

**KAUNO TECHNOLOGIJOS UNIVERSITETAS
ELEKTROS IR ELEKTRONIKOS FAKULTETAS**

Mindaugas Vaičiūnas

ŽMOGAUS VEIKSMŲ APTIKIMO 3D DUOMENYSE TYRIMAS

Baigiamasis magistro projektas

Vadovas

lekt. dr. Kęstas Rimkus

KAUNAS, 2017

**KAUNO TECHNOLOGIJOS UNIVERSITETAS
ELEKTROS IR ELEKTRONIKOS FAKULTETAS**

ŽMOGAUS VEIKSMŲ APTIKIMO 3D DUOMENYSE TYRIMAS

Baigiamasis magistro projektas
Valdymo technologijos (kodas 621H66001)

Vadovas

(parašas) lekt. dr. Kęstas Rimkus

(data)

Recenzentas

(parašas) doc. prof. dr. Arūnas Lipnickas

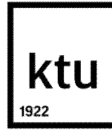
(data)

Projektą atliko

(parašas) Mindaugas Vaičiūnas

(data)

KAUNAS, 2017



KAUNO TECHNOLOGIJOS UNIVERSITETAS
ELEKTROS IR ELEKTRONIKOS

(Fakultetas)

Mindaugas Vaičiūnas

(Studento vardas, pavardė)

Valdymo technologijos, 621H66001

(Studijų programos pavadinimas, kodas)

„Žmogaus veiksmų aptikimo 3D duomenyse tyrimas“

AKADEMINIO SAŽININGUMO DEKLARACIJA

20 17 m. Gegužės 23 d.
Kaunas

Patvirtinu, kad mano, **Mindaugo Vaičiūno**, baigiamasis projektas tema „Žmogaus veiksmų aptikimo 3D duomenyse tyrimas“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

(vardą ir pavardę įrašyti ranka)

(parašas)

Vaičiūnas, Mindaugas. Žmogaus veiksmų aptikimo 3D duomenyse tyrimas. *Valdymo sistemų Magistro* baigiamasis projektas / vadovas Kęstas Rimkus; Kauno technologijos universitetas, Elektros ir Elektronikos fakultetas

Mokslo kryptis ir sritis: Elektros ir Elektronikos Inžinerija, Technologiniai Mokslai.

Reikšminiai žodžiai: veiksmų aptikimas, klasifikavimas, *TreeBagger*, Neuroniniai tinklai, *LibSVM*, *Microsoft Kinect*.

Kaunas, 2017. 53 p.

SANTRAUKA

Darbe apžvelgiama literatūra nagrinėjanti veiksmų atpažinimą, kai naudojami skirtingų dimensijų duomenys (2, 3, 4 ir n dimensijų) bei skirtingi metodai požymių vektorių sudarymui.

Atlikus literatūros analizę pasirinkta didelės apimties duomenų bazė *Human3.6M*, su jau išskirtais pozų požymių vektoriais. Šiems duomenims klasifikuoti pasirinkti literatūroje apžvelgti *TreeBagger*, Neuroninių tinklų ir *LibSVM* klasifikatoriai.

Darbe duomenų klasifikavimui pritaikomos strategijos leidžiančios manipuluoti didesniais duomenų kiekiais. Tokios, kaip didelės apimties duomenų klasifikavimas, balsavimas, klasifikavimo modelių taisyklių apjungimas, vieno veiksmo išskyrimas iš visų likusių, atpažįstamų veiksmų sumažinimas.

Paskutiniame etape atliekamas duomenų grupavimas, tam panaudojant didžiausią bendrąjį visų klasių atpažinimo tikslumą pasiekusį modelį. Šiuo atveju apdorojami jau atpažinti duomenys, jie grupuojami į tam tikras sekas, kiekvienos sekos metu nusprendžiant, kurios klasės atpažinimo tikimybė pasirinktoje sekoje didžiausia. Pasiektas 82,63 % bendras 15 klasių atpažinimo tikslumas, kurių duomenys nebuvo naudojami apmokymo procese.

Vaičiūnas, Mindaugas. *Research of 3D data based human event detection: Master's thesis in Control Systems* / supervisor Kęstas Rimkus. The Faculty of Electrical and Electronics Engineering, Kaunas University of Technology.

Research area and field: Electrical and Electronic Engineering, Technological Sciences

Key words: event detection, classification, *TreeBagger*, *Neural Network*, *LibSVM*, *Microsoft Kinect*.

Kaunas, 2017. 53 p.

SUMMARY

In this paper reviewing literature that deals with action recognition using various dimensional data (2, 3, 4, and n-dimensional) and different methods for forming feature vectors.

Selected large-scale Human3.6M database with excluded postures feature vectors after literary analysis. For these data classification chosen *TreeBagger*, *Neural network* and *LibSVM* classifiers.

To allow manipulation of large amounts of data adapted strategy such as large-scale data classification, vote, bundling rules of classification models, one class isolation from all the rest, reduction of recognizable actions.

In the last stage for grouping data chosen model with the highest accuracy in overall recognition of all classes. In this case working with classified data, these data are grouped into certain sequences, deciding which class detection probability is the highest for each sequence. Reached 82,63 % overall recognition accuracy of 15 classes for which data were not used in the training process.

Turinys

PAVEIKSLŲ SĄRAŠAS.....	7
LENTELIŲ SĄRAŠAS	8
TERMINAI IR AKRONIMAI	9
IVADAS.....	10
1. Literatūros apžvalga	11
1.1. Žmogaus veiksmų atpažinimas panaudojant 2D duomenis	13
1.2. Žmogaus veiksmų atpažinimas panaudojant 3D, 4D duomenis.....	15
2. Teorinė dalis	24
2.1. Human3.6M duomenų bazė	24
2.2. Klasifikatoriai.....	27
2.2.1. TreeBagger	27
2.2.2. Neuroniniai tinklai	28
2.2.3. LibSVM	29
3. Tiriamoji dalis.....	31
3.1. Klasifikavimas naudojant pozos požymių vektorių.....	31
3.1.1. Klasifikatorių parametrų, požymių vektoriaus dydžio parinkimas	31
3.1.2. Klasifikavimo strategijos	33
3.1.2.1. Dideli duomenų kiekiai.....	33
3.1.2.2. Apmokytų modelių sujungimas.....	35
3.1.2.3. Balsavimas	37
3.1.2.4. Vienas su visais	39
3.1.2.5. Atpažįstamų veiksmų mažinimas.....	40
3.2. Klasifikavimas naudojant pozos požymių vektorių kadrų seką	41
3.2.1. Klasifikavimo tikslumo priklausomybė nuo kadrų skaičiaus	41
3.2.2. Balsavimas	43
3.2.3. Keturių klasių atpažinimas	45
3.2.4. Suklasifikuotų duomenų grupavimas	46
REZULTATAI IR IŠVADOS.....	49
INFORMACIJOS ŠALTINIŲ SĄRAŠAS	50

PAVEIKSLŲ SĄRAŠAS

1.1 pav. Kinect veikimo principas [1].....	12
1.2 pav. Kinect panaudojimas [1]	13
1.3 pav. Algoritme naudoti skeleto taškai pažymėti žaliai a) naudoti 7 taškai, b) 11 taškų, c) 15 taškų d) visi 20 taškų, raudonai pažymėti taškai buvo nenaudojami sudarinėjant požymių vektorių.....	16
1.4 pav. (a) Skeleto taškai atvaizduoji HOJ3D plokštumoje. (b) Modifikuota sferinė koordinatinių sistema taškų vietoms suskirstyti [7].....	18
1.5 pav. Informatyviausi skeletų taškai ir atitinkamų taškų judesiai atliekant tam tikrus veiksmus Teniso padavimas, (b) Boksavimasis, (c) Bėgiojimas, (d) Pakėlimas ir metimas (e) Horizontalus rankos mojavimas [8]	19
1.6 pav. Plokštumos vektorių vizualizacija a) fonas b) fonas + objektas gylio vaizde [24]	22
1.7 pav. a) Silueto suskirstymas į keturis ketvirčius b) Veiksmo aptikimo hierarchija pozų apraše [24]	22
2.1 pav. Human3.6M duomenų bazės veiksmų rinkinys [26].....	25
2.2 pav. Požymių vektoriaus sudarymo paaiškinimas, nuotraukos suskirstymas lygmenimis, erdvės padalijimas į dalis kas 40 laipsnių	26
2.3 pav. Dirbtinio neuroninio tinklo struktūra [30]	28
3.1 pav. Priklausomybė tarp TreeBagger klasifikatoriaus medžių skaičiaus ir atpažinimo tikslumo	31
3.2 pav. Priklausomybė tarp NNet klasifikatoriaus neuronų skaičiaus ir atpažinimo tikslumo ...	32
3.3 pav. Priklausomybė tarp požymio vektoriaus dydžio ir atpažinimo tikslumo naudojant TreeBagger ir NNet klasifikatorius.....	33
3.4 pav. TreeBagger klasifikatoriaus testavimo sumaišymo matrica, kai apmokymui naudoti pirmojo subjekto duomenys, testuojama su 25 % visų duomenų (iš visų subjektų).....	34
3.5 pav. NNet klasifikatoriaus testavimo sumaišymo matrica, kai apmokymui naudoti pirmojo subjekto duomenys, testuojama su 25 % visų duomenų (iš visų subjektų).....	35
3.6 pav. Modelio Nr. 2 sumaišymo matrica	36
3.7 pav. Modelio Nr. 5 sumaišymo matrica	37
3.8 pav. Balsavimo modelio sumaišymo matrica	39
3.9 pav. Klasifikavimo tikslumo priklausomybė nuo kadru skaičiaus	41
3.10 pav. TreeBagger765 modelio sumaišymo matrica, kai kadru sk. – 35	42
3.11 pav. NNet765 modelio sumaišymo matrica, kai kadru sk. – 20.....	42
3.12 pav. Balsavimo strategijos algoritmas	43
3.13 pav. Modelio Nr. 2 sumaišymo matrica	44
3.14 pav. Balsavimo modelio sumaišymo matrica.....	45
3.15 pav. Duomenų grupavimo algoritmas.....	47
3.16 pav. Tikslumo priklausomybė nuo duomenų grupelės dydžio	47
3.17 pav. Grupavimo sumaišymo matrica, kai grupelės dydis 205.....	48

LENTELIŲ SĄRAŠAS

<i>1.1 Lentelė. Nagrinėtų vienmačių algoritmų apžvalga</i>	14
<i>1.2 Lentelė. Laiko išnaudojimas vieno kadro apdorojimui (laikas pateikiamas milisekundėmis) [6]</i>	17
<i>2.1 Lentelė. LibSVM klasifikatoriaus apmokymui, testavimui byloje naudojama duomenų struktūra</i>	30
<i>3.1 Lentelė. Apmokytų bei apjungtų modelių atpažinimo tikslumai</i>	36
<i>3.2 Lentelė. Modelių sudarytų naudojant pozų požymių vektorius naudotų balsavimui atpažinimo tikslumai</i>	38
<i>3.3 Lentelė. Vieno veiksmo atpažinimas lyginant su visais likusiais veiksmais</i>	39
<i>3.4 Lentelė. 4 klasių atpažinimas naudojant pozų požymių vektorius</i>	40
<i>3.5 Lentelė. Modelių sudarytų naudojant kadru sekas naudotų balsavimui atpažinimo tikslumai</i>	44
<i>3.6 Lentelė. 4 klasių atpažinimas naudojant kadru sekas</i>	46

TERMINAI IR AKRONIMAI

- 2D, 3D, 4D – plokštumą, trimatę arba keturmatę erdvę nusakantis žymėjimas (angl. *two/three/four dimensions*);
- RGB–D – trijų kanalų spalvų erdvės naudojamos vaizdų kodavimui gylio modelis (angl. *Red Green Blue color depth model*);
- HOJ3D – Trimačių taškų histogramos (angl. *Histograms of 3D Joints*);
- LCSS – Ilgiausia bendra seka (angl. *Longest common subsequence*);
- PMM – Paslėptieji Markovo modeliai (angl. *Hidden Markov model – HMM*);
- GMM – Gauso pasiskirstymo modelis (angl. *Gaussian mixture model*);
- Mocap – Judesio fiksavimo sistema (angl. *Motion capture system*);
- SVM – Atraminių vektorių klasifikavimo modelis (angl. *Support Vector Machine*);
- RAM – operatyvioji atmintis (angl. *Random-access memory*);
- TreeBagger – Medžių pakavimo klasifikavimo modelis;
- NNet – Neuroninių tinklų klasifikavimo modelis (angl. *Neural Network*);
- LibSVM – SVM klasifikavimo modelio modifikacija;
- GHz – gigahercas (angl. *Gigahertz*);
- GB – gigabaitas (angl. *Gigabyte*);
- SSD – puslaidininkinis diskas (angl. *Solid-state drive*).

IVADAS

Pastaraisiais metais yra sukurta daug būdų kaip įgyvendinti žmogaus veiksmų atpažinimą. Dažniausiai tai sudėtingi kompleksiniai uždaviniai, prasidedantys nuo tokių elementarių dalykų kaip žmogaus silueto išskyrimas nuotraukoje, triukšmų pašalinimas geresniam atpažinimui, pereinant tokius etapus, kaip žmogaus atkūrimas trimatėje erdvėje, reikšmingų taškų atpažinimas (galva, koja, ranka, pečiai, alkūnės ir pan.), šiuos taškus sujungiant linijomis, tokiu būdu sukuriamas ir išskiriamas žmogaus skeletas. Baigiamajame etape stengiamasi aptikti įprastus žmonių veiksmus: sėdėjimas,ėjimas, gulėjimas, mojavimas ir daugelį kitų. Išanalizavus bei išskyrus įprastus žmonių veiksmus galima pereiti prie neįprastų reiškinių, tokių, kurie nėra natūraliai būdingi žmogaus elgsenai, pvz., kai kurie stengiasi atpažinti būsimus žmogaus kritimus, taip norėdami prognozuoti žmogaus elgseną ir apsaugoti juos nuo patiriamų sužalojimų ateityje. Kita tyrimų sritis yra senjorai. Kadangi mūsų visuomenėje jaunesni žmonės tampa vis labiau užimti, o noras ir galimybės prižiūrėti senyvus žmones blėsta arba nėra tam galimybių, tai skatina kurti tokias atpažinimo sistemas, kurios atpažindamos kasdienę ar natūralią jų veiklą, nustatydamos žmogaus patogumo, komfortiškumo laipsnį ir tirdamos neįprastą jų veiklą, gebėtų juos apsaugoti, pagerinti jų gyvenimo kokybę. Vaizdo analizė gali pagelbėti ir tokiais atvejais, kaip vagystės, užpuolimai, patekimai į teritoriją be bilieta ar net pabėgimai iš kalėjimų, psichiatrijos ligoninių ir pan., šiuos neleistinus veiksmus stengiamasi atpažinti ir traktuoti, kaip neįprastus, o atpažinus greičiau informuoti budintį personalą. Išplėtojus sistemą į didesnę regioną, pvz. miesto tinklą, neleidžiamą veiksmą atlikusį asmenį galima sekti, tokiu būdu sumažinama tikimybė iki minimalios, kad sekamas asmuo pabėgs. Dar viena iš galimų vaizdo analizės taikymo sričių – medicina:

- Reabilitacija – žmonių, atliekančių potrauminius reabilitacijos pratimus stebėjimas bei informavimas apie neteisingai atliktą veiksmą;
- Virtualus treneris – pratimo atvaizdavimas ekrane, asmens darančio pratimą sekimas bei korekcinį veiksmų nurodymas, pratimą atliekant netaisyklingai;
- Pacientų stebėjimas – sunkios būklės pacientų, nepajėgių išsikviesti pagalbą į palatą ir įvykus nelaimėi (pasijutus blogai, stojant širdžiai, negalint kvėpuoti ir pan.) atpažinimas ir prižiūrinčiojo personalo informavimas.

Šiuos veiksmus traktuojant, kaip neįprastus galima pagerinti žmonių reabilitacijos procesą, sumažinti traumų tikimybę sporte ar išgelbėti sunkios būklės pacientų gyvybes, laiku informavus prižiūrintįjį personalą apie nelaimę.

1. Literatūros apžvalga

Literatūroje gausu pavyzdžių kaip atpažinti įprastus ir neįprastus žmogaus veiksmus, tam panaudojant vienmačius (2D – dviejų dimensijų), trimačius (3D – trijų dimensijų), keturmačius (4D – keturių dimensijų) ar n–mačius vaizdus. Pagrindinės problemos kuriant algoritmus vaizdo analizėje:

- Techninės galimybės atliekant skaičiavimus (t. y. nors ir sparčiais šuoliais vyksta technologijų raida, tačiau vaizdų apdorojimas išlieka ganėtinai sudėtingas uždavinys, nes jam reikia nemažai techninių resursų, tai kyla dėl tokių dalykų kaip n–mačių matricų skaičiavimai, duomenų dideliais kiekiais apdorojimas per ganėtinai trumpus, milisekundžių eilės laiko tarpus);
- Greitaveika (dažniausiai norima, jog duomenys būtų apdorojami realiu laiku, filmuojant apie 30 kadrų per sekundę);
- Didelė duomenų bazė apmokymams (dažniausiai norint gauti geresnių rezultatų baigtiniame produkte, jis turi būti apmokytas su skirtingais duomenimis, skirtingais žmonėmis, skirtingais veiksmais esant skirtingoms aplinkos sąlygoms ir kitiems trikdžiams);
- Tikslumas (stengiamasi pasiekti kuo didesnę teisingai atpažintų veiksmų skaičių).

Dažniausiai norima pasiekti kuo didesnę greitaveiką, užtikrinant kuo didesnę tikslumą. Kartais klaidingas sistemos veikimas gali būti nereikšmingas, nesukeliantis nei didelės grėsmės, nei pavojaus, pvz. žaidimo metu neatpažintas vienas ar kitas veiksmas, tačiau kartais klaidos negali būti toleruojamos, pvz. medicinoje, kai klaidingas sistemos veikimas lemia žmogaus mirtį ar teisėsaugoje, kai sukeliama grėsmė aplinkiniams bei visuomenei.

Atpažinimas gali būti vykdomas panaudojant bet kurią vaizdo kamerą, taip gaunant 2D vaizdą. Dažniausiai dviejų dimensijų vaizdas naudojamas elementariems veiksmais atpažinti: bėgimui, važiavimui dviračiu ar mašina. Tokio atpažinimo procese dažnai naudojami judėjimo vektoriai, kurie gaunami lyginant du kadrus, dabartinį su prieš tai buvusiu. Lyginant informaciją tarp kadrų atkreipiamas dėmesys į kurią pusę ir kiek pasislinko kiekvienas taškas (pikselis), tokiu būdu sudarant krypties vektorių [10, 11, 18, 21, 24]. Norint padidinti tam tikrų žmogaus veiksmų atpažinimo tikslumą, reiktų naudoti bent jau trimatę erdvę, kurioje galima būtų išskirti mums įprastą trimatį žmogaus vaizdą, jam priskiriant skeletą.

Šiam tikslui galima naudoti *Mocap* (angl. *Motion capture system*) algoritmą, nutolusius jutiklius, stereoviziją (t. y. turi būti panaudotos bent 2 kameros nutolusios viena nuo kitos tam tikru atstumu vienoje plokštumoje), struktūrizuotos šviesos technologiją (analogiškai kaip ir stereovizijoje naudojamos 2 kameros, tačiau viena kamera siunčia infraraudonuosius lazerio spindulius, kita priima jau atsispindėjusius nuo paviršiaus). Tiek stereovizijoje, tiek

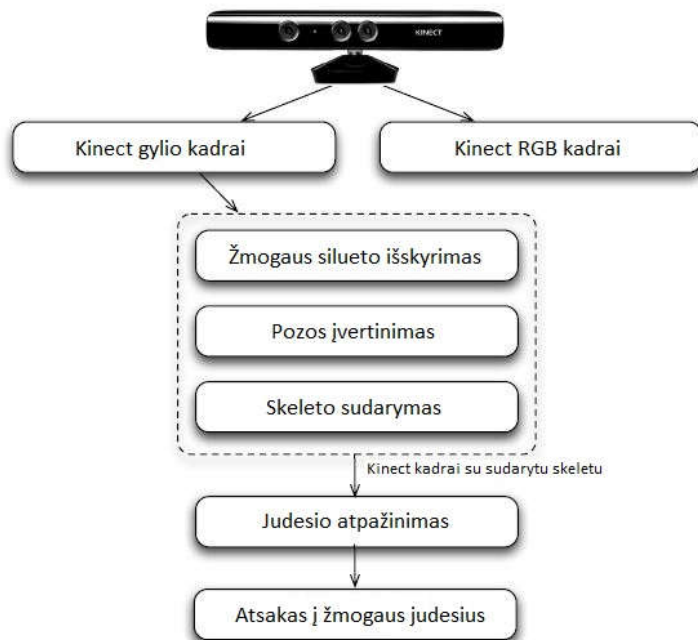
struktūrizuotos šviesos technologijoje naudojant trigonometriją išskaičiuojami atstumai iki kiekvieno pikselio, gaunama gylio (angl. *depth*) matrica, kur kiekvienas matricos narys reiškia atstumą nuo kamerų iki tam tikro taško erdvėje [2]. Dažnai tai vadinama ne 3D vaizdu, bet 2.5D vaizdu, nes turimas gylio matricos vaizdas teikia informaciją tik apie vieną scenos pusę.

Dažniausiai susiduriama su keturiais sunkumais naudojant veiksmų aptikimo metodus [2]:

1. Netvarkingas fonas, šešėliai ir kintančios apšvietimo sąlygos.
2. Veiksmų sekimas, tie patys veiksmai gali generuoti skirtingą vaizdą iš skirtingų pozicijų.
3. Skirtingi objektų dydžiai priklausomai nuo to kiek jie nutolę nuo kameros.
4. Panašių veiksmų kitimas, objektai gali atlikti tuos pačius veiksmus skirtingomis kryptimis, skirtingais stebėjimo kampais, taip pat, tas pats judesys kiekvieną kartą gali truputį skirtis.

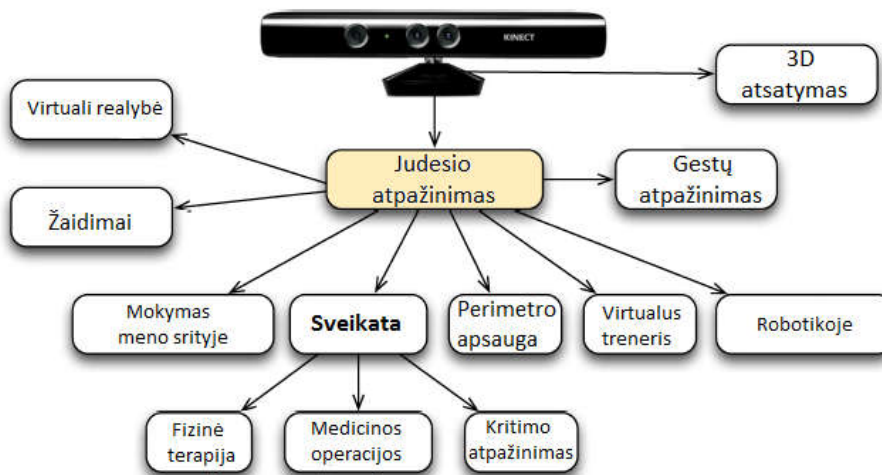
Dažnai netgi tokie dalykai kaip šviesos šaltinis, paviršiaus atspindžiai ar naudojama kamera gali apsunkinti atpažinimo situaciją.

Šiuo metu rinka gali pasiūlyti gaminius atliekančius stereoviziją, struktūrizuotos šviesos technologiją, suskaičiuojančius gylio matricą ar net išskiriančius ir sudarančius žmogaus skeletą, tokius kaip *ZED* kamera ar *Microsoft Kinect*. 1.1. paveiksle pateikiamas *Kinect* veikimo principas, naudojant įtaisą kartu su *Xbox* žaidimų konsole.



1.1 pav. *Kinect* veikimo principas [1]

Nors *Kinect* buvo kuriamas žaidimams, jis tapo patrauklus tyrinėtojams dėl ankščiau aptartų funkcijų (tai galima matyti 1.2. pav.), taip pat dėl savo nedidelės kainos bei gan tikslių algoritmų, todėl darbe bus plačiau apžvelgiami žmogaus veiksmų atpažinimo metodai panaudojant *Kinect*.



1.2 pav. *Kinect* panaudojimas [1]

Trimatį vaizdą galima gauti panaudojant judesį. Veikimo principas pagrįstas žymeklių panaudojimu, optiniai žymekliai sukabinami ant kūno, taip pat naudojamos kameros sekančios žymeklių pozicijas. Tokiu būdu yra galimybė atkurti vaizdą trimatėje erdvėje. Šis metodas labiau paplitęs fizinėse terapijose, nes didžiausias trūkumas yra tas, jog trimatėje erdvėje atkuriami tik žymeklių taškai, todėl dažniausiai jie apjungiami tiesėmis ir panaudojami skeletui sukurti.

Per pastaruosius kelis metus žmogaus veiksmams atpažinti buvo taikyti algoritmai panaudojant gylio matricas: atpažinimas iš 3D siluetų, skeleto (kūno dalių), vietos erdvės laiko modelis (4 dimensijų algoritmas apjungiant trimatę RGBD matricą su laiku), vietinio užimtumo modelis, 3D optinis srautas.

1.1. Žmogaus veiksmų atpažinimas panaudojant 2D duomenis

Apžvelgtuose [12–22] straipsniuose matyti, jog veiksmų: tiek įprastų, tiek neįprastų atpažinimu susidomėta ne taip ir seniai. Nuo 1994 iki 2004 metų buvo išleidžiama vos po kelis straipsnius per metus, o nuo 2004 metų straipsnių šia tematika pradėjo daugėti iki 6–18 straipsnių per metus [13, 14]. Tokią tendenciją galima būtų paaiškinti per pastarąją dekadą padidėjusiais techniniais skaičiavimo įrangos parametrais bei proveržiu kompiuterių inžinerijoje, kadangi vaizdų apdorojimas ganėtinai sudėtingas procesas, reikalaujantis per ganėtinai trumpą –

milisekundžių laiko tarpą atlikti nemažai skaičiavimų. Dėl norimos greitaveikos buvo kuriami, apjungiami įvairiausi algoritmai.

Vienmačiai atpažinimo algoritmai veiksmui atpažinti naudoja ribotą informacijos kiekį – vaizdo plokštumą, kurioje yra galimybė išskirti tik žmogaus siluetą, neturint informacijos apie objektą iš šonų, nugaros, viršaus. Dėl to didžioji dalis algoritmų pagrįsti optinių srautų analize [12–14, 16–22], tiriamas pokytis tarp kadru, todėl norint tiksliai aptikti ir išskirti tam tikrą veiksmą reikalinga nors kelių kadru istorija. Lyginant kadru seką išskiriami tie taškai (pikseliai), kurie keičiasi per tam tikrą laiką. Kiti, truputį mažiau paplitę metodai būtų – histogramų (kuriose išskiriama: judesys, pikselio, spalvos, tekstūros pokyčiai), formų (stovinčio žmogaus forma skiriasi nuo sėdinčio, atsitūpusio), erdvės laiko pokyčių. Labiausiai paplitęs PMM klasifikatorius, kadangi daugiausiai tyrimų 2D srityje paremti dinamika [13].

Dėl ribotos informacijos norint pasiekti geresnių rezultatų reikalingi metodai, naudojančys sudėtingus matematinius veiksmus, ir kas dažniausiai turi įtakos greitaveikai. Iš apžvelgtų metodų literatūroje šaltiniuose tik vienas [15] yra realaus laiko, metodas pagrįstas kūno dalių išskyrimu bei krypties vektorius sudarymu, priimant, kad vektorius pradžia – klubai, vektorius pabaiga – pėdos. Atpažinimo procesas vykdomas pagal vektorius ilgį su sąlyga, jeigu vektorius ilgis tampa ilgesnis nei nustatyta leistina riba, tai traktuojama kaip bėgimas, jeigu per trumpas – tupėjimas. Nors metodas ir yra realaus laiko greitaveikos, tačiau atpažįsta labai nesudėtingus veiksmus.

1.1 Lentelė. Nagrinėtų vienmačių algoritmų apžvalga

Eil. Nr.	Metodas	Sritis/ atpažįstamas objektas	Tikslumas	Greitaveika
1.	Bendros jėgos poveikio metodas (angl. <i>Social force</i>) [16]	Minios veiksmų atpažinimas	96 %	–
2.	Optinio srauto metodas (angl. <i>Pure optical flow</i>) [16]	Minios veiksmų atpažinimas	84 %	–
3.	Aukštų dažnių ir laiko erdvės požymių metodas (angl. <i>high-frequency and spatio-temporal (HFST) features</i>) [18]	Minios veiksmų atpažinimas	85 %	1 kadras per 0.258 s
4.	Skirtingų mastelių laiko erdvės požymių metodas (angl. <i>Multi scale spatio-temporal features</i>) [20]	Neįprasto veiksmo minioje aptikimas	28 %	1 kadras per 25 s
5.	Laiko erdvės intensyvumų metodas (angl. <i>spatio-temporal volumes</i>) [21]	Anomalijų aptikimas (t. y. neleistinių veiksmų)	84,9 %	1 kadras per 0.22 s
6.	Paslėptieji Markovo modeliai [19]	Įtartinų veiksmų aptikimas	91 %	–
7.	Krypties vektorius naudojimas [15]	Neįprasto veiksmo aptikimas	88 %	Realaus laiko

Apžvelgti metodai nors ir pasiekia ganėtinai aukštus atpažinimo rezultatus, tačiau atpažįstami veiksmai gana grubūs, pvz., vieno žmogaus (bėgimas, tupėjimas) ar minios judesys (panika). Šie veiksmai dažniausiai leidžia daryti prielaidas, kurios ne visada gali būti teisingos, pvz., bėganti minia ne visada gali reikšti paniką, kartais tai gali būti sportuojančių bėgikų grupelė. Arba minioje sprunkantis žmogus ne visada gali būti vagis, galbūt žmogus kažkur vėluoja.

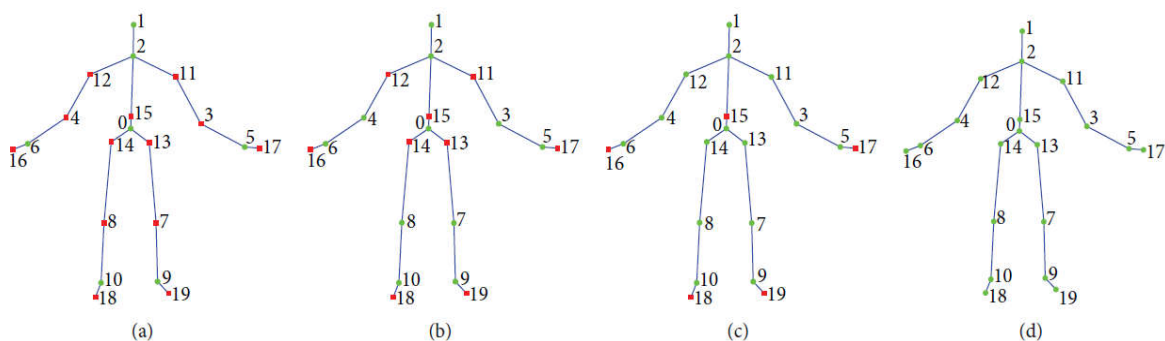
1.2. Žmogaus veiksmų atpažinimas panaudojant 3D, 4D duomenis

Per pastaruosius kelis metus žmogaus veiksmams atpažinti buvo taikyti algoritmai panaudojant gylio matricas: atpažinimas iš 3D siluetų, skeleto (kūno dalių), vietos erdvės laiko modelis (4 dimensijų algoritmas apjungiant trimatę RGBD matricą su laiku), vietinio užimtumo modelis, 3D optinis srautas.

Literatūros šaltinyje [3] pateikiamas kasdienių žmogaus veiksmų atpažinimas panaudojant Gauso pasiskirstymo modelį pagrįstą paslėptuoju Markovo modeliu. Panaudojant šių metodų apjungimą buvo siekiama didesnio tikslumo, kad robotas galėtų atpažinti neįprastus reiškinius, tokius kaip ant grindų gulintis arba per ilgai nei įprastai miegantis senyvas žmogus ir apie tai praneštų prižiūrinčiam personalui. Atpažinimas įgyvendintas panaudojant tik skeleto modelį sudarytą iš 15 taškų, kurį sugeneravo RGB-D kamera *Microsoft Kinect*. Algoritmas buvo testuojamas su *Cornell'io* įvykių duomenų baze CAD60. Autoriaus sudarytas modelis buvo apmokytas ir testuojamas dviem variantais: su duomenimis, kurie buvo naudojami apmokymų metu ir su apmokymų metu nenaudotais duomenimis. Naudojant duomenis tokius pačius kaip apmokymuose buvo pasiektas 84 % tikslumas, o atpažinimui naudojant žmones, kurie nebuvo apmokymo procese, pasiektas vos 78 % tikslumas.

Literatūros šaltinyje [4] apžvelgiamas kasdienės žmogaus veiklos atpažinimas panaudojant skeleto modelį pagrįstą kūno požymiais. Autorius kurdamas algoritmą naudojo žmogaus siluetą išgautą iš gylio matricos (tiek iš priekinio, tiek iš šoninio vaizdų). Silueto gavimas susidėjo iš triukšmų, tokių kaip fono (pvz. siena) ir nereikšmingų daiktų esančių už tiriamojo objekto panaikinimo, žmogaus aptikimo ir pačio silueto išskyrimo. Sekančiame etape išskyrė skeleto sujungimus, tam panaudodamas odos spalvą. Šis veiksmas autoriui leido gauti penkis reikšmingus taškus (1 tašką galvos, 2 taškus rankų ir 2 – kojų) ir sukurti charakteringą vektorį. Apmokymo procese parinktas SOM (angl. *Self-Organized Map*) neuroninių tinklų apmokymo algoritmas. Testavimas atliktas su autoriaus literatūros analizėje apžvelgta „*IM-DailyDepthActivity*“ duomenų baze, tai leido pastebėti, kad sukurtas algoritmas pasiekė 92,43 % tikslumą bei palyginti su kitais duomenų bazės rezultatais ir sužinoti, kad 8,97 % pagerino kitų autorių rezultatus pasiektus su šia duomenų baze.

Straipsnyje [5] apžvelgiamas žmogaus veiksmų atpažinimas naudojant *Microsoft Kinect* pagalba sukuriama žmogaus skeleto algoritmą iš RGBD duomenų. Tokiu būdu gaunamos pagrindinės žmogaus pozos pagal kurias formuojami veiksmų klasteriai (tam tikras rinkinys pozų priskiriamas vienam ar kitam veiksmui, kad neapsiriboti tik su viena poza kažkuriam vienam veiksmui). Sudarius veiksmų klasterius iš jų formuojamas požymių vektorius, kuris baigiamajame veiksmo klasifikuojamas SVM klasifikavimo algoritmu. Pagal nutylėjimą *Kinect* naudojamas algoritmas išskiria 20 skeleto taškų (sąnarių) (1.3 pav. d)), autorius tyrimams naudojo kelis skirtingus variantus (1.3 pav. a), b), c) variantai) siekdamas surasti tikslumo sąsają tarp skirtingo skeleto taškų skaičiaus, naudojamo gaunant pozų požymių vektorių.



1.3 pav. Algoritme naudoti skeleto taškai pažymėti žaliai a) naudoti 7 taškai, b) 11 taškų, c) 15 taškų d) visi 20 taškų, raudonai pažymėti taškai buvo nenaudojami sudarinėjant požymių vektorių

Autorius tyrimą atliko su 5 skirtingomis duomenų bazėmis (*Florence3D*, *MSRAction3D*, *UTKinect*, *CAD-60*, *KARD*), kurios buvo testuojamos naudojant skirtingas sąlygas, t. y. panaudojant skirtingą klasterių skaičių bei 1.3 paveiksle pateiktus skirtingus skeletų modelius. Vertinant tyrimo rezultatus, kaip nekeista, bet tyrėjas įrodė, kad daugiau ne visada geriau. Geriausi rezultatai gauti panaudojant mažiausią sujungimo taškų skaičių turinčius skeleto modelius, t. y. tuos, kurie turėjo išskirtus tik 7 ar 11 taškų. Klasterių parinkime pastebėta, kad geresni rezultatai gauti naudojant didesnę klasterių skaičių, t. y. jeigu duomenų bazė leido išskirti nuo 5 iki 35 klasterių, geriausi rezultatai gauti, kai klasterių skaičius svyravo nuo 25 iki 30 klasterių. Turint galimybę pasirinkti tarp 2 ir 8 klasterių, geriausi rezultatai buvo gaunami su 6 klasteriais. Tyrimo metu pasiekti rezultatai:

- *Florence3D* – 86,1 % tikslumas (su 11 skeleto sujungimo taškų);
- *MSR Action3D* – 81,2 % tikslumas (su 7 skeleto sujungimo taškais);
- *UTKinect* – 95,1 % tikslumas (su 7 skeleto sujungimo taškais);

- *CAD-60* – 93,9 % tikslumas (su 11 skeleto sujungimo tašku), 92,7 % tikslumas (su 7 skeleto sujungimo taškais);
- *KARD* – 95,1 % tikslumas (su 11 skeleto sujungimo tašku), 95 % tikslumas (su 7 skeleto sujungimo taškais).

Autoriaus teigimu kiekviena iš šių bazių yra ganėtinai „triukšmingos“, todėl norint pasiekti tikslesnių atpažinimo rezultatų reikėtų naudoti papildomus algoritmus triukšmų pašalinimui. Šis tyrimas parodė, kad naudojant mažesnę skeleto taškų skaičių galima gauti ne ką prastesnius rezultatus, tokiu būdu pagreitinant skaičiavimus bei patį atpažinimą.

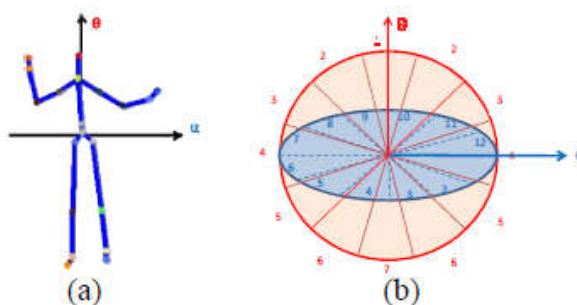
Vienas iš ganėtinai svarbių aspektų vaizdo atpažinimo sistemose galimybė dirbti realiu laiku. Vėlavimai turi būti kuo mažesni arba išvis netoleruoti, skaičiavimai turi būti atliekami kaip įmanoma greičiau, nenaudojant superkompiuterių. Pagrindinė realaus laiko atpažinimo sistemų taikymo tendencija įžvelgiama robotikoje. Taip pat, tai gali pagelbėti ir sistemose, kurios privalo informuoti apie pablogėjusią žmogaus savijautą, tokiu būdu prisidedant prie žmonių sveikatos gerinimo, gyvybės gelbėjimo. Būtent [6] straipsnio autorius atliko tyrimą siekiant gauti realaus laiko sistemą, pagrįstą *Kinect* naudojamu skeleto sudarymo algoritmu. Algoritmas pagrįstas sferinių kampų tarp skeleto modelio taškų ir atitinkamų kampinių pagreičių skaičiavimu. Metodas skiriasi nuo įprastinės praktikos tuo, kad koordinatinių sistemos atskaitos taškas priimamas, kaip liemens sritis kurios erdvinė pozicija atliekant veiksmą kinta mažiausiai iš visų kitų sričių. Dėl greitaveikos naudojamas *OpenNI* skeleto sekimo algoritmas, kuriame atsisakyta 5 papildomų skeleto taškų, todėl naudojama tik 15 iš galimų 20. Taip pat algoritme naudojama horizontali simetrija, kuri pagrįsta judesio energija, t. y. norint išskirti koks judesys atliekamas lyginami dviejų tų pačių, horizontalioje ašyje esančių skeleto taškų kampiniai pagreičiai, kaip pavyzdys galėtų būti lyginami kairės ir dešinės rankų, kaip taškų kampiniai pagreičiai, taip išskiriant, kuri žmogaus pusė dalyvauja atpažįstamame veiksmo (kaklo, galvos, liemens taškai šiame etape nelyginami). Paskutiniame žingsnyje – veiksmo atpažinime naudojami paslėptieji Markovo modeliai. Bandymai atlikti su duomenų baze (*Huawei/3DLife 3D, Grand Challenge*). Atpažinimo tikslumo vidurkis siekė tik 76,03 %, tačiau buvo pasiekta užsibrėžta realaus laiko greitaveika (1.2 lentelė).

1.2 Lentelė. Laiko išnaudojimas vieno kadro apdorojimui (laikas pateikiamas milisekundėmis) [6]

Kinect	Panaudoto algoritmo žingsniai		
	Skeleto sekimas	Požymių išskyrimas	PMM–paremtas atpažinimas
46.821	45.350	0.01	0.106

Laiko sąnaudos pateiktos 1.2 lentelėje gautos naudojant kompiuterį su Intel I7 procesoriumi, dirbančiu 2,67 GHz taktiniu dažniu ir 6 GB RAM atminties. Didžiausia dalis algoritmo vykdymo laiko, net 99,74 % buvo skiriama sekti skeletui, likusiai algoritmo daliai reikėjo tik 0,26 % viso laiko.

Literatūros šaltinis [7] labai panašus į prieš tai nagrinėtą [6] šaltinį, kadangi algoritme naudojamas skeleto taškų modelis, atpažinimui taikomi paslėptieji Markovo modeliai, o pats algoritmas vykdomas realiu laiku. Metodo išskirtinumas su prieš tai lygintu - veiksmų aptikimui taikomas trimačių skeleto modelio taškų padėtis histogramomis aprašantis algoritmas (*HOJ3D*). *HOJ3D* apskaičiuojamas naudojant gylio matricos veiksmų seką, kuri projektuojama panaudojant tiesinę diskriminantinę analizę. Sekančiame etape veiksmų seka suskirstoma į tam tikrą klasterių skaičių k , kur kiekvienas iš klasterių atitinka tam tikras pozas, tinkančias kuriam nors iš atpažįstamų veiksmų. Veiksmų atpažinimas ir klasifikacija vykdoma taikant diskretinius paslėptuosius Markovo modelius. Šiame darbe naudojamos sferinės histogramų koordinatės. Tokia metodika pasižymi tuo, kad kiekvienos pozos aprašai yra labai panašūs, nesvarbu iš kurios pusės jie būtų užfiksuoti. Tai pasiekama suderinant darbe naudojamas sferines koordinates su asmens tikrąją kryptimi (1.4 pav. (a)), koordinačių centras apibrėžiamas, kaip gautasis taškas ties klubais. Autorius suskirsto 3D plokštumą į tam tikras sritis: α plokštumą į 12 lygių sričių kas 30 laipsnių, θ plokštumą į 7 sritis nuo 0 iki 180 laipsnių kaip pavaizduoja 1.4 paveiksle (b), taip suskirstyta 3D plokštuma padalijama į 84 zonas.

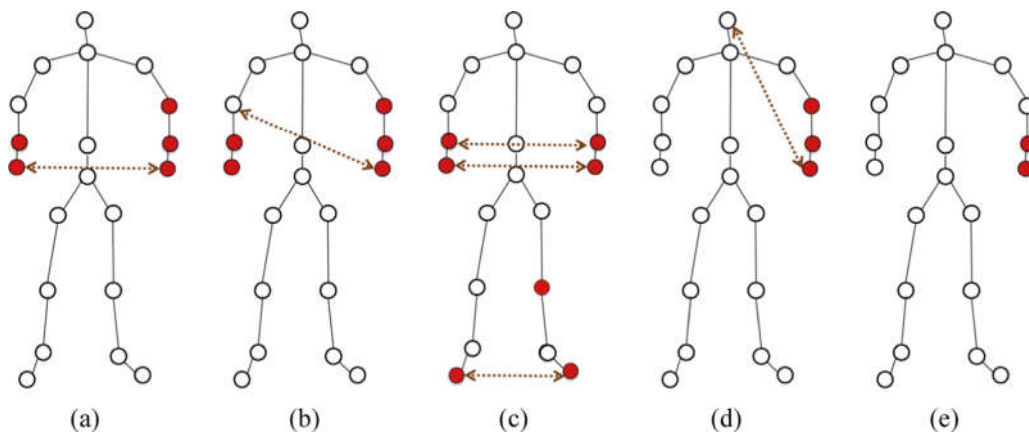


1.4 pav. (a) Skeleto taškai atvaizduoji HOJ3D plokštumoje. (b) Modifikuota sferinė koordinačių sistema taškų vietoms suskirstyti [7]

Sekančiame etape gaunamos jau suskirstytų pozų histogramos, kurios balsavimo būdu panaudojant Gauso svorių funkciją priskiriamos skirtingoms pozoms. Eksperimentai atlikti su dvejomis duomenų bazėmis. Viena iš jų sukurta autoriaus, kita vieša – *MSR Action3D* duomenų bazė. Naudojant autoriaus kurtą duomenų bazę pasiektas 90,92 % tikslumas, atpažįstant tokius veiksmus kaip ėjimas, sėdėjimas, nešimas ir pan. Sukurtą metodą testuojant su viešąją duomenų baze pasiektas net 97,15 % tikslumas. Autoriaus teigimu geresnis tikslumas galėjo būti gautas

dėl to, kad subjektai stovėjo visu kūnu priešais kamerą ir, jeigu veiksmas buvo atliekamas, kaip pavyzdys kairiąją ranka, tuomet jis būdavo pakartojamas ir dešiniąją. Tyrejo teigimu 3D skeleto modelio taškų išskyrimas ir HOJ3D skaičiavimas atliekamas realiu laiku, nes pasiektas vidutinis vienos sekos testavimo laikas 12,5 ms, naudojant *Matlab* programą bei kompiuterį su 2,93 GHz taktiniu dažniu veikiančiu Intel i7 CPU procesoriumi.

Straipsnyje [8] pateiktu metodu stengiamasi atpažinti ne tik paprastus veiksmus, kurie dažniausiai atpažįstami tik pagal kurią nors vieną žmogaus kūno dalį, pvz. mojavimas ranka ar dantų valymas, bet ir sudėtingesnius, kur gali dalyvauti ne viena kūno galūnė, kaip pavyzdys, tokie veiksmai galėtų būti: žmogaus ėjimas, einant dalyvauja abi rankos, judėdamos į abi puses, šis papildomas veiksmas negalėtų būti atpažintas, jeigu vertintume kiekvieną iš galūnių atskirai. Šiam tikslui tyrėjas sujungė laiko erdvės modelį, paremtą skeleto modelio požymiais ir naują panašumo funkcijomis paremtą LCSS algoritmą. Sudarytame skeleto modelyje atsisakoma kelių taškų, todėl algoritmas realizuojamas su 11 skeleto taškų, koordinačių pradžią priimant kaip klubų tašką. Atlikus šiuos veiksmus tam, kad charakterizuoti kūno pozas ir atitinkamų šių taškų judesių sekas, tyrėjas išskyrė charakteringąjį vektorių, sudarytą iš skeleto modelio taškų kampų sekos, tai padėjo aprašyti kaip sąveikauja modelio taškai kintant laikui. Šis metodas ypač prasmingas kompleksinių veiksmų išskyrimo. Charakteringajame vektoriuje išskiriamos daugiausiai informacijos suteikiančios skeleto modelio taškų kampų ir atitinkamų šių taškų judesių sekos, kurios toliau lyginamos su etaloninėmis ir klasifikuojamos LCSS algoritmu.



1.5 pav. Informatyviausi skeletų taškai ir atitinkamų taškų judesiai atliekant tam tikrus veiksmus Teniso padavimas, (b) Boksavimasis, (c) Bėgiojimas, (d) Pakėlimas ir metimas (e) Horizontalus rankos mojavimas [8]

1.5 paveiksle pateikiami jau išskirti informatyvūs skeleto modelio taškai (pažymėti raudonai) bei tų taškų judesiai (pažymėti rodyklėmis) iš kurių atpažįstami veiksmai. Eksperimentai atlikti su dvejomis duomenų bazėmis: *MSRAction3D* ir *HDM05*. Naudojant

MSRAction3D duomenų bazę šis metodas pasiekė geriausius rezultatus tarp skeleto modelių paremtų metodų, naudotų būtent su šia duomenų baze, kurie siekė 91,2 %, geriau pasirodė tik puikios normalės vektoriaus (angl. *super normal vector*) metodas [9], kuris paremtas gylio matrica, jo rezultatai siekė 93,09 %. Taip pat šis metodas [9] pasiekė geriausius rezultatus ir su kitomis *MSR* duomenų bazėmis *Gesture3D* – 94,72 %, *ActionPairs* – 98,89 %, *DailyActivity3D* – 86,25 %. Atliekant eksperimentus su *HDM05* duomenų baze pasiektas 85,23 % tikslumas [8]. Taip pat autorius atliko tyrimą kaip įtakojamas LCSS klasifikatoriaus tikslumas užtriukšminant vaizdo medžiagą Gauso ir impulsiniais triukšmais. Tiek vienu, tiek kitu atveju triukšmas didelės įtakos klasifikavimui nepadarė, kai triukšmo lygis neviršijo 35 %, įnešant daugiau triukšmo nuo 35 % iki 60 % klasifikavimo tikslumas krito nuo 90 % iki 60 % Gauso triukšmo atveju ir iki 70 % impulsinio triukšmo atveju.

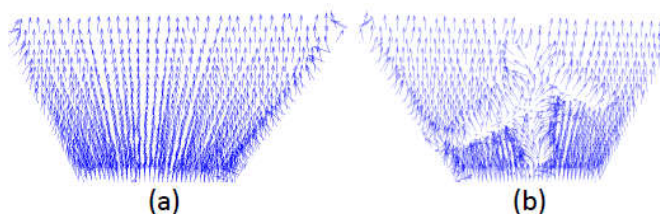
Literatūros šaltinyje [10] veiksmų atpažinimui įvertinami lokalaus laiko erdvės modelio požymiai. Šis metodas ganėtinai sparčiai populiarėjantis tarp tyrėjų, kurie stengiasi atpažinti veiksmus vaizdo medžiagoje. Autoriai straipsnyje apžvelgia keturis atpažinimo algoritmus: *Harris3D*, *Cuboid*, *Hessian*, *Dence*. Šiuos algoritmus kombinuoja su šešiais skirtingais aprašančiais algoritmais ar jų modifikacijomis: *HOG/HOF*, *HOG3D*, *ESURF*, *Cuboid*, *HOG*, *HOF*. *Cuboid* aprašantysis algoritmas galėjo būti pritaikytas tik su *Cuboid* atpažinimo algoritmu, analogiškai *ESURF* aprašantysis algoritmas su *Hessian*. Bandymams naudotos trys duomenų bazės: *KTH*, *UCF* sporto ir *Hollywood2*. Tyrimą atliekant su *KTH* duomenų baze geriausius rezultatus pasiekė *Harris3D+HOF* algoritmų derinys, pasiektas tikslumas siekė 92,1 %, su sekančia *UCF sporto* duomenų baze didžiausias tikslumas pasiektas naudojant *Dense+HOG3D* algoritmų derinį, kuris siekė 85,6 % bei *Hollywood2* duomenų bazės tyrimo didžiausias tikslumas siekė tik 47,4 % panaudojant *Dense+HOG/HOF* algoritmų derinį. Taip pat autoriai atliko ir greitaveikos tyrimą, naudojant Dell Precision T3400 dviejų branduolių kompiuterį dirbantį 2,66 GHz taktiniu dažniu, turintį 4 GB operatyviosios atminties (RAM). Pasiekta 4,6 kadro per sekundę greitaveika naudojant ne pačius geriausius tikslumo rezultatus pasiekusį *Hessian+ESURF* algoritmų derinį, tačiau į kadra įėjo nedaug, tik 19 požymių, pagal kuriuos vyko atpažinimas. Sekantis derinys *Dense+HOG/HOF* pasiekė tik 1,2 kadro per sekundę greitaveiką, tačiau šiuo metu kadre buvo net 643 požymiai pagal kuriuos vyko atpažinimas. Nors tyrimas buvo atliktas prieš 7 metus, su to meto technika buvo pasiekti ganėtinai neprasti rezultatai, tačiau netinkantys realaus laiko sistemai. Sparčiai gerėjant, greitėjant technologijoms šiuo metu be abejo galima būtų pasiekti geresnius rezultatus naudojant tuos pačius autoriaus apžvelgtus metodus, kurie šiai dienai galbūt galėtų patenkinti realaus laiko sistemų keliamus reikalavimus.

Dar vienas tvirtas (angl. *robust*) metodas apžvelgiamas [11] straipsnyje. Autorius realizavo trimatėje erdvėje išskirto žmogaus suradimą, sekimą ir veiksmų atpažinimą, tam panaudodamas gylio matricų seką, kad sudarytų laiko erdvės požymius, pačiam veiksmo atpažinimui pritaikė modifikuotą PMM algoritmą. Tam, kad būtų galima be didesnių triukšmų išskirti žmogaus siluetą, pašalinamas fonas (panaudojant mažiausių kvadratų metodą) ir grindys (eliminuojant y koordinatę gylio matricoje) iš neapdirbtos gylio matricos. Silueto parinkimo patikslinimui ir triukšmų sumažinimui bei rezultatų pagerinimui priderinami optinio srauto požymiai su skirtingų verčių normalės vektoriais, kurie gaunami iš gylio matricų istorijos, išskiriant judesius (t. y. gylio matricos kitimą bėgant laikui). Gautame siluete išskiriama penkiolika 3D skeleto taškų. Norint atpažinti veiksmą sekami tik tie skeleto taškai, kurie atlieka judesį, t. y. tie skeleto taškai, kurių erdvinė padėtis keičiasi kintant laikui, šie požymiai grupuojami panaudojant *LBG (Linde–Buzo–Gray)* algoritmą. Sugrupuoti duomenys naudojami testavimui bei atpažinimui pritaikant modifikuotus paslėptuosius Markovo modelius, kurių esmė – atkreipti dėmesį į labiausiai judančias žmogaus kūno sritis, tokias kaip judanti koja ir klubai simbolizuojantys ėjimą ar judančią ranką, kas reikštų mojavimą ir pan. Tyrimams naudotos viešai publikuojamos duomenų bazės: *MSRDailyActivity3D*, *IM–DailyDepthActivity*, *SMMC–10*. Pirmosios dvi duomenų bazės naudotos metodo tikslumo tyrimui, trečioji *SMMC–10* greitaveikos tyrimui. Naudojant *MSRDailyActivity3D* duomenų bazę pasiektas atpažinimo tikslumas – 91,3 %, o *IM–DailyDepthActivity* duomenų bazę – 68,3 %. Nors su antrąją duomenų bazę pasiektas tikslumas nėra labai aukštas, tačiau tiek vienu, tiek kitu atveju jie viršijo kitų metodų pasiektus tikslumo rezultatus, kuriuos apžvelgė autorius savo darbo įžangoje. Tiriant greitaveiką su *SMMC–10* duomenų bazę pasiekta net 289,9 kadrų per sekundę greitaveika.

Straipsnyje [23] nagrinėjami metodai žmogaus padėties (gulėjimas, sėdėjimas, stovėjimas, pasilenkimas) nustatymui. Sukurta ganėtinai paprasta sistema, kurios sudedamosios dalys yra duomenų surinkimas, apdorojimas ir požymių išskyrimas, pagal kuriuos atpažįstama žmogaus padėtis. Kadangi sprendžiamas uždavinys nėra toks sudėtingas, todėl atsisakoma pusės sudaryto skeleto modelio taškų, kurie šioje situacijoje nėra informatyvūs bei sulėtina skaičiavimus, tokių kaip klubai, rankos ir pėdos pirštai. Autorius pats sudarė duomenų bazę sudarytą iš 4 padėčių scenarijų su 5 skirtingais asmenimis. Visais atvejais *Kinect* nuo objekto buvo nutolęs 110 cm. Su šia duomenų bazę atliko 7 eksperimentus atjungtame (angl. *off–line*) režime: naudojant skeleto modelį sudarytą iš 7, 9, 17, 20 taškų, vienu atveju duomenims panaudojant mastelį tarp 1 ir –1, kitu atveju be mastelio, vėlesniame etape juos klasifikuojant SVM algoritmo pagalba. Šie eksperimentai parodė, kad netaikant duomenims mastelio nebuvo atpažįstama gulėjimo padėtis, nepaisant to pasiektas atpažinimo tikslumas nuo 65,26 % iki 73,43 %, tačiau naudojant mastelį tikslumas siekė net 98,2–100 %. Prijungtame (angl. *on–line*) režime tyrėjas atliko tyrimą, kaip

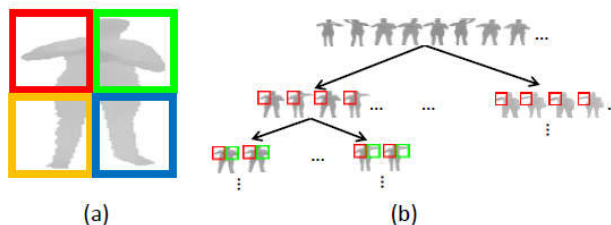
tikslumą įtakoja skirtingi atstumai (270, 320, 370 cm) ir kampai (0, 90° kairiau, 90° dešiniau) tarp objekto ir *Kinect*. Tiek kampų, tiek atstumo pasikeitimai padarė neigiamą įtaką rezultatams. Sunkiai sekėsi atpažinti gulėjimą ir susilenkimą, atpažinimo tikslumo vidurkis siekė vos 49,57 % – susilenkimui ir 45,31 % – gulėjimui, sėdėjimą ir stovėjimą atpažino didesniu nei 90 % tikslumu.

Straipsnyje [24] žmogaus veiksmų atpažinimas realizuotas naudojant tik gylio matricas. Metodas pagrįstas žmogaus silueto išskyrimu ir analize gylio matricoje. Silueto išskyrimui naudotos plokštumos normalės, kurių panaudojimas tampa efektyvus, kai objektas yra per arti fono, pvz. žmogaus gulėjimas ant žemės/lovos, objektai labai arti sienos (1.6 pav.)



1.6 pav. Plokštumos vektorių vizualizacija a) fonas b) fonas + objektas gylio vaizde [24]

Iš gautų siluetai, pritaikant taškų debesies (angl. *point cloud*) metodą gylio vaizduose formuojami pozų aprašai. Kadangi kiekvieno vaizdo registravimui trimatės erdvės taškų debesyje reikalinga atlikti ganėtinai nemažai skaičiavimų, todėl dėl greitaiveikos stengiamasi apsiriboti tik tais regionais į kuriuos įeina išskirtas siluetas. Siluetas grupuojamas į keturis ketvirčius. Algoritmas pagrįstas vieno ketvirčio stebėjimu, jei stebimajame ketvirtyje aptiktas apraše apibrėžtas veiksmas, einama prie sekančio ketvirčio bei kartojami aptikimo veiksmai (1.7 pav.).



1.7 pav. a) Silueto suskirstymas į keturis ketvirčius b) Veiksmo aptikimo hierarchija pozų apraše [24]

Metodo testavimui pasirinkta žmogaus veiklos¹ duomenų bazė su kuria pasiektas 78,1 % tikslumas, kai žmogus nebuvo prieš tai matytas ir 94,6 % tikslumas, kai žmogus buvo prieš tai matytas.

Straipsnyje [25] žmogaus ir objekto sąveikai apibrėžti naudojamas 4D erdvės metodas, apjungiantis dvi užduotis, papildančias viena kitą. Pirmoji užduotis susideda iš trimatėje erdvėje žmogaus pozos ir objektų išskyrimo geometrinėmis figūromis, antroji – įvykių perėjimai ir sąsaja tarp objektų vienmatėje erdvėje, tokiu būdu gaunamas keturių matmenų laiko erdvės modelis. Metodo esmė sekti ir atpažinti ne tik žmogų, bet ir objektą (pvz. puoduką, kėdę, kompiuterį). Pagal šių objektų tarpusavio sąveiką nustatomas veiksmas. Objektus reikalinga sekti, nes kai kuriuos iš jų gali pernešti žmogus. Analizei autorius sudarė duomenų bazę susidedančią iš 3815 vaizdo sekų ir 383,036 RGBD kadru, gautą panaudojant *Kinect* kameras. Lygčių sprendimui pritaikytas dinaminio programavimo spindulio paieškos (angl. *Dynamic programming beam search*) algoritmas. Šiuo metodu naudojant autoriaus sukurtą duomenų bazę pasiekti 87 % įvykių atpažinimo ir 78,3 % objektų atpažinimo tikslumo vidurkiai. Objektų atpažinimo tikslumas gautas ganėtinai žemas dėl to, kad mažesni objektai, tokie kaip telefonas, pelė buvo uždengiami žmonių arba naudojama rezoliucija buvo per žema.

Pirmojo skyriaus išvados ir apibendrinimai

1. Apžvelgus naudojamą įrangą vaizdų atpažinimui pasirinktas perspektyviausias, nebrangus *Microsoft* korporacijos produktas – *Kinect*.
2. Literatūros analizėje apžvelgti 2D algoritmai dažniausiai taikomi tik nesudėtingiems veiksmai atpažinti, sudėtingesnių veiksmų atpažinimui reikalinga atlikti itin daug skaičiavimų, kas turi stiprios įtakos algoritmų greitaveikai.
3. Algoritmai pasirinktus uždavinius sprendžiantys erdvėje didesnėje nei 3 dimensijų, pasižymi tiksliais atpažinimo vidurkiais, tačiau vis dar apriboti techninėmis skaičiavimų galimybėmis, kas įtakoja sistemos greitaveiką.
4. *Microsoft Kinect* siūlomas algoritmas išskiria skeleto modelį sudaryta iš 20 taškų, daugelyje sprendžiamų uždavinių siekiant padidinti skaičiavimų greitaveiką užtenka ženkliai mažiau, 7 ar 11 taškų skeleto modelio.

¹ J. Sung, C. Ponce, B. Selman, and A. Saxena. Human activity detection from rgbd images. AAAI workshop on Pattern, Activity and Intent Recognition, 2011.

2. Teorinė dalis

2.1. Human3.6M duomenų bazė

Atliekamas tyrimas orientuotas į 3D duomenis, dėl to pasirinkta viena iš didžiausių bei autorių teigimu viena iš sudėtingiausių duomenų bazių [26]. Kaip galima pastebėti iš pavadinimo duomenų bazę sudaro 3,6 milijono trimačių žmogaus pozų bei atitinkamų vaizdų. Šios duomenų bazės sudaryme dalyvavo net 11 skirtingų subjektų – profesionalių aktorių, iš kurių 6 vyrai ir 5 moterys atrinkti pagal kūno masės indeksą tarp 17 ir 29, kurie suvaidino net 15 scenarijų (2.1 pav.):

1. Krypties rodymas;
2. Diskutavimas;
3. Valgymas;
4. Pasisveikinimas;
5. Šnekėjimas telefonu;
6. Pozavimas;
7. Apsipirkinėjimas;
8. Sėdėjimas ant kėdės;
9. Veiksmų atlikimas sėdimoje pozicijoje;
10. Rūkymas;
11. Fotografavimas;
12. Laukimas;
13. Ėjimas;
14. Šuns vedžiojimas;
15. Pasivaikščiojimas drauge su įsivaizduojamu subjektu.



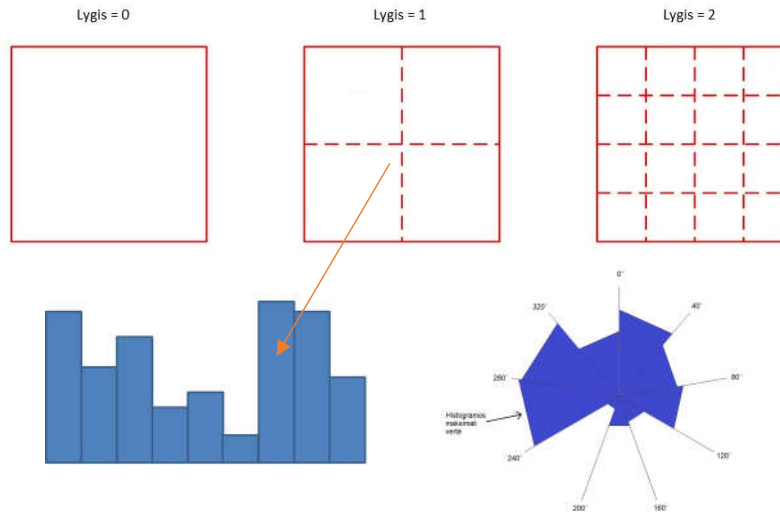
2.1 pav. *Human3.6M* duomenų bazės veiksmų rinkinys [26]

Norint sudaryti kuo realistiškesnę duomenų bazę aktoriai turėjo vilkėti savo kasdienes rūbas bei suvaidinti skirtingas scenarijų scenas be jokio iš ankstinio pasiruošimo. Tokiu būdu kiekviename scenarijuje yra keletas skirtingų scenų, kurios atitinka paruoštą scenarijų, tačiau patys atliekami veiksmai yra ganėtinai skirtingi tarpusavyje. Autoriai atskyrė duomenų bazę į apmokymo (7 subjektai iš kurių 3 moterys ir 4 vyrai) ir testavimo (4 subjektai iš kurių 2 moterys ir 2 vyrai) duomenis.

Šios duomenų bazės sukūrimui naudota scena su keturiomis aukštos rezoliucijos kameromis, išdėstytomis skirtinguose patalpos kampuose. Kadangi subjektai turėjo vilkėti jutiklius, tuo pačiu trimatėje erdvėje buvo atkuriami ir skeleto taškai (sąnariai) bei nustatinėjami kampai tarp kiekvieno iš jų, tokiu būdu vėlesniame etape buvo atkuriamas trimatis žmogaus skeletas.

Ši duomenų bazė tyrimams pasirinkta dėl to, jog jos kūrėjai su duomenų baze pateikia jau išskirtus pozų požymių vektorius. Duomenų bazės sudėtyje pateikiami dviejų rūšių požymiai, vieni iš jų išskirti atmetant foną, tokiu būdu paliekant tik subjekto siluetą, sekantieji – ties subjektu išskiriant aprėpties langelį į kurį pateko subjektas. Pats požymių vektorius sudarytas panaudojant

SIFT aprašą tinklų piramidėmis (angl. *pyramids of SIFT grids*), kuris sudarytas iš trijų lygmenų (2x2, 4x4 ir 8x8), kiekvieno iš lygmens kvadratą dalijant į 9 zonas, taip gaunant 360 laipsnių erdvę padalintą į dalis kas 40 laipsnių (2.2 pav.).



2.2 pav. Požymių vektoriaus sudarymo paaiškinimas, nuotraukos suskirstymas lygmenimis, erdvės padalijimas į dalis kas 40 laipsnių

2.2 pav. pavaizduotas dviejų lygių ir 9 zonų pavyzdys, tačiau šioje duomenų bazėje naudojamas trijų lygių aprašas, dėl to gaunami $1+4+4*4+8*8=85$ mazgai. Kiekvienam iš šių mazgų sukuriama histograma orientuota gradiento kryptimi, kurios kaip ir prieš tai minėta suskirstytos į 9 zonas, kas 40 laipsnių, tokiu būdu gautas požymių vektorius sudarytas iš $85*9=765$ reikšmių.

Kadangi duomenų bazė ganėtinai didelė, todėl susiduriama su techninėmis problemomis, dėl kurių apmokinti klasifikatorių su visais duomenimis reikia ne tik galingos techninės įrangos, turinčios nemažai RAM atminties, bet ir laiko problemomis, kadangi apmokymo ir testavimo procesai su tokiais duomenų kiekiais vyksta ganėtinai ilgai. Dėl šių priežasčių stengiamasi apžvelgti specifinius klasifikatorius, kurie gali būti taikomi didelės apimties duomenims apdoroti.

Straipsnyje [27] pateikiama klasifikatorių apžvalga, palyginant juos tarpusavyje. Pateiktame straipsnyje geriausi rezultatai gauti su *TreeBagger*, *SVM* ir *NNet* klasifikatorių modifikacijomis, dėl šių priežasčių nuspręsta tyrime naudoti klasikinius *TreeBagger*, *NNet* klasifikatorius ir modifikuotą *SVM* klasifikatorių – *LibSVM*, kuris skiriasi nuo savo pirmtako, tuo, kad gali būti pritaikytas daugiau nei dviejų klasių klasifikacijai.

2.2. Klasifikatoriai

2.2.1. TreeBagger

Atsitiktinio miško apmokymo algoritmas medžių apmokinimui naudoja pakavimo metodiką.

Pseudo algoritmas [28]:

Duotas mokymo rinkinys $X=x_1, \dots, x_n$ su atsakais $Y=y_1, \dots, y_n$ pakartotinai supakuojami (B kartų) parenkant atsitiktines imtis ir pakeičiant mokymo rinkinius, taip pritaikant medžius prie šių imčių:

Nuo $b=1$ iki B :

Atrinkti su pakeitimu n apmokymo pavyzdžių iš X, Y , pavadinant juos X_b, Y_b .

Apmokinti sprendimų ir regresijos medį f_b su duomenimis X_b, Y_b .

Po apmokymo nematytų duomenų x' atpažinimas gali būti daromas vidurkinant atpažinimus iš visų individualių regresijos medžių.

$$\hat{f} = \frac{1}{B} \sum_{b=1}^B \hat{f}_b(x') \quad (1)$$

Arba taikant balsavimo metodą, išrenkant dažniausiai pasikartojusį variantą.

Pakavimo procedūra veda prie geresnių modelio charakteristikų, nes tai sumažina modelio variaciją, nepadidinant poslinkio. Tai reiškia, kad tuo atveju, kai vieno medžio spėjimai yra labai jautrūs triukšmui atsirandančiame apmokymo duomenyse, tokiu atveju daugelio medžių spėjimo vidurkis yra tvirtesnis, nes nėra toks jautrus triukšmams, tol kol šie medžiai nėra susiję tarpusavyje. Paprasčiausiai apmokinant vis daugėjantį medžių kiekį su tais pačiais mokymo duomenimis gaunami stipriai koreliuojantys medžiai (jeigu mokymo algoritmas yra deterministinis užtenka naudoti tą patį medžių skaičių, apmokymą kartojant daugybę kartų). Metodo pagrindas yra medžių koreliavimas, panaudojant skirtingus duomenų rinkinius [29].

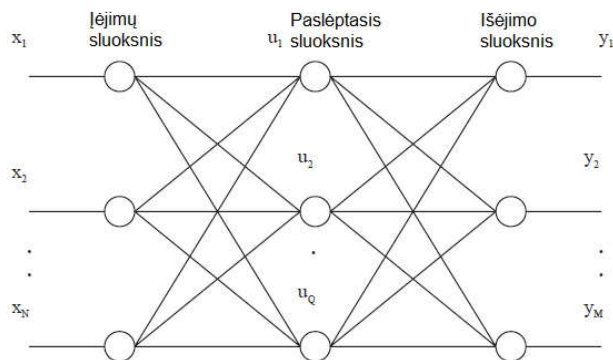
Medžių skaičius B yra laisvai pasirenkamas parametras. Paprastai naudojama nuo kelių šimtų iki kelių tūkstančių medžių, priklausomai nuo apmokymo duomenų pobūdžio ir dydžio. Optimalus medžių B skaičius gali būti surandamas naudojant kryžminį patvirtinimą (*angl. Cross-validation*) arba stebint „*out-of-bag*“ paklaidą: vidutinė kiekvieno apmokymo rinkinio x_i atpažinimo paklaida, panaudojant tik tuos medžius, kuriems nebuvo naudojamas x_i duomenų rinkinys. Mokymo ir bandymo paklaidos linkusios stabilizuotis suradus reikiamą kiekį medžių, todėl kartais mažesnis medžių kiekis gali duoti netinkamą tikslumą arba didesnis medžių kiekis gali sąlygoti ilgesnį apmokinimo procesą [29].

Su tam tikrais parametru rinkiniais šį klasifikavimo algoritmą galima būtų padaryti analogišką atsitiktinio miško (angl. *Random Forest*) klasifikavimo algoritmui.

2.2.2. Neuroniniai tinklai

Pagal savo struktūrą dirbtinis neuroninis tinklas atitinka biologinį žmogaus smegenų mokymosi procesą. Dirbtinių neuroninių tinklų yra ganėtinai daug skirtingų struktūrų, viena iš jų daugiasluoksniu perceptrono neuroniniai tinklai [30].

Dažnai apie neuroninius tinklus galima išgirsti, kad jie veikia juodosios dėžės principu, kai į modelį paduodami įėjimų duomenys ir juodosios dėžės, paslėptąjo sluoksnio pagalba gaunami išėjimo duomenys. Modelio efektyvumas priklauso nuo paslėptajame sluoksnyje naudojamų neuronų skaičiaus, su nedideliu neuronų skaičiumi galimai gautas tikslumas bus netenkinantis, tačiau parinkus per didelį neuronų skaičių išskyla rizika, jog mokymo duomenys persidengs ir naujų duomenų atpažinimas taps netenkinantis.



2.3 pav. Dirbtinio neuroninio tinklo struktūra [30]

Dėl prieš tai minėtų problemų, išskylančių dėl neuronų skaičiaus paslėptajame sluoksnyje sugalvota nemažai metodų, kaip galima parinkti korektiškiausią neuronų kiekį [30]. Tarkim turima trijų sluoksnių struktūra su N įėjimų, M paslėptųjų ir Q išėjimų sluoksnių. Įėjimų vektorius $\mathbf{x}=(x_1x_2\dots x_N)^T$ transformuojamas į tarpinį „paslėptąjį“ kintamąjį \mathbf{u} , kuris naudojamas φ_1 funkcijoje. Paslėptajame sluoksnyje j -ojo mazgo išėjimas u_j gaunamas pagal tokią lygtį:

$$u_j = \varphi_1 \left(\sum_{i=1}^N w_{i,j}^1 x_i + b_j^1 \right) \quad (2)$$

šioje formulėje b_j^1 ir $w_{i,j}^1$ atitinkamai jungties tarp j -tojo mazgo paslėptajame sluoksnyje ir i -ojo mazgo įėjimų mazgo poslinkis ir svoris. Šio tinklo išėjimų vektorius $\mathbf{y}=(y_1y_2\dots y_Q)^T$, panaudojant tarpinį kintamąjį \mathbf{u} , kuris naudojamas išėjimo sluoksnio φ_2 funkcijoje. Kaip pavyzdys, k neurono išėjimas gali būti gaunamas pagal tokią lygtį:

$$u_k = \varphi_2 \left(\sum_{i=1}^M w_{i,k}^2 u_i + b_k^2 \right) \quad (3)$$

Šiose formulėse aprašomas sąryšis tarp visų tinklo sluoksnių, (1) lygtimi aprašomas sąryšis tarp įėjimo ir paslėptojo sluoksnių, (2) lygtimi aprašomas sąryšis tarp paslėptojo ir išėjimo sluoksnių. Pateikiama keletas aktyvavimo funkcijų φ_1 ir φ_2 formų, tokios kaip loginės, hiperbolinio tangento ir tiesinės funkcijos, atitinkamai (4)–(6) lygtyse:

$$\varphi(v) = \frac{1}{1+e^{-v}} \quad (4)$$

$$\varphi(v) = \frac{1-e^{-2v}}{1+e^{-2v}} = \frac{2}{1+e^{-2v}} - 1 \quad (5)$$

$$\varphi(v) = v \quad (6)$$

Dirbtinio neuroninio tinklo mokymas apima jungčių svorio reikšmių paiešką, kurios atitinkamai kuo labiau sumažintų paklaidos funkciją tarp faktinio išėjimo reikšmės tinkle ir ieškomos reikšmės apmokymų imtyje. Kaip pavyzdys viena iš labai plačiai naudojamų paklaidos mažinimo funkcijų - vidutinė kvadratinė paklaida [30].

2.2.3. LibSVM

LibSVM klasifikatorius [31] yra SVM (angl. *Support Vector Machine*) klasifikatoriaus atmaina, kadangi SVM klasifikatorius skirtas klasifikuoti tik dvi klases, buvo sugalvotas šio klasifikatoriaus patobulinimas, pritaikant jį daugiau nei dviejų klasių problemoms spręsti. Galimi algoritmai „vienas su visais“ arba „vienas su vienu“.

Kaip ir minėta viena iš LibSVM modifikacijų daugiau nei vienai klasei klasifikuoti yra „vienas su vienu“. Jeigu k yra klasių skaičius, tai sukuriama $k(k-1)/2$ klasifikatorių iš kurių kiekvienas apmokomas su duomenimis iš dviejų klasių. Apmokymo duomenims nuo i -osios ir j -osios klasių, sprendžiama dviejų klasių klasifikavimo problema [31]:

$$\min_{w^{ij}, b^{ij}, \xi^{ij}} \frac{1}{2} (w^{ij})^T w^{ij} + C \sum_t (\xi^{ij})_t \quad (7)$$

$$\text{priklausomai nuo } (w^{ij})^T \varphi(x_t) + b^{ij} > 1 - \xi_t^{ij}, \text{ jeigu } x_t \text{ priklauso } i - \text{ajai klasei}, \quad (8)$$

$$(w^{ij})^T \varphi(x_t) + b^{ij} \leq -1 + \xi_t^{ij}, \text{ jeigu } x_t \text{ priklauso } j - \text{ajai klasei}, \quad (9)$$

$$\xi_t^{ij} \geq 0. \quad (10)$$

Duomenų klasifikavime naudojama balsavimo strategija: kiekvienas dviejų klasių klasifikatorius paskutiniame etape atpažįsta visą duomenų imtį x , sulyginami visų klasifikatorių rezultatai, o duomenys priskiriami tai klasei, kuri surinko didžiausią balsų skaičių. Žinoma galima sąlyga, jog dvi identiškios klasės surinko tokį patį balsų skaičių, šiuo atveju tai nėra pati geriausia strategija, tačiau pasirenkama ta klasė, kuri buvo atpažįstama pirmesnė (pasirinkimas vykdomas tik tarp dviejų klasių kurios surinko vienodą skaičių balsų) [31].

2.1 Lentelė. LibSVM klasifikatorio apmokymui, testavimui byloje naudojama duomenų struktūra

Vieta byloje	1	2	3	N
Duomenų struktūra	Klasės nr.	Požymio nr.:Požymis	Požymio nr.:Požymis	Požymio nr.:Požymis
Pavyzdys	3	1:0.4578	2:1.5241	N-1:2.1425

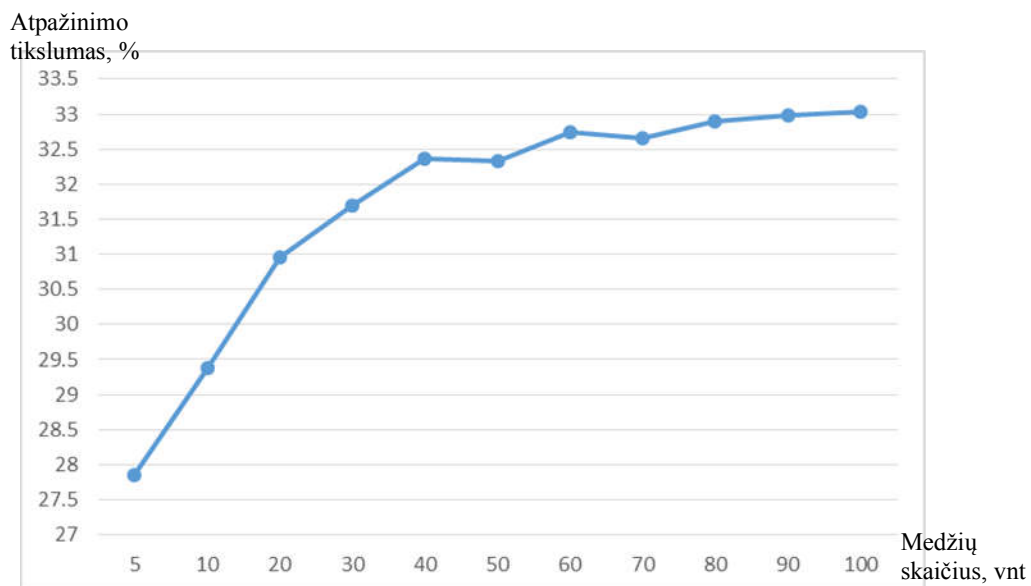
3. Tiriamoji dalis

Kadangi tyrimui atlikti turima bazė yra ganėtinai didelė, susiduriama su pirminiais sunkumais, kurie atsigręžia į naudojamą techninę įrangą, t. y. kompiuteris su *Intel Core i5–2410M* procesoriumi dirbančiu 2,30 GHz taktiniu dažniu ir turinčiu 8 GB operatyviosios atminties RAM, dirbantis su *Windows 7* operacine sistema, todėl pirmiausia pasirenkama strategija, kuri leidžia kompiuteryje sukurti virtualią atmintį, taip padidinant operatyviosios atminties dydį, tam panaudojant kietojo disko SSD atmintį [32].

3.1. Klasifikavimas naudojant pozos požymių vektorių

3.1.1. Klasifikatorių parametrų, požymių vektoriaus dydžio parinkimas

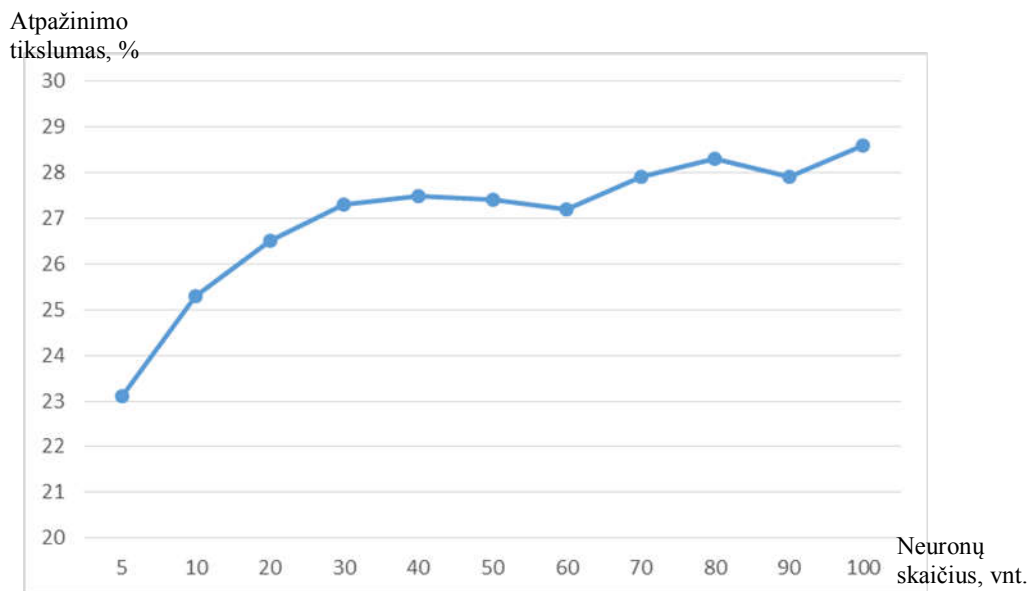
Siekiant geresnių atpažinimo rezultatų, tam paskiriant mažiau laiko atliekamas *TreeBagger* klasifikatoriaus medžių skaičiaus priklausomybės atpažinimo tikslumui tyrimas. Klasifikatorius apmokomas su pirmojo subjekto požymiais, testuojamas su visų likusių subjektų 25 % turimų požymių (3.1 pav.).



3.1 pav. Priklausomybė tarp *TreeBagger* klasifikatoriaus medžių skaičiaus ir atpažinimo tikslumo

3.1 paveiksle pateiktoje priklausomybėje galima pastebėti, kad esant medžių skaičiui didesniai nei 80 daroma įtaka atpažinimo tikslumui ganėtinai maža, šimtųjų eilės, priklausomybėje medžių skaičius apribojamas iki 100 dėl skaičiavimų greitaveikos. Dėl 3.1 paveiksle gautos priklausomybės tolimesniems tyrimams pasirenkamas 100 medžių *TreeBagger* klasifikatorius.

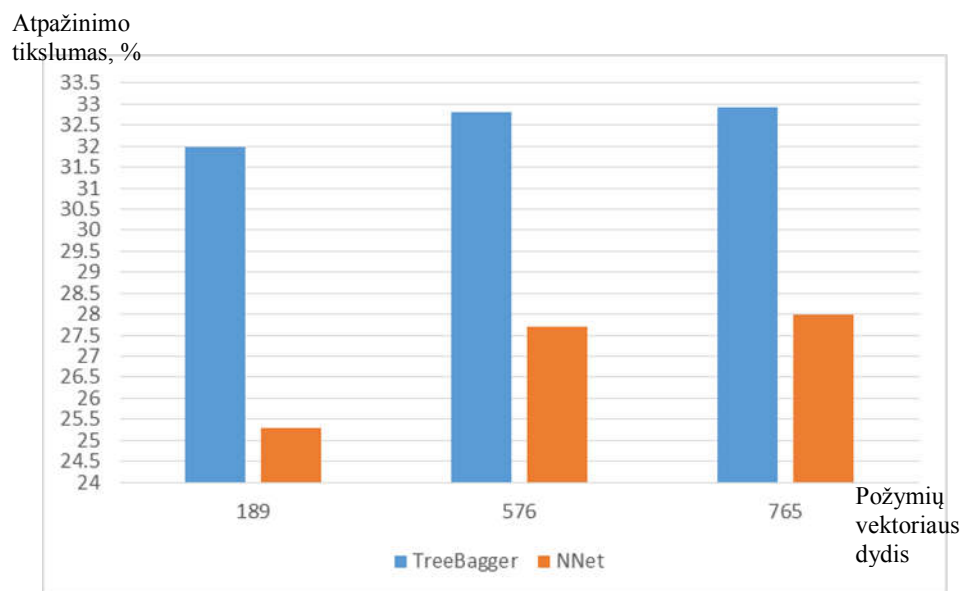
Taip pat atliekama ir *NNet* klasifikatoriaus analizė, kaip neuronų skaičius įtakoja turimos duomenų bazės atpažinimo tikslumą, dėl greitaiveikos tyrimas apribojamas iki 100 neuronų. Tyrimo rezultatai pateikiami 3.2 pav., analogiškai kaip ir 3.1 paveiksle pateiktoje priklausomybėje klasifikatoriai apmokomi su pirmojo subjekto požymiais, testuojami su visų likusių subjektų 25 % turimų požymių.



3.2 pav. Priklausomybė tarp *NNet* klasifikatoriaus neuronų skaičiaus ir atpažinimo tikslumo

3.2 paveiksle pateiktoje priklausomybėje galima matyti, kad ženklus atpažinimo tikslumo augimas pastebimas naudojant iki 40 neuronų, toliau didėjant neuronų skaičiui gaunami neženkliš pokyčiai įtakojantys atpažinimo tikslumą tiek į didesnę, tiek į mažesnę puses. Nors šiuo tyrimu geriausias 28,6 % atpažinimo tikslumas gautas naudojant 100 neuronų paslėptajame sluoksnyje, tačiau dėl ilgų skaičiavimų pasirenkama toliau naudoti 50 neuronų, ties kuriais buvo gaunamas neprastas 27,4 % atpažinimo tikslumas.

Sekančiu etapu, norint pasiekti didesnę greitaiveiką, bandoma mažinti požymių vektoriaus dydį. Pristatant prie turimo *Human3.6M* duomenų bazėje pateikto pozos požymio vektoriaus dydžio, kuris sudarytas iš 765 reikšmių. Jį atitinkamai galima sumažinti iki 576 (paliekamas tik trečiasis požymių išskyrimo lygis žr. 2.1. dalyje 2.2 pav.) arba 179 (paliekami tik du pirmieji požymių išskyrimo lygiai žr. 2.1. dalyje 2.2 pav.) reikšmių.



3.3 pav. Priklausomybė tarp požymio vektoriaus dydžio ir atpažinimo tikslumo naudojant *TreeBagger* ir *NNet* klasifikatorius

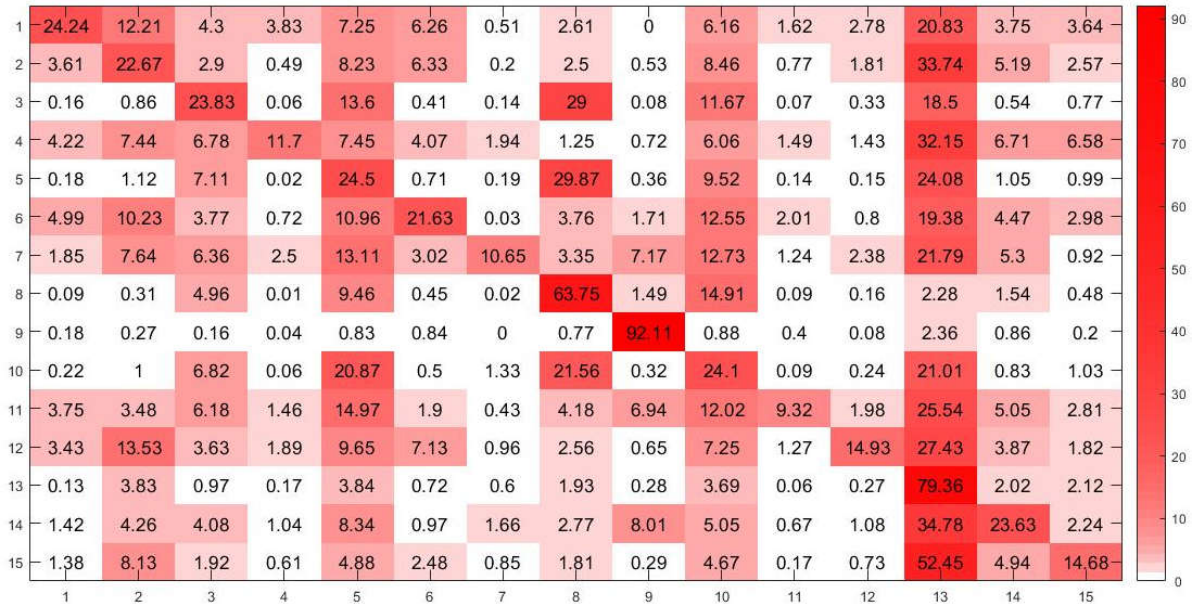
3.3 paveiksle pateiktoje diagramoje duomenims gauti naudojami *TreeBagger* klasifikatorius su medžių skaičiumi – 100 ir *NNet* klasifikatorius su 50 neuronų. Abiem atvejais geriausias bendras visų klasių atpažinimo tikslumas gaunamas naudojant 765 reikšmių požymio vektorius, nors atpažinimo tikslumo skirtumas naudojant 576 ar 765 reikšmių požymių vektorius neženklus, t. y. atitinkamai *TreeBagger* klasifikatoriaus atveju 32,82 % ir 32,93 %, o *NNet* klasifikatoriaus atveju 27,7 % ir 28 %. Tolimesniems tyrimams pasirinktas ilgiausias vektorius, kadangi šiuo atveju nebuvo gaunama didelė įtaka klasifikatorių apmokymo, testavimo greitaveikai.

3.1.2. Klasifikavimo strategijos

Siekiant kuo geriau išnaudoti turimus techninius resursus su dideliais duomenų kiekiais reikalinga išsikelti tam tikras strategijas, kaip tikslingai apdoroti bei suklasifikuoti turimus duomenis, siekiant gauti kuo aukštesnius, tvirtesnius tyrimo rezultatus.

3.1.2.1. Dideli duomenų kiekiai

Ši strategija atliekama su kuo didesniu duomenų kiekiu, tam, kad klasifikatorius galėtų apsimokinti bei prisitaikyti prie įvairesnių duomenų imčių. Klasifikavimui naudojamas *TreeBagger* klasifikatorius su 100 medžių, kadangi sistema apribota techninėmis galimybėmis apmokymui buvo naudojami tik vieno, pirmojo subjekto veiksmų pozų požymiai (3.4 pav.).



3.4 pav. *TreeBagger* klasifikatoriaus testavimo sumaišymo matrica, kai apmokymui naudoti pirmojo subjekto duomenys, testuojama su 25 % visų duomenų (iš visų subjektų)

3.4 paveiksle pateiktoje sumaišymo matricos pagrindinėje įstrižainėje galima matyti, kaip sekėsi atpažinti kiekvieną veiksmą (veiksmų numeraciją žr. 2.1. skyriuje, veiksmai pateikiami ta pačia numeracija), duomenys matricoje pateikiami procentine išraiška. Kaip galima matyti geriausiai sekėsi atpažinti sėdėjimą ant kėdės (8) – 63,75 % tikslumu, veiksmų atlikimą sėdimose pozicijoje (9) – 92,11 % tikslumu ir ėjimą (13) – 79,36 % tikslumu, nors šio veiksmo atpažinimo tikslumas ir buvo gaunamas ganėtinai aukštas, kaip matyti iš matricos 13 stulpelio šis veiksmas buvo painiojamas beveik su visais kitais veiksmais. Likusių veiksmų atpažinimo tikslumas neviršijo net 25 %.

Taip pat su analogiškais duomenimis išbandomas ir *NNet* klasifikatorius, gauti rezultatai pateikiami 3.5 pav.



3.5 pav. *NNet* klasifikatoriaus testavimo sumaišymo matrica, kai apmokymui naudoti pirmojo subjekto duomenys, testuojama su 25 % visų duomenų (iš visų subjektų)

3.5 paveiksle pateiktoje sumaišymo matricos pagrindinėje įstrižainėje galima matyti, kaip sekėsi atpažinti kiekvieną veiksmą naudojant *NNet* klasifikatorių su 50 neuronų paslėptajame sluoksnyje (veiksmų numeraciją žr. 2.1. skyriuje, veiksmai pateikiami ta pačia numeracija). Geriausi rezultatai gauti bandant atpažinti valgymo veiksmą (3) – 35,20 %, apsipirkinėjimą (7) – 71,96 %, sėdėjimą ant kėdės (8) – 63,05 % tikslumais, kitų veiksmų atpažinimas nesiekė net 26 %. Lyginant 3.4 pav. *TreeBagger* ir 3.5 pav. *NNet* klasifikatoriais pateiktus rezultatus galima pastebėti, jog *TreeBagger* klasifikatoriaus atveju geriau nei *NNet* klasifikatoriaus sekėsi atpažinti veiksmų atlikimą sėdimoje pozicijoje (9) ir ėjimą (13), tačiau *NNet* klasifikatorius geriau atpažino valgymą (3) ir apsipirkinėjimą (7).

3.1.2.2. Apmokytų modelių sujungimas

Dėl techninių galimybių apribojimo pasirinkta *TreeBagger* klasifikavimo algoritmo strategija, kuri leidžia sujungti kelis skirtingus modelius apmokytus su tais pačiais veiksmais. Šiam algoritmo įgyvendinimui sukurti penki modeliai, iš jų trys modeliai apmokyti, sekantys du gauti sujungiant prieš tai apmokytus modelius:

1. Modelis Nr. 1 – gautas apmokant klasifikatorių su pirmojo subjekto duomenimis;
2. Modelis Nr. 2 – gautas apmokant klasifikatorių su penktojo subjekto duomenimis;
3. Modelis Nr. 3 – gautas apmokant klasifikatorių su šeštojo subjekto duomenimis;

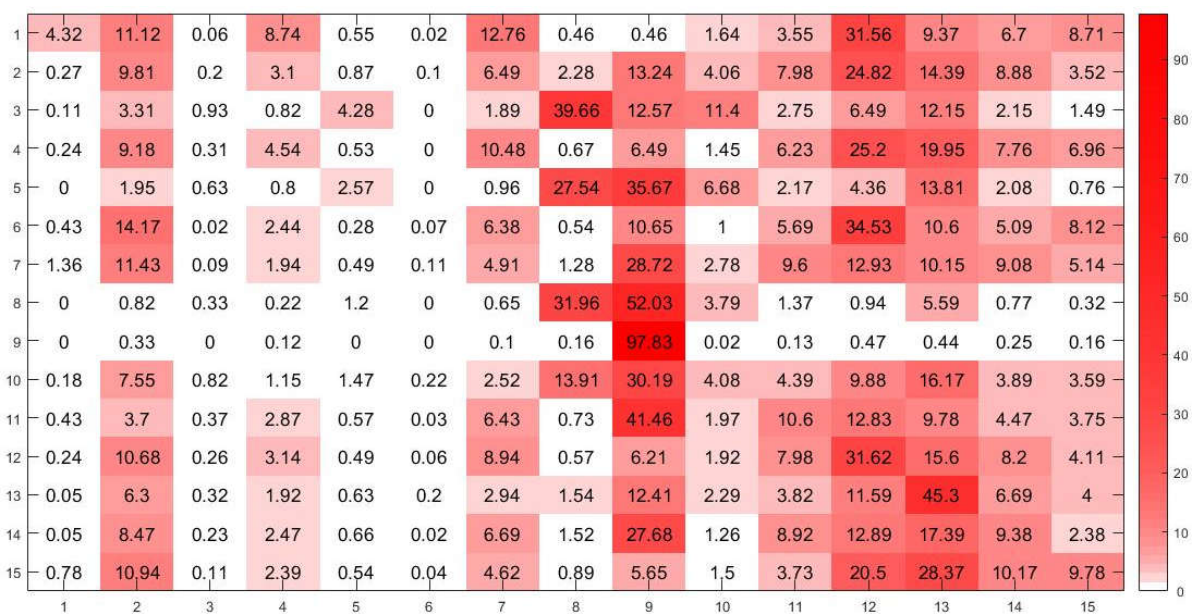
4. Modelis Nr. 4 – gautas sujungiant modelio Nr. 2 klasifikavimo taisykles su modelio Nr. 3 taisyklėmis;
5. Modelis Nr. 5 – gautas sujungiant modelio Nr. 4 klasifikavimo taisykles su modelio Nr. 1 taisyklėmis.

Modelių testavimas atliekamas su antrojo subjekto požymių vektorių duomenimis (gauti rezultatai pateikiami 3.1 lentelėje).

3.1 Lentelė. Apmokytų bei apjungtų modelių atpažinimo tikslumai

Modelio Nr.	Bendras visų veiksmų atpažinimo tikslumas, %
1.	15,49
2.	17,85
3.	12,41
4.	14,61
5.	15,15

3.1 lentelėje pateiktuose duomenyse galima matyti, kad *TreeBagger* klasifikatorių modelių taisyklių sujungimas davė rezultatą artimą individualių modelių atpažinimo vidurkiui. Modelių įvertinimui 3.6, 3.7 paveiksluose pateikiamos geriausią rezultatą pasiekusio antrojo modelio ir sujungtojo, penktojo modelio sumaišymo matricos.



3.6 pav. Modelio Nr. 2 sumaišymo matrica



3.7 pav. Modelio Nr. 5 sumaišymo matrica

3.6, 3.7 paveiksluose pateikiamose sumaišymo matricose galima matyti, kad sujungimo strategija davė neigiamą poveikį, kadangi išryškėjo tik dviejų (9), (13) klasių atpažinimas, visos likusios klasės pagrinde buvo priskiriamos (13) klasei, dalis (9) ir labai nedidelė likusi dalis (14). Tai galėjo lemti *TreeBagger* klasifikatorių taisyklių sujungimas, kadangi didžioji dalis taisyklių buvo „silpnos“, t. y. turėjo nedidelę įtakos dalį reikalingos klasės atpažinimui, tokiu būdu maišantis tarp kelių klasių, tikrosios buvo užgožtos, atitinkamai išryškinant stipriąsias taisykles.

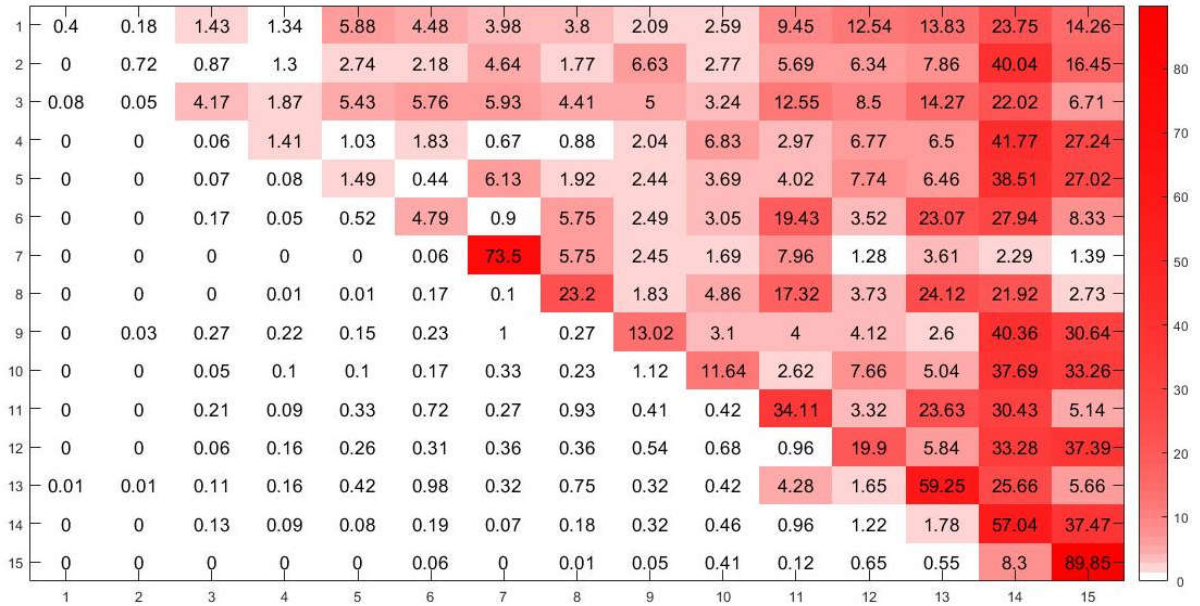
3.1.2.3. Balsavimas

Taip pat negavus patenkinamo rezultato buvo bandoma balsavimo strategija, t. y. balsavimui pasirinktos septynių apmokymo subjektų (žr. 2.1 Skyriuje) duomenų imtys, padalintos į 8 dalis po 15 tų pačių veiksmų. Apmokyti devyni skirtingi modeliai, visi modeliai testuoti su apmokyte nenaudotomis dvejomis duomenų imtimis. Galutiniame rezultate vyko balsavimas, kuri klasė atpažinta dažniausiai, su didžiausia sumine tikimybe t. y. pirmuoju etapu išrenkama viena klasė surinkusi daugiausiai balsų arba kelios klasės surinkusios vienodą kiekį balsų, antruoju etapu išrenkamos ir sudedamos tikimybės tų klasių, kurios buvo išrinktos pirmajame etape, laimėjusi klasė išrenkama su didžiausia šių tikimybių suma, tokiu būdu panaikinant galimybę kelioms klasėms surikti vienodą kiekį balsų. Tyrimo rezultatai bei kiekvieno modelio duomenų rinkiniai pateikiami 3.2 lentelėje.

3.2 Lentelė. Modelių sudarytų naudojant pozų požymių vektorius naudotų balsavimui atpažinimo tikslumai

Modelio Nr.	Modelio apmokymui/testavimui naudoti duomenų rinkiniai	Bendras visų veiksmų atpažinimo tikslumas, %
1.	Apmokymas su rinkiniais: 1, 2 Testavimas su rinkiniais: 7, 8	45,6
2.	Apmokymas su rinkiniais: 3, 4 Testavimas su rinkiniais: 7, 8	20
3.	Apmokymas su rinkiniais: 5, 6 Testavimas su rinkiniais: 7, 8	64
4.	Apmokymas su rinkiniais: 1, 3 Testavimas su rinkiniais: 7, 8	30,6
5.	Apmokymas su rinkiniais: 2, 4 Testavimas su rinkiniais: 7, 8	30,4
6.	Apmokymas su rinkiniais: 3, 5 Testavimas su rinkiniais: 7, 8	41,7
7.	Apmokymas su rinkiniais: 4, 6 Testavimas su rinkiniais: 7, 8	41,7
8.	Apmokymas su rinkiniais: 1, 4 Testavimas su rinkiniais: 7, 8	29,1
9.	Apmokymas su rinkiniais: 2, 5 Testavimas su rinkiniais: 7, 8	53,8
10.	Balsavimo modelis Testavimas su rinkiniais: 7, 8	26,3

3.2 lentelėje pateiktuose duomenyse galima pastebėti, kad 8 iš 9 atskirai apmokyti modelių bendras visų veiksmų atpažinimo tikslumas, buvo geresnis negu balsavimo modelio, kai kurių netgi ~2 kartus geresnis. Kai kurių atskirai apmokyti modelių tikslumas viršijo balsavimo modelio tikslumą, tai įtakojo silpna taisyklių bazė, maži atpažįstamų veiksmų tikslumai, dėl šių priežasčių stipresnes taisykles turintys modeliai nusvėrė silpnesnes taisykles turinčius. Balsavimo modelio sumaišymo matrica pateikiama 3.8 pav.



3.8 pav. Balsavimo modelio sumaišymo matrica

3.8 paveiksle pateiktoje sumaišymo matricoje matoma, jog gana gerai sekėsi atpažinti apsipirkinėjimą (7) – 73,50 %, fotografavimą (11) – 34,11 %, ėjimą (13) – 59,25 %, šuns vedžiojimą (14) – 57,04 %, pasivaikščiojimą drauge su įsivaizduojamu subjektu (15) – 89,85 %, likusiųjų veiksmų atpažinimas ganėtinai žemas nesiekiantis nei 25 %.

3.1.2.4. Vienas su visais

Norint patikrinti, kaip sekasi išskirti vieną veiksmą iš visų veiksmų visumos buvo apmokomi skirtingi klasifikatoriai išskiriant tik vieną ieškomą klasę. Apmokymui naudojant pirmojo subjekto duomenis, testavimui visos duomenų imties 1/4 duomenų, *TreeBagger* klasifikatoriaus atveju naudojama 100 medžių, o *NNet* atveju neuronų skaičius lygus 50, abu klasifikavimo metodai apmokyti ir testuoti su 765 reikšmių požymio vektoriumi, bandymo rezultatai pateikiami 3.3 lentelėje.

3.3 Lentelė. Vieno veiksmo atpažinimas lyginant su visais likusiais veiksmais

Eil. Nr.	Veiksmas	Veiksmo atpažinimas, %		Veiksmo priskyrimas prie likusių klasių, %	
		<i>TreeBagger</i>	<i>NNet</i>	<i>TreeBagger</i>	<i>NNet</i>
1.	Krypties rodymas	10,27	13,98	89,73	86,02
2.	Diskutavimas	14,49	20,12	85,51	79,88
3.	Valgymas	15,43	22,89	84,57	77,11
4.	Pasisveikinimas	8,99	9,02	91,01	90,98
5.	Šnekėjimas telefonu	11,93	16,25	88,07	83,75
6.	Pozavimas	9,24	17,01	90,76	82,99

Eil. Nr.	Veiksmas	Veiksmo atpažinimas, %		Veiksmo priskyrimas prie likusių klasių, %	
		<i>TreeBagger</i>	<i>NNet</i>	<i>TreeBagger</i>	<i>NNet</i>
7.	Apsipirkinėjimas	10,57	9,98	89,43	90,02
8.	Sėdėjimas ant kėdės	70,63	74,26	29,37	25,74
9.	Veiksmų atlikimas sėdimoje pozicijoje	23,56	62,22	76,44	37,78
10.	Rūkymas	11,03	25,51	88,97	74,49
11.	Fotografavimas	100	9,78	0	90,22
12.	Laukimas	8,85	9,01	91,15	90,99
13.	Ėjimas	24,82	18,71	75,18	81,29
14.	Šuns vedžiojimas	13,18	7,65	86,82	92,35
15.	Pasivaikščiavimas drauge su įsivaizduojamu subjektu	9,06	8,53	90,94	91,47

3.3 lentelėje pateiktuose duomenyse galima matyti, kad didžioji dalis veiksmų geriau atskiriami nuo likusių panaudojant *NNet* klasifikatorių. Naudojant *TreeBagger* klasifikatorių geriau atskirti veiksmai: apsipirkinėjimas, fotografavimas, ėjimas, šuns vedžiojimas ir pasivaikščiavimas drauge su įsivaizduojamu subjektu. *TreeBagger* sugebėjo idealiai atskirti fotografavimą – 100 %, abu klasifikatoriai gerai atskyrė sėdėjimą kėdėje. *NNet* klasifikatorius lyginant su *TreeBagger* geriau pasirodė klasifikuojant veiksmų atlikimą sėdimoje pozicijoje.

3.1.2.5. Atpažįtamų veiksmų mažinimas

Norint patikrinti kaip atpažinimas vykdomas su keliomis klasėmis, buvo palikti tik 4 bendroje visumoje geriau atpažinti veiksmai iš visų 15 veiksmų:

1. Valgymas;
2. Sėdėjimas ant kėdės;
3. Veiksmų atlikimas sėdimoje pozicijoje;
4. Ėjimas.

Strategijai patikrinti sudaryti trys modeliai: *TreeBagger* (medžių sk. 100, požymių vektorius dydis – 765 reikšmės), *NNet* (neuronų sk. 50, požymių vektorius dydis – 765 reikšmės) ir *LibSVM* (požymių vektorius dydis – 189 reikšmės), visi modeliai apmokyti su pirmojo subjekto duomenimis, testuoti su 1/4 visų duomenų, rezultatai pateikiami 3.4 lentelėje.

3.4 Lentelė. 4 klasių atpažinimas naudojant pozų požymių vektorių

Eil. Nr.	Klasifikatorius	Veiksmo atpažinimas, %			
		1 veiksmas	2 veiksmas	3 veiksmas	4 veiksmas
1.	<i>TreeBagger</i>	32,33	91,02	81,25	95,65
2.	<i>NNet</i>	58,34	77,95	78,82	39,89
3.	<i>LibSVM</i>	16,75	66,16	44,67	99,50

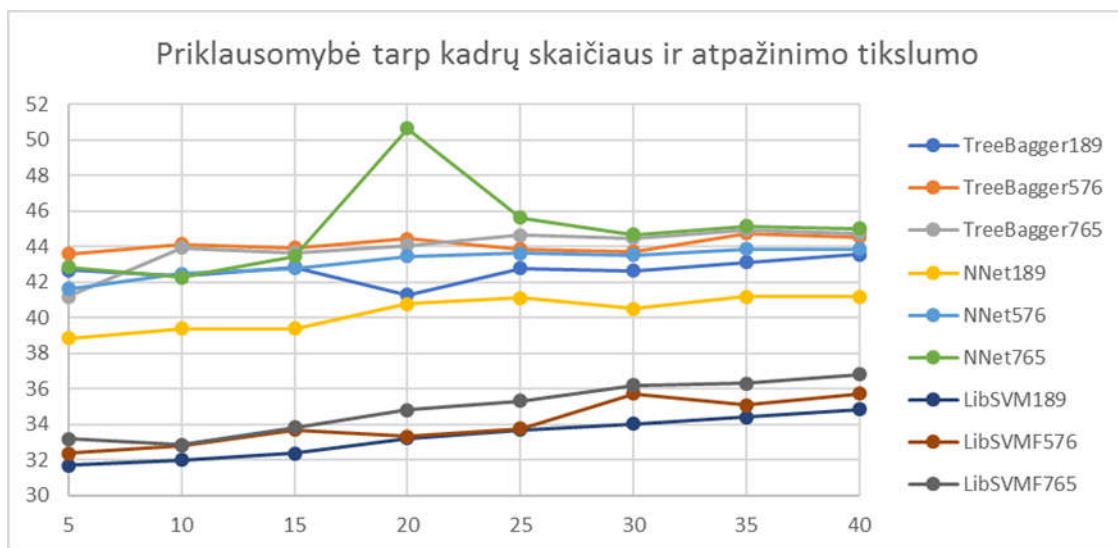
Analizuojant 3.4 lentelėje pateiktus duomenis galima pastebėti, kad nepaisant mažiausio požymių vektoriaus ilgio *LibSVM* klasifikatorius net 99,50 % tikslumu atpažino ėjimą, *NNet* klasifikatorius geriausiai iš visų trijų atpažino valgytą – 58,34 %, nors tikslumas sąlyginai nedidelis, bet net 25,99 % aukštesnis už to pačio veiksmo atpažinimą *TreeBagger* klasifikatoriumi. *TreeBagger* klasifikatorius puikiai atskyrė 3 iš 4 veiksmų aukštesniu nei 80 % tikslumu.

3.2. Klasifikavimas naudojant pozos požymių vektorių kadru seką

Dėl didelio duomenų kiekio veiksmams atpažinti išbandytos tam tikros strategijos, panaudojant tik pozas (žr. 3.1.2. Skyriuje), norint gauti didesnę sistemos veiksmų atpažinimo tikslumą tikslinga naudoti pozų sekas. Kaip parodė prieš tai atliktas tyrimas, tikslingiausia naudoti didelių duomenų kiekio strategiją (žr. 3.1.2.1. Skyriuje), kurios metu gautas tikslumas siekė 30,74 % *TreeBagger* klasifikatoriaus atveju, taip pat išbandoma neblogus rezultatus pasiekusi balsavimo strategija (žr. 3.1.2.3. Skyriuje).

3.2.1. Klasifikavimo tikslumo priklausomybė nuo kadru skaičiaus

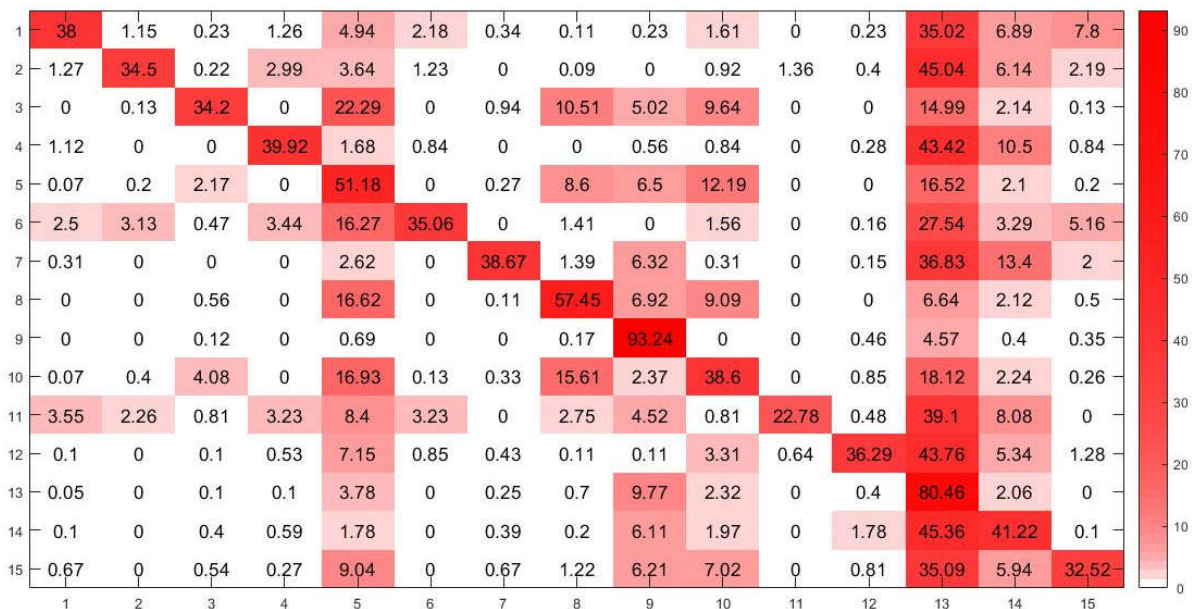
Tyrimas atliekamas norint rasti tikslingiausią kadru skaičių, klasifikatorių bei požymių vektoriaus dydį tolimesnei tyrimų eigai, rezultatai pateikiami 3.9 pav.



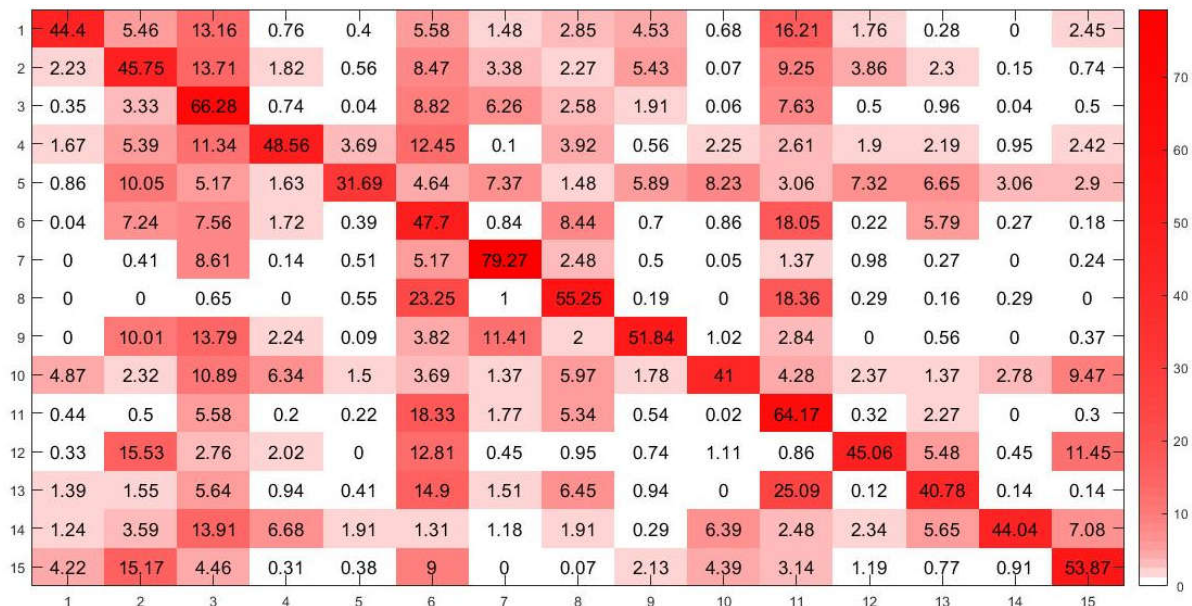
3.9 pav. Klasifikavimo tikslumo priklausomybė nuo kadru skaičiaus

Iš 3.9 paveiksle pateiktų duomenų matyti, kad didinant kadru skaičių didėja ir atpažinimo tikslumas. Prasčiausias atpažinimo tikslumas gautas naudojant *LibSVM* klasifikatorių, geriausias naudojant *NNet*. Visais atvejais geriausi tikslumo rezultatai gauti, kai požymių vektorių sudarė 765 reikšmės. Geriausius tikslumo rezultatus pasiekusių *TreeBagger765* (kadru skaičius – 35,

atpažinimo tikslumas – 44,94 %) ir *NNet765* (kadrų skaičius – 20, atpažinimo tikslumas – 50,64 %) modelių sumaišymo matricos pateikiamos 3.10 ir 3.11 pav.



3.10 pav. *TreeBagger765* modelio sumaišymo matrica, kai kadrų sk. – 35



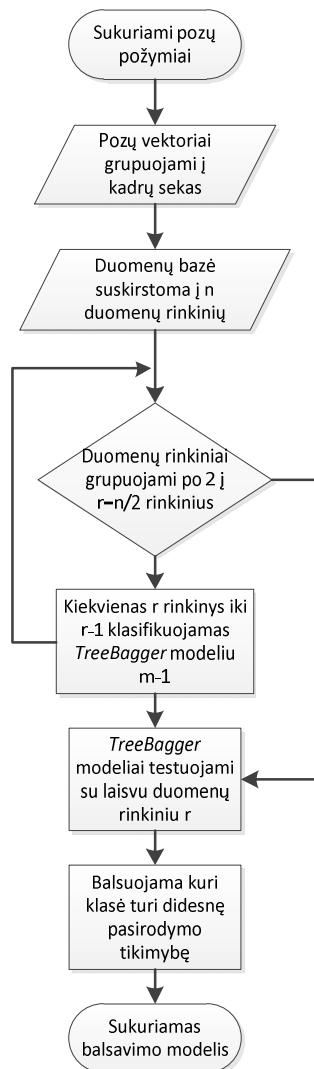
3.11 pav. *NNet765* modelio sumaišymo matrica, kai kadrų sk. – 20

3.10 paveiksle pateiktoje sumaišymo matricoje galima pastebėti, kad *TreeBagger* modelis labiausiai maišo visas klases su 13-ąja klase, ėjimu. Lyginant 3.10 ir 3.11 paveiksluose gautus rezultatus galima pastebėti, kad skirtingi klasifikatoriai geriau atpažįsta skirtingus veiksmus, *TreeBagger* klasifikatorius geriausiai atpažino 5, 8, 9, 13 klases, kai tuo tarpu *NNet* klasifikatorius

geriausiai pasirodė atpažįstant 3, 7, 11, 15 klases (veiksmų numeraciją žr. 2.1. skyriuje, veiksmai pateikiami ta pačia numeracija).

3.2.2. Balsavimas

Kadangi atpažįstant veiksmus iš pozų, ši strategija (žr. 3.1.2.3. Skyriuje) pasiekė ganėtinai gerą rezultatą, dėl to ji išbandoma ir su pozų kadrų seka. Šios strategijos algoritmas pateikiamas 3.12 pav.



3.12 pav. Balsavimo strategijos algoritmas

Kadangi 3.2.1. skyriuje atliktame tyrime stabiliausius ir ganėtinai gerus rezultatus parodė *TreeBagger765* klasifikatorius su 35 kadrų seka, dėl to jis ir pasirenkamas šios strategijos išbandymui. Tyrimo rezultatai bei kiekvieno modelio duomenų rinkiniai pateikiami 3.5 lentelėje.

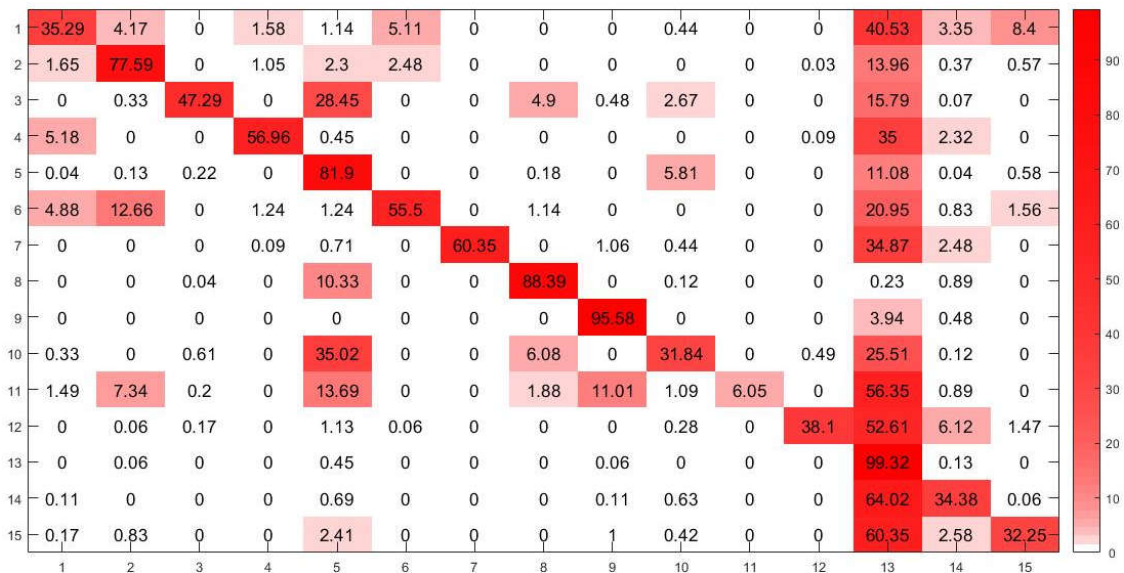
3.5 Lentelė. Modelių sudarytų naudojant kadrų sekas naudotų balsavimui atpažinimo tikslumai

Modelio Nr.	Modelio apmokymui/testavimui naudoti duomenų rinkiniai	Bendras visų veiksmų atpažinimo tikslumas, %
1.	Apmokymas su rinkiniais: 1, 2 Testavimas su rinkiniais: 7, 8	19,55
2.	Apmokymas su rinkiniais: 3, 4 Testavimas su rinkiniais: 7, 8	66,83
3.	Apmokymas su rinkiniais: 5, 6 Testavimas su rinkiniais: 7, 8	24,34
4.	Apmokymas su rinkiniais: 1, 3 Testavimas su rinkiniais: 7, 8	45,03
5.	Apmokymas su rinkiniais: 2, 4 Testavimas su rinkiniais: 7, 8	43,25
6.	Apmokymas su rinkiniais: 3, 5 Testavimas su rinkiniais: 7, 8	43,49
7.	Balsavimo modelis Testavimas su rinkiniais: 7, 8	56,05

3.5 lentelėje pateiktuose duomenyse galima pastebėti, kad balsavimo modelis pasiekė ganėtinai neprastą 56,05 % atpažinimo tikslumą, geresnį rezultatą pasiekė tik antrasis modelis, kuris siekė 66,83 %. Pirmasis ir trečiasis modeliai davė labai prastą rezultatą lyginant su kitais modeliais, tačiau tas per daugiausiai neįtakoją balsavimo modelio, kadangi naudojant balsavimo strategiją eliminuojamos klasės su mažesne atpažinimo tikimybe, tų modelių, kurie turi silpnesnes taisyklės, tokiu būdu išryškinant stipriąsias taisykles. Dėl šios sąlygos naudojant tą patį modelį su kitais testavimo duomenimis didesnė tikimybė, kad jie bus suklasifikuoti tiksliau. Geriausius rezultatus pasiekusių antrojo ir balsavimo modelio sumaišymo matricos pateikiamos 3.13, 3.14 pav.



3.13 pav. Modelio Nr. 2 sumaišymo matrica



3.14 pav. Balsavimo modelio sumaišymo matrica

3.14 paveiksle pateiktoje sumaišymo matricoje galima pastebėti, kad prasčiausiai atpažinta 11 klasė – fotografavimo veiksmas, atpažinimo tikslumas tik 6,05 %. 3.13 paveiksle pateiktoje antrojo modelio sumaišymo matricoje matyti, kad beveik visi veiksmai atpažinti geriau nei balsavimo modelyje. Balsavimo modelyje nors ir 13 veiksmas atpažintas 99,32 % tikslumu, tačiau beveik visi likusieji veiksmai maišomi su juo.

3.2.3. Keturių klasių atpažinimas

Tam, kad patikrinti, kaip tarpusavyje atpažįstami keli veiksmai, šiame tyrimo etape parenkami 4 prieš tai tyrimuose tiksliau atskirti veiksmai:

1. Diskutavimas;
2. Sėdėjimas ant kėdės;
3. Veiksmų atlikimas sėdimoje pozicijoje;
4. Ėjimas.

Veiksmams klasifikuoti naudojami trys modeliai: *TreeBagger* (medžių sk. 100), *NNet* (neuronų sk. 50) ir *LibSVM*. Visais atvejais klasifikuojami veiksmai sudaryti iš 30 kadru sekos, požymių vektorių dydis – 765 reikšmės. Modeliai apmokyti su pirmojo subjekto duomenimis, testuoti su penktojo subjekto duomenimis, tyrimo rezultatai pateikiami 3.6 lentelėje.

3.6 Lentelė. 4 klasių atpažinimas naudojant kadru sekas

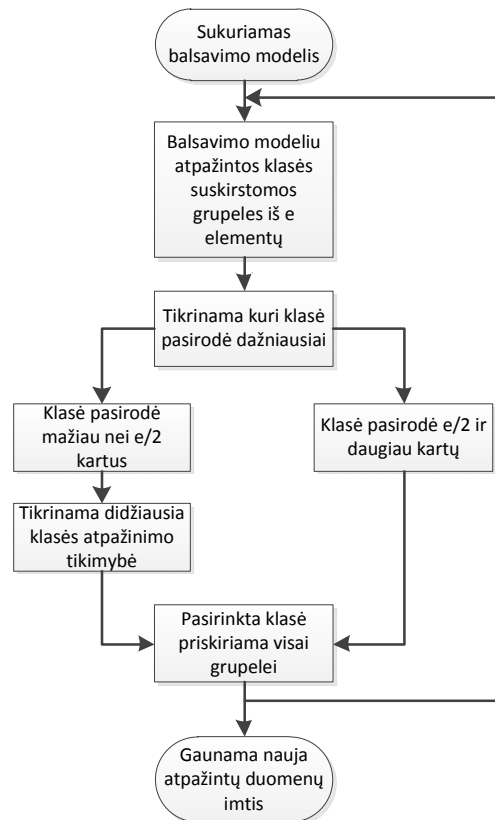
Eil. Nr.	Klasifikatorius	Veiksmo atpažinimas, %			
		1 veiksmas	2 veiksmas	3 veiksmas	4 veiksmas
1.	<i>TreeBagger</i>	16,83	95,36	92,71	98,56
2.	<i>NNet</i>	94,20	92,14	84,89	23,99
3.	<i>LibSVM</i>	32,13	82,60	69,21	89,57

Analizuojant 3.6 lentelėje pateiktus duomenis galima pastebėti, kad geriausiai pasirodė *TreeBagger* klasifikavimo algoritmo modelis pasiekęs 75,87 % bendrą visų veiksmų atpažinimo tikslumą, ne daug nusileido *NNet* modelis – 73,81 %, o prasčiausiai pasirodė *LibSVM* modelis pasiekęs tik 68,38 % tikslumą. Iš pateiktų duomenų matyti, kad 1-ąjį veiksmą iš visų klasifikatorių geriausiai išskiria *NNet* klasifikatorius, o ęjimą – *TreeBagger*.

3.2.4. Suklasifikuotų duomenų grupavimas

Analizuojant atpažįstamų klasių duomenis su realiomis pastebėta, kad atpažinimo procese pasitaiko trikdžių, esant pasikartojančiam veiksmui, pavyzdžiui ęjimui besitęsiant minutę, kai kada atpažįstama klasė pasitaiko ne ęjimas, bet šuns vedžiojimas. Tokių trikdžių atsiradimo priežastis dažniausiai galima dėl duomenų supanašėjimo (tam tikru momentu gauti duomenys aprašantys veiksmus ganėtinai panašūs vieni į kitus). Bandant anuluoti šiuos trikdžius priimama, kad esant pasikartojančiam veiksmui, jis negalėjo staiga pakisti į kitą veiksmą ir vėl sugrįžti, kadangi žmogus, kaip judanti sistema ganėtinai inertiška ir einant negali per kelias milisekundes atsisėsti, atsistoti ir vėl eiti.

Šiuo metodu stengiamasi paimti tam tikrą atpažintų klasių seką ir balsuoti, lyginti kurios klasės pasirodymo tikimybė labiausiai tikėtina. Didžiausia metodo spraga yra ta, kad perėjimuose tarp skirtingų veiksmų mažiau nei puse grupelės duomenų gali būti priskirta neteisingam veiksmui. Tiriamuoju atveju ši spraga ganėtinai nereikšminga, kadangi dirbama su dideliais duomenų kiekiais ir palyginus su grupelės dydžiu veiksmi kinta retai. Duomenų grupavimui pasirenkamas balsavimo strategijos modelis (žr. 3.2.3. Skyriuje), kadangi šis modelis pasiekė didžiausią atpažinimo tikslumą, tarp skirtingose strategijose gautų modelių, kuris siekė 56,05 %. Duomenų grupavimas atliekamas pagal 3.15 pav. pateiktą algoritmą.



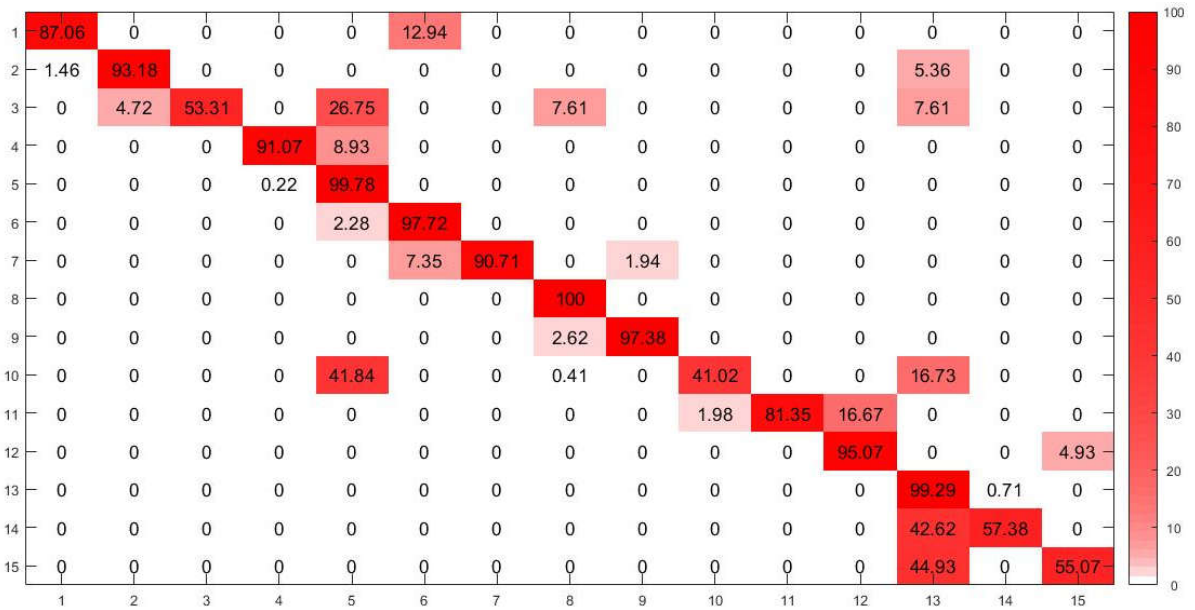
3.15 pav. Duomenų grupavimo algoritmas

Norint surasti tinkamiausią grupelės dydį, atliekamas sekantis tyrimas nustatyti priklausomybę tarp grupelės dydžio ir bendrojo visų veiksmų atpažinimo tikslumo. Tyrimo rezultatai pateikiami 3.16 pav.



3.16 pav. Tikslumo priklausomybė nuo duomenų grupelės dydžio

3.16 paveiksle pateiktoje priklausomybėje matyti, kad didėjant duomenų grupelės dydžiui atpažinimo tikslumas taip pat didėja, tačiau atsiranda svyravimai iki 4 % atpažinimo tikslumo. Geriausias 82,63 % atpažinimo tikslumas gautas, kai grupelės dydis 205, šio modelio sumaišymo matrica pateikiama 3.17 pav.



3.17 pav. Grupavimo sumaišymo matrica, kai grupelės dydis 205

Analizuojant 3.17 paveiksle pateiktą sumaišymo matricą, galima pastebėti, kad prasčiausiai atpažintos 3, 11, 14, 15 klasės (veiksmų numeraciją žr. 2.1. skyriuje, veiksmai pateikiami ta pačia numeracija), atitinkamai 53,31 %, 41,02 %, 57,38 %, 55,07 % tikslumais. Labiausiai painiojamos 5 ir 13 klasės su kitomis klasėmis. 5-oji klasė labiausiai painiojama su 3 ir 10 klasėmis, 26,75 % trečios klasės ir 41,84 % dešimtos klasės duomenų klaidingai priskirti penktai klasei. 13-oji klasė labiausiai painiojama su 10, 14 ir 15 klasėmis, 16,73 % dešimtos klasės, 42,62 % keturioliktos klasės ir 44,93 % penkioliktos klasės duomenų klaidingai priskirti tryliktai klasei. Likusios klasės atpažintos su didesne tikimybe nei 80 %. 5, 8, 13 klasės atpažintos beveik 100 % arba 100 % tikimybe. Bendrasis visų penkiolikos klasių atpažinimo vidurkis 82,63 %.

REZULTATAI IR IŠVADOS

1. Pasirinkti *TreeBagger*, *NNet*, *LibSVM* klasifikatoriai tinkami didelių duomenų klasifikavimui, tačiau kiekvienas iš jų geriausiai atpažįsta skirtingas klase.
2. Pozų požymių vektoriai nėra tinkami veiksmų klasifikavimui, kadangi gautas didžiausias atpažinimo tikslumas 32,93 %.
3. Naudojant strategijas gauti geresni atpažinimo tikslumo rezultatai, paprastai suklasifikuoto modelio atpažinimo tikslumas 44,94 %, pritaikius balsavimo strategiją atpažinimo tikslumas išaugo iki 56,05 %.
4. Pritaikius duomenų grupavimo metodiką klasifikavimo modelio tikslumas padidėjo iš 56,05 % iki 82,63 %.

INFORMACIJOS ŠALTINIŲ SĄRAŠAS

1. Roanna Lun and Wenbing Zhao, A survey of applications and human motion recognition with microsoft kinect, International Journal of Pattern Recognition and Artificial Intelligence 29 leidimas, Nr. 5, 2015 m. (48 lapai), [žiūrėta: 2017 05 14]. Prieiga per internetą: <http://www.worldscientific.com/doi/abs/10.1142/S0218001415550083>.
2. J.K. Aggarwala, Lu Xiaa, Human Activity Recognition From 3D Data: A Review, [žiūrėta: 2017 05 14]. Prieiga per internetą: <http://www.sciencedirect.com/science/article/pii/S0167865514001299>.
3. Lasitha Piyathilaka and Sarath Kodagoda, Gaussian Mixture Based HMM for Human Daily Activity Recognition Using 3D Skeleton Features. 2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA), 567–572 psl., [žiūrėta: 2017 05 14]. Prieiga per internetą: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6566433&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D6566433.
4. Ahmad Jalal, Yeonho Kim, Shaharyar Kamal, Adnan Farooq, Daijin Kim, Human Daily Activity Recognition with Joints plus Body Features Representation Using Kinect Sensor, [žiūrėta: 2017 05 14]. Prieiga per internetą: <https://www.researchgate.net/publication/279298309>.
5. Enea Cippitelli, Samuele Gasparrini, Ennio Gambi, and Susanna Spinsante, A Human Activity Recognition System Using Skeleton Data from RGBD Sensors, [žiūrėta: 2017 05 14]. Prieiga per internetą: <http://www.hindawi.com/journals/cin/2016/4351435/>.
6. Georgios Th. Papadopoulos, Apostolos Axenopoulos and Petros Daras, Real-time Skeleton-tracking-based Human Action Recognition Using Kinect Data, [žiūrėta: 2017 05 14]. Prieiga per internetą: http://link.springer.com/chapter/10.1007/978-3-319-04114-8_40.
7. Lu Xia, Chia-Chih Chen, and J. K. Aggarwal, View Invariant Human Action Recognition Using Histograms of 3D Joints, [žiūrėta: 2017 05 14]. Prieiga per internetą: http://cvrc.ece.utexas.edu/Publications/Xia_HAU3D12.pdf.
8. Hossein Pazhoumand-Dar, Chiou-Peng Lam, Martin Masek, Joint movement similarities for robust 3D action recognition using skeletal data, [žiūrėta: 2017 05 14]. Prieiga per internetą: <http://www.sciencedirect.com/science/article/pii/S1047320315000462>.
9. Xiaodong Yang and YingLi Tian, Super Normal Vector for Activity Recognition Using Depth Sequences, [žiūrėta: 2017 05 14]. Prieiga per internetą: http://www.cv-foundation.org/openaccess/content_cvpr_2014/html/Yang_Super_Normal_Vector_2014_CVP_R_paper.html.

10. Heng Wang, Muhammad Muneeb Ullah, Alexander Kläser, Ivan Laptev, Cordelia Schmid, Evaluation of local spatio-temporal features for action recognition, [žiūrēta: 2017 05 14]. Prieiga per internetą: http://www.irisa.fr/vista/Papers/2009_bmvc_wang.pdf.
11. Shaharyar Kamal, Ahmad Jalal and Daijin Kim, Depth Images-based Human Detection, Tracking and Activity Recognition Using Spatiotemporal Features and Modified HMM, [žiūrēta: 2017 05 14]. Prieiga per internetą: <http://www.jeet.or.kr/LTKPSWeb/uploadfiles/be/201605/230520161417111253750.pdf>.
12. Jose M. Chaquet, Enrique J. Carmona, Antonio Fernandez-Caballero, A Survey of Video Datasets for Human Action and Activity Recognition, [žiūrēta: 2017 05 14]. Prieiga per internetą: <http://www.sciencedirect.com/science/article/pii/S1077314213000295>.
13. Oluwatoyin P. Popoola, Kejung Wang, Video-Based Abnormal Human Behavior Recognition–A Review, [žiūrēta: 2017 05 14]. Prieiga per internetą: https://www.researchgate.net/publication/270790763_Video-Based_Abnormal_Human_Behavior_Recognition–A_Review.
14. Joshua Candamo, Matthew Shreve, Dmitry B. Goldgof, Deborah B. Sapper, Rangachar Kasturi, Understanding Transit Scenes A Survey Human Behavior–Recognition Algorithms, [žiūrēta: 2017 05 14]. Prieiga per internetą: <https://pdfs.semanticscholar.org/a52f/8a7bdc19fd9d80da68cdb79a19f0c2d0f466.pdf>.
15. Xinyu Wu, Yongsheng Ou, Huihuan Qian, Yangsheng Xu, Detection System for Human Abnormal Behavior, [žiūrēta: 2017 05 14]. Prieiga per internetą: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1545205&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D1545205.
16. Ramin Mehran, Alexis Oyama, Mubarak Shah, Abnormal Crowd Behavior Detection using Social Force Model, [žiūrēta: 2017 05 14]. Prieiga per internetą: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5206641&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D5206641.
17. Yufeng Chen, Guoyuan Liang, Ka Keung Lee, Yangsheng Xu, Abnormal Behavior Detection by Multi–SVM–Bases Bayesian Network, [žiūrēta: 2017 05 14]. Prieiga per internetą: <http://ieeexplore.ieee.org/document/4295746/>.
18. Bo Wang, Mao Ye, Xue Li, Fengjuan Zhao, Jian Ding, Abnormal crowd behavior detection using high–frequency and spatio-temporal features, [žiūrēta: 2017 05 14]. Prieiga per internetą: <http://link.springer.com/article/10.1007%2Fs00138-011-0341-0?LI=true>.

19. Kan Ouivirach, Shashi Gharti, Matthew N. Dailey, Incremental behavior modeling and suspicious activity detection, [žiūrēta: 2017 05 14]. Prieiga per internetą: <http://www.sciencedirect.com/science/article/pii/S0031320312004426>.
20. Marco Bertini, Alberto Del Bimbo, Lorenzo Seidenari, Multi-scale and real-time non-parametric approach for anomaly detection and localization, [žiūrēta: 2017 05 14]. Prieiga per internetą: <http://www.sciencedirect.com/science/article/pii/S1077314211002104>.
21. Mehrosan Javan Roshtkhari, Martin D. Levine, An on-line, real-time learning method for detecting anomalies in videos using spatio-temporal composition, [žiūrēta: 2017 05 14]. Prieiga per internetą <http://www.sciencedirect.com/science/article/pii/S1077314213001239>.
22. Mehrosan Javan Roshtkhari, Martin D. Levine, Online Dominant and Anomalous Behavior Detection in Videos, [žiūrēta: 2017 05 14]. Prieiga per internetą: http://www.cv-foundation.org/openaccess/content_cvpr_2013/html/Roshtkhari_Online_Dominant_and_2013_CVPR_paper.html.
23. Thi-Lan Le, Minh-Quoc Nguyen, Thi-Thanh-Mai Nguyen, Human posture recognition using human skeleton provided by Kinect, [žiūrēta: 2017 05 14]. Prieiga per internetą: http://www.redes.unb.br/lasp/files/events/COMMANTEL13/PDFs/Papers/63_60567.pdf.
24. Raj Gupta, Alex Yong-Sang Chia, Deepu Rajan, Human Activities Recognition using Depth Images, [žiūrēta: 2017 05 14]. Prieiga per internetą: <http://dl.acm.org/citation.cfm?id=2502099&dl=ACM&coll=DL&CFID=633629284&CFTOKEN=59017633>.
25. Ping Wei, Yibiao Zhao, Nanning Zheng, and Song-Chun Zhu, Modeling 4D Human-Object Interactions for Event and Object Recognition, [žiūrēta: 2017 05 14]. Prieiga per internetą: http://vcla.stat.ucla.edu/papers_vcla/2013/human_computer_interaction_iccv2013.pdf.
26. Catalin Ionescu, Dragos Papava, Vlad Olaru, Cristian Sminchisescu, Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments, [žiūrēta: 2017 05 14]. Prieiga per internetą: <http://ieeexplore.ieee.org/abstract/document/6682899/>.
27. Manuel Fernandez-Delgado, Eva Cernadas, Senen Barro, Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?, [žiūrēta: 2017 05 14]. Prieiga per internetą: <http://www.jmlr.org/papers/volume15/delgado14a/delgado14a.pdf>.
28. Gareth James; Daniela Witten; Trevor Hastie; Robert Tibshirani (2013). An Introduction to Statistical Learning. Springer. pp. 316–321.
29. Grant Izmirlian, Application of th Random Forest Classification Algorithm to a SELDI-TOF Proteomics Study in the Setting of a Cancer Prevention Trial, [žiūrēta: 2017 05 14]. Prieiga per internetą:

- https://www.researchgate.net/publication/8500332_Application_of_the_Random_Forest_Classification_Algorithm_to_a_SELDI-TOF_Proteomics_Study_in_the_Setting_of_a_Cancer_Prevention_Trial.
30. B. Samanta, Gear fault detection using artificial neural networks and support vector machines with genetic algorithms, [žiūrēta: 2017 05 14]. Prieiga per internetu: <http://www.sciencedirect.com/science/article/pii/S0888327003000207>.
31. Chih-Chung Chang and Chih-Jen Lin, LIBSVM: A Library for Support Vector Machines, [žiūrēta: 2017 05 14]. Prieiga per internetu: <http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf>.
32. Ke Liu, Xuechen Zhang, Kei Davis, Song Jiang, Synergistic Coupling of SSD and Hard Disk for QoS-aware Virtual Memory, [žiūrēta: 2017 05 14]. Prieiga per internetu: <http://ieeexplore.ieee.org/document/6557143/>.