

TARTU ÜLIKOOL  
Arvutiteaduse instituut  
Informaatika õppekava

Cardo Kambla

# Tegevuse tuvastamine kodukeskkonnas siluettide põhjal

Bakalaureusetöö (9 EAP)

Juhendaja: Meelis Kull, PhD

Tartu 2018

## **Tegevuse tuvastamine kodukeskkonnas siluettide põhjal**

### **Lühikokkuvõte:**

Tervishoius on suureks probleemiks kulutused ja inimressurss. Inimressurssi on piiratud kogus ja ei ole alati võimalik suunata inimest sündmuskohale abistama. Osa inimressurssi läheb selle peale, et abistada inimesi kodus, kes põevad vaimset või füüsilist haigust.

Selle töö eesmärk on abistada tervishoiutöötajaid, luues abistava masinõppe mudeli, mis suudab kaamera kaudu tuvastada inimese tegevust. Kaadritest eraldatakse siluett, millel on tausta ja inimese detailid eemaldatud, et hoida inimese anonüümsust. Alles jääb mustal taustal valge siluett.

Töö eesmärgile lähenetakse siluettide seeria liikuvate ja mitteliikuvate tunnuste eraldamisega, mille abil õpetati välja masinõppe mudel. Masinõppe mudelitena kasutati otsustusmetsa ja K lähima naabri algoritmi. Lõpliku tulemusena leiti, et on võimalik tuvastada tegevust siluetti kaudu, kuid tulemus sõltub palju andmetest ja kasutusjuhtumitest.

### **Võtmesõnad:**

K lähima naabri algoritm, otsustusmets, tegevuse tuvastus, pildianalüüs

**CERCS:**P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

## **Activity recognition in a home environment based on silhouettes**

### **Abstract:**

The amount and cost of human resources are big problems in healthcare. The availability of these resources is limited and therefore, it is not always possible to send help when someone needs assistance. A part of human resources goes into helping people at home who suffer from mental or physical disorders.

The purpose of this Bachelor's thesis is to help health professionals by creating a machine learning model, which can recognize human activity from a camera. The recorded material is extracted in a way, that the human and background details are removed to keep people's identity anonymous, resulting in a picture, where there is a white silhouette on a black background.

The purpose was achieved by extracting stationary and non-stationary features for the machine learning model. K-Nearest neighbors and Random Forest classifier were used by the new features. The final results showed that it is possible to recognize activities from silhouettes however, the result depends a lot on the data and the use cases.

### **Keywords:**

K-Nearest neighbors algorithm, Random Forest classifier, activity recognition, image analysis

**CERCS:**P170 Computer science, numerical analysis, systems, control

# Sisukord

<b>1</b>	<b>Sissejuhatus</b>	<b>5</b>
<b>2</b>	<b>Sarnased tööd</b>	<b>6</b>
<b>3</b>	<b>Andmed</b>	<b>7</b>
3.1	SPHERE projekt . . . . .	7
3.2	Andmete kogumine . . . . .	7
3.3	RGB-D kaamera andmed . . . . .	7
3.4	Kasutatud andmed . . . . .	8
<b>4</b>	<b>Märgendid</b>	<b>9</b>
4.1	Trepp . . . . .	9
4.2	Kummardamine . . . . .	10
4.3	Põlvitamine . . . . .	10
4.4	Lamamine . . . . .	10
4.5	Istumine . . . . .	11
4.6	Kõndimine . . . . .	11
4.7	Seismine . . . . .	11
4.8	Probleemid märgendamisel . . . . .	12
4.9	Märgendamise tulemused . . . . .	12
<b>5</b>	<b>Andmestiku töötlus</b>	<b>14</b>
5.1	Eeltöötlus . . . . .	14
5.2	Andmehulkade jaotus . . . . .	15
5.3	Läheneemisviis . . . . .	15
5.4	Andmestikust informatsiooni eraldamine . . . . .	16
5.5	Pildilt andmete eraldamine . . . . .	16
5.5.1	Statsionaarsete andmete eraldamine . . . . .	16
5.5.2	Mittestatsionaarsete andmete eraldamine . . . . .	17
<b>6</b>	<b>Analüüs</b>	<b>18</b>
6.1	Mudelid . . . . .	18
6.1.1	Parameetrid . . . . .	18
6.1.2	Otsustusmets . . . . .	18
6.1.3	K lähimat naabrit . . . . .	19
6.2	Tulemused . . . . .	20
6.2.1	Tüüp 1 - Pildilt otsene info . . . . .	20
6.2.2	Tüüp 2 - Mittestatsionaarne info pildiseeriast . . . . .	23
6.2.3	Tüüp 3 - Statsionaarne info pildiseeriast . . . . .	25

6.2.4	Tüüp 4 - Statsionaarne ja mittestatsionaarne info pildiseeriast .	27
<b>7</b>	<b>Järeldused</b>	<b>29</b>
<b>8</b>	<b>Kokkuvõte</b>	<b>31</b>
	<b>Viited</b>	<b>33</b>
	<b>Lisa</b>	<b>34</b>
	I. Lähtekood . . . . .	34
	II. Litsents . . . . .	34

# 1 Sissejuhatus

Tervishoid on valdkond, kus on innovaatilistest lahendustest olnud palju kasu nii tervishoiutöötajatele kui ka patsientidele. Masinõppes on tehtud mitmeid lahendusi meditsiini valdkonnas, kus suudetakse hinnata tõenäosust, kas inimesel on näiteks vähk või mitte. Sellised lahendused soosivad uute ja innovaatiliste lahenduste leidmist tervishoiu valdkonnades.

Üks sellistest lahenduste pakkujatest on SPHERE projekt. SPHERE on otsetõlkes tervishoiutarbeline kodukeskkonna sensorite platvorm (inglise keeles *Sensor Platform for HEalthcare in Residential Environment*). SPHERE projekti eesmärk on muuta paremaks ja lihtsustada tervishoiuga seotud ülesandeid, mille hulka kuulub näiteks inimese aktiivsuse hindamine ja füüsiline suutlikkus.

SPHERE projekti raames on salvestatud töödeldud kujul andmed, millest on välja eraldatud vaid inimese siluett, et hoida projektist osavõtjate anonüümsust (Twomey jt, 2016). Siluett on must-valge pilt, kus inimene on valge ja taust on must ehk ära on kaotatud detailid inimesel ja detailid taustal.

Selle töö eesmärk on luua meditsiinitöötajatele abistav masinõppe mudel, mille abil saaks tuvastada silueti kaudu inimese tegevust nagu näiteks kõndimist ja kummardamist. Tegevusi hakatakse tuvastama automaatselt, mille tõttu pole ka arstil või tervishoiu spetsialistil vaja ligipääsu siluettide andmetele, vaid saab teha järeldusi tuvastatud tegevuste järgi. Sellise tegevuste tuvastamisviisiga saaks näiteks arst dementsuse all kannatava patsiendi kohta teada, kui kaua tal läheb igapäevaste tegevuste peale ja kas on tegevustes korrapäratuid juhtumeid.

Ekspimenteeriti SPHERE projekti tulemusena saadud kahe päeva andmestikuga. Märkendid olid andmestikul olemas, kuid need polnud piisavalt spetsiifilised, et antud töös kasutusele võtta. Selle asemel otsiti töös märgendeid trepist kõndimisele, kummardamisele, istumisele, seismisele ja kõndimisele. Töö koosnes kahest etapist, millest esimene pool seisnes kavandamises ja teostamises, ning teises pooles loodi tegevuse tuvastamise mudel, kasutades eelmise etapi andmeid.

Andmete peatükis kirjeldatakse täpsemini SPHERE projekti, kuidas andmeid koguti ning eelnevalt töödeldi. Märkendamise peatükis kirjeldatakse silueti võimalikke tegevusi ja tuuakse näiteid. Andmete töötamise peatükis räägitakse täpsemini, kuidas uusi tunnuseid sisse tuuakse. Analüüsi peatükis kirjeldatakse mudeleid, parameetreid ja tulemusi, mis eksperimenteerimise käigus välja tulid. Järelduste peatükis kirjeldatakse analüüsi käigus leitud probleeme ja kitsaskohti.

## 2 Sarnased tööd

Silueti tegevuse tuvastamise kohta on leitud mitmeid erinevaid viise, kuidas tuvastada tegevust. Antud peatükis kirjeldatakse kahte artiklit. Mõlematel artiklitel on oma viis, kuidas eraldati siluette videost, antud töös salvestati ainult video töödeldud kujul, kus inimese siluett oli valgelt eraldatud musta tagataustaga.

D.K.Vishwakarma ja Rajiv Kapoor (Vishwakarma ja Kapoor, 2015) on toonud välja viisi siluetilt tegevuse tuvastamiseks. Seda viisi prooviti kolme erineva andmestiku peal. Nende eesmärk oli tuvastada, kui hästi tuvastab antud lähenemisviis tegevusi pildiseeriate alusel. Andmeid treeniti mitme mudeli peal ning tulemused olid väga head. Peamiselt taheti katsetada, kui hästi töötab hübriidmudel siluettide tegevuse tuvastamises. Tulemused olid väga optimistlikud, kuna ühe andmestikuga tuli täpsuseks 100%, teisega 96.4% ja kolmandaga 94%. Puudu olid täpsused iga märgendi kohta, mille tõttu ei saanud ka tuvastada, mille tõttu täpsus langes.

M. Singh, A. Basu ja Mrinal Kr. Mandal (Singh jt, 2008) tegid algoritmi silueti aktiivsuse tuvastamiseks. Katsetati seda meetodit nelja andmestiku peal, mis olid saadud erinevatel viisidel Southamptoni ülikooli ja Carnegie-Melloni ülikooli andmebaasist. Seisev asend (jalutamine ja seismine), istumine, kükitamine, näpuga näitamine ja pikali heitmine olid andmestikus olevad märgendid, millest osasid tegevusi kaeti mõnes andmestikus eraldi. Täpsused olid väga head seisval asendil, istumisel, pikali heitmisel ja kükitamisel. Täpsus üleüldse oli 95,5%.

Lähtekoodi ei ole olemas kummaski artiklis, selle tõttu ei saanud ka läbi proovida, kui hästi oleks SPHERE andmestik töötanud antud viiside peal. Võrreldes tsiteeritud töödega on autori poolt loodud lahendus erinev, kuna tegevuse tuvastamisel võetakse arvesse statsionaarset ja mittestasionaarset informatsiooni, mis aitab kaasa liikumise ja mitteliikumise seotud märgendite valimisel. Andmed on sarnastel töödel kõik erinevad ja neid ei saaks läbi katsetada töös käsitletud lähenemisega.

## 3 Andmed

See peatükk kirjeldab täpsemini SPHERE projekti ja selle projekti tulemusena loodud andmeid ning kuidas need saadud olid. Järgnevates alapeatükkides tuginetakse andmetele SPHERE projektist (Twomey jt, 2016).

### 3.1 SPHERE projekt

SPHERE on otseses tõlkes tervishoiutarbeline kodukeskkonna sensorite platvorm (Twomey jt, 2016). SPHERE projekti käigus plaanitakse abistada kodukeskkonnas elavaid inimesi, andes teadlastele ligipääsu projekti raames kogutud andmetele ning leides lahendusi, mis lihtsustaksid tervishoiutöötajate tööd, tõstes tervishoiu kvaliteeti kodukeskkondades. Projekt on rahastatud Suurbritannia valitsuse poolt, läbi organisatsiooni EPSRC, mille inglise keelne täisnimetus on *The Engineering and Physical Sciences Research Council* (EPSRC, 2018).

### 3.2 Andmete kogumine

Andmete kogumise jaoks (ja süsteemi testimiseks) oli kasutusele võetud maja, mille sees koguti informatsiooni jälgitava inimese randme peale käiva kiirendusanduriga, tupp paigaldatud RGB-D kaameratega ja sensoritega, mis saatsid passiivselt andmeid keskkonna kohta. Osavõtjad olid värvatud, et täita igapäeva tegevusi SPHERE majas, kus andmeid lindistati töödeldud kujul ja salvestati andmebaasi. Inimesed olid kirjutanud alla andmete kogumise jaoks määratud dokumendile, mis oli ette valmistatud Bristol ülikooli eetikakomitee poolt, et anda nõusolek andmete kasutamiseks. Esimeses andmete kogumise faasis oli projektist osavõtjatele ette antud skript, mille järgi tehakse igapäevaseid tegevusi. Teises andmete kogumise faasis lindistati osavõtjaid ja annoteeriti andmed ilma skriptita loomulikus keskkonnas.

### 3.3 RGB-D kaamera andmed

Bakalaureusetöö jaoks kasutati andmeid, mis olid saadud SPHERE projekti raames RGB-D kaameratega. Projektis olid kasutusel *ASUS Xtion PRO RGB-Depth* kaamerad (ASUS, 2012), mis suudavad tuvastada ka sügavust. Automaatne inimeste tuvastus teostati *OpenNI* (Hendel, 2012) teeki kasutades. Kaamerad olid paigutatud elutuppa, koridori ja kööki. Ühtegi teist kaamerat ei olnud kuskil mujal majas. Osavõtjate anonüümsuse hoidmiseks ei salvestatud töötlemata videomaterjali, selle asemel töödeldi kaadri pealt välja valge siluett mustal taustal, millel on olemas koordinaadid 2D ja 3D ümbritsevale raamile (inglise keeles *bounding box*) ning nende raamide keskpunktidele. 2D ümbritseva raami ja keskpunkti andmed on pikslites ja väärtus sõltub sellest, kus asub ümbritsev raam pildil, mille suurus on 640 x 480 pikslit. 3D koordinaatsüsteemi andmed on joondatud

2D ümbritseva raami järgi. Lisaks on üks täiendav dimensioon, mis mõõdab kaugust kaamerast. Esimesed kaks dimensiooni kirjeldavad vertikaalset ja horisontaalset nihkumist keskpunkti vektori suhtes millimeetrites ja viimane dimensioon kirjeldab objekti kaugust kaamerast.

### **3.4 Kasutatud andmed**

Antud töös on kasutusel kahe päeva kaadrite andmed SPHERE projektist, milles oli välja töödeldud valge siluett mustal taustal. Andmete sisse kuulusid kaadrid siluettidega ja informatsioon kaadri kohta (sh. 2D ja 3D koordinaatsüsteemide ümbritsev raam ja raamide keskpunktid). Andmestikus esines ka müra ning pilte, millel polnud piisavalt informatsiooni, et tuvastada pildi peal olevat tegevust. Kahe päeva andmete puhul saadi umbkaudselt 473 000 pilti, millest osa hiljem märgendati. Andmeid on kasutatud SPHERE projekti loal.

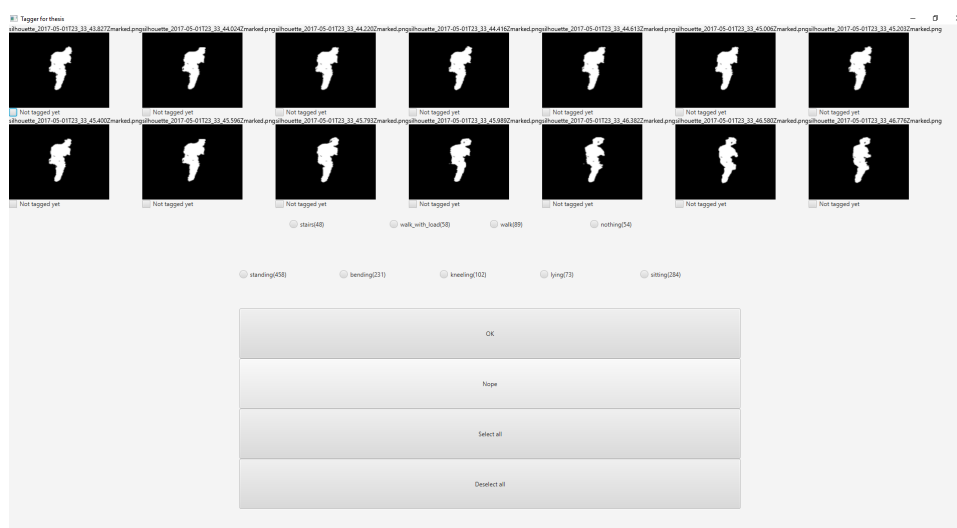


## 4 Märgendid

Andmestikus oli kahe päeva kohta suur hulk pilte, millel puudusid tegevuste märgendid. Käsitsi märgendamise lihtsustamiseks arendati *JavaFX* (Dea jt, 2014) abil rakendus, mis suutis käsitleda mitu pilti korraga ning seejärel järjestada ja salvestada vastavatesse kaustadesse. (Joonis 1) Märgendaja lõi iga märgendi jaoks eraldi kausta, mille sees on iga pildiseeria jaoks kaust (Kambla, 2018b). Märgendid määratakse võeti iga pilt järjestikuliselt. Märgendamise protsess käis järgnevalt:

- Valitakse antud pildiseeriast pildid, milles on näha järjestikku liikumist.
- Määratakse valitud piltidele märgend.
- Salvestatakse valitud pildiseeria määratud märgendiga.

Märgendamisel kasutati järgnevaid märgendeid: trepp, kummardamine, istumine, seisimine ja kõndimine.



Joonis 1: Märgendamise programm

### 4.1 Trepp

Trepist (inglise keeles *stairs*) üles või alla liikumisel kallutab inimene enda ülakeha ettepoole, et hoida tasakaalu trepist liikumisel. Trepist liikumisel on samuti ka põlved kõverdunud seisundis ning käed ja jalad on liikumises. Märgendamisel võeti kriteeriumiks pildilt silmnähtavat vertikaalset liikumist pildiseeria vältel ning jalgade liigutamist. (Joonis 2)



Joonis 2: Trepil kõndimise seeria

## 4.2 Kummardamine

Kummardamine (ingl. keeles *bending*) on inimese puhul olukord, kus ülakeha kallutatakse ettepoole ja põlved lähevad veidi kõverdunud olekusse. Märghendamisel võeti kriteeriumiks, et pildiseerialt on näha siluetti kõverdunud selga või ülakeha ettepoole kallutamist. Kummardamise märghend antakse sellisele pildiseeriale, millel on näha siluetti selga painutamas ja pea ei paikne pildi suhtes horisontaalses asendis. Kummardamise pildiseerias ei toimu jalgadega suuremat liikumist ehk jalad püsivad enamuse ajast paigal. (Joonis 3)



Joonis 3: Kummardamise seeria

## 4.3 Põlvitamine

Põlvitamine (ingl. keeles *kneeling*) on inimese puhul olukord, kus üks või mõlemad põlved on maas ning ülakeha on painutatud olekus. Siluetiga kujutades kuuluvad põlvitamise alla pildiseeriad, kus on näha, et üks põlv on kõverdunud ja teine põlv on maas ja ülakeha on madalamal kui kummardamisel. Põlvitamise pildiseerias ei toimu suuremat jalgadega liikumist ehk jalad püsivad enamuse ajast paigal. Selle töö käigus käsitleti lihtsustamise mõttes põlvitamist ja kummardamist koos ühise märghendiga "kummardamine". (Joonis 4)



Joonis 4: Põlvitamise seeria

## 4.4 Lamamine

Lamamine (ingl. keeles *lying*) on inimese puhul olukord, kus keha ei toetu otseselt jalgade peale, vaid on kontaktis pinnaga, mis toetab ülakeha. Lamamise märghendit kirjeldab siluett, mille keha püsib paigal ning on sirgjoonelisel paigutatud mittevertikaalses

asendis. Lamav asend eeldab seda, et ülakeha ja alakeha paiknevad umbkaudselt samal tasandil. Võib ette tulla erandeid, kus näiteks ülakeha on kõrgemal kui jalad. Selle töö käigus käsitleti lihtsustamise mõttes lamamist ja istumist koos märgendina "istumine". (Joonis 5)



Joonis 5: Lamamise seeria

## 4.5 Istumine

Istumine (ingl. keeles *sitting*) on inimese puhul olukord, kus põlved on kõverdatud ning ülakeha toetab altpoolt toetav pind. Istumise märgendit kirjeldab silueti seisund, kus põlved on kõverdatud ja jalad on kas maas või kõrgemal tasapinnal. Silueti keskmine piirkond on toetumas pinnale. Istumise pildiseerias ei toimu jalgadega suuremat liikumist ehk jalad püsivad enamuse ajast paigal. (Joonis 6)



Joonis 6: Istumise seeria

## 4.6 Kõndimine

Kõndimine (ingl. keeles *walk*) on inimese puhul olukord, kus keha on enamjaolt sirge ning käed ja jalad on liikumas. Kõndimisel on siluutil seisund, kus siluett on liikumas ja samal ajal seismas. Täpsemini, siluett liigub, kasutades jalgu ehk üks või mõlemad jalad on eristatavas asendis. Kõndimise pildiseerias toimub jalgadega liikumine. (Joonis 7)



Joonis 7: Kõndimise seeria

## 4.7 Seismine

Seismine (ingl. keeles *standing*) on inimese puhul olukord, kus keha toetub jalgade peale ning põlved võivad olla kõverdatud ja selg võib olla kumerdatud. Seismise all on silueti puhul mõeldud seisundit, kus siluett püsib paigal enamuse ajast. Täpsemini, kui

siluett seisab kahe jala peal ning suurt liikumist ei toimu, siis on siluett seismas. Seismise pildiseerias ei toimu jalgedega liikumist ehk jalad püsivad enamuse ajast paigal. (Joonis 8)



Joonis 8: Seismise seeria

## 4.8 Probleemid märgendamisel

Pildiseeriad said märgendatud käesoleva töö autori poolt. Tekkisid olukorrad, kus piltide pealt ei saanud piisavalt informatsiooni, et otsustada, mis märgend sobiks. Tihti juhtus ka olukordi, kus andmed kordusid väga pikkade lõikudena, sest siluett oli jäänud istuma pikemaks perioodiks ja siis suuri muudatusi ei toimunud pikemas ajavahemikus. Projekti raames on kirjeldatud 20 tegevuse märgendit, millest bakalaureusetöös on võetud arvesse ainult osad (Twomey jt, 2016). Selle peamiseks põhjuseks oli tegevuste näidete puudulikkus. Märgendamisel on probleem, kus üks pildiseeria võib tähendada mitut sarnast märgendit. Sellisel juhul on mõistlik kasutusele võtta ühine märgend, mis katab kõiki sobivaid märgendeid. Seda on tehtud ka märgendamisel, kus on kokku viidud kummardamise ja põlvitamise märgend, ning samuti ka lamamise ja istumise märgend.

## 4.9 Märgendamise tulemused

Tabelis 1 on näidatud märgendamise tulemused. Tabelist saab järeldada, et märgendite andmed ei olnud jaotatud ühtlaselt, sest trepi ja kõndimise märgendil on vähem seeriaid kui ülejäänutel märgenditel. Keskmine pildiseeria oli umbkaudselt 13 pilti seeria kohta. Üldiselt on umbes 9-10 siluetti sekundi kohta. Iga silueti vahel on umbes 0,11-0,13 sekundit vahet.

Märgend	Seeriade arv	Keskmine seeria pikkus
trepp	48	11.2
kummardamine	333	13.4
istumine	357	13.9
seismine	458	13.6
kõndimine	147	10.1
$\Sigma$	1343	13.1

Tabel 1: Märgendamise tulemused. Iga märgendi kohta on toodud välja märgendis olevate seeriade arv ja keskmise seeria pikkus. Viimasel real on märgendite seeriade summa ja keskmise seeria pikkus.

Aega kulus märgendamisele umbkaudselt 20 tundi, millele lisaks käsitsi märgendamise programmi arendamisele kulus 10 tundi.

## 5 Andmestiku töötlus

Peale piltide märgendamist eraldatakse pildiseeriatest informatsioon. Selleks prooviti kasutada *OpenPose* raamistikku, mis suutis tuvastada mitme inimese keha, näo ja käte võtmepunkte (OpenPose, 2018). *OpenPose* raamistikku prooviti SPHERE andmete peal, aga saadud tulemused olid vähelubavad, sest *OpenPose*'i poolt genereeritud andmed olid siluetis reaalistest võtmepunktidest väga erinevad. Sellepärast ei võetud *OpenPose* raamistikku antud töös kasutusele. Järgmistes alapeatükkides kirjeldatakse täpsemini, kuidas andmeid töötleti.

### 5.1 Eeltöötlus

Andmestikus olevad 2D ja 3D andmed ei ole konstantse suurusega iga pildi puhul. Masinõppe mudeliga töötamisel on vajalik, et kõik andmed, mis sisendiks antakse, oleksid konstantse suurusega. Lisaks konstantsele suurusele on probleemiks ka arvutamise kulu. Mida suurem on pilt, seda rohkem aega kulub arvutamisele. Arvutamise kulu ja konstantse suurusega piltide arvutamiseks on vaja skaleerida pilte. 3D ümbritsev raam on sama raam, mis on 2D ümbritsev raam, ainult 3D puhul on lisatud juurde sügavus ehk 3D raam on saadud 2D raamist sügavuse lisamise teel. 2D koordinaadid on olemas millimeetrites ja pikslites. Seega on meil 2D raami koordinaadid olemas kahes erinevas koordinaatsüsteemis ja see võimaldab silueti teisendada samasse koordinaadistikku, mida kasutab 3D koordinaatsüsteem. Konverteerida on vaja selle jaoks, et varieeruvate suurustega 2D raami sisse jääv info saaks mahutada konstantse suurusega pildi sisse. Andmestiku eeltöötlemiseks kasutati programmeerimiskeelt *Python* (Python Software Foundation, u), versioon 2.7.15 ja 3.6.3 ning teke *OpenCV* ((Bradski, 2000), versioon 3.3.1) ja *NumPy* ((Oliphant, 2006), versioon 1.12.1). *OpenCV* abil töödeldi pilte ja *NumPy*'ga tehti keerukamaid arvutusi. SPHERE projekti kaudu saadud andmete töötlemisel võeti arvesse pildile algoritmi poolt genereeritud andmed (2D ja 3D ümbritsev raam). Piltide skaleerimiseks peab olemas olema konstantne 3D koordinaatsüsteemi väärtus ja konstantne 2D koordinaatsüsteemi väärtus, et saaks soovitud 3D koordinaatsüsteemist skaleerida pildi 2D koordinaatsüsteemi. Pildi skaleerimiseks on vaja teha järgmist:

- Orginaalselt pildilt (suurusega 640x480 pikslit) on vaja välja lõigata 2D silueti ümbritseva raami koordinaatide sisse jääv pilt. Väljalõigatud pildi suurus on sama, mis on 2D ümbritseva raami kõrgus ja laius.
- Eelnevalt mainitud konstantset 3D koordinaatsüsteemi väärtust ja 2D süsteemi väärtust kasutatakse konverteerimiseks. Konverteerimisega saab teada, kui palju millimeetreid on ühes pikslis.
- Väljalõigatud pilti skaleeritakse, kasutades selleks eelnevalt arvutatud konversiooni.

Skaleerimise jaoks leiti 3D süsteemi kõrgus ja laius ning jagati need konversiooniga, et saada skaleeritud pildi uus kõrgus ja laius 2D koordinaatsüsteemis.

- Väljalõigatud pilt skaleeritakse uue kõrguse ja laiuse järgi, kasutades selle jaoks *OpenCV* skaleerimise meetodit. Skaleerimise viisiks on valitud parameeter `INTER_AREA`, mida kasutatakse peamiselt piltide alla skaleerimiseks, kuna selle skaleerimisviisi puhul ei teki kattuvust teiste pikslitega.
- Luuakse uus pilt, mis on konstantse 2D koordinaatsüsteemi suuruses.
- Skaleeritud pilt lisatakse tühja pildi keskele.
- Tulemuseks on konstantses suuruses 2D koordinaatsüsteemis skaleeritud pilt.

## 5.2 Andmehulkade jaotus

Tavaliselt jaotatakse masinõppes andmestiku andmed juhuslikus järjekorras treening- ja testandmestikuks. Pildiseeriade puhul on võimalik, et üks pildiseeria on väga sarnane teise pildiseeriaga, kui seeriad on märgendatud üksteise järel. Sellisel juhul, kui andmestik jaotatakse juhuslikult, on võimalus, et treeningandmestikus on andmed, mis on väga sarnased testandmestikus olevate andmetega. Selle olukorra tulemusena võib masinõppe mudel anda väga optimistlikke tulemusi. Selle probleemi ennetamiseks on eelnevalt käsitsi jaotatud andmestik treening- ja testandmestikuks. Treeningandmestikuks on ühe päeva andmed ja testandmestikuks on teise päeva andmed.

## 5.3 Lähenemisviis

Oli vaja leida viis, kuidas saaks eristada tegevust piltidel. Iga piksli väärtus on kas must või valge. Selleks, et eristada, kas pildil on liikumine, seismine või mitte midagi, on vaja leida viis, kuidas tunnuseid eraldada. Esialgu, et liikumine saaks toimuda, peab olema mitu pilti (st. peab olema olema mingis ajavahemikus läbitud pildiseeria). Täpsemini, peab võtma ajavahemiku, millal seerias toimus tegevus. Ajavahemik ei pea olema konstantne, vaid sõltub, kuidas piltidel on siluett liikumas. Idee poolest võetakse pildiseeria, millele määratakse märgend. Pildiseeria piltidest loetakse sisse pikslite väärtused koordinaadi põhisealt. Kui kokku lugeda iga pildi piksli asukoht kindla punkti kohta, siis saab genereerida kahte liiki andmeid. Esimeseks saadakse pilt, mis näitab ära pildil olevad koordinaadid, kus siluett ei püsi paigal. Seda saab välja kutsuda funktsiooniga, mis sarnaneb operaatoriga XOR. Teiseks saadakse pilt, mis näitab ära pildil olevad koordinaadid, kus siluett püsib paigal terve seeria vältel. Seda saab välja kutsuda funktsiooniga, mis sarnaneb operaatoriga AND.

## 5.4 Andmestikust informatsiooni eraldamine

Pildiseeriatest on võimalik eraldada mittestatsionaarset ja statsionaarset informatsiooni. Pildi iga piksel on kas musta või valget värvi, kus valge värv määrab silueti. Töö raames on käsitletud musta piksilt väärtusega 0 ja valget piksilt väärtusega 1.

## 5.5 Pildilt andmete eraldamine

Masinõppega tegelemiseks on vaja sisendiks andmehulka, kus on olemas tunnused ja treenimiseks/testimiseks ka lõpptulemus. Sisseloetava pildi kuju on näidatud maatriksis (1).

$$\mathbf{P}^{m \times n} = \begin{bmatrix} x_{1,1} & x_{2,1} & x_{3,1} & \dots & \dots & x_{m,1} \\ x_{1,2} & x_{2,2} & \dots & \dots & \dots & \dots \\ x_{1,3} & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{1,n} & \dots & \dots & \dots & \dots & x_{m,n} \end{bmatrix} \quad (1)$$

Tunnuste eristamiseks võetakse sisendpildi maatriksilt (1) kõik read ja laotatakse üheks reaks:

$$\mathbf{P} = ( x_{1,1}, x_{2,1}, \dots, \dots, x_{m,1}, x_{1,2}, x_{2,2}, \dots, \dots, x_{m,2}, \dots, \dots, x_{m,n} ). \quad (2)$$

Kui on olemas pildiseeria  $Se$ , siis tuleb lugeda kõik selle seeria pildid  $P_1, P_2, \dots, P_m$  sisse ja paigutada ühte massiivi:

$$Se = ( P_1, P_2, \dots, P_m ). \quad (3)$$

### 5.5.1 Statsionaarsete andmete eraldamine

Statsionaarsed andmed näitavad pildiseeria raames informatsiooni paigal püsimise kohta. Statsionaarse informatsiooni saamiseks peab lugema kõik pildiseeria pildid ühte massiivi ning eraldama välja pikslid iga rea ja veeru koordinaadi kohta terve massiivi raames, kasutades valemit (4), kus  $Se$  on pildiseeria massiiv,  $m$  on pildiseerias olevate piltide arv,  $x$  on rea indeks,  $y$  on veeru indeks ja  $l$  on lävend. Lävend on lubatud vea määr. Statsionaarsete andmete tulemusena tekib uus pilt, mille väärtused sõltuvad pildiseerias olevatest piltidest ja lävendist:

$$s(Se, x, y, l) = \begin{cases} 1, & \text{kui } \frac{\sum_{i=1}^m Se_{i,x,y}}{m} \geq l \\ 0, & \text{muidu} \end{cases} \quad (4)$$

Valemis (4) tähistab  $Se_{i,x,y}$  seerias asuvat  $i$ -nda pildi piksli koordinaati  $x, y$ . Statsionaarne on piksel siis, kui vähemalt  $l$  osakaalul piltidel on seerias  $m$  antud kohal valge piksel.



### 5.5.2 Mittestatsionaarsete andmete eraldamine

Mittestatsionaarsed andmed näitavad pildiseeria raames informatsiooni liikumise kohta. Mittestatsionaarse informatsiooni saamiseks peab lugema kõik pildiseeria pildid sisse ja tekitama ühe pildiseeria massiivi. Mittestatsionaarsete andmete tulemusena tekib uus pilt, mille väärtused sõltuvad pildiseerias olevatest piltidest ja lävendist. Mittestatsionaarsete andmete eraldamiseks on vaja iga rea ja veeru koordinaadid läbida terve pildiseeria massiivi raames, kasutades valemit (5), kus  $Se$  on pildiseeria,  $x$  on rea indeks,  $y$  on veeru indeks,  $l_1$  on alamlävend ja  $l_2$  on ülemlävend. Lävend on lubatud vea määr. Mittestatsionaarsete andmete tulemusena tekib uus pilt, mille väärtused sõltuvad pildiseerias olevatest piltidest ja lävendist:

$$t(Se, x, y, l) = \begin{cases} 1, & \text{kui } l_2 \geq \frac{\sum_{i=1}^m Se_{i,x,y}}{m} \geq l_1 \\ 0, & \text{muidu} \end{cases} \quad (5)$$

Valemis (5) tähistab  $Se_{i,x,y}$  seerias asuvat  $i$ -nda pildi piksli koordinaati  $x, y$ . Mittestatsionaarne on piksel siis, kui antud kohal on seeria  $m$  valgete pikslite osakaal vähemalt  $l_1$ , kuid mitte suurem kui  $l_2$ .

## 6 Analüüs

Analüüsimiseks kasutati programmeerimiskeelt *Python* ((Python Software Foundation, u), versioon 3.6.3), teeki *scikit-learn* ((Pedregosa jt, 2011), versioon 0.19.1), *OpenCV* ((Bradski, 2000), versioon 3.3.1), *NumPy*'t ((Oliphant, 2006), versioon 1.12.1) ja *pandas* ((McKinney, 2010) versioon 0.20.3).

Andmestikust saab eraldada 4 tüüpi informatsiooni. Need tüübid on järgnevad:

- Tüüp 1 - Pildilt otsene info - Andmestikust tuvastatakse iga pilt eraldiseisvalt. Selline lähenemine on kasulik, kui on vaja saada infot vaid ühe kindla pildi kohta ning see ei nõua terve pildiseeria analüüsimist.
- Tüüp 2 - Mittestatsionaarne info pildiseeriast - Andmestikust võetakse iga pildiseeria pealt mittestatsionaarne info.
- Tüüp 3 - Statsionaarne info pildiseeriast - Andmestikust võetakse iga pildiseeria pealt statsionaarne info.
- Tüüp 4 - Statsionaarne ja mittestatsionaarne info pildiseerias - Andmestikust võetakse iga pildiseeria kohta mittestatsionaarne ja statsionaarne info.

### 6.1 Mudelid

#### 6.1.1 Parameetrid

Antud masinõppe mudelite parameetrite suuruste valimisel on võetud arvesse olemasolevaid treenimisandmeid. Treenimisandmetena kasutatakse üle 50 näite, mis on väike hulk treenimiseks. Võttes arvesse, et treenimishulk on pool andmetest ja sellest treenimishulgast on osa valideerimishulk, jääb treenimise jaoks väike osa masinõppe mudeli treenimiseks. Analüüsimise käigus kasutatakse nelja andmestiku kasutusviisi.

#### 6.1.2 Otsustusmets

Ühe masinõppe mudelina kasutati otsustusmetsa, mis realiseerides teegi *sklearn* (Pedregosa jt, 2011) otsustusmetsa (inglise keeles *Random forest*) (Liaw, Wiener jt, 2002) meetodiga. Sellel on mitmeid parameetreid, mida saab muuta, et sobituks andmetega paremini. Üks muudetavatest parameetritest on otsustuspuude arv. Mida rohkem otsustuspuud on otsustusmetsas, seda täpsem on otsustusmets. Selle miinuspool on ajaline kulu, sest mida rohkem on puud, seda kauem läheb mudelil vastuse andmisega, kuna peab läbi-ma enne kõik puud. Parameetrite tuunimise jaoks kasutati *scikit-learn* teegist meetodit *GridSearchCV*'d (Pedregosa jt, 2011). Eksperimentides on kasutatud mittestatsionaarse info eraldamiseks alamlävendit 0,4 ja ülemlävendit 0,6. Statsionaarse info eraldamiseks kasutati lävendit 0,6. Piltide suurus on 32x32 pikslit. Mittestatsionaarne ja statsionaarne

info olid eelnevalt ära töödeldud, et lihtsustada mudeli treenimist. Eelneval töötlemisel genereeritakse uued pildid, kus on olemas statsionaarne info ja mittestatsionaarne info. *GridSearchCV* meetod on kasulik, sest selle meetodi sees tehakse ka andmestiku valideerimine automaatselt ära. Prooviti järgnevate parameetritega:

- Otsustuspuude arv - 1000
- Maksimaalne sügavus metsal (MSA) - väärtustega [3, 4, 5, ..., 11, 12]
- Maksimaalne tunnuste arv (MFA) - väärtustega [50, 75, 100, 125, 150]
- Minimaalne näidete arv haru tekitamiseks (MNAHT) - väärtustega [5, 10, 20]
- Minimaalne näidete arv lehe tekitamiseks (MNALT) - väärtustega [3, 5, 7, 10]

Peale *GridSearchCV* jooksutamist antakse ette optimaalsed parameetrid. Optimaalseid parameetreid on näha tabelis 2.

Tüüp	OA	MSM	MFA	MNAHT	MNALT
1	1000	7	100	3	6
2	1000	7	50	3	5
3	1000	7	75	3	5
4	1000	12	135	3	6

Tabel 2: Optimaalsed parameetrid, mis saavutati *GridSearchCV*ga, kui kasutati otsustusmetsa. Tähiste tähendused: OA - otsustuspuude arv, MSM - maksimaalne sügavus metsal, MFA - maksimaalne tunnuste arv, MNAHT - minimaalne näidete arv haru tekitamiseks ja MNALT - minimaalne näidete arv lehe tekitamiseks

### 6.1.3 K lähimat naabrit

Lähima naabri klassifikaator (inglise keeles *k-nearest neighbors*) (Peterson, 2009) töötab selle põhimõtte järgi, et otsitakse sisendiks antud tulemusele K lähimat naabrit ning selle järgi otsustatakse, mis on parim tegevus sisendile. Lähima naabri klassifikaatoril on peamiseks parameetriks K. Töö jaoks kasutati *scikit-learn* K lähima naabri meetodit. (Pedregosa jt, 2011)

- Mitme lähima naabri järgi ennustab? - väärtustega [3, 4, 5, ..., 11, 12]

Peale *GridSearchCV* jooksutamist antakse ette optimaalsed parameetrid. Optimaalseid parameetreid on näha tabelis 3.

Tüüp	Lähimaid naabreid arvesse võetud
1	12
2	3
3	3
4	3

Tabel 3: Optimaalsed parameetrid, mis saavutati *GridSearchCV*ga, kui kasutati K lähima naabri klassifikaatorit

## 6.2 Tulemused

### 6.2.1 Tüüp 1 - Pildilt otsene info

Andmestikus on pildiseeriad, mida on võimalik sisendiks jaotada mitmel viisil. Need viisid on järgnevad:

- Eraldada iga pilt ning tuvastada igale pildile märgend
- Tuvastada pildiseerias igale pildile märgend ning tagastada kõige rohkem esinev märgend

#### Tüüp 1.1 - Igale pildile määratakse eraldi märgend

Igal pildil märgendi tuvastamisega on näha, et enamus pilte, mis õigesti märgendatakse, on varustatud märgendiga, mis ei ole liikumisega seotud.

Tegelik	Tuvastatud märgend						Skoorid		
	TR	KU	I	SE	KÕ	$\sum$	T	S	FS
TR	138	0	0	31	0	169	0.88	0.82	0.85
KU	0	1472	56	315	4	1847	0.52	0.8	0.63
I	0	944	1429	30	1	2404	0.94	0.59	0.73
SE	3	346	34	2740	7	3130	0.73	0.88	0.8
KÕ	15	67	3	631	15	731	0.56	0.02	0.04
$\sum$	156	2829	1522	3747	27	8281	Täpsus: 70.0%		

Tabel 4: Tüüp 1.1 täpsused otsustusmetsaga, kasutatud on parameetreid tabelist 2. Tähi-  
sed: T - täpsus (inglise keeles *precision*), S - saagis (inglise keeles *recall*), FS - F-skoor  
(inglise keeles *F-score*), TR - trepp, KU - kummardamine, I - istumine, SE - seismine,  
KÕ - kõndimine.

Jooniselt 9 on näha kahte siluetti, mis on õigesti märgendatud. Vasakpoolsel pildil on siluett kõndimas, sest ta jalad on liikumise asendis ja ta on horisontaalsel tasandil. Parempoolsel pildil on siluett trepist üles minemas, seda saab kindlamalt öelda, sest

siluett ei asu enam pildi keskel, vaid on liikumas kaadrist välja ja üks põlvedest on kõverdatud.



Joonis 9: Õigesti märgendatud kõndimise ja trepi pildid

Probleemaatiliseks muutuvad mudeli ennustused siis, kui pildiseerias toimuvad vahepeal liigutused, mida tuvastatakse teise märgendiga. Kõndimise pildiseerias esineb see probleem tihedamini, kuna osadel seeria pildidel tundubki üksikult, et inimene seisab. Selle probleemi tõttu võib ka mudel ennustada, et seismine on trepist liikumine või kõndimine. (Joonis 10 )



Joonis 10: Seisvate märgendite pildid tuvastatud kõndimiseks ja trepist liikumiseks

Sarnane probleem esineb ka K lähima naabri klassifikaatoriga, kuid mitte trepist liikumise märgendi puhul. Peamine probleem tekib seismise märgendamisel, kuna mudel märgib tihti seismist kõndimise märgendiga. Silueti tuvastamises on seismine ja kõndimine väga sarnased, kuna kõndimise seeria jooksul esineb pilte, kus siluett on "seismas". Käsitsi seeria märgendamisel on sellele pildile pandud kõndimise märgend, mistõttu sarnase pildi esinemisel mudelis võidakse panna märgendiks "kõndimine", kuigi tegelikult pilt oli seismise seeriast. Selle tõttu võib K lähima naabri klassifikaator tuvastada seisvaid pilte kõndimisena.

Tegelik	Tuvastatud märgend						Skoorid		
	TR	KU	I	SE	KÕ	$\Sigma$	T	S	FS
TR	149	0	0	20	0	169	0.76	0.88	0.82
KU	0	1482	51	286	28	1847	0.48	0.8	0.6
I	0	1060	1309	30	5	2404	0.95	0.54	0.69
SE	24	427	19	2513	147	3130	0.76	0.8	0.78
KÕ	23	103	2	473	130	731	0.42	0.18	0.25
$\Sigma$	196	3072	1381	3322	310	8281	Täpsus: 67.4%		

Tabel 5: Tüüp 1.1 täpsused K lähima naabri algoritmiga, kasutatud on parameetreid tabelist 3. Tähisted: T - täpsus, S - saagis, FS - F-skoor, TR - trepp, KU - kummardamine, I - istumine, SE - seismine, KÕ - kõndimine.

### Tüüp 1.2 - Igale pildile määratakse märgend ning tagastatakse pildiseerias kõige rohkem esinev märgend

Pildilt otsene info ei anna informatsiooni liikumise kohta, mis teeb mudeli ebaefektiivseks, kui märgenditeks on tegevused, mis sõltuvad palju liikumisest ja on sarnased osade märgenditega, mis ei sõltu liikumisest. Tabelist 6 on näha, et mudel kõndimise märgendil ei suutnud õigesti ühtegi kõndimise märgendit ennustada.

Tegelik	Tuvastatud märgend						Skoorid		
	TR	KU	I	SE	KÕ	$\Sigma$	T	S	FS
TR	14	0	0	3	0	17	0.88	0.82	0.85
KU	0	112	1	28	1	142	0.55	0.79	0.65
I	0	66	107	2	0	175	0.98	0.61	0.75
SE	0	21	1	215	0	237	0.66	0.91	0.77
KÕ	2	5	0	77	0	84	0.0	0.0	0.0
$\Sigma$	16	204	109	325	1	655	Täpsus: 68.4%		

Tabel 6: Tüüp 1.2 ennustamise tulemused otsustusmetsaga, kui igale pildile on ennustatud märgend ja tagastatud on pildiseeriast kõige rohkem esinev märgend. Mudeli loomisel on kasutatud parameetreid tabelist 2. Tähistes: T - täpsus, S - saagis, FS - F-skoor, TR - trepp, KU - kummardamine, I - istumine, SE - seismine, KÕ - kõndimine.

Mudel ennustas ühe kummardamise näite kõndimisena (Joonis 11). Selle näite puhul on kummardamine määratud märgendiks, kuna pea on ettepoole kallutatud, aga mudel tuvastas pilte üksikult seismise ja kõndimise märgenditega.



Joonis 11: Kummardamise seeria tuvastatud kõndimisena

K lähima naabri klassifikaator suudab paremini tuvastada liikumisega seotud märgendeid kui otsustusmets.

Tegelik	Tuvastatud märgend						Skoorid		
	TR	KU	I	SE	KÕ	$\Sigma$	T	S	FS
TR	16	0	0	1	0	17	0.89	0.94	0.91
KU	0	116	3	22	1	142	0.52	0.82	0.64
I	0	72	100	3	0	175	0.97	0.57	0.72
SE	1	25	0	202	9	237	0.69	0.85	0.76
KÕ	1	9	0	64	10	84	0.5	0.12	0.19
$\Sigma$	18	222	103	292	20	655	Täpsus: 67.8%		

Tabel 7: Tüüp 1.2 ennustamise tulemused K lähima naabri klassifikaatoriga, kui igale pildile on ennustatud märgend ja tagastatud on pildiseeriast kõige rohkem esinev märgend. Mudeli loomisel on kasutatud parameetreid tabelist 3. Tähistes: T - täpsus, S - saagis, FS - F-skoor, TR - trepp, KU - kummardamine, I - istumine, SE - seismine, KÕ - kõndimine.

Mudel tuvastas kõndimise märgendit väga hästi (Joonis 12), kuigi mudel märgendas ka paar seismise pildiseeriat kõndimise alla, mis teeb kõndimise tuvastamise ebakindlaks (Joonis 13).



Joonis 12: Kõndimise seeria tuvastatud kõndimisena



Joonis 13: Seismise seeria tuvastatud kõndimisena

### 6.2.2 Tüüp 2 - Mittestatsionaarne info pildiseeriast

Mittestatsionaarse info eraldamisega tuuakse sisse uued tunnused, mida eelnevalt polnud olemas. Mittestatsionaarne info tähendab, et on olemas liikumise kohta informatsioon, kus toimub kõige rohkem aktiivsust pildiseerias.

Tegelik	Tuvastatud märgend						Skoorid		
	TR	KU	I	SE	KÕ	$\Sigma$	T	S	FS
TR	11	0	2	4	0	17	1.0	0.65	0.79
KU	0	69	48	25	0	142	0.57	0.49	0.52
I	0	28	144	3	0	175	0.58	0.82	0.68
SE	0	17	56	164	0	237	0.62	0.69	0.65
KÕ	0	8	0	70	6	84	1.0	0.07	0.13
$\Sigma$	11	122	250	266	6	655	Täpsus: 60.2%		

Tabel 8: Tüüp 2 täpsused otsustusmetsaga, kasutatud on parameetreid tabelist 2. Tähhised: T - täpsus, S - saagis, FS - F-skoor, TR - trepp, KU - kummardamine, I - istumine, SE - seismine, KÕ - kõndimine.

Mittestatsionaarse infoga mudeli ennustamise tulemustest on näha, et võrreldes eelmiste mudelitega, on istumise ja seismise täpsused langenud. See on peamiselt sellepärast, et mudel ennustab kõndimist seismiseks. Seda põhjustab peamiselt pildiseerias olevate piltide vähesus ja ka pildiseerias toimuv vähene liikumine. Jooniselt 14 on näha erinevusi seismisel ja kõndimisel. Kõndimine ja seismine on mittestatsionaarse info poolest väga sarnased, kuna keha kõigub ja tekitab sarnaseid liigutusi nagu kõndimisel. See probleem sõltub palju kaamera nurgast ja kui pikk pildiseeria on. Kaamera nurgast sõltub, kui hästi võtab kaamera kõndimist kaadrist, see tähendab, et kui kaamera pildi pealt on raske eristada kõndimist, siis on ka keeruline eraldada mittestatsionaarset infot, sest seda jääb vähe alles.



Joonis 14: Mittestatsionaarse infoga õigesti märgendatud pildid. Vasakul seismise pilt ning paremal kõndimise pilt.

K lähima naabri algoritmil on peale istumise ja seismise langenud ka kummardamise märgendi täpsus. (Tabel 9) See on selle tõttu, et mudel ennustab istumist ja seismist ka kummardamisena. Mittestatsionaarne info on nende 3 märgendi puhul väga sarnased ja kuna algoritm otsib ennustamise ajal mittestatsionaarse info põhjal lähimat naabrit, siis ei ole märgendid väga täpsed.



Tegelik	Tuvastatud märgend						Skoorid		
	TR	KU	I	SE	KÕ	$\Sigma$	T	S	FS
TR	10	3	2	2	0	17	1.0	0.59	0.74
KU	0	90	36	16	0	142	0.27	0.63	0.38
I	0	103	70	2	0	175	0.43	0.4	0.41
SE	0	106	47	84	0	237	0.59	0.35	0.44
KÕ	0	32	8	38	6	84	1.0	0.07	0.13
$\Sigma$	10	334	163	142	6	655	Täpsus: 39.7%		

Tabel 9: Tüüp 2 täpsused K lähima naabri algoritmiga, kasutatud on parameetreid tabelist 3. Tähsed: T - täpsus, S - saagis, FS - F-skoor, TR - trepp, KU - kummardamine, I - istumine, SE - seismine, KÕ - kõndimine.

Jooniselt 15 on näha, et kuna suurt liikumist ei toimu, siis saab ka vähe informatsiooni liikumise kohta, selle tõttu ei saa täpselt eristada enamust seisvaid märgendeid.



Joonis 15: Vasakul on mudeli poolt istumisena tuvastatud pilt, keskel on seismisena tuvastatud pilt ja paremal on kummardamisena tuvastatud pilt. Kõik pildid on tegelikult kummardamise seeriast.

### 6.2.3 Tüüp 3 - Statsionaarne info pildiseeriast

Statsionaarse infoga töödeldakse pildiseeriast välja informatsioon paigal püsimisest. See on sarnane pildilt otse info eraldamisele, ainult statsionaarse info puhul võetakse arvesse terve pildiseeria pildid ja luuakse uus pilt, mida kasutada mudelis. Probleemid on sarnased tüüp 1 mudeliga, kuna mittestatsionaarne info eemaldatakse pildiseeriast, siis alles jäi informatsioon paigal püsimise kohta ehk liikumise kohta pole enam infot. Tabelist 10 on näha, et mudeli poolt tuvastatud kõndimise märgendid on päriselt seismise ja kõndimise märgenditega. Sellest saab järeldada, et mudel ei tuvasta täpselt märgendit, ning tuvastab kõndimist enamasti seismisena.

Tegelik	Tuvastatud märgend						Skoorid		
	TR	KU	I	SE	KÕ	$\Sigma$	T	S	FS
TR	14	0	0	3	0	17	0.93	0.82	0.87
KU	0	115	10	17	0	142	0.56	0.81	0.66
I	0	61	114	0	0	175	0.86	0.65	0.74
SE	0	18	8	207	4	237	0.71	0.87	0.78
KÕ	1	11	1	66	5	84	0.56	0.06	0.11
$\Sigma$	15	205	133	293	9	655	Täpsus: 69.5%		

Tabel 10: Tüüp 3 täpsused otsustusmetsaga, kasutatud on parameetreid tabelist 2. Tähistes: T - täpsus, S - saagis, FS - F-skoor, TR - trepp, KU - kummardamine, I - istumine, SE - seismine, KÕ - kõndimine.

Jooniselt 16 on märgatav, et kõndimise ja seismise pildid on erinevad, kõndimise märgendiga pildil on osa jalgade pikslitest puudu ja see peabki olema, sest jalad on liikumas, aga kuna tavaliselt liikumine on nii minimaalne pildiseeriatel, siis suurt erinevust ei teki ja mudel ennustabki sarnaselt sellise infoga pilte.



Joonis 16: Vasakul on mudeli poolt tuvastatud kõndimise märgend statsionaarse infoga pildil, mis on päriselt seismise märgendiga pildiseeriast eraldatud. Paremal on mudeli poolt õigesti tuvastatud kõndimise pilt.

Tabelist 11 on näha K lähima naabri klassifikaatoriga tuvastatud tulemusi. Mudelis on rohkem näiteid seismise ja kõndimise kohta, aga probleem on sama, mis joonisel 16 ja kuna K lähima naabri algoritm töötab selle järgi, et leiab lähima naabri, on problemaatiline ennustada kindlalt liikuvaid märgendeid.

Tegelik	Tuvastatud märgend						Skoorid		
	TR	KU	I	SE	KÕ	$\Sigma$	T	S	FS
TR	14	0	0	2	1	17	0.82	0.82	0.82
KU	0	116	12	14	0	142	0.47	0.82	0.59
I	0	74	101	0	0	175	0.81	0.58	0.67
SE	1	33	12	178	13	237	0.75	0.75	0.75
KÕ	2	26	0	44	12	84	0.46	0.14	0.22
$\Sigma$	17	249	125	238	26	655	Täpsus: 64.3%		

Tabel 11: Tüüp 3 täpsused K lähima naabri algoritmiga, kasutatud on parameetreid tabelist 3. Tähistes: T - täpsus, S - saagis, FS - F-skoor, TR - trepp, KU - kummardamine, I - istumine, SE - seismine, KÕ - kõndimine.

#### 6.2.4 Tüüp 4 - Statsionaarne ja mittestatsionaarne info pildiseeriast

Statsionaarse ja mittestatsionaarse infoga kombineerimisel treenitakse mudel koos liikuva pildiseeria ja paigal seisva pildiseeria informatsiooniga. Sisendandmeteks on statsionaarse infoga pilt ja mittestatsionaarse infoga pilt. Tabelitest 12 ja 13 on näha, et liikuvaid märgendeid ennustatakse paremini kui eelmiste mudelitega. See on selle tõttu, et liikuv ja paigalseisev informatsioon on kokku kombineeritud, mis annab kindlama tegevuse tuvastamise liikuvate märgendite osas.

Tegelik	Tuvastatud märgend						Skoorid		
	TR	KU	I	SE	KÕ	$\sum$	T	S	FS
TR	14	0	0	3	0	17	0.93	0.82	0.87
KU	0	116	8	18	0	142	0.55	0.82	0.66
I	0	65	110	0	0	175	0.87	0.63	0.73
SE	0	20	7	208	2	237	0.71	0.88	0.79
KÕ	1	10	1	63	9	84	0.82	0.11	0.19
$\sum$	15	211	126	292	11	655	Täpsus: 69.8%		

Tabel 12: Tüüp 4 täpsused otsustusmetsaga, kasutatud on parameetreid tabelist 2. Tähistes: T - täpsus, S - saagis, FS - F-skoor

Tegelik	Tuvastatud märgend						Skoorid		
	TR	KU	I	SE	KÕ	$\sum$	T	S	FS
TR	13	0	1	3	0	17	1.0	0.76	0.87
KU	0	106	18	18	0	142	0.48	0.75	0.58
I	0	62	113	0	0	175	0.75	0.65	0.69
SE	0	36	15	185	1	237	0.72	0.78	0.75
KÕ	0	18	4	52	10	84	0.91	0.12	0.21
$\sum$	13	222	151	258	11	655	Täpsus: 65.2%		

Tabel 13: Tüüp 4 täpsused K lähima naabri algoritmiga, kasutatud on parameetreid tabelist 3. Tähistes: T - täpsus, S - saagis, FS - F-skoor, TR - trepp, KU - kummardamine, I - istumine, SE - seismine, KÕ - kõndimine.

Joonisel 9 on näha probleemi andmestikus. Täpsemini, pildiseeriad on lühikesed ja siluettidelt on keeruline eristada liikuvaid osasid. Seda saaks osaliselt parandada pikemate pildiseeriatega. Lisaks saaks andmete kvaliteeti muuta, kui tõsta kaamera allapoole nii, et ta vaatevälja suund oleks paralleelne maapinnaga. Nii saab paremini eristada erinevaid siluetti osasid.



Joonis 17: Õigesti märgendatud kõndimise näide. Vasakul on statsionaarse infoga pilt ja paremal on mittestatsionaarse infoga pilt.

## 7 Järeldused

Analüüsi tulemustest on näha, et järeldada saab tegevuse tuvastamise osas palju. Erinevat tüüpidest saab välja tuua järgmist:

1. Pildilt otsene informatsioon - liikumise märgendeid suudeti tuvastada ainult siis, kui on kergemini silmapaistev liikumise faktor olemas, nagu näiteks jalutamisel on üks jalg eespool kui teine.
2. Statsionaarne informatsioon - liikumise märgendeid on keerukas tuvastada, kuna olemas on ainult paigal püsimise kohta informatsioon.
3. Mittestatsionaarne informatsioon - liikumise märgendeid on kergem tuvastada, kuid paigalpüsimisega seotud märgendid on keeruline tuvastada, kuna mittestatsionaarse infoga on enamuse paigalseismise informatsioon väga sarnane teiste andmetega.
4. Mittestatsionaarne ja statsionaarne informatsioon - kombineerides mõlemad andmed saab olla kindlam informatsiooni osas, kuna märgendi tuvastamisel saab teada, kui palju on pildiseeria jooksul siluett liikunud ja mis piirkonnad paigal püsivad.

Analüüsides tuli välja mitu probleemi:

1. Kõndimine ja seismine on väga sarnased - see on tingitud mitmest põhjusest, peamiselt silueti väiksest liikumisest ja pildiseeria väiksest ajavahemikust.
2. Andmeid oli vähe - märgendamine toimus antud töö autori poolt, selle tõttu on ka andmeid vähe.
3. Paljude pildiseeriade käsitsi märgendamises on raske eristada, mis tegevus reaalselt pildi peal toimub ilma eelneva informatsioonita.
4. Märgendite andmestikude osakaal oli ebauhtlane - mõnel märgendil oli kahekohaline arv testandmeid, mõnel kolmekohaline.
5. Pildiseeriad olid lühikesed - osades pildiseeriates oli liiga vähe informatsiooni, et eraldada tunnuseid. Selle tõttu ka tuvastati osa märgendeid valesti.
6. Ühes pildiseerias võib toimuda mitu tegevust - näiteks kõndimise seerias tekib vahepeal silueti selline pilt, kus siluett on seismas. See on peamiselt probleemiks siis, kui andmed võetakse tervest seeriast. Eraldi piltide märgendamisel seda probleemi ei tekiks.
7. Kaamera suunatud nurk oli ebaefektiivne mõnes olukorras - kaamera kaudu eraldatud piltide seeriast oli vahepeal arusaamatu, mis tegevus toimus pildiseerias (ka inimsilmale), kuna kaamera nurk oli kehv ja suutis muuta liikuva märgendi mudelile arusaamatuks.

Probleemidest sõltumata, oli pildiseeriast bakalaureusetöös kasu, kuna nende abil sai eraldada tähtsat informatsiooni, mis aitas kaasa liikumisega seotud märgendite tuvastamisel. Arvestada tuleb ka, et SPHERE projektis on kirjeldatud kakskümmend erinevat tegevuse märgendit (Twomey jt, 2016), millest bakalaureusetöös kaeti 5. Selle töö tulemustest on näha, et treenides mudelit piisavate näidetega, on reaalne saavutada rahuldava tuvastamisega tulemusi ka kõigi 20ne märgendiga. Väga head tulemust garanteerida ei saa, kuna osad märgendid on väga sarnased ning nendega on vaja rohkem vaeva näha mudeli treenimisel. Tegevusi on võimalik tuvastada, aga see sõltub palju sellest, mis andmed on kasutusel ja kui palju ressursi on andmete märgendamiseks. Tegevusi saab ka tuvastada otse piltide pealt, aga kindlamad vastused saab, kui kasutada pildiseeriaid.

## 8 Kokkuvõte

Antud töö raames avastati meetodeid, millega tuvastada silueti tegevust etteantud pildiseeriade põhjal. Märghendamise kavandamise ja teostamise käigus sai valmis märghendatud andmestik. Märghendamise abistamiseks loodi rakendus, mis lihtsustas märghendamist. Töös toodi välja 5 märghendit (trepp, kummardamine, istumine, seismine, kõndimine), mida hiljem masinõppe mudelitega kasutati.

Ekspimenteerimise käigus võrreldi mudeli tulemuslikkust statsionaarse ja mittestatsionaarse lähenemise viisi puhul. Mõlemad viisid prooviti läbi otsustusmetsa ja K lähima naabri algoritmiga. Lisaks võrreldi, mis tulemusi saavutatakse, kui märghendada pilte otse pildiseeriast antud märghendiga, võrreldes pildiseeriast tunnuste eraldamisega. Töös toodi välja, et masinõppe mudel, mis on treenitud ja testitud üksikute piltidega, ei saanud hästi hakkama piltidega, millelt ei ole võimalik tuvastada kindla faktori (nagu näiteks kõndimisel jala ette tõstmine) järgi tegevust. Lahendusena eraldati tervest pildiseeriast mittestatsionaarsed ja statsionaarsed tunnused. Nende tunnuste kombineerimisel saab kindlam olla tulemustes, kuna pildiseeria jooksul saab teada, kui palju siluett on liikunud ja mis silueti osad paigal püsivad.

Töös kirjeldati ka probleeme, mis esinesid tunnuste eraldamisega. Analüüsis toodi välja, et andmeid oli vähe, kuna märghendamine toimus käsitsi antud töö autori poolt ning paljudes seeriates oli keeruline tuvastada, mis tegevus realselt seerias toimub. Andmete vähesuse tõttu oli andmestike osakaal ebahühtlane, kuna mõnel tegevusel oli rohkem näiteid kui teisel. Lisaks andmete vähesusele olid pildiseeriad osaliselt lühikesed, mille tõttu ei olnud piltide peal piisavalt informatsiooni tunnuste eraldamiseks.

Mudeli täpsus jäi 70% lähedale, mis ei ole kõige parem. Tulevikus võib täpsus tõusta, kui luua pikemad pildiseeriad ning suuremad andmestikud. Lisaks sellele võib sisse tuua rohkem märghendeid täpsemate tegevuste tuvastamiseks ning proovida ka närvivõrke.

## Viited

- ASUS. (2012). Xtion pro. Vaadatud 05.05.2018 allikast [https://www.asus.com/3D-Sensor/Xtion\\_PRO/](https://www.asus.com/3D-Sensor/Xtion_PRO/). (Viidatud lehel 7)
- Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*. (Viidatud leheküljed 14, 18).
- Dea, C., Heckler, M., Grunwald, G., Pereda, J. ja Phillips, S. (2014). *JavaFX 8: Introduction by Example* (2nd). Berkely, CA, USA: Apress. (Viidatud lehel 9).
- EPSRC. (2018). The Engineering and Physical Sciences Research Council. Vaadatud 13.05.2018 allikast <https://epsrc.ukri.org/>. (Viidatud lehel 7)
- Hendel, Z. (2012). OpenNI. Vaadatud 05.05.2018 allikast <https://github.com/OpenNI/OpenNI>. (Viidatud lehel 7)
- Kambla, C. (2018). Bachelor-source-code. Vaadatud 14.05.2018 allikast <https://github.com/CardoKambla/bachelor-source-code>. (Viidatud lehel 34)
- Kambla, C. (2018). Tagger. Vaadatud 21.04.2018 allikast <https://github.com/CardoKambla/tagger>. (Viidatud lehel 9)
- Liaw, A., Wiener, M. jt. (2002). Classification and regression by RandomForest. *R news*, 2(3), 18–22. (Viidatud lehel 18).
- McKinney, W. (2010). Data Structures for Statistical Computing in Python, Proceedings of the 9th Python in Science Conference. AQR Capital Management, LLC. (Viidatud lehel 18).
- Oliphant, T. E. (2006). A guide to NumPy. USA: Trelgol Publishing. (Viidatud leheküljed 14, 18).
- OpenPose. (2018). OpenPose: Real-time multi-person keypoint detection library for body, face, and hands estimation. Vaadatud 19.04.2018 allikast <https://github.com/CMU-Perceptual-Computing-Lab/openpose>. (Viidatud lehel 14)
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... ja Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830. (Viidatud leheküljed 18, 19).
- Peterson, L. E. (2009). K-nearest neighbor. *Scholarpedia*, 4(2), 1883. (Viidatud lehel 19).
- Python Software Foundation. (u). Python Language Reference. Vaadatud allikast <https://www.python.org/>. (Viidatud leheküljed 14, 18)
- Singh, M., Basu, A. ja Mandal, M. K. (2008). Human Activity Recognition Based on Silhouette Directionality. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(9), 1280–1292. doi:10.1109/TCSVT.2008.928888. (Viidatud lehel 6)
- Twomey, N., Diethe, T., Kull, M., Song, H., Camplani, M., Hannuna, S., ... ja Craddock, I. (2016). The SPHERE Challenge, Activity Recognition with Multimodal Sensor Data. Vaadatud 20.04.2018 allikast <https://arxiv.org/pdf/1603.00797v2.pdf>. (Viidatud leheküljed 5, 7, 12, 30)



Vishwakarma, D. ja Kapoor, R. (2015). Hybrid classifier based human activity recognition using the silhouette and cells. *Expert Systems with Applications*, 42(20), 6957–6965. doi:<https://doi.org/10.1016/j.eswa.2015.04.039>. (Viidatud lehel 6)

# Lisa

## I. Lähtekood

Lähtekood on saadaval *GitHub*'is (Kambla, 2018a).

## II. Litsents

### **Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks**

Mina, **Cardo Kambla**,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose  
**Silueti tegevuse tuvastamine kodukeskkonnas**  
mille juhendaja on Meelis Kull
  - 1.1 reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
  - 1.2 üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, 14.05.2018