupené do tzv. sietí (*angl. meshes*). Tieto siete umožňujú vzájomné prepojenie zariadení, ktoré spolupracujú na preposielaní správ k bráne. V literatúre sa často uvádza realistická priepustnosť jedného BLE zariadenia ako približne 233 Kbps [33], no v prípade siete, kde zariadenia musia spolupracovať na preposielaní správ od ostatných sa táto hodnota môže znížiť.

Nevýhodou použitia takejto siete je samotná integrácia týchto zariadení pre komunikáciu so serverovou časťou. Z takéhoto riešenia by vznikla nevyhnutnosť vytvoriť zariadenie slúžiace ako brána pre všetky správy z Bluetooth siete.

Zigbee

Podobne ako v prípade BLE, aj Zigbee zariadenia sú organizované do sietí, čo zvyšuje ich možný dosah. Zigbee je špeciálne navrhnutý pre bezdrôtovú komunikáciu v oblasti senzorov, inteligentných domácností a ďalších IoT (Internet of Things) aplikácii. Jeho výhody zahŕňajú nízku spotrebu energie a schopnosť tvorby rozsiahlych sietí. Rovnako ako pri BLE, zariadenia sú pomocou týchto sietí schopné prenášať dáta medzi sebou do väčších vzdialeností, kde každé zariadenie spolupracuje na posielaní a prijímaní správ, čím vytvárajú spoľahlivé a flexibilné bezdrôtové ekosystémy. Prenosová rýchlosť Zigbee zapojenej v sieti sa pohybuje okolo 4 Kbps [6], čo je dostatočné pre väčšinu aplikácii v oblasti senzorov a inteligentných zariadení k prenášaní zmeny svojho stavu. Takáto priepustnosť je pri používaní viacerých senzorov nedostatočná, a to najmä kvôli charakteru prenášaných dát.

Wi-Fi

Wi-Fi je bezdrôtová technológia, ktorá umožňuje pripojenie zariadení k lokálnej sieti a prístup k internetu bez potreby fyzického pripojenia káblami. Je často využívaná v domácnostiach, podnikoch a verejných priestoroch. Wi-Fi ponúka vysoké prenosové rýchlosti a dostatočný dosah pre bežné použitie.

Použitie Wi-Fi však prináša nevýhodu najmä v oblasti spotreby energie, a to v porovnaní s technológiami, ako je Bluetooth Low Energy (BLE) alebo Zigbee. Wi-Fi spotrebuje oveľa viac energie, čo môže byť nevýhodné v prípade batériových zariadení, ako sú senzory alebo nositeľné technológie. Spotreba energie je závislá od konkrétneho štandardu Wi-Fi. Novšie štandardy ako Wi-Fi 6 (802.11ax) spotrebu výrazne znižujú [37].

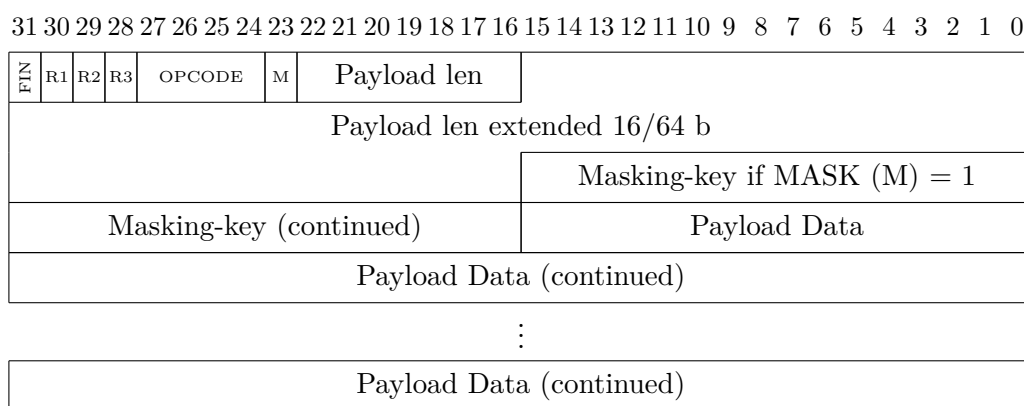
Veľkou výhodou senzorov s technológiou Wi-Fi, je možnosť pripojenia zariadení k sieti Internet, čo umožňuje posielanie informácií do vzdialených serverov. V zmysle tejto práce to znamená, že server môže byť umiestnený aj mimo fitness centra, čo môže byť pre niektoré priestory výhodou. Pre problém s dosahom je možné využiť viaceré siete či access pointov.

Prenos dát – WebSockets

Výhoda vybratia komunikačného protokolu Wi-Fi prináša možnosť použitia prenosu dát pomocou štandardu RFC6455 – známym pod označením ako WebSocket (skr. WS). Tento protokol umožňuje obojsmerný prenos dát nad TCP protokolom štvrtej vrstvy modelu OSI (Open Systems Interconnection Reference). Ten zabraňuje vzniku chyby v dátach počas prenosu implementáciou kontrolného súčtu. Dizajn tohoto protokolu je postavený na princípe, že by *“malo existovať minimálne rámcovanie dát (jediné rámcovanie dát ktoré sa tu nachádza zabezpečuje to, aby bol protokol založený na prenášaní rámcov a nie ako stream a zároveň aby pomáhal rozlišovať medzi prenášaním binárnych dát a textu”* ([23], str. 8). Takže takmer jediné čo tento protokol robí je to, že pridáva adresovanie a pomenovanie protokolu tak, aby umožňoval súbeh viacerých služieb na jednom porte (mienené najmä pre službu HTTP), vrství a rozdeľuje dáta tak, aby boli kompatibilné s požiadavkami TCP protokolu a pridáva doplnujúci *closing handshake* (ktorý je zároveň platná HTTP Upgrade žiadosť). Týmto minimalistickými úpravami sa snaží dosiahnuť to, aby prijímané dáta boli čo najbližšie k TCP vrstve [23].

Podrobný proces vytvárania a uzatvárania pripojenia je v texte tejto práce irelevantný, keďže tento protokol implementujú použité knižnice. Avšak niektoré aspekty tohoto protokolu je vhodné zdôrazniť, pretože jeho mechanizmy sú výrazne využité.

Na obrázku 4.1 je zobrazený odosielaný WS rámec. V úvode rámca bit FIN značí, že tento rámec je v správe posledný. V implementačnej časti senzorov, je opísaná vytvorená forma odosielaných aplikačných dát (z pohľadu WebSocket protokolu). Tie je nutné prenášať s operačným kódom rámca 0x2, čo značí prenos binárnych dát. V ďalších dátach patriacich danej správe je už kód rámca nastavený na 0x0, čo označuje, že dáta sú pokračovanie predošlej správy. Kvôli korektnému použitiu je dôležité dávať pozor na to, aby knižnice mali výpočtový priestor na zaslanie tzv. *keep-alive* správy. Tento mechanizmus slúži na uistenie oboch strán, že nedošlo k prerušeniu komunikácie – v prípade že ani jedna strana nechce posilať dáta, no ani jedna nepožaduje uzavrenie spojenia. Server periodicky zasiela rámce s op. kódom 0x9, na ktoré mu musí koncový bod odpovedať rámcom s op. kódom 0xA. Ukončenie spojenia je implementované ako obojstranný *handshake* a môžu ho iniciovať obe strany zaslaním ukončujúceho rámca s op. kódom rámca 0x8.



Obr. 4.1: Formát WebSocket rámca podľa RFC6455.

4.1.2 Micro Controller Unit (MCU)

Momentálny trh disponuje veľkým počtom výrobcov mikrokontrolérov. Tí každým rokom zlepšujú výkon a znižujú spotrebu ich čipov. Populárnymi výrobcami MCU pri inteligentných zariadeniach sú výrobcovia ako NXP, Espressif, Raspberry, STMicroelectronic alebo Atmel. Tie sa líšia architektúrou procesoru (ARM, Tensilica Xtensa, ...), dostupnými perifériami, počtom GPIO (General Pin Input Output) pinov či veľkosťou dostupných pamätí. Pre návrh senzorov bolo zvolených a implementovaných niekoľko MCU od výrobcu Espressif – konkrétne ESP32-S3, ESP32-C3, ESP-C3-MINI-1 a ESP32-WROOM32 a to hneď pre niekoľko dôvodov:

1. Platforma esp-idf natívne podporuje používanie RTOS (Real Time Operating System).
2. Dôležitým faktorom pri vytváraní zariadení, ktorých bude potreba niekoľko je práve nákupná cena. Nákupná cena je pri týchto čipoch relatívne nízka vzhľadom ku konkurencii s podobným výkonom.

3. ESP32-S3 podporuje akceleráciu neurónových sietí priamo na čipe. V prípade použitia neurónovej siete by bolo možné v budúcnosti niektoré dáta vyhodnocovať priamo na strane zariadenia [2].
4. Tento mikroprocesor poskytuje rozhranie pre Wi-Fi, Zigbee aj Bluetooth. Je teda možné neskôr po nasadení zmeniť komunikačný protokol jednoduchou zmenou firmvéru.
5. Dve jadrá procesora na relatívne vysokej frekvencii, kde jedno jadro je typicky vyhradené na komunikáciu. (Pri typoch S3 a WROOM32)
6. Existencia vývojovej dosky s MCU ESP32-WROOM32, ktorá má zabudovaný držiak na batériu, jej využitie urýchli prvotný vývoj a zber dát.

Veľkou výhodou je dostupnosť modulu a vývojovej dosky s týmto MCU. Modul je kompaktné balenie obsahujúce čip MCU spolu so všetkými dôležitými súčastami ako sú kryštál, rezistory, kondenzátory a anténa. Tie sú zapracované do kovového puzdra, ktoré je certifikované. To značne zjednodušuje vývojový proces výsledného zariadenia a samotnej dosky plošných spojov. Vývojová doska umožňuje vyskúšať svoj firmvér na prototypy, ktorý je zostrojený na nespájkovacom kontakte poli. Vývojová doska umožňuje aj UART-USB rozhranie umožňujúce nahráť firmvér do pamäte a debug zariadenia cez sériový port. Konkrétne zvolené moduly sú ESP32-S3-WROOM-1-N16R8 a ESP32-C3-MINI-1, ktorých dôležité vlastnosti sú zobrazené v tabuľke 4.1. Mimo iné S3 a C3 verzia obsahuje aj nízkoenergetický co-processor, ktorý sleduje periférie počas vypnutia oboch jadier. Takto vie MCU dosahovať v “Light-sleep” móde iba $240 \mu A$ [1, 2].

Tabuľka 4.1: Špecifikácia modulov.

Meno	ESP32-S3-WROOM-1-N16R8	ESP32-C3-MINI-1
MCU	ESP32-S3	ESP32-C3
Konektivita	Wi-Fi, Bluetooth, Zigbee	Wi-Fi, Bluetooth, Zigbee
CPU	2x RISC-V 32b až 240 MHz	RISC-V 32b na frekv. až 160 MHz
Pamäť flash	16 MB QSPI	4 MB QSPI
RAM	384 KB + 8 MB OSPI PSRAM	400 KB
GPIO Piny	45	15
ADC Kanály	20	5

4.1.3 Detekcia zdvihnutia a veľkosti váhy

Pre získavanie dát k určovaniu opakovaní bol v práci vybraný akcelerometer. Jednoduchšou metódou by bolo použitie nejakého typu binárneho senzora, ktorý by určoval stav váhy (zdvihnutá/nezdvihnutá). Takáto možnosť by však ochudobnila odosielané dáta a možnosť ich ďalšej analýzy, chýbala by napr. informácia o rýchlosti zdvihnutia, čase zotrvania v zdvihnutej polohe apod. Táto možnosť je zároveň zo všetkých ostatných diskutovaných možností najkompaktnejšia, keďže ostatné diskutované riešenia vyžadujú istý typ pomocnej štruktúry pripevnenej na stroj. Využitie akcelerometra však so sebou prináša aj nevýhody, a to v podobe umiestnenia zariadenia priamo na váhu stroja, ktorá sa bude hýbať. K zariadeniu je teda veľmi zložité pripojiť pevný zdroj napätia, alebo akékoľvek ďalšie senzory

na stroji, ktoré by zotrvali v statickej polohe. V nasledujúcich sekciách sú diskutované rôzne možnosti pre meranie opakovaní hardvérovou cestou. Niektoré možnosti uvažujú aj spôsoby pre meranie nastavenej váhy, ako doplňujúci a prínosný údaj pre zaznamenanie.

Rad Hallových senzorov

Táto možnosť umožňuje merať jedným radom Hallových senzorov oboje – váhu aj zdvih váhy. Hallov senzor je založený na využití tzv. Hallového javu ktorý spôsobuje, že napätie generované v Hallovom snímači je úmerné zložke magnetického poľa pozdĺž jeho normály [36].

Toto riešenie spočíva v umiestnení pomocnej konštrukcie za zdvíhanú váhu a umiestnením magnetu na koniec kolíka, ktorým sa váha nastavuje. Týmto spôsobom by bolo možné určovať jeho polohu, z ktorej by bola zjavná nastavená váha a pri lineárnom pohybe by bolo možné detegovať aj zdvih váhy binárnym spôsobom.

Tenzometer a load cells

Riešením pre snímanie váhy je aj tenzometer a silomer (*angl. load cells*). Odpor týchto zariadení sa mení v závislosti od váhy, ktorá na tieto zariadenia pôsobí [24]. Určovanie váhy by spočívalo v umiestnení takýchto senzorov pod celé závažie. V momente kedy by bolo závažie nadvihnuté, bolo by možné určiť váhu zostatku závažia a teda aj zvolenú váhu. Toto riešenie má niekoľko nevýhod, hlavným nedostatkom tohto prístupu je, že tenzometer môže byť poškodený opakovanými nárazmi váhy, čo môže viesť k jeho poškodeniu alebo potrebe častej výmeny. Týmto riešením nie je možné spoľahlivo merať opakovania, pretože pri bežnom použití stroja sa len málokedy stáva, že sa závažie vráti do bodu, kedy by bolo opäť vo východzej pozícii.

Detekcia infračerveným svetlom

Pri tejto možnosti je nutné postaviť okolo zásobníku s váhou až 2 pomocné konštrukcie, jedna bude mať senzory na infračervené svetlo a tá druhá bude toto svetlo vyžarovať. Týmto spôsobom by sme mohli sledovať pohyb váhy, kedy zdvih váhy bude odkrývať infračervené senzory, ktoré budú zachytávať svetlo z druhej strany. Podľa toho, koľko senzorov svetlo zachytí vieme určiť, kde sa takáto váha nachádza a aká časť váhy zostala v zásobníku. Nevýhodou je že svetlo sa môže náhodne odrážať a dopadať na snímače. Toto riešenie je však na integráciu do stroja príliš komplikované.

ToF senzor

Senzory ToF (Time-of-Flight) fungujú na princípe merania doby, ktorú svetelný signál potrebuje na cestu od senzora k objektu a späť. Tieto senzory vysielajú krátky impulz svetla (často v infračervenej oblasti spektra) a následne detegujú odrazené svetlo. Na základe času, ktorý svetelný impulz strávi na tejto ceste, je možné určiť vzdialenosť objektu od senzora [20].

Pri použití ToF senzora na meranie vzdialenosti kolíka na nastavovanie váhy z vrchnej strany zásobníka s váhou, je možné získať presné údaje o nastavenej veľkosti váhy. ToF senzor meria čas, ktorý svetlo potrebuje na cestu od senzora k objektu a späť, čím umožňuje určiť vzdialenosť. Tento prístup poskytuje vysokú presnosť a je odolný voči vonkajším rušivým faktorom, avšak vyžaduje presné umiestnenie senzora a kolíka. Výhoda tohoto

senzora je, že je možné ho umiestniť na rôzne miesta – môže merať výšku kolíka od zeme aj z vrchu. To z tohoto senzora robí výbornú možnosť kombinovať ho s akcelerometrom, pretože je možné ho umiestniť na pohybujúcu sa váhu. Rovnako tak je možné ho použiť samostatne v prípade merania vzdialenosti kolíka od zeme. V prípade zdvíhania váhy sa kolík od zeme vzdaluje na základe čoho je možné určovať opakovania. Vyžaduje to však vysokú presnosť, pretože sledovaný kolík je veľmi malý bod.

4.1.4 Napájanie

Keďže senzor bude umiestnený v časti stroja, ktorá sa hýbe (kvôli akcelerometru), je nutné senzor napájať bez dodatočných drôtov. Ako zdroj napätia bola vybraná lítiová batéria. Tá ponúka vysokú energetickú hustotu, dlhú životnosť, kapacitu podľa fyzickej veľkosti medzi 100 *mAh* - 3500 *mAh* a napätie medzi 3.2 *V* - 4.2 *V*. V kombinácii so správnym použitím spánkových režimov mikrokontroléra, by mala takáto batéria vydržať niekoľko dní až týždňov.

4.2 Virtuálne kamerové senzory

Tam, kde nie je možné umiestniť hardvérový senzor na pohybujúcu sa váhu, bude nutné získavať dáta analýzou obrazu. Jedna z možností je obraz spracovávať v serverovej aplikácii, toto však vytvára zaťaženie serverovej aplikácie, ktoré je možno mitigovať len zlepšovaním hardvéru. Riešením tohoto problému a decentralizáciou výpočtu je emulácia senzorov pomocou inej výpočtovej jednotky, ktorá bude z pohľadu analyzačného servera dáta odovzdávať rovnakým spôsobom, ako hardvérové senzory.

Virtuálny senzor bude prijímať obraz z kamery, tie bude následne pomocou neurónovej siete premieňať na určené body na človekovi, ktoré budú v čase meniť polohu. Tieto body môžu byť napríklad jednotlivé kĺby, alebo pozícia rúk. Počet emulovaných senzorov bude závisieť od počtu zón nastavených v obraze. Každá zóna bude vytvárať miesto, kde ľudia cvičia – lavička, miesto na státie apod. Virtuálny senzor bude rozoznávať polohu ľudí na obraze, určovať či sa nachádzajú v nejakej zo zón a následne získavať polohy ich kĺbov. Dáta z viacerých zón budú odosielané zvlášť cez nové WS spojenie, tým vznikne (z pohľadu servera) jeden senzor na jedno miesto. To poskytne dostatočne veľa dát analyzačnému serveru na určenie počtu opakovaní vykonávanom na takomto mieste.

4.2.1 Vybraný model siete

Virtuálne kamerové senzory by mali dostatočne rýchlo spracovávať živé video z kamier. V tomto prípade bude nutné brať do úvahy, že aj napriek modulárnemu riešeniu, v realite bude pravdepodobné, že jeden systém bude spracovávať viacero kamier naraz a emulovať viacero senzorov. Tu vzniká potreba vybrať model, ktorý dokáže rozpoznávať kĺby dostatočne rýchlo. Pre započítanie uhla a pozície kamery by bolo ideálne, ak by bola použitá sieť, ktorá dokáže určovať pozíciu človeka v 3D priestore. Priestor by bolo možné jednoduchými matematickými transformáciami a informáciou o pozícii kamery normalizovať a dáta mať zakaždým z pohľadu prednej strany užívateľa. Avšak, ako bolo v sekcii 3.1.2 naznačené, inferencia pri týchto modeloch je pomalá pre spracovávanie väčšieho objemu obrazu v rozumnej rýchlosti. Tu prichádza na rad použitie modelu RTMPose (Real-Time Multi-Person Pose) opísaného v sekcii 3.1.2, ktorý dokáže spolu s pomerne dostupnou grafickou kartou vykonávať inferenciu v rýchlosti až 430+ snímok za sekundu. Jeden z hlavných

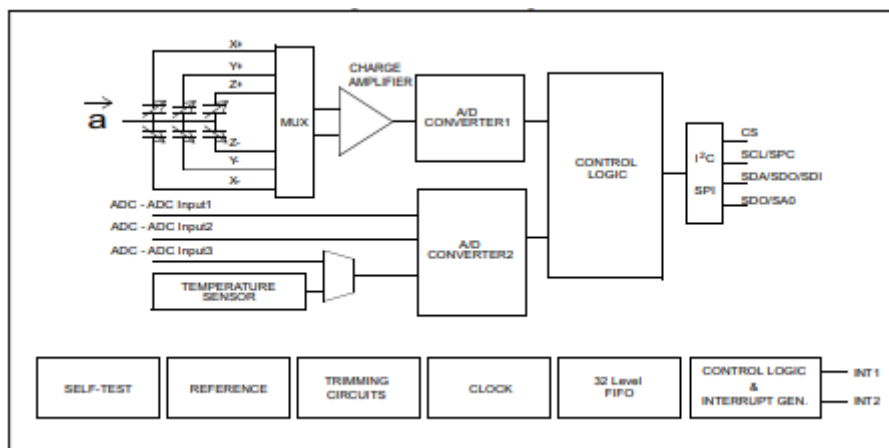
dôvodov, prečo nie je použitý model RepNet, ktorý na prvý pohľad túto problematiku rieši, je jeho rýchlosť inferencie. Tá je prakticky nezlučiteľná so spracovaním kamerového streamu v reálnom čase.

4.3 Návrh senzora C3

V prílohe B je zobrazená schéma zapojenia hardverového modulu. V tejto sekcii sú opísané použité integrované obvody a časti schémy.

4.3.1 Akcelerometrický senzor

Pre vývoj hardvérových senzorov bolo zvolené použitie akcelerometrického čipu. Pri procese vývoja boli uvažované viaceré integrované obvody od rôznych výrobcov. Medzi ne patrili IMU jednotky – to konkrétne BNO055, a BNO080 (firma Bosch), MPU-6050 (IvenSense), ďalej senzory ADXL325 (Analog Devices) a LIS3DHTR (STMicroelectronics). Tieto senzory boli na začiatku implementované a zbežne otestované pri použití na strojoch, pričom merané dáta pri ich charaktere nevykazovali zásadné rozdiely. Tie sú viditeľné najmä pri používaní IMU na sledovanie objektu v priestore, no pri jednoduchom určovaní akcelerácie váhy sú tieto rozdiely zanedbateľné. Dôležité sú však parametre ako cena senzoru, rozlíšenie a citlivosť. Väčšina takýchto senzorov je postavených na technológii MEMS (Micro-Electro-Mechanical Systems), čo znamená, že samotné čidlo senzoru je tvorené jemnou vrstvou substrátu formovanou do mechanického tvaru a často o veľkosti ľudského vlasu [10]. Tá sa pri pôsobiacej mechanickej sile hýbe a mení svoje elektrické vlastnosti, ktoré sú následne merané obvodom čipu. To spôsobuje, že prudký náraz (ako je napríklad po voľnom páde váhy), by mohol tento senzor zničiť. Pre túto prácu bol vybraný senzor LIS3DHTR kvôli jeho nízkej cene, odolnosti voči šoku až do výšky desiatich kilogramov. Na obrázku 4.2 je zobrazená bloková schéma tohoto integrovaného čipu. Umožňuje nastavenie až 2 signálov prerušenia, a to podľa nastavenej udalosti, ktorá takéto prerušenie vyvolá. Podporuje taktiež pripojenie ďalších troch analógových signálov a ich premenu na digitálnu hodnotu, čo však nie je v tejto práci potrebné, keďže samotný MCU obsahuje dostatok pinov s funkčnosťou AD prevodníka. Rovnako tak obsahuje aj FIFO front, ktorý umožňuje vyčítanie viacerých hodnôt naraz. Tieto dáta dokáže prenášať pomocou SPI a I2C rozhrania.



Obr. 4.2: Bloková schéma LIS3DH [30].

4.3.2 Senzor pre meranie váhy

Ako vylepšenie zadania je možné fyzicky k obvodu pripojiť ToF senzor s I2C rozhraním. Takýto senzor je možno pripojiť pomocou konektoru typu JST-6. V práci bol zvolený VL53L0X, ktorý vie s vysokou presnosťou určiť vzdialenosť kolíka až na 2 m. Pre zvýšenie presnosti merania je možné použiť verziu VL6180X, ktorá určuje vzdialenosť kolíka do 50 cm. Tento čip je obsiahnutý v module, ktorý je dostupný v obchodoch pod názvom TOF200C.

4.3.3 Napájací obvod

Vzhľadom k relatívne veľkému napäťovému rozsahu Li-Po batérie je vcelku zložité použitie lineárnych napäťových regulátorov (LDOs). V nižších napäťových polohách batérie by mohli spôsobiť nesprávne fungovanie, keďže MCU vo svojej špecifikácii udáva, že zdroj má byť schopný dodávať prúd minimálne o veľkosti 500 mA – napríklad kvôli nárazovému prúdu pri odosielaní dát cez anténu, pričom LDO s nízkou stratou napätia vedia iba v zriedkavých prípadoch poskytnúť takýto prúd. Jedno z častých riešení, ktoré je relatívne lacné, je použitie obvodu na zvýšenie napätia batérie na 5 V a následného zníženia pomocou LDO na $3,3\text{ V}$. Pri tomto postupe ale vzniká strata – ako pri zvyšovaní napätia, tak aj pri jeho znižovaní, čo môže výrazne znížiť výdrž zariadenia.

Ako napájací obvod bol vybraný integrovaný čip od Texas Instruments TPS63051 typu Buck-Boost. V Buck režime tento typ obvodu operuje na fixnej frekvencii nastaviteľnej cievkou – typicky 2.5 MHz – a zmenou modulácie šírky pulzov reguluje výstupné napätie. Naopak, ak je vstupné napätie nižšie, automaticky sa prepne do Boost režimu a napätie zvýši na nominálnu hodnotu výstupu. Pri Boost móde dokáže tento obvod dodávať až 500 mA a pri Buck móde až 1 A , čo je pre MCU a periférie dostačujúce. Kvôli zvolenému typu napájacieho obvodu je k perifériám pripojené väčšie množstvo kondenzátorov na od-filtrovanie vysokých frekvencií a stabilizáciu obvodu. Efektivita takéhoto obvodu je podľa špecifikácie 90% pri Boost móde a až 95% pri Buck móde.

Ako nabíjací obvod batérie bol zvolený integrovaný obvod TP4056. Rýchlosť nabíjania, resp. veľkosť nabíjacieho prúdu je programovateľná veľkosťou rezistoru. Tá bola v obvode nastavená na 500 mA , kvôli obmedzeniu tepla, ktoré vzniká pri nabíjaní. Stav nabitia signalizujú dve LED (Light-Emitting Diode) diódy – pri nabíjaní červená, pri plnom nabití zelená.

4.3.4 Meranie stavu batérie

Obvod implementuje možnosť sledovať stav batérie, a to pomocou dvojice komplementárnych MOSFET (Metal Oxide Semiconductor Field Effect Transistor) tranzistorov v kombinácii s napäťovým deličom, ktorý merané napätie znižuje na rozsah hodnôt merateľných AD prevodníkom MCU. Pomocou tohoto obvodu môže takto spomínané rozpätie batérie $4,2 - 3,2\text{ V}$ lineárne mapovať na rozsah $1,6 - 2,1\text{ V}$, a keďže zostávajúca kapacita Li-Po batérie je viazaná na napätie batérie, senzor si vie vypočítať približný percentuálny zostatok v batérii. Výhodou tohoto obvodu oproti použitiu iba napäťového deliča je ten, že ak na tranzistor nie je privedené aktivačné napätie z MCU, neodoberá žiaden prúd – presnejšie iba zanedbateľný únikový prúd podľa charakteristiky tranzistora medzi emitorom a kolektorom. Samotné privedené napätie batérie v kombinácii s tranzistorom typu P potom drží tento obvod v zatvorenom stave.

4.3.5 Ostatné periférie

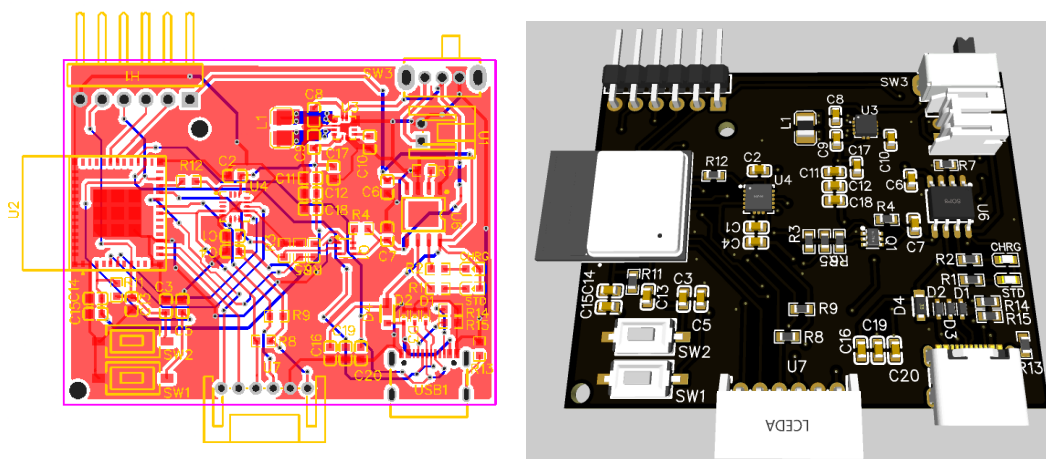
Obvod taktiež obsahuje USB-C konektor slúžiaci na nabíjanie. Vďaka pripojeniu dátových liniek k JTAGu MCU je možné cez tento konektor aj nahráť firmvér. Dátové linky sú chránené proti elektro-statickému výboju ochrannými ESD (Electro-static discharge) diódami. Dve tlačidlá (v schéme v prílohe B označené SW1 a SW2) sú určené na manuálny reštart MCU a manuálne prepnutie MCU do boot módu. SW3 je posúvací vypínač, ktorý odpojí napätie batérie od obvodu, čím vypína obvod. S MCU je dodatočne možná komunikácia a nahratie firmvéru pomocou externého programovacieho obvodu pripojeného ku konektoru H1.

4.3.6 Doska plošných spojov a obal

Na základe opísaných súčastí a schémy boli navrhnuté a posunuté do výroby dve dosky plošných spojov.

Dvojvrstvá doska

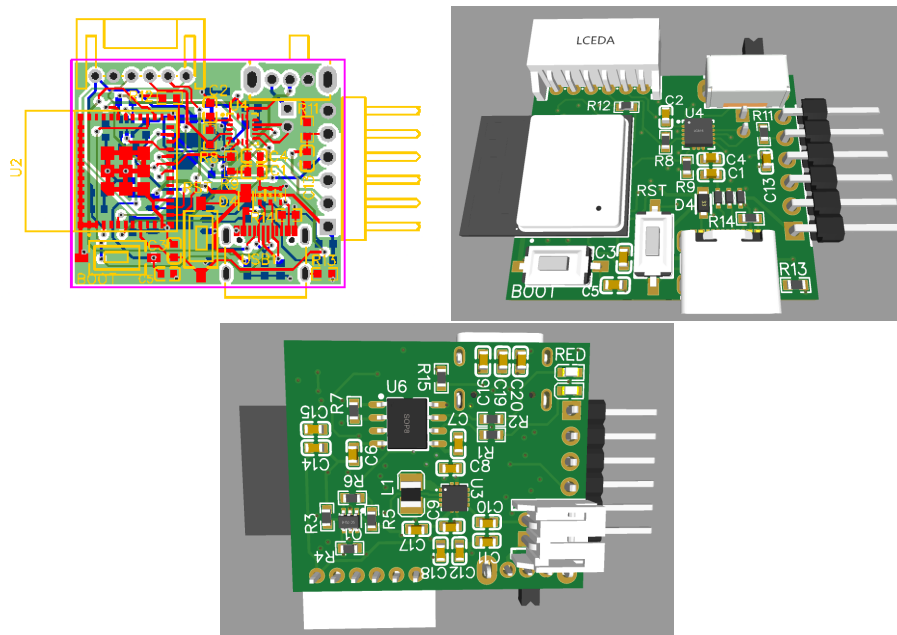
Prvá z dosiek je väčšia a súčiastky sú uložené len na jednej strane. Je tak vhodná pre jednoduchšie osadenie a overenie funkčnosti. Táto doska má rozmery 5x4 cm.



Obr. 4.3: DPS a 3D model dvojvrstvej dosky.

Štvorvrstvá doska

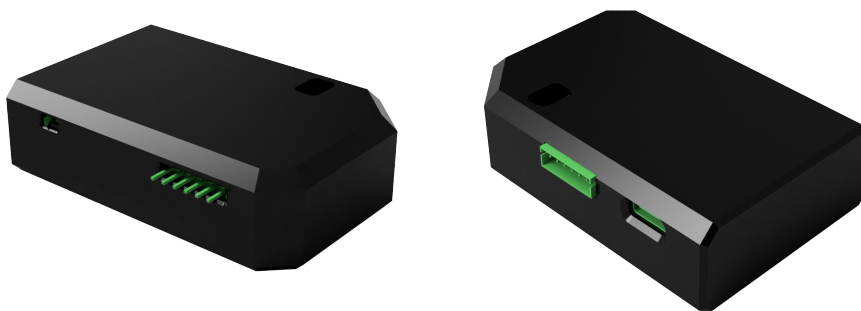
Pre kompaktnejšiu veľkosť dosky bolo nutné pridať ďalšie dve signálové vrstvy. Na obrázku 4.4 je zobrazený 3D model a DPS, ktorej veľkosti sú 2.5x3 cm. Je nutné podotknúť že časti ktoré presahujú hranicu dosky je možné nahradiť pevnými spojmi – okrem USB-C portu a samotnej antény MCU modulu. Toho umiestnenie musí na doske rešpektovať definovanú špecifikáciu.



Obr. 4.4: DPS a 3D model štvorvrstvovej dosky.

Obal pre dvojvrstvovú dosku

Na obrázku 4.5 je zobrazené navrhnuté puzdro pre dvojvrstvovú verziu DPS. Táto veľkosť sa bez problémov zmestí na merané stroje. Puzdro je zložené z dvoch častí, spodná je určená ako priestor pre batériu rozmerov 50x40x6 mm a druhá pre uchytenie dosky. Doska sa upevňuje pomocou M2x5 skrutiek, uzavretie celého puzdra je realizované skrutkou M3x15.



Obr. 4.5: Puzdro pre dvojvrstvovú verziu DPS.

K puzdru je taktiež navrhnutý držiak pre modul (obr. 4.6), ktorý meria vzdialenosť kolíka. Je možné ho k tomuto držiaku uchytiť dvomi M2x5 skrutkami. Zadná strana sa tak môže prilepiť na stroj.



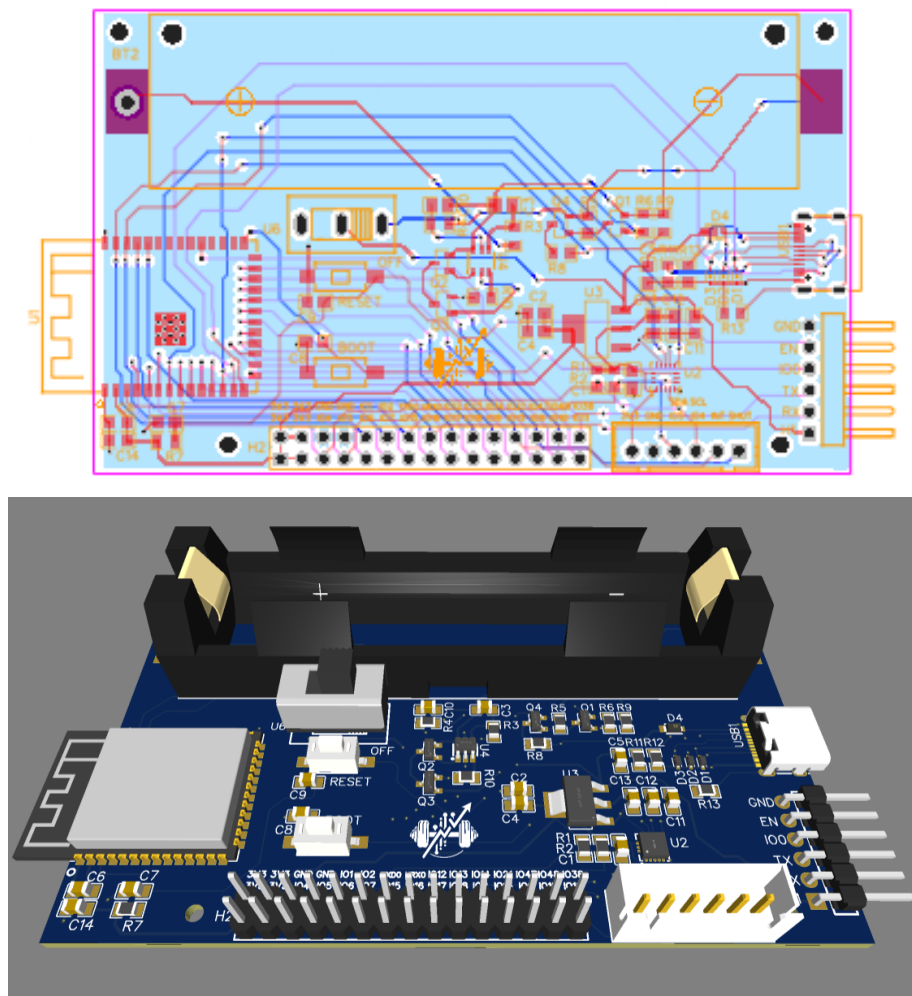
Obr. 4.6: Držiak pre ToF modul.

4.4 Návrh vývojového setu S3

V prílohe C je zobrazené zapojenie MCU verzie S3. V tom je možno vidieť niekoľko rozdielov. Toto zapojenie je koncipované ako vývojová doska so silnejším výkonom, preto je na nej k dispozícii až 30 voľných pinov pripojených ku GPIO (General Pin Input Output) pinom MCU.

4.4.1 Rozdiely oproti verzii C3

V prvom rade je verzia osadená dvomi držiakmi na batérie typu 18650, tie v sérii poskytujú napätie vhodné pre použitie LDO. Pre možnosť rýchlej výmeny týchto bateriek nie je použitý nabíjací obvod a namiesto neho je použitý obvod HY2120, ktorý ochráni batérie pred príliš veľkým vybitím, ktoré by mohlo batériám uškodiť. Návrh samotnej DPS a 3D model je možné vidieť na obrázku 4.7 (*pozn. držiak druhej batérie je na druhej strane dosky*).



Obr. 4.7: DPS a 3D model verzie S3.

Kapitola 5

Implementácia softvéru

5.1 Firmvér hardvérového senzora

Firmvér senzora bol implementovaný pomocou platformy esp-idf s využitím FreeRTOS (Real-Time Operating System). V platforme je možné prepnúť implementačné prostredie, ktoré pomocou makra mení typ cieľového MCU spolu s jeho zapojením.

5.1.1 FreeRTOS

FreeRTOS je kernel operačného systému pracujúceho v reálnom čase. Jeho použitie je zamerané najmä na malé vstavané systémy. Keďže väčšina jeho kódu je napísaná v jazyku C, jeho výhodou je veľká prenositeľnosť [9]. Oproti ostatným OS, kernel systému ponúka menej funkcií. Medzi hlavné tri funkcie patrí:

- Plánovač – pre plánovanie úloh (v preemptívnej alebo kooperatívnej verzii) .
- Obsluha frontu pre správy medzi úlohami.
- Semaforey pre synchronizáciu.

5.1.2 Použité knižnice

Na implementáciu boli použité štandardné knižnice jazyka C, knižnice pre testované senzory lis3dh [30], bmi160 [11] a knižnice pre MPU6050, FreeRTOS a periférie MCU poskytované esp-idf platformou.

5.1.3 Hlavná rutina firmvéru

Vstupná funkcia programu `app_main()` je definovaná v súbore `main.c`. Táto rutina je volaná pri štarte MCU, ako aj pri prebudení z hlbokého spánku, do ktorého sa senzor prepína po dokončení merania. V nej inicializuje v tomto poradí: front správ (zdieľaný medzi všetkými úlohami), pamäť flash, GPIO piny, I2C rozhranie, úlohu zabezpečujúcu ukladanie dát z akcelerometra, Wi-Fi pripojenie, úlohu zodpovednú za odosielanie dát a úlohu ktorá meria vzdialenosť váhy. Hlavná rutina potom inicializuje perifériu AD prevodníka a získa hodnotu napätia batérie, ktorú lineárne mapuje na percentá a odošle serveru.

Po týchto krokoch zotrúva v slučke a každú sekundu a zisťuje, kedy bolo zachytené posledné prerušenie z akcelerometrického senzora. Ak je to dlhšia doba ako bola definovaná v

súbore `defines.h`, zašle správu úlohe odosielajúcej dáta, ktorá uzavrie Websocket spojenie, nastaví RTC perifériu pre detekciu prerušenia a uspí MCU do hlbokého spánku.

5.1.4 Meranie dát

Úloha na meranie dát `lis3dh_task(void *arg)`, je definovaná v súbore `acceleration.c`. Vďaka jej skorej inicializácii dokáže merať začatie cviku už po približne $\sim 70ms$ od zachytenia prerušenia pohybu váhy, resp. po zapnutí MCU. Inicializuje všetky potrebné nastavenia použitého senzora (na doske C3 to je `lis3dh`) vrátane nastavenia odosielaných prerušení zo senzora, citlivosť, rýchlosť vzorkovania na senzore a jeho presnosť. Nasleduje nekonečná slučka posúvania dát úlohe pomocou štruktúry `send_item` a frontu, ktorá správy odosiela serveru. Na konci funkcie dynamicky počíta čas, ktorý jej výpočet zaberie a potom sa uspí tak, aby bola výsledná vzorkovacia frekvencia podľa makra `FREQUENCY_FS`. Na merané dáta je aplikovaná implementácia Kalmanového filtru [32], ktorá vyhladí malé oscilácie v dátach.

Nastavenie senzora

LIS3DH je vybavený dvomi nezávislými generátormi prerušenia. Každý z nich je možné nastaviť na udalosť voľného pádu a presiahnutia prahu, pričom pri každom z nich vygeneruje prerušenie v nastavenom trvaní – trvanie sa udáva pomocou vzťahu N/ODR , kde ODR predstavuje nastavenú vzorkovaciu frekvenciu a N je nastavovaná hodnota trvania prerušenia. To znamená, že trvanie prerušenia je priamo úmerné nastavenej rýchlosti vzorkovania. Nastavuje sa aj informácia, ktoré z osí môžu vyvolať prerušenie. Najdôležitejšou hodnotou je v tomto nastavení prahová hodnota. Tá sa nastavuje v registri `INT1_THS` a nastavuje hodnotu, ktorá závisí na nastavenej citlivosti. V prípade zvolenej najnižšej citlivosti, najnižší bit predstavuje 16 mg presahu. Tu bola nastavená konštanta 69, ktorá vyšla z testovania ako najlepšia možnosť vzhľadom k miere citlivosti na zdvih váhy – senzor je tak dostatočne citlivý na to, aby zachytil jemný zdvih váhy, keď sa cvičenec dostáva do správnej pozície a zároveň dosť veľká na to, aby nevyvolal prerušenie pri jemnom zachvení so strojom. V prepočte to teda je $69 \cdot 16 = 1104 mg$, pričom záleží hlavne na samotnom kuse senzoru, pretože v prípade práve toho použitého, gravitácia na senzore spôsobuje východziu akceleráciu 1,18 g

Citlivosť je nastavená na najnižší možný interval a to $\pm 2 g$, pričom rozlíšenie dát je 12 bitov a vzorkovacia frekvencia je 100 Hz. Pri testovaní týchto senzorov bolo zhodnotené, že v ďalšej verzii by bolo ideálne použiť senzory vyššou citlivosťou – napríklad 1,5 g.

5.1.5 Meranie váhy a odosielanie dát

MCU získava dáta o vzdialenosti kolíka váhy od pripojeného senzora ako 16 bitové číslo v súbore `dist.c`. Urobí niekoľko (predom definovaný počet) meraní, ktoré následne odošle. Na strane serveru sú tieto dáta spracované tak, aby vylúčili možné chyby pri meraní a odchýlené hodnoty. Následne je táto vzdialenosť podľa kalibračných údajov stroja premenená na váhu.

Definícia odosielania dát ako aj inicializácia Wi-Fi spojenia sa nachádza v súbore `wifi.c`. Táto úloha čaká v nekonečnej slučke, pričom používa neblokujúce volanie FreeRTOS API (Application programming interface) `xQueueReceive`. Prijaté správy následne odošle cez Websocket protokol serveru. V prípade prijatia správy s operačným kódom `OPCODE_WS_DISCONNECT` ukončí spojenie so serverom a pošle tzv. *closing frame*, ktorý je súčasťou Websocket protokolu ako ukončenie komunikácie. Táto funkcia taktiež udržiava spojenie so serverom a rieši prípadné výpadky opätovným nadväzovaním spojenia.

Samotná forma odosielaných dát je opísaná neskôr v sekcii 5.3.1.

5.2 Softvér virtuálneho senzora

Softvér virtuálneho senzora je vytvorený ako Python skript využívajúci knižnice ako OpenCV, mmdeploy či numpy. Na inferenciu dát z obrazu využíva platformu ONNX Runtime.

5.2.1 Prerekvizity

Pre správne fungovanie tohoto softvéru je nutné nainštalovať ONNX Runtime a potrebné python knižice zo súboru `prerequisites.txt`. Ďalej je nutné mať nainštalované prostredie mmdeploy alebo jeho inferenčné SDK (software development kit) pre jazyk Python.

Konfiguračný súbor

Skriptu je nutné poskytnúť konfiguračný súbor obsahujúci adresu kamerového streamu – preferovaný je protokol RTSP (podporuje ale aj protokoly ako HTTP a RTMP). Ďalej je nutné poskytnúť zoznam zón a ich umiestnenie v obraze pomocou súradníc ľavého horného a pravého spodného rohu tejto zóny. Súčasťou je súbor `config-template.yaml`, ktorý slúži ako ukážka takéhoto konfiguračného súboru.

5.2.2 ONNX Runtime

ONNX Runtime [26] je cross-platformné riešenie pre zjednodušuje nasadenie modelu hlbokého učenia. Podporuje modely z platformami ako sú TorchPy a TensorFlow. Je kompatibilný s veľkým množstvom hardvéru a jednoduchým prepnutím parametra dokáže modely inferovať na procesore alebo grafickej karte. Pred samotným nasadením modelu pomocou tejto platformy je nutné model najskôr exportovať do `.onnx` formátu.

5.2.3 Implementácia

Skript využíva dva paralelne bežiacie procesy, jeden na načítavanie a inferenciu dát z obrazu a druhý na odosielanie inferovaných dát. Obidva sú konfigurovatelné nastaveniami konfiguračného súboru skriptu `config.yaml`. Procesy sú spojené jednosmernou komunikáciou zrefazeneho spracovania (*angl. pipelining*).

Proces spracovania obrazu

Proces spracovania obrazu využíva konfiguračným súborom zvolený model na detekciu ľudí v obraze z kamery. Predvoleným modelom je `rtmdet-nano`. Tento model vráti súradnice obdĺžnikov, v ktorom deteguje ľudí (*angl. bounding boxes*). Následne na základe predom definovaných zón vyberú len tie obdĺžniky s ľuďmi, ktoré sa nachádzajú vo vnútri týchto zón. Vybrané obdĺžniky opisujúce detegovaného človeka sú spolu s obrazom z kamery používané ako vstup modelu definovaného v konfiguračnom súbore, ktorý ďalej inferuje ľudskú pózu. Ako predvolený model je použitý `rtmpose-t`.

Proces spracovania obrazu sa za cieľom ušetrenia výpočtového výkonu dá spomaliť. Na to slúži parameter `max_fps` v konfiguračnom súbore, ktorým sa definuje maximálny počet spracovaných obrázkov za sekundu. Rovnako tak je možné ušetriť výpočtový výkon na detekcii ľudí v obraze a to pomocou čísla `x` v parametre `people_det_optimize`, ktorý spôsobí

vykonanie inferencie ľudí len každý x -tý snímok (inferencia póz sa vykonáva na predošlom obdĺžniku). Myšlienka za týmto riešením je taká, že detegované štvorce sa nezvyknú tak rýchlo meniť, zatiaľ čo pohyb kĺbov v nich áno.

Proces odosielania dát

Spracovanie obrazu tvorí funkciu polohy kĺbov $x(t)$ v čase t . Tieto dáta je nutné odoslať na analýzu rovnako ako dáta z hardvérových senzorov. Pre zachovanie uniformného správania senzorov sú sety zachytávané priamo procesom spracovania obrazu. Je preto nutné, pre každú zónu vytvoriť vlastné spojenie cez WebSocket. V rámci procesu sa o tieto spojenia stará funkcia `manage_connections()`. Pomocou Python knižnice `threading` je táto funkcia multitaskovaná spolu s funkciou, ktorá prijíma dáta od procesu spracovania obrazu pomocou zrefazovaného spracovania.

Proces vytvára hlavičku paketu na obr. 5.1, a následne používa metódy knižnice `numpy` na serializáciu poľa dát. Takýto paket následne zašle so správnym kódom správy na server, ktorý sa stará o ďalšiu analýzu.

5.3 Server pre spracovanie dát

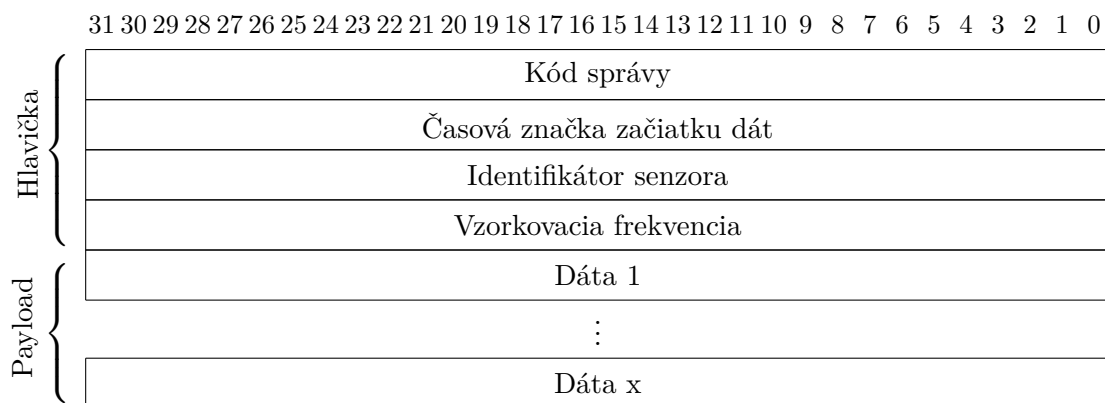
Všetky senzory zasielajú dáta, ktoré reprezentujú vykonávanie cviku. Tieto dáta je nutné spracovať a výsledky uložiť. Na tento úkon slúži implementácia analyzačného servera. Server bol napísaný v jazyku Python, ktorý poskytuje veľa knižníc určených na spracovanie signálu a dát. Server implementuje niekoľko užitočných utilít pre jednoduché nasadenie, vrátane mechanizmu mDNS [13], ktorý zaručuje jednoduché nájdenie serveru senzormi v lokálnej sieti podľa lokálneho doménového mena.

5.3.1 Komunikácia

Server otvára port 5000 pre TCP komunikáciu pomocou protokolu WebSocket. Táto komunikácia je charakteristická vlastnou formou prijímaných dát. Na obrázku 5.1 je možné vidieť formu prijímaných dát. Výhoda prenosu dát pomocou vlastného binárneho formátu spočíva v zmenšení objemu prenášaných dát. Pre príklad – ak by sa dáta prenášali ako serializovaný JSON objekt, bolo by nutné všetky čísla typu `float` prenášať ako reťazce. To by v kombinácii s nutnosťou použitia oddeľovacích a ďalších charakteristických znakov pre tento objekt spôsobilo nárast prenášaného objemu dát.

V hlavičke komunikácie každej správy je uvedený **kód správy**, ktorý určuje obsah a spôsob spracovania priložených dát. V **časovej značke** je uložená hodnota časovej značky začiatku vytvárania dát podľa štandardu ISO 8601 [25]. **Identifikátor senzora** nesie jedinečné číslo pridelené senzoru – a teda je z neho možné odvodiť aj konkrétny stroj, na ktorom je umiestnený. Posledným atribútom hlavičky je **vzorkovacia frekvencia** meraných dát. Tá je pri niektorých typoch správ nepotrebná. Všetky atribúty sú vo veľkosti 4 bajtov, a to najmä kvôli jednoduchšiemu spracovaniu.

Samotná správnosť prenosu správ je zabezpečená na vrstve sieťového rozhrania v rámci protokolu TCP.



Obr. 5.1: Formát správy prijatej cez WebSocket.

5.3.2 Typ prijímaných správ

Ako bolo v predošlej sekcii 5.3.1 naznačené, server prijíma rôzne typy správ. Podľa hodnoty typu správy server určuje, ako bude spracovávať danú správu. Hodnoty pre hardverové senzory sú enumerované od 0, zatiaľ čo pre virtuálne senzory sú enumerované od 100.

Server prijíma tieto správy:

- `OPCODE_SENDFLOAT_XYZ` sú akcelerometrické dáta zo všetkých osí vyjadrené pomocou typu `float`, jedna vzorka tak obsahuje tri 4 bajtové čísla z osí `x`, `y` a `z`.
- `OPCODE_SENDFLOAT_X`, `OPCODE_SENDFLOAT_Y`, `OPCODE_SENDFLOAT_Z` obdobne obsahujú rovnaké dáta, avšak iba v jednej osi na vzorku.
- Pri kóde `OPCODE_SENDDIST` dáta obsahujú niekoľko meraní vzdialenosti kolíka na nastavovanie váhy v milimetroch ako 32 bitové čísla typu `int`.
- `OPCODE_STATUS` obsahuje pomocné informácie o senzore ako napríklad zostávajúcu kapacitu batérie.
- `OPCODE_ERR` obsahuje reťazec, ktorý predstavuje chybu senzora. Tú server zapisuje k chybovým správam do databázy spolu s identifikátorom senzora a časom výskytu.
- `OPCODE_JOINTS_XYP` sú dáta z virtuálnych sensorov obsahujúce pozíciu kĺbov zachytených v prednastavených zónach.

5.3.3 Databáza

Zadanie definuje potrebu vytvárania štatistík. Pre tento dôvod je v konfigurácii servera možno zadať pripojenie k MariaDB databáze. K interakcii s ňou používa server triedu `MariaDBManager`, ktorá obsahuje štruktúru databázy. Po úvodnej inicializácii dokáže trieda zistiť, či je na databázovom serveri vytvorená správna štruktúra tabuliek. V prípade že tomu tak nie je, vytvorí ju. Na obrázku 5.2 je zobrazená štruktúra vytvorenej databázy.

excercise_log	
PK	id
	sensor_id
	timestamp
	reps
	weight
	det_quality
	rep_quality
	average_rep_t

battery_status	
PK	id
	sensor_id
	timestamp
	battery

error_log	
PK	id
	sensor_id
	error_msg
	timestamp

Obr. 5.2: Štruktúra tabuliek v databáze.

5.3.4 Postup spracovania dát

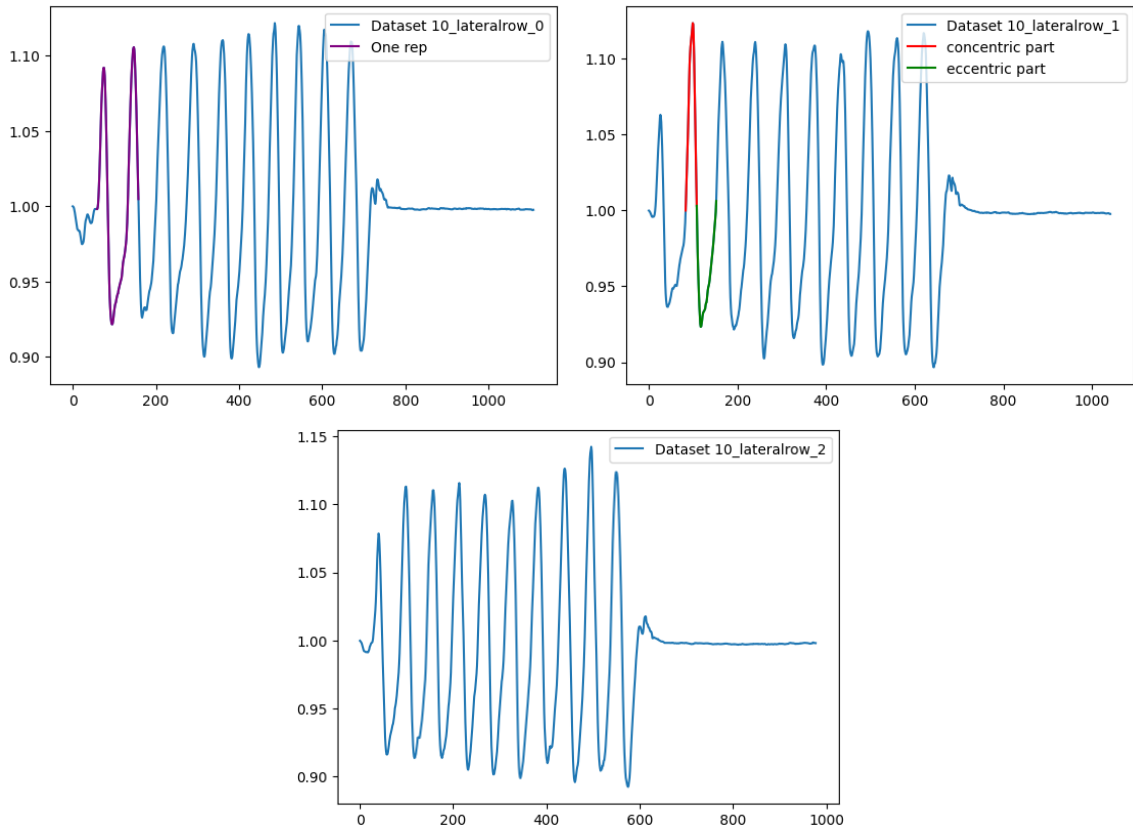
V momente nového spojenia so serverom, je takáto komunikácia asynchrónne spracovávaná serverom pomocou knižnice `asyncio`. Server vytvorí objekt `SensorComm` reprezentujúci komunikáciu a spracuje kód správy. Následne po trvanie spojenia vkladá prijaté dáta zo senzora pomocou metód objektu. Po odpojení senzora zozbierané dáta podrobí analýze a získané hodnoty vloží do databázy. V prípade prijatých chybových hlásení, tieto zapíše do tabuľky chýb.

5.4 Určovanie opakovaní z akcelerometrických dát

Za účelom vybrania metódy pre určovanie opakovaní bolo pomocou senzorov zozbieraných niekoľko stoviek opakovaní, súbory boli následne označené s reálnym počtom opakovaní, ktoré boli počas merania vykonané. Tento základný set bol použitý pre vývoj algoritmu, ktorý počíta opakovania.

5.4.1 Charakter dát

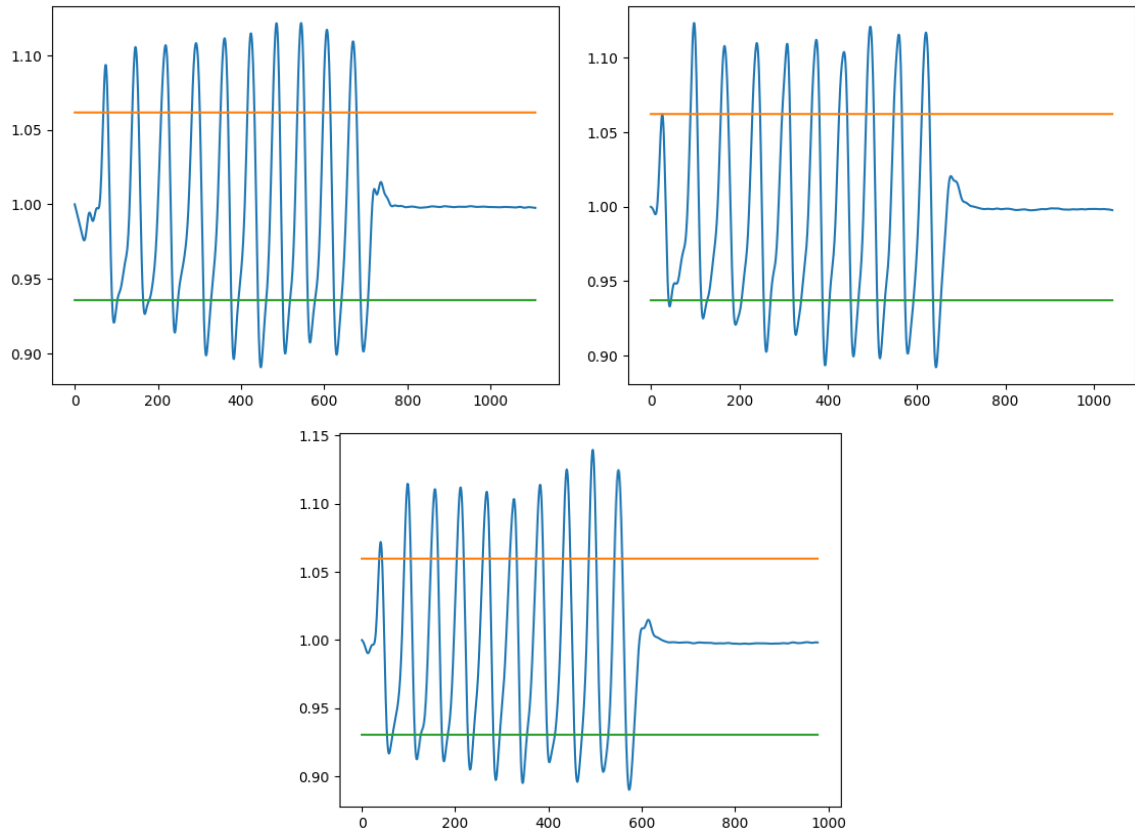
Na obrázku 5.3 sú zobrazené tri sety po 10 opakovaní zachytených z hardvérového senzora. Tento konkrétny cvik je vybraný na uvedenie do tejto problematiky najmä z dôvodu, že výstup z tohoto stroja je kvôli charakteru vyžadovaného pohybu “najčistejší”. Inými slovami, takýmto spôsobom vyzerajú najdokonalejšie dáta opakovaní, ktoré je možno zo stroja cvičením dostať.



Obr. 5.3: Tri sety desiatich opakovaní cviku Lateral Row.

Z charakteru dát je zrejmé, že bude vhodné použiť algoritmické riešenie. V prípade tréovania neurónovej siete by bolo nutné zozbierať veľké kvantum dát pre tréovanie, ktoré by ale ani tak nemuselo zaručovať správne výstupy. Je možné si všimnúť malé vlnenie na začiatku a konci dát. Tieto vlnenia predstavujú prvotné vzatie váhy a nastavenie sa do pozície pre cvičenie a následné odloženie váhy. Jedno opakovanie predstavuje vlnu sínusového tvaru (vyznačená fialovou farbou v prvom grafe) – v prvej polovici (teda v koncentrickej časti cviku) ide váha smerom hore a naberá akceleráciu kde ku koncu spomalí na pôvodnú gravitačnú hodnotu. Druhá časť polvlny predstavuje pohyb váhy smerom dole (excentrická časť cviku). Na druhom grafe obrázku sú tieto polvlny oddelené červenou a zelenou farbou.

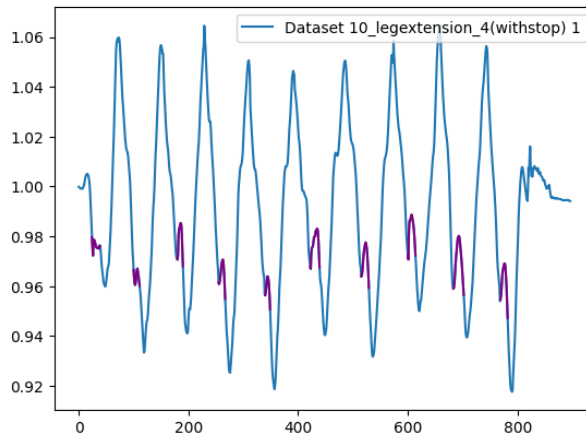
Pre takto dokonalé dáta je možné uvažovať o použití naivného algoritmu presiahnutia prahovej hodnoty. Táto prahová hodnota môže byť nastavená napríklad ako hodnota x -tého percentilu signálu. Aplikácia takéhoto algoritmu s percentilom 85 je zobrazená na obrázku 5.4. V druhom prípade prah tesne pretína krivku signálu, takže vrchol je započítaný. Hodnota po vydelení počtu skrížení signálu prahovou hodnotou vyjadruje počet vykonaných opakovaní. V prípade nesúlady počtu horného a spodného prahu by bolo možné prikloniť sa výhradne k tomu, ktorý je štatisticky presnejší pri danom cviku – pozorovaním bolo zistené, že pri niektorých strojoch majú ľudia tendenciu akcelerovať rýchlejšie počas koncentrickej časti cviku a pri niektorých zas počas excentrickej.



Obr. 5.4: Aplikovanie prahového algoritmu.

5.4.2 Cviky s izometrickou časťou

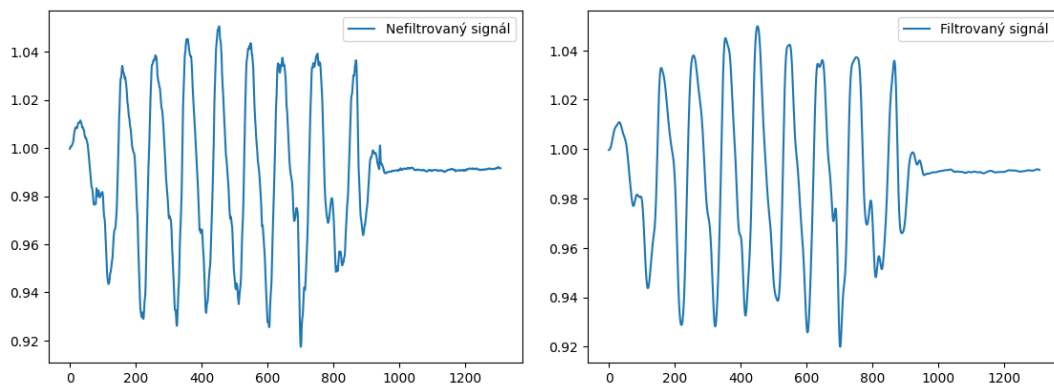
Pre zlepšenie efektivity sa v cvičení často zaraďuje izometrická časť cviku medzi koncentrickú a excentrickú časť. Inými slovami – po zdvihnutí váhy sa v tejto polohe zotrváva zhruba sekundu. V dátach sa to potom prejaví ako náhly nárast akcelerácie v strede opakovania. Tieto časti je možné vidieť na obrázku 5.5 vyznačené fialovou farbou. Tento údaj pomáha určovať kvalitu opakovaní a tento jav je žiadaný.



Obr. 5.5: Cvik so zadržiavaním váhy.

5.4.3 Algoritmus hľadania vrcholov

Knižnice `scipy` a `numpy` poskytujú všetky potrebné nástroje pre analýzu spracovanie prijímaného signálu. V prvom rade je nutné dáta filtrovať, na to je použitý Butterworthov filter. Pri dostatočne veľkých parametroch vyhladí dáta a odstráni miesta s nulovou deriváciou v prechodoch. To je dôležité kvôli aplikovaniu ďalšieho postupu. Na obrázku 5.6 je možno vidieť aplikovanie tohoto filtru na dáta ôsmich opakovaní cviku zhybu. Ten býva často “nepekčný” kvôli tomu, že v tomto prípade váha na stroji nie je použitá ako odpor, ale naopak, ako odľahčenie tela vo vykonávaní cviku.

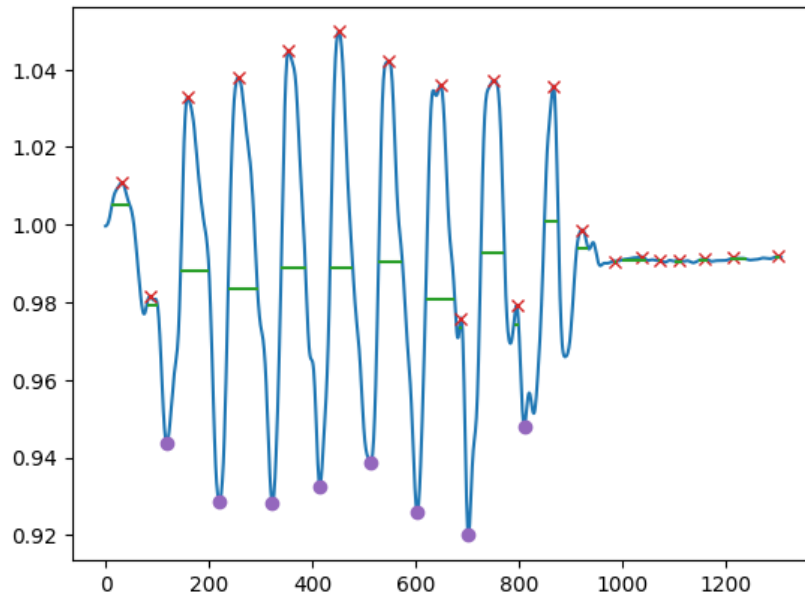


Obr. 5.6: Filtrovanie signálu.

V ďalšom kroku sú pomocou funkcie `scipy.find_peaks` nájdené všetky vrcholy v dátach. Tie sa vždy hľadajú v okolí približne FPS vzoriek. Funkcia vylúči ostatné menšie vrcholy ktoré nedosahujú výšku aspoň `baseline + 0,02`, kde `baseline` predstavuje konštantu východzej hodnoty senzoru – v prípade dát v Z osi, na tie pôsobí gravitácia a hodnota sa pohybuje okolo 1. V prípade polohy na šikmej rovine, 1 bude súčet všetkých `baseline` hodnôt. Pre maximálny zisk údajov z dát je funkcia negovaná a vrcholy sú znova nájdené rovnakým spôsobom.

Na obrázku 5.7 sú nájdené vrcholy označené krížikmi a inverzné vrcholy bodkami. Získané vrcholy sú vložené na vstup algoritmu určujúcu ich šírku v polovici ich celkovej veľkosti,

tie sú na grafe zobrazené zelenou čiarou. Z tejto šírky a informácii o rýchlosti vzorkovania vieme približnú dobu vykonávania fáze cviku v relatívnej polovici jeho trvania.



Obr. 5.7: Nájdené vrcholy a inverzné vrcholy.

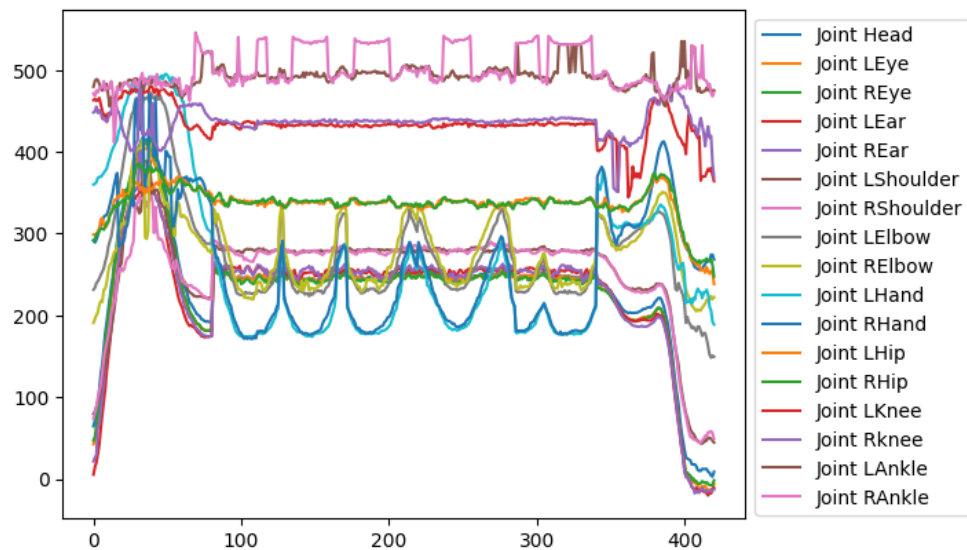
Finálne, algoritmus je založený na myšlienke, že v prípade zdvihnutia váhy musí existovať vrchol s najvyššou akceleráciou nasledovaný prepadom, prípadne naopak. Tieto výšky sa následne zaznačia a je možné aplikovať štatistické metódy na odstránenie odchylených hodnôt. Z týchto informácií sa získa počet opakovaní.

5.5 Určovanie opakovaní z vizuálnych dát

Dáta z vizuálnych senzorov sú privedené na vstup reťazca úprav, zloženého zo základných metódik spracovania signálu.

5.5.1 Charakter dát

Tak ako bolo vysvetlené v sekcii 5.2.3, virtuálny senzor odosiela pakety so zachyteným pohybom, tie sú následne v objekte spravujúcom komunikáciu nahromadené pre celý set. Taktak vzniknú dáta potrebné na analýzu s dimenziami $(x, 17, 3)$, kde x je počet vzoriek, 17 predstavuje počet kľbov a 3 sú súradnice bodov spolu s pravdepodobnosťou detekcie. Nespracované získané dáta kľbov sú zobrazené na grafe 5.8.

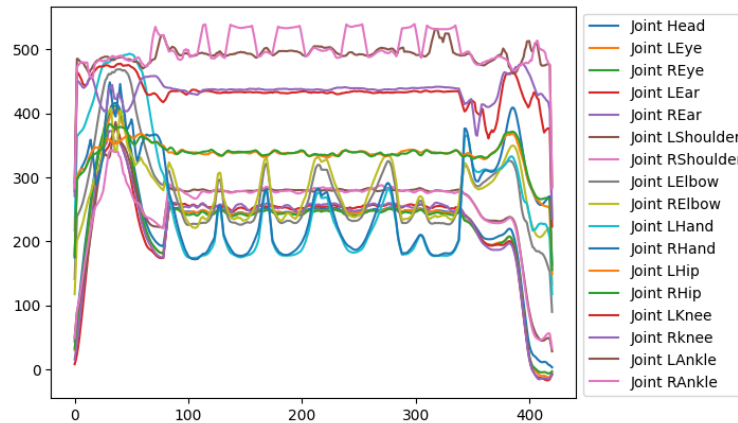


Obr. 5.8: Nespracované dáta.

Graf znázorňuje pohyb kĺbov v ose Y. Z takéhoto grafu by mal algoritmus rozpoznať 6 opakovaní. Je možné si všimnúť niektoré problémy, ktoré bude nutné adresovať. Prvým hlavným problémom je rozpoznanie časti, kedy cvičenec vykonáva opakovania. Rôzne cviky spôsobujú pohyb rôznych kĺbov. Ďalšími problémami, ktoré je na prvý pohľad vidno je šum týchto dát a nepravdivé oscilácie. Nepravdivými osciláciami sú myslené pohyby kĺbov, ktoré nenastali. V grafe táto situácia nastáva primárne v prípade pravého členka. Ružová čiara na vrchnej časti grafu mení svoju hodnotu (pripomínajúc digitálny signál), ktorá môže evokovať opakovania. Pri preskúmaní týchto dát je však zrejماً príčina, a síce neurónová sieť určuje polohu kĺbu len s nízkou pravdepodobnosťou – resp. nemusí sa vôbec nachádzať na obraze. Ďalšie sekcie sa teda zaoberajú riešením týchto problémov.

5.5.2 Filtrovanie dát

Z testovaných filtrovacích metód ako bol Butterworthov, Savitzky-Golay a Gausov filter, bola vybraná práve triviálna verzia Savitzky-Golayového filtra – kľzavý priemer (*angl. moving average*). A teda namiesto prekladania n vzoriek dát polynómom sa použije aritmetický priemer týchto vzoriek. Výsledok tohoto filtra je zobrazený na obrázku 5.9, k porovnaniu je možné použiť obrázok s jeho nespracovanou verziou v predošlej sekcii.



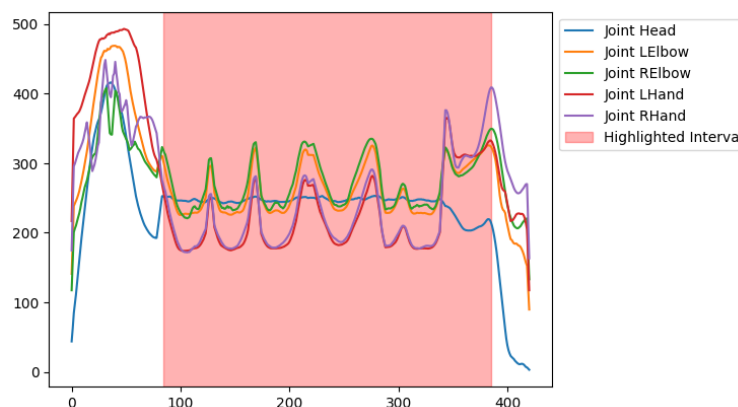
Obr. 5.9: Aplikovaný filter kľzavého priemeru.

5.5.3 Určovanie časti s opakovaniami

Najväčším problémom je určiť interval dát, kedy cvičenec vykonáva opakovania v sete. Zo spôsobu prijímania dát vyplýva, že v prípade zotrvania človeka v sledovanej zóne sú dáta automaticky posielané na server bez ohľadu na to, kedy daný set vykonáva. Toto sa pri hardvérových senzoroch diať nemôže, pretože dáta sú prenášané len vtedy, ak je zaznamenaný pohyb váhy, a teda vzniká predpoklad, že na stroji sa cvičí.

Použitie riešenie spočíva v niekoľkých krokoch a je založené na fakte, že v drivej väčšine cvikov je pohyb hlavy minimálny. Napríklad pri použití lavičky je hlava vo väčšine prípadov opretá o lavičku – či už v sedavej polohe alebo pri ležaní. Aplikovaním jednoduchého algoritmu, ktorý hľadá najväčší interval z dát dostaneme sekciu, kde s veľkou šancou môže byť vykonávaný cvik. Algoritmus funguje na presiahnutí prahovej hodnoty bodu v osiach x a y, ak je tento prah prekročený, zapamätá si daný interval (ak bol väčší ako predošlý) a pokračuje ďalej.

Výsledok tohoto algoritmu je zobrazený na obrázku 5.10 opäť na rovnakých dátach, kde červená časť predstavuje najdlhší interval, kedy sa hlava nehýbe. Tie sú zredukované pre lepší prehľad na pohyb hlavy a rúk v ose Y.



Obr. 5.10: Algoritmus pre hľadanie intervalu.

5.5.4 Určovanie periodicity pomocou Fourierovej transformácie

Jeden z relatívne jednoduchších spôsobov, ako ohodnotiť periodicitu signálu predstavuje aplikovanie Fourierovej transformácie na signál. A teda na vstup privádzame diskretný signál $x[n]$ vzorkovaný na frekvencii f_s – v tomto prípade FPS prijatých dát. Ten je pomocou vzorca 5.1, kde n je počet vzoriek, k predstavuje komponenty frekvencii a X je výsledok Fourierovej transformácie.

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-i2\pi kn/N} \quad (5.1)$$

V jazyku Python je možno použiť funkciu knižnice `numpy fft`, ktorá implementuje algoritmus FFT (Fast Fourier Transform) na vypočítanie diskretnej Fourierovej transformácie. Následne je potrebné určiť spektrum magnítúd (pozn. vzorec 5.2), kde $\text{Re}(X[k])$ and $\text{Im}(X[k])$ sú imaginárne a reálne jednotky $X[k]$, z ktorého je možno vyčítať dominantné frekvencie.

$$|X[k]| = \sqrt{\text{Re}(X[k])^2 + \text{Im}(X[k])^2} \quad (5.2)$$

Na určenie periodicity sa identifikujú dominantné frekvencie nájdením maximálnej hodnoty v spektre. Periodicita je potom vypočítaná ako pomer magnítúd dominantných frekvencii k celkovej magnítúde spektra.

$$\text{Periodicita} = \frac{|X[\text{Dominantné frekvencie}]|}{\sum_{k=0}^{N/2} |X[k]|} \quad (5.3)$$

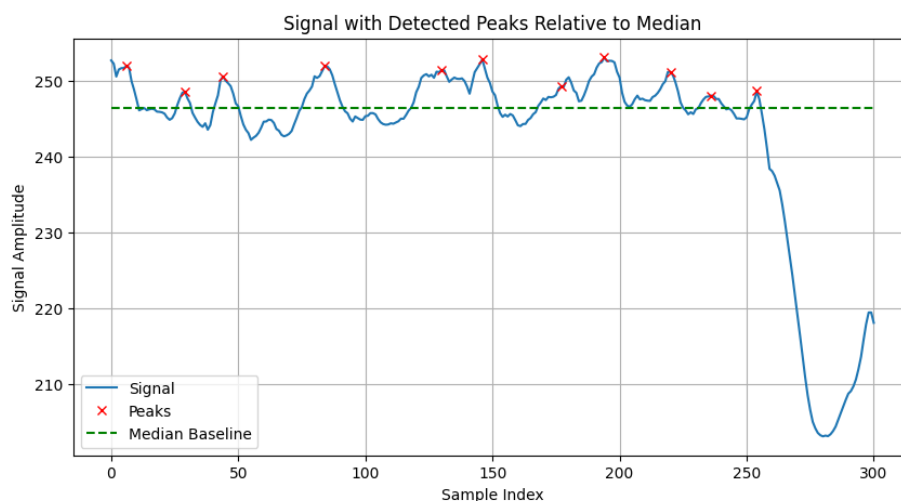
Periodicita teda udáva ako veľmi je energia signálu zhromaždená do jeho dominantných frekvencii. Tento prístup však po testovaní nevykazoval uspokojivé výsledky. Pár z dôvodov bol samotný fakt, že vzorkovanie signálu nebolo úplne periodické – v jednom prípade mala sekunda 16 vzoriek, inokedy 11 – to najmä kvôli spomaleniu inferencie v prípade detekcie viacerých osôb v obraze a transformácií potrebných na odstránenie daných osôb v tomto prípade.

Opakovania vykonávané sledovaným subjektom tiež neboli úplne periodické, síce túto periodicitu je voľným okom vidieť, je len ťažko izolovateľná metódami, ktoré zohľadňujú časovú informáciu. V konečnom dôsledku sa ukázalo, že periodicitu signálu s charakteristikou cvičenia je len veľmi ťažko všeobecne izolovať. Je možné použiť rôzne prahové hodnoty a ďalšie na mieru kované filtre tak, aby postup na väčšine testovacích dát fungoval, no preskúvané metódy, či už z oblasti štatistických metód alebo spracovania signálu zlyhávali. Boli vyskúšané aj rôzne filtre typu low-pass, ktoré mali parazitný signál odstrániť, no ten sa v menších amplitúdach podobal na signál opakovaní.

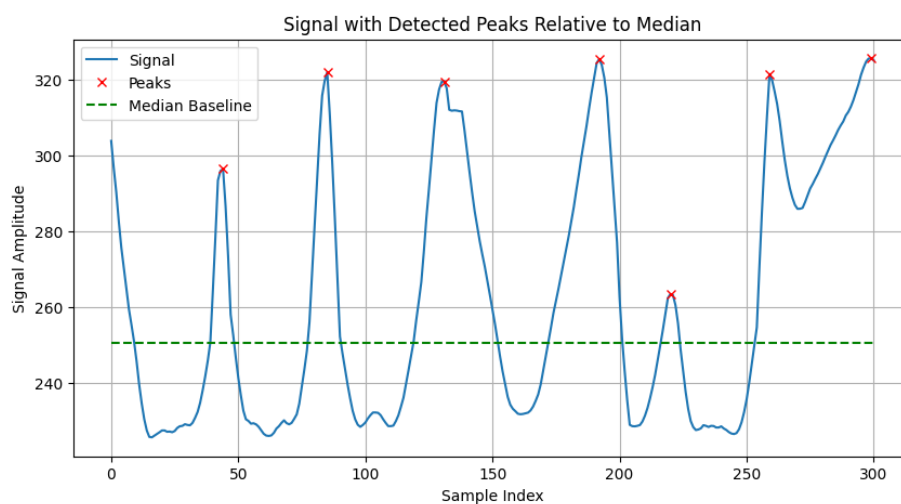
5.5.5 Určovanie opakovaní pomocou metódy hľadania vrcholov

Na akési ohodnotenie kvality odhadu opakovaní však stále môžeme použiť hľadanie vrcholov, ktorým je možné určiť aj počet opakovaní. Tento prístup spočíva v nájdení všetkých vrcholov v dátach vzdialených od seba aspoň 1 s (resp. FPS počtu vzoriek). Pre všetky vrcholy sa následne vypočíta výška od mediánu v dátach, čo by mala byť zhruba úroveň, kam sa klby vracajú. Následne sa vypočíta výška všetkých vrcholov. S týmito dátami je možné potom pracovať, a síce vo väčšine klbov ktoré sa nehýbu bude veľa malých vrcholov – v prípade šumu, alebo skoro žiadne vrcholy – opäť s malou výškou. Na tieto hodnoty sú aplikované výpočty odchýlok a v porovnaní so zadanými hodnotami pre očakávané veľkosti a možnej tolerancii, je vrátená dvojica počtu opakovaní (vrcholov) a pravdepodobnosť

správnosti odhadu (koľko dát sa odchyľuje od očakávaných). Pre príklad na reálnych dátach je možno na obrázku 5.11 vidieť šum ohodnotený algoritmom. Pre tieto dáta algoritmus vrátil dvojicu (0,0, 11), čím vyjadrí, že vidí 11 vrcholov ale pŕst že sú očakávané je 0. Naopak pre dáta z opakovaní, konkrétne bodu pravej ruky zobrazeného na obrázku 5.12 určil pŕst približne 0,6. Určenie opakovaní je už potom na jednoduchom výbere kľbu - prípadne priemeru viacerých, s najväčšou vypočítanou pravdepodobnosťou prítomnosti opakovaní.



Obr. 5.11: Šum zachytený algoritmom.



Obr. 5.12: Dáta opakovaní spracované algoritmom.

Kapitola 6

Zobrazovanie štatistík a testovanie

V tejto kapitole je opísaný spôsob testovania a zobrazovania štatistík, resp. ich sprístupnenia užívateľovi.

6.1 Štatistiky a prehľad senzorov

Zadanie presne nešpecifikuje, akým spôsobom má prehľad štatistík vyzerieť. Server preto implementuje jednoduchý API interface, cez ktorý sa dajú získať dáta, z ktorých je možné zostaviť rôzne štatistiky. K riešeniu práce je v zložke `frontend` priložený súbor `statistics.html`, ktorý pomocou jazyku Javascript komunikuje so serverovou API a zobrazuje základné dáta zo senzorov.

6.1.1 Serverové API

Serverové API funguje na protokole HTTP a je spustené na porte 8080. Obsahuje nasledujúce volania, ktoré môže užívateľ využiť k tvoreniu ľubovoľných štatistík. Forma odpovede je fixná a jej tvar je objekt typu JSON (JavaScript Object Notation). Obsahuje kľúče `result` a `error`. V prípade výskytu chyby je hodnota pod kľúčom `error` nastavená na hodnotu `True` a hodnota `result` udáva reťazec s chybou. V prípade správneho uskutočnenia transakcie je hodnota `error` nastavená na `False` a `result` obsahuje výsledok transakcie.

`/sensors`

Volanie vracia ID všetkých existujúcich senzorov, ktoré už zapísali do databázy.

`/sensor_name/<id>`

Volanie otvára konfiguračný súbor a hľadá, či bolo senzoru s daným ID nastavené pole `machine_name`. Týmto spôsobom je možné anotovať, ako sa volá stroj, na ktorom sa senzor nachádza.

Získanie tabuľky pre senzor

API taktiež implementuje niekoľko ciest, ktorými je možno z databázy dostať všetky záznamy pre dané ID senzoru. Sú to konkrétne:

- `/exercise_log/<id>`

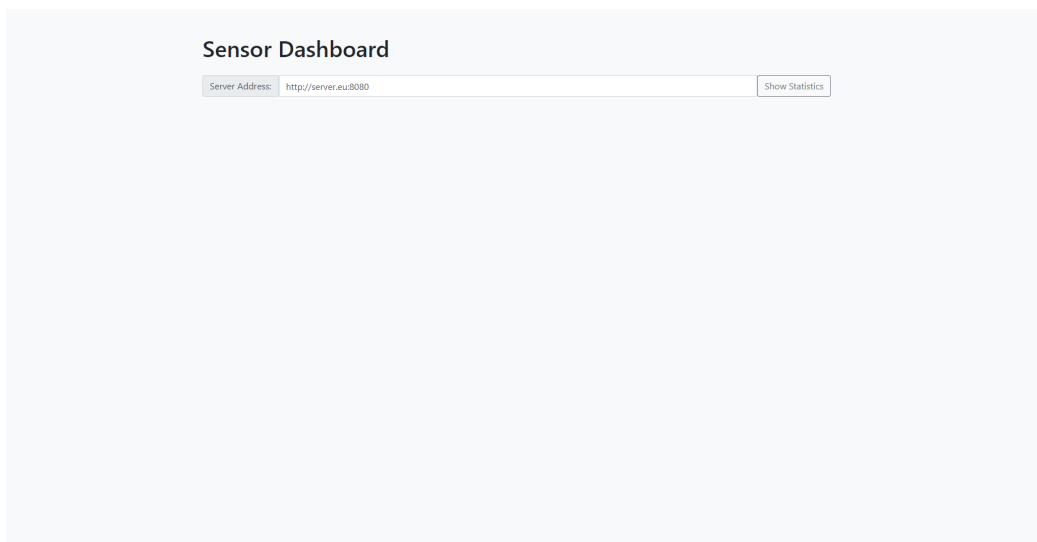
- /battery_log/<id>
- /error_log/<id>

Dáta opakovaní a váhy

Pre rýchle získanie poľa so stĺpcami váhy a počtu opakovaní z tabuľky `exercise_log` je možno použiť volania `/reps/<id>` a `/weight/<id>`. Preferované použitie je aj napriek existencii týchto volaní načítanie celej tabuľky, vďaka čomu je možno v grafoch uvádzať ďalšie doplnujúce dáta k danému setu.

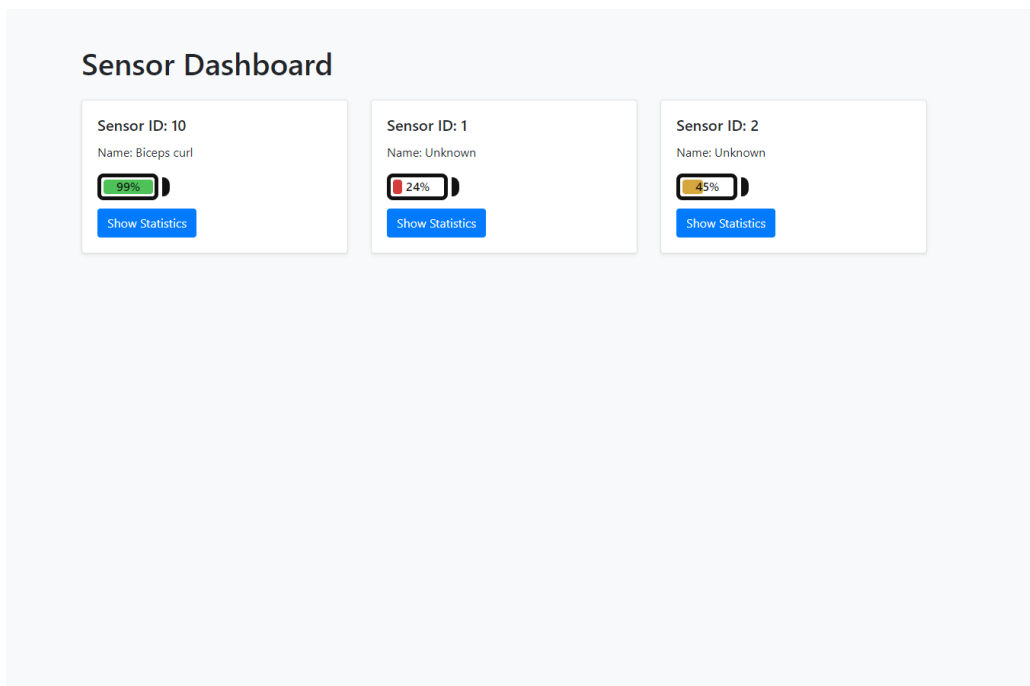
6.1.2 Zobrazenie štatistík

Pre serverové API bol vytvorený pomocou jazykov HTML, CSS a Javascript zobrazovací panel pre senzory. Ten využíva open-source knižnice `Chart.js` a `Bootstrap`. Na základnej obrazovke užívateľ zadá adresu Flask servera, následne sa mu zobrazia všetky senzory s menami strojov, ku ktorým sú tieto senzory priradené. Toto zobrazenie je možno vidieť na obrázku 6.1.



Obr. 6.1: Úvodné zobrazenie.

Prehľad sensorov je zobrazený na obrázku 6.2.



Obr. 6.2: Prehľad senzorov.

Na obrázku 6.3 je zobrazené “pop-up” okno. To sa zobrazí po stlačení tlačidla “Show Statistics” a otvorí graf zobrazujúci počet opakovaní, ktoré senzor zachytil. Po namierení kurzora na jeden z bodov sa zobrazia dodatočné údaje o tomto sete.



Obr. 6.3: Prehľad opakovaní.

6.2 Testovanie zamerané na presnosť detekcie počtu opakovaní

Navrhnutý systém bol otestovaný v reálnom prostredí, použité boli 2 kusy navrhnutých hardvérových senzorov a jeden z obrazu vytvorený virtuálny senzor. Jeden z hardvérových senzorov mal pripojený aj prídavný senzor pre snímanie zvolenej váhy. Kamerový stream bol spustený pomocou telefónu umiestneného pred lavičkou na cvičenie. Skript pre virtuálny senzor bol spustený na pomerne pomalom a úspornom počítači s procesorom Intel N100, ktorý disponuje štyrmi jadrami s maximálnou frekvenciou 3,4 GHz. Ten dokázal dáta kľbov detegovať a odosielať priemernou rýchlosťou 15 snímok za sekundu. Všetky dáta sa k analýze odosieli na VPS server umiestnený mimo lokálnu sieť fitness centra.

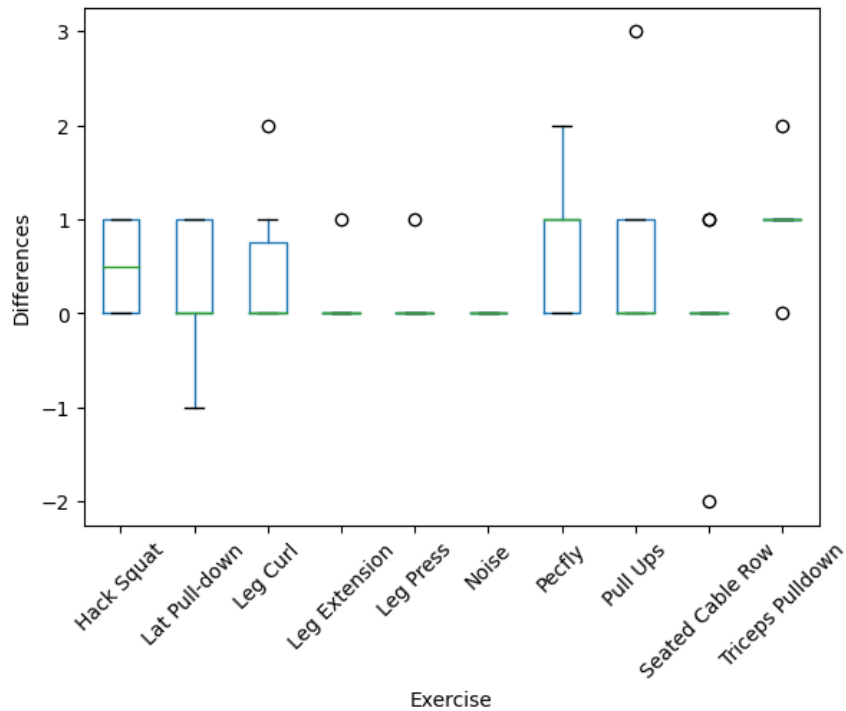
6.2.1 Spôsob testovania

Stroje s nainštalovanými senzormi boli sledované po dobu 2 hodín a boli sprístupnené návštevníkom fitness centra. V každej zo sledovaných zón boli človekom počítané opakovania, pričom boli následne s časom vykonania zaznamenané. Vykonávané sety boli spravidla v dĺžke 8 až 15 opakovaní. Po uplynutí hodiny času sa senzory umiestnili na ďalšie dva stroje, pre väčšie pokrytie množstva testovaných strojov. Z dvojice dát (x_1, x_2), kde x_1 je určený počet opakovaní systémom a x_2 je zapísaný počet opakovaní, bol vypočítaný ich rozdiel udávajúci presnosť algoritmu. Všetky výsledky boli získané pomocou predvolených hraničných hodnôt s rovnakou konfiguráciou senzora.

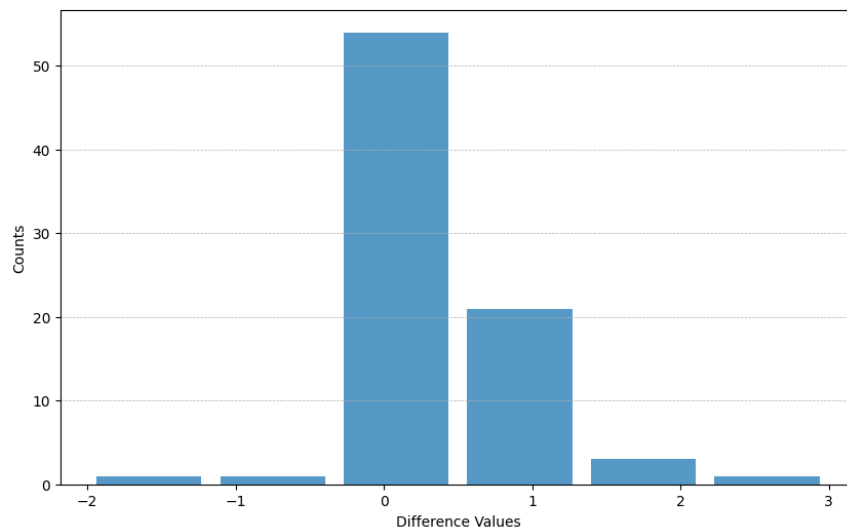
Ako bolo v sekcii 5.4 zmienené, pre vývoj algoritmu bol vrámci práce vytvorený aj dataset zozbieraných dát z akcelerometrického senzora. Každý súbor má v názve počet opakovaní a cvik, opisujúci jeho obsah. Tieto dáta boli postupne zozbierané v rovnakom fitness centre počas celého obdobia riešenia práce. Spôsob ich zberu bol totožný s procesom testovania (jediný rozdiel bol v tom, že analýza neprebíhala automaticky okamžite, ale až neskôr manuálnym spustením), preto na ne bol aplikovaný algoritmus analýzy serverovou aplikáciou a výsledky zahrnuté do výsledkov celkového testovania.

6.2.2 Výsledky presnosti hardvérových senzorov

Na obrázku 6.4 je zobrazený krabicový graf reprezentujúci výsledky rozdielov, medzi očakávaným výstupom algoritmu a jeho reálnou hodnotou z dát hardvérových senzorov. Je možné vidieť, že pri šume je 100% úspešnosť detekcie 0 opakovaní. Šum môže vzniknúť napríklad pri situácii, že akcelerometer prebudí MCU kvôli pohybu stroja (napr. v dôsledku opretia sa o stroj), ale neodohráva sa tam momentálne žiadne cvičenie. Ďalej je vidno, že algoritmus sa správa viacej pesimisticky – ak sa pomýli, je to skôr odhadom menšieho počtu opakovaní, aký bol v skutočnosti vykonaný. Na nasledujúcom obrázku 6.5 je zobrazený histogram rozdielov všetkých setov. Je na ňom vidieť vysoká úspešnosť systému odhadnúť správny počet opakovaní. V neposlednom rade za zmienku stojí aj úspešnosť vylepšenia v podobe zaznamenávania váhy. To dokázalo s takmer **90 % úspešnosťou** určiť nastavenú váhu. Táto úspešnosť bola dosiahnutá na strojoch, kde sa kolík posúval o 30 mm na každých 5 kg váhy. Najväčšie problémy merania váhy tvorili najmä krajné prípady, kedy kolík bol vzdialený čo najďalej. To je možné z časti vyriešiť presnejším upevnením senzoru ku konštrukcii stroja, avšak na takéto zásahy do upevnenia nebol pri testovaní priestor.



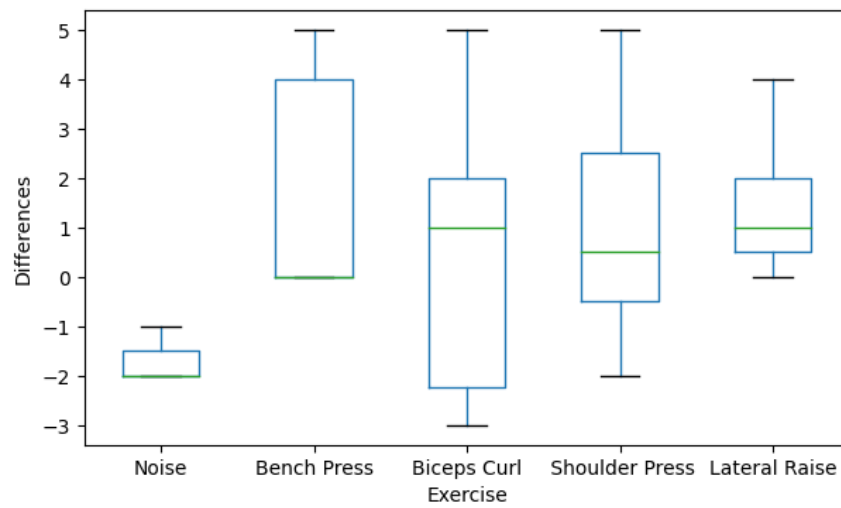
Obr. 6.4: Krabicový graf z meraní opakovaní hardvérových senzorov.



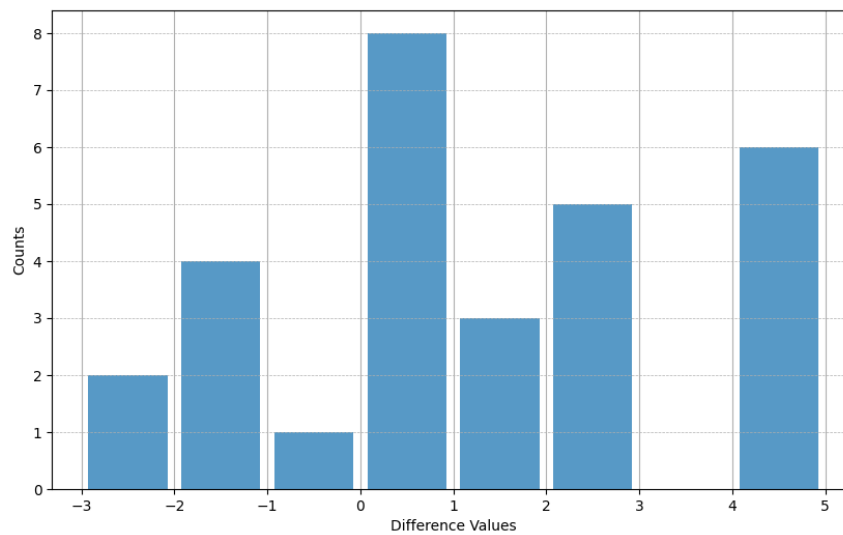
Obr. 6.5: Histogram rozdielov výsledkov všetkých setov.

6.2.3 Výsledky presnosti kamerových senzorov

Rovnako ako pri hardvérových senzoroch, na obrázku 6.6 je krabicový graf výsledkov získaných počas testovania vo fitness centre. Pri týchto dátach je už chybovosť väčšia, čo je vidieť aj na obrázku 6.7, na ktorom je zobrazený histogram chybovosti všetkých cvikov. Napriek vyššej chybovosti tu aj tak prevláda stĺpec s nulovou chybovosťou.



Obr. 6.6: Výsledky analýzy opakování zo softvérových senzorov.



Obr. 6.7: Histogram rozdielov výsledkov všetkých setov.

Kapitola 7

Záver

V práci sa podarilo navrhnúť a implementovať fungujúci modulárny systém, ktorý je schopný poskytovať spätnú väzbu užívateľom fitness centier. Navrhnuté hardvérové senzory boli schopné určiť počet vykonaní cviku na stroji s pomerne vysokou presnosťou. Ako rozšírenie bol navrhnutý a implementovaný spôsob merania kvality opakovaní spolu s určovaním nastavenej váhy na fitness stroji. Pre pokrytie miest, kde nie je možné hardvérové senzory umiestniť bol vyvinutý program, ktorý z obrazu kamerových dát a predom definovaných zón v obraze extrahuje telesné body užívateľa. Tie ďalej odosiela k analýze na server. Telesné body extrahuje pomocou existujúcich moderných riešení v podobe modelov neurónových sietí. Tento prístup vykazuje už s pomerne nižším dostupným výkonom dobré výsledky, ktoré sa však nedajú porovnávať s hardvérovým prístupom merania tohoto problému. Všetky získané výsledky sú dostupné cez serverovú API. K nej bolo vytvorené aj jednoduché grafické užívateľské rozhranie, ktoré zobrazuje stav jednotlivých sensorov, stav ich nabitia a informácie o posledných vykonaných setoch.

V tejto časti je možné reflektovať aj nedostatky, ktoré si čitateľ mohol všimnúť. Jeden z nich je v návrhu hardvérového senzora a spočíva v odpájaní batérie od nabíjacieho modulu. Toto riešenie zapínania a vypínania senzoru zabraňuje nabíjaniu počas vypnutého stavu. Ďalším z odhalených nedostatkov je zvolenie kondenzátora s malou kapacitou pri pin EN MCU. Ten slúži na zdržanie zapnutia MCU pred tým, než má zapnuté a inicializované ostatné periférie. To spôsobovalo po zapnutí nutnosť stlačenia reset tlačidla či uvedenie MCU do vypnutého a následne znova zapnutého stavu za sebou.

V softvérovej časti nedostatky systému boli najmä v spôsobe určovania konca setu. V oboch prípadoch to bolo riešené pomocou časových limitov neaktivity. Aj napriek preukázanej efektívnosti, v reálnych podmienkach husto frekventovaného fitness centra môže byť takéto riešenie nedostačujúce. Vyriešenie tohoto problému ale nie je bez implementovania detekcie cvičenia triviálne.

Ako ďalšie rozšírenie tejto práce by mohla byť jej implementácia do obalov kruhového tvaru, ktoré by mohli byť nasadené na tyče rovnako, ako kotúčové závažie. Sensory by tak mohli merať širokú škálu cvikov, v ktorých sa takéto kotúče používajú. O niečo zložitejším rozšírením by bola implementácia rozpoznávania tvárí, na základe ktorej by bolo možné priradiť jednotlivé výkony priamo ku konkrétnemu užívateľovi bez jeho interakcie so systémom počas cvičenia.

Literatúra

- [1] *ESP32-C3-MINI-1 Datasheet* [online]. Espressif Systems, 2022 [cit. 2024-5-14]. Dostupné z: https://www.espressif.com/sites/default/files/documentation/esp32-c3-mini-1_datasheet_en.pdf.
- [2] *ESP32-S3 Series Datasheet* [online]. Espressif Systems, 2023 [cit. 2024-5-14]. Dostupné z: https://www.espressif.com/sites/default/files/documentation/esp32-s3_datasheet_en.pdf.
- [3] Opening the black box of fitness tracking: understanding the mechanisms of feedback in motivating physical activity among older Singaporeans. *Behaviour & Information Technology*. 2023.
- [4] ABRIL, E. P. Tracking Myself: Assessing the Contribution of Mobile Technologies for Self-Trackers of Weight, Diet, or Exercise. *Journal of Health Communication*. Taylor & Francis. 2016, zv. 21, č. 6, s. 638–646. DOI: 10.1080/10810730.2016.1153756. PMID: 27168426. Dostupné z: <https://doi.org/10.1080/10810730.2016.1153756>.
- [5] ADDOLORATO, S., GARCÍA FERNÁNDEZ, J., GALLARDO, L. a GARCÍA UNANUE, J. An Overview of the Origins and Effectiveness of Commercial Fitness Equipment and Sectoral Corporate Settings: A Critical Review of Literature. *Applied Sciences*. 2020, zv. 10, č. 4. DOI: 10.3390/app10041534. ISSN 2076-3417. Dostupné z: <https://www.mdpi.com/2076-3417/10/4/1534>.
- [6] ADI, P. D. P., SIHOMBING, V., SIREGAR, V. M. M., YANRIS, G. J., SIANTURI, F. A. et al. A Performance Evaluation of ZigBee Mesh Communication on the Internet of Things (IoT). In: *2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT)*. 2021, s. 7–13. DOI: 10.1109/EIConCIT50028.2021.9431875.
- [7] AROGANAM, G., MANIVANNAN, N. a HARRISON, D. Review on Wearable Technology Sensors Used in Consumer Sport Applications. *Sensors*. 2019, zv. 19, č. 9. DOI: 10.3390/s19091983. ISSN 1424-8220. Dostupné z: <https://www.mdpi.com/1424-8220/19/9/1983>.
- [8] ASHRAF, F. B., ISLAM, M. U., KABIR, M. T. a UDDIN, J. YONET: A neural network for yoga pose Classification. *SN Computer Science*. Február 2023, zv. 4, č. 2. DOI: 10.1007/s42979-022-01618-8. Dostupné z: <https://doi.org/10.1007/s42979-022-01618-8>.
- [9] BARRY, R. *FreeRTOS reference manual: API functions and configuration options*. Real Time Engineers Limited, 2009.

- [10] BISHOP, D., GILES, C. a AUSTIN, G. The Lucent LambdaRouter: MEMS technology of the future here today. *IEEE Communications Magazine*. 2002, zv. 40, č. 3, s. 75–79. DOI: 10.1109/35.989760.
- [11] BOSCH SENSORTEC GMBH. *BMI160 SensorAPI* [online]. GitHub, 2021 [cit. 2024-4-11]. Dostupné z: https://github.com/boschsensortec/BMI160_SensorAPI.
- [12] CHANG, K.-h., CHEN, M. a CANNY, J. Tracking Free-Weight Exercises. In: 2007, s. 19–37. ISBN 978-3-540-74852-6. Dostupné z: https://doi.org/10.1007/978-3-540-74853-3_2.
- [13] CHESHIRE, S. a KROCHMAL, M. *Multicast DNS* [RFC 6762]. RFC Editor, február 2013. DOI: 10.17487/RFC6762. Dostupné z: <https://www.rfc-editor.org/info/rfc6762>.
- [14] DWIBEDI, D., AYTAR, Y., TOMPSON, J., SERMANET, P. a ZISSERMAN, A. Counting Out Time: Class Agnostic Video Repetition Counting in the Wild. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.
- [15] GEANTA, V. A. a PETRU, A. Improving muscle size with Weider’s principle of progressive overload in non-performance athletes. *Timisoara Physical Education and Rehabilitation Journal*. December 2021, zv. 14, s. 27–32. DOI: 10.2478/tpelj-2021-0011.
- [16] JIANG, T. *RTMPose: Real-Time Multi-Person Pose Estimation based on MMPose*. Marec 2023. Dostupné z: <https://arxiv.org/abs/2303.07399>.
- [17] KAHN, E., RAMSEY, L., BROWNSON, R., HEATH, G., HOWZE, E. et al. The effectiveness of interventions to increase physical activity: A systematic review 1 and 2. *American Journal of Preventive Medicine*. Elsevier BV. máj 2002, zv. 22, č. 4, s. 73–107. DOI: 10.1016/s0749-3797(02)00434-8. Dostupné z: [https://doi.org/10.1016/s0749-3797\(02\)00434-8](https://doi.org/10.1016/s0749-3797(02)00434-8).
- [18] KHURANA, R., AHUJA, K., YU, Z., MANKOFF, J., HARRISON, C. et al. GymCam: Detecting, Recognizing and Tracking Simultaneous Exercises in Unconstrained Scenes. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* New York, NY, USA: Association for Computing Machinery. dec 2018, zv. 2, č. 4. DOI: 10.1145/3287063. Dostupné z: <https://doi.org/10.1145/3287063>.
- [19] KRUGER, J., BLANCK, H. M. a GILLESPIE, C. *International Journal of Behavioral Nutrition and Physical Activity*. Springer Science and Business Media LLC. 2006, zv. 3, č. 1, s. 17. DOI: 10.1186/1479-5868-3-17. ISSN 1479-5868. Dostupné z: <http://dx.doi.org/10.1186/1479-5868-3-17>.
- [20] LAKOVIĆ, N., BRKIĆ, M., BATINIĆ, B., BAJIĆ, J., RAJS, V. et al. Application of low-cost VL53L0X ToF sensor for robot environment detection. In: *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*. 2019, s. 1–4. DOI: 10.1109/INFOTEH.2019.8717779.
- [21] LIN, T.-Y., MAIRE, M., BELONGIE, S., BOURDEV, L., GIRSHICK, R. et al. *Microsoft COCO: Common Objects in Context*. 2015.

- [22] MAKATSERCHYK, A. *Gym visual* [online]. [cit. 2024-5-14]. Dostupné z: <https://gymvisual.com/>.
- [23] MELNIKOV, A. a FETTE, I. *The WebSocket Protocol* [RFC 6455]. RFC Editor, december 2011. DOI: 10.17487/RFC6455. Dostupné z: <https://www.rfc-editor.org/info/rfc6455>.
- [24] MULLER, I., BRITO, R. M. de, PEREIRA, C. E. a BRUSAMARELLO, V. Load cells in force sensing analysis – theory and a novel application. *IEEE Instrumentation & Measurement Magazine*. 2010, zv. 13, č. 1, s. 15–19. DOI: 10.1109/MIM.2010.5399212.
- [25] NEWMAN, C. a KLYNE, G. *Date and Time on the Internet: Timestamps* [RFC 3339]. RFC Editor, júl 2002. DOI: 10.17487/RFC3339. Dostupné z: <https://www.rfc-editor.org/info/rfc3339>.
- [26] ONNX RUNTIME DEVELOPERS. *ONNX Runtime* [online]. 2021 [cit. 2024-5-14]. Dostupné z: <https://onnxruntime.ai>.
- [27] PAVLLO, D., FEICHTENHOFER, C., GRANGIER, D. a AULI, M. 3D human pose estimation in video with temporal convolutions and semi-supervised training. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.
- [28] RIZA ALP GULER, I. K. DensePose: Dense Human Pose Estimation In The Wild. In: . 2018.
- [29] SCHOENFELD, B. The Mechanisms of Muscle Hypertrophy and Their Application to Resistance Training. *Journal of strength and conditioning research / National Strength & Conditioning Association*. Október 2010, zv. 24, s. 2857–72. DOI: 10.1519/JSC.0b013e3181e840f3.
- [30] SCHORCHT. *LIS3DH ESP-IDF* [online]. GitHub, 2017 [cit. 2024-5-14]. Dostupné z: <https://github.com/gschorcht/lis3dh-esp-idf>.
- [31] SEEGER, C., BUCHMANN, A. a VAN LAERHOVEN, K. MyHealthAssistant: A Phone-based Body Sensor Network that Captures the Wearertextquoterights Exercises throughout the Day. In: . Január 2011.
- [32] SIMON, D. Kalman filtering. *Embedded systems programming*. 2001, zv. 14, č. 6, s. 72–79.
- [33] TOSI, J., TAFFONI, F., SANTACATTERINA, M., SANNINO, R. a FORMICA, D. Performance Evaluation of Bluetooth Low Energy: A Systematic Review. *Sensors*. 2017, zv. 17, č. 12. DOI: 10.3390/s17122898. ISSN 1424-8220. Dostupné z: <https://www.mdpi.com/1424-8220/17/12/2898>.
- [34] ULTRALYTICS. *COCO*. Dostupné z: <https://docs.ultralytics.com/datasets/segment/coco/#dataset-yaml>.
- [35] WANG, J., TAN, S., ZHEN, X., XU, S., ZHENG, F. et al. Deep 3D human pose estimation: A review. *Computer Vision and Image Understanding*. 2021, zv. 210, s. 103225. DOI: <https://doi.org/10.1016/j.cviu.2021.103225>. ISSN 1077-3142. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S1077314221000692>.

- [36] WOUTERS, C., VRANKOVIĆ, V., RÖSSLER, C., SIDOROV, S., ENSSLIN, K. et al. Design and fabrication of an innovative three-axis Hall sensor. *Sensors and Actuators. A, Physical*. január 2016, zv. 237, s. 62–71. DOI: 10.1016/j.sna.2015.11.022. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0924424715302363?casa_token=NQc0gwvDWKIAAAAA:um_jwWk0IQszBpsLoQkVQe8axDHMrqCJrHkKP1QuehXDEXSZjFnZJTOBvPuX3mSoDkY_UIriIg.
- [37] YANG, H., DENG, D.-J. a CHEN, K.-C. On Energy Saving in IEEE 802.11ax. *IEEE Access*. 2018, zv. 6, s. 47546–47556. DOI: 10.1109/ACCESS.2018.2865763.
- [38] YANG, W. a LYONS, T. Leveraging the Path Signature for Skeleton-based Human Action Recognition. *ArXiv preprint arXiv:1707.03993*. 2017.

Príloha A

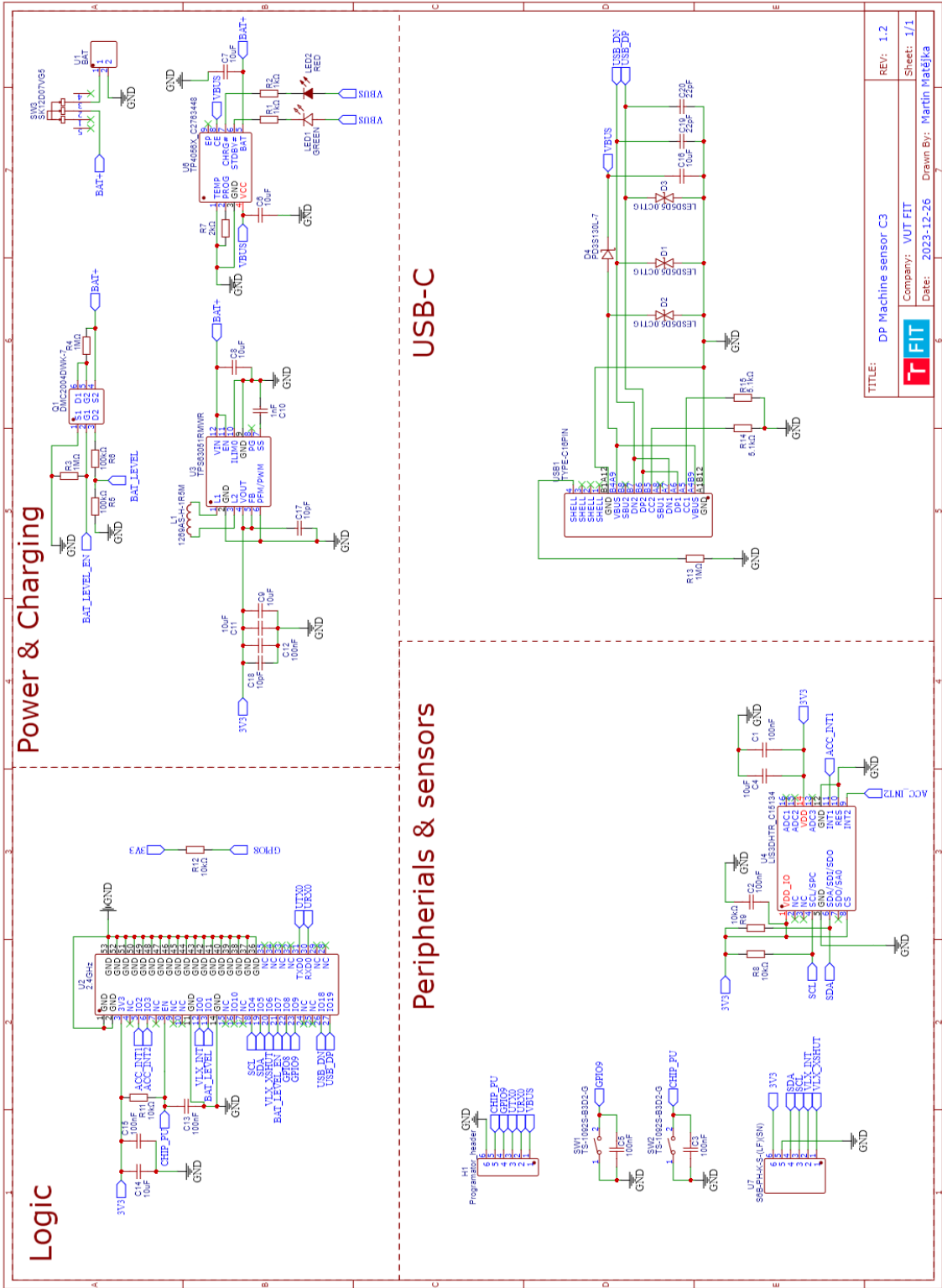
Obsah pamäťového média

1. `xmatej55.pdf` je tento dokument vo formáte PDF
2. `xmatej55_latex.zip` sú zdrojové súbory technickej správy
3. `xmatej55_source.zip` je archív so zdrojovými súborami tejto práce spolu s 3D modelmi a PCB doskami.

Príloha B

Schéma zapojenia obvodu senzora C3

Schéma nasleduje na ďalšej strane, kvôli lepšej čitateľnosti.

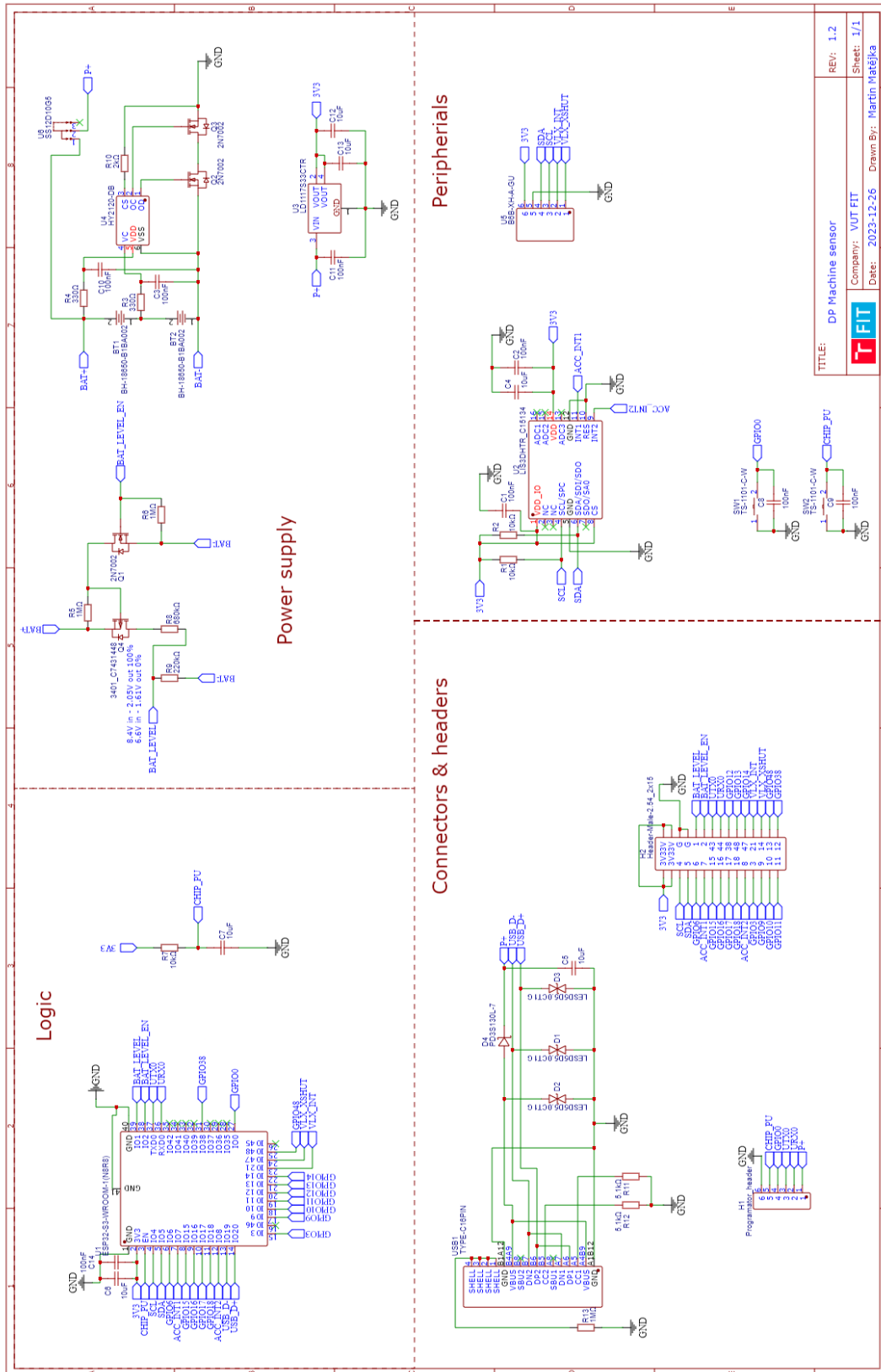


TITLE:	DP Machine sensor C3	REV:	1.2
Company:	VUT FIT	Sheet:	1/1
Date:	2023-12-26	Drawn By:	Martin Matějka

Príloha C

Schéma zapojenia obvodu senzora S3

Schéma nasleduje na ďalšej strane, kvôli lepšej čitateľnosti.



Logic

Power supply

Connectors & headers

Peripherals

TITLE: Dp Machine sensor		REV: 1.2
Company: VUT FIT		Sheet: 1/1
Date: 2023-12-26		Drawn By: Martin Matějka