KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

Liutauras Butkus

ROBOTŲ REGOS NAUDOJANT GILŲJĮ MOKYMĄSI DEMONSTRACIJA

Baigiamasis magistro projektas

Vadovas
Dr. Mantas Lukoševičius

KAUNAS, 2018

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

PROGRAMŲ INŽINERIJOS KATEDRA

ROBOTŲ REGOS NAUDOJANT GILŲJĮ MOKYMĄSI DEMONSTRACIJA

Baigiamasis magistro projektas

Programų sistemų inžinerija (kodas 621E16001)

Vadovas
(parašas) Dr. Mantas Lukoševičius
(data)

Recenzentas
(parašas) Prof. dr. Robertas Damaševičius
(data)

Projektą atliko
(parašas) Liutauras Butkus
(data)

KAUNAS, 2018

KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos
(Fakultetas)
Liutauras Butkus
(Studento vardas, pavardė)
Programų sistemų inžinerija, 6211BX011
(Studijų programos pavadinimas, kodas)

„Robotų regos naudojant gilųjį mokymąsi demonstracija“

AKADEMINIO SĄŽININGUMO DEKLARACIJA

20 18 m. Gegužės 31 d.
Kaunas

Patvirtinu, kad mano, **Liutauro Butkaus**, baigiamasis projektas tema „Robotų regos naudojant gilųjį mokymąsi demonstracija“ 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)*

SANTRAUKA

Magistrinio projekto tikslas sukurti demonstracinę sistemą, kurioje mažas humanoidinis robotas, pamatęs ištiestą ranką, priima tai kaip pasisveikinimą ir reaguoja į tai. Vaizdinis rankos atpažinimas, kuris yra pagrindinė sistemos dalis, pasirodė esanti nelengva užduotis. Taigi šiuo darbu išbandžiau, kaip gerai galima išspręsti šią problemą naudojant gilųjį mokymąsi. Skirtingai nuo daugybės gestų atpažinimo tyrimų, šio darbo metu nebuvo naudojama gylio informacija, ar vaizdo įrašai, užduotis buvo atlikta naudojant statinius vaizdus. Dėl šios priežasties naudota paprasta kamera, kadangi gestas, kurį reikia atpažinti yra gana statinis. Šiai užduočiai atlikti surinkau specialų duomenų rinkinį. Buvo išbandytos skirtingos konvoliucinio neuroninio tinklo konfigūracijos ir mokymosi algoritmai. Tačiau didžiausias proveržis buvo gautas, kuomet buvo pašalintas fonas ir modelis buvo sutelktas tik į priekyje esantį asmenį.

Sukurta demonstracinė sistema gali būti naudojama kaip edukacinė priemonė siekiant pritraukti ir sudominti tiksliaisiais mokslais, mašininiu mokymusi, robotika besidominčius jaunuosius būsimus tyrėjus bei specialistus.

Butkus Liutauras. Robotų regos naudojant gilųjį mokymąsi demonstracija. Magistro baigiamasis projektas / vadovas Mantas Lukoševičius; Kauno technologijos universitetas, Informatikos fakultetas, Programų inžinerijos katedra.

Reikšminiai žodžiai: vaizdo atpažinimas, kompiuterinė rega, gilusis mokymasis, konvoliucinis neuroninis tinklas, robotika

Kaunas 2018. 52 puslapiai.

SUMMARY

The purpose of the master's project is to create a demonstration system in which a small humanoid robot, having seen an extended hand, accepts it as a greetings and reacts to it. The visual recognition of the hand, which is the main part of the system, has proven to be a difficult task. I've described in this work how to solve this problem using deep learning. Unlike many other gesture recognition research, this work did not use depth information, video clips. The task was done using static images. For this reason, a simple camera was used, since the gesture that needs to be recognized is quite static. I have collected a special dataset for this task. Various convolutional neural network configurations and learning algorithms have been tested. However, the biggest breakthrough came when the background was removed from the images and the model was focused only on the person in front.

The created demonstratior can be used as an educational tool for attracting and engaging young future researchers and specialists in exact sciences, machine learning, and robotics.

Butkus Liutauras. Machine Vision Demonstration Using Deep Learning. Master thesis in Software engineering / supervisor Dr. Mantas Lukoševičius. The Faculty of Informatics, Kaunas University of Technology.

Study field and area: Informatics, Software engineering

Key words: image recognition, computer vision, deep learning, convolutional neural networks, robotics

Kaunas, 2018. 52 p.

Turinys

1. ĮŽANGA 11
   1.1. Dokumento paskirtis 11
   1.2. Santrauka 11
2. ANALITINĖ DALIS 12
   2.1. Projekto tikslas ir adresatas 12
   2.2. Įranga 12
      2.2.1. Aparatinė įranga 12
      2.2.2. Programinė įranga 14
   2.3. Bibliotekos pasirinkimas 14
   2.4. Egzistuojantys sprendimai 14
      2.4.1. The Wolfram Language Image Identification Project 15
      2.4.2. Clarifai 16
      2.4.3. Imagga 16
      2.4.4. Gestų atpažinimo sistema, naudojant gilų mokymąsi 17
   2.5. Situacijos Lietuvoje įvertinimas 18
3. PROJEKTINĖ DALIS 19
   3.1. Projekto paskirtis 19
   3.2. Sistemos paskirtis ir panaudos atvejai 19
   3.3. Bendrieji apribojimai 20
      3.3.1. Reikalavimai sistemos išvaizdai 20
      3.3.2. Reikalavimai panaudojamumui 21
   3.4. Reikalavimai sistemai 24
   3.5. Sistemos architektūra 25
   3.6. Sistemos statinis vaizdas 26
      3.6.1. Duomenų valdymo paketas 26
      3.6.2. GUI paketas 27
      3.6.3. Apmokymo valdymo paketas 27
      3.6.4. Atpažinimo valdymo paketas 27
      3.6.5. Roboto valdymo paketas 28
   3.7. Veiklos diagrama 28
   3.8. Būsenų diagramos 31
   3.9. Sekų diagramos 32
4. TYRIMO IR EKSPERIMENTINĖ DALYS 34
   4.1. Esamas funkcionalumas 34
   4.2. Neuroninio tinklo sukūrimas 34
   4.3. Neuroninio tinklo apmokymo procesas 35
   4.4. Validacijos procesas 36

4.5. Testavimo procesas ... 36
4.6. Duomenų rinkinio paruošimas ... 36
4.7. Fono šalinimas ... 37
4.8. Eksperimentų rezultatai ir jų analizė ... 39
4.9. Realus sistemos išbandymas ... 41
5. IŠVADOS ... 43
6. LITERATŪRA ... 44
7. PRIEDAI ... 46

Paveikslėlių sąrašas

Pav. 2.1 Robotas ir jo panaudojimo pavyzdys 13
Pav. 2.2 Roboto sąnarių valdymo sąsaja 13
Pav. 2.3 “The Wolfram Language Image Identification Project” vartotojo sąsaja 15
Pav. 2.4 “Clarifai” vartotojo sąsaja 16
Pav. 2.5 „Imagga“ vartotojo sąsaja 17
Pav. 2.6 Pavyzdinės sistemos duomenų pavyzdžiai 17
Pav. 3.1 Sistemos panaudos atvejai 20
Pav. 3.2 Sistemos apmokymo modelis 25
Pav. 3.3 Sistemos paleidimo modelis 26
Pav. 3.4 Abstraktus sistemos skaidymas į paketus 26
Pav. 3.5 Paketo Duomenų valdymas klasių diagrama 27
Pav. 3.6 Paketo GUI klasių diagrama 27
Pav. 3.7 Paketo Apmokymo valdymas klasių diagrama 27
Pav. 3.8 Paketo Atpažinimo valdymas klasių diagrama 28
Pav. 3.9 Paketo Roboto valdymas klasių diagrama 28
Pav. 3.10 Apmokymo veiklos diagrama 29
Pav. 3.11 Atpažinimo veiklos diagrama 30
Pav. 3.12 Apmokymo ir paleidimo būsenų diagrama 31
Pav. 3.13 Roboto valdymo būsenų diagrama 32
Pav. 3.14 Apmokymo sekų diagrama 32
Pav. 3.15 Sistemos paleidimo sekų diagrama 33
Pav. 4.1 Konvoliucinio neuroninio tinklo architektūra 35
Pav. 4.2 Nuotraukų pavyzdžiai: teigiami viršuje, neigiami apačioje 37
Pav. 4.3 Tos pačios nuotraukos prieš ir po fono nuėmimo 38
Pav. 4.4 Apmokymo duomenimis, be fono pašalinimo, rezultatų grafikas 39
Pav. 4.5 Apmokymo duomenimis, be fono pašalinimo, rezultatų grafikas 40
Pav. 4.6 Sistemos pristatymas "KTU Technorama 2018" technologijų parodoje 42

Lentelių sąrašas

Lentelė 2.1 Reikalinga programinė įranga 12
Lentelė 3.1 Nefunkcinis reikalavimas išvaizdai 20
Lentelė 3.2 Nefunkcinis reikalavimas išvaizdai 20
Lentelė 3.3 Nefunkcinis reikalavimas panaudojamumui 21
Lentelė 3.4 Nefunkcinis reikalavimas panaudojamumui 21
Lentelė 3.5 Nefunkcinis reikalavimas panaudojamumui 21
Lentelė 3.6 Nefunkcinis reikalavimas panaudojamumui 22
Lentelė 3.7 Nefunkcinis reikalavimas panaudojamumui 22
Lentelė 3.8 Nefunkcinis reikalavimas panaudojamumui 22
Lentelė 3.9 Nefunkcinis reikalavimas panaudojamumui 23
Lentelė 3.10 Nefunkcinis reikalavimas panaudojamumui 23
Lentelė 3.11 Nefunkcinis reikalavimas panaudojamumui 23
Lentelė 3.12 Nefunkcinis reikalavimas panaudojamumui 23
Lentelė 3.13 Funkcinis reikalavimas 24
Lentelė 3.14 Funkcinis reikalavimas 24
Lentelė 3.15 Funkcinis reikalavimas 24
Lentelė 3.16 Funkcinis reikalavimas 25
Lentelė 4.1 Skirtingų apmokymų validacijos lentelė 41

SANTRUMPŲ IR ŽENKLŲ AIŠKINIMO ŽODYNAS

API – apibūdininmų, protokolų, įrankių rinkinys (ang. Application programing interface)

CNN – konvoliucinis neuroninis tinklas (ang. Convolutional neural network) - susideda iš daugybės sluoksnių, ir, apžiūrinėdamas paveikslėlį, pamažu atvaizdo fragmentus priskiria požymiams.

Dirbtinis neuroninis tinklas - tarpusavyje sujungtų dirbtinių neuronų grupė. Ši technologija mėgdžioja žmogaus galvos smegenų darbą – tiksliau neuronų veikimą.

Duomenų padidinimas – (ang. Data augmentation) – kalbant apie paveikslėlius, tai reiškia, paveikslėlių skaičius padidėjimą duomenų rinkinyje.

FPS – kadrai per sekundę (ang. Frames per socond)

Gilusis mokymasis – mašininio mokymo būdas (ang. Deep learning) – teorinių apribojimų neturintis algoritmas, besiremiantis tuo, kaip dirba žmogaus smegenys. Kuo daugiau jam duodi duomenų ir laiko jiems apskaičiuoti, tuo jis geresnis.

RGB - spalvų maišymo sistema, kurioje naudojamos trys, žmogaus akių receptorius atitinkančios spalvos: raudona (ang. Red), žalia (ang. Green) ir Mėlyna (ang. Blue).

RNN – rekurentinis neuroninis tinklas (ang. Recurrent neural network) – yra dirbtinis neuroninis tinklas, apimantis kryptinius ciklus atmintyje. Vienas iš rekurentinių neuroninių tinklų aspektų yra gebėjimas remtis ankstesniais tinklais su fiksuoto dydžio įvesties vektoriais ir išvesties vektoriais.

SSH - tai tinklo protokolas, aprašantis apsaugotą kliento prisijungimą prie serverio aplinkos (shell) ir komandų vykdymą.

Vidutinė kvadratinė klaida - Statistikoje vidutinė kvadratinė klaida (MSE) arba vidutinis kvadrato nuokrypis (MSD) nustato klaidų ar nukrypimų kvadratų vidurkį, ty skirtumą tarp įvertinimo priemonės ir kas apskaičiuojama.

1. ĮŽANGA

1.1. Dokumento paskirtis

Šio dokumento paskirtis supažindinti su magistrinio darbo tema, aprašyti jo eigą, tyrimą ir apžvelgti gautus rezultatus.

1.2. Santrauka

Dirbtinis neuroninis tinklas, tai yra struktūra skirta apdoroti informacijai, kuri yra sukurta remiantis biologinės nervų sistemos analogu [1]. Ši struktūra yra sudaryta iš daugelio tarpusavyje susijusių skaičiavimus atliekančių komponentų, kurie yra vadinami neuronais. Tam, kad neuroninis tinklas galėtų išspręsti uždavinius, jis turi būti apmokomas iš turimų pavyzdžių. Šie pavyzdžiai turi būti kruopščiai parenkami, nes kitaip neuroninis tinklas gali ilgai mokytis arba išvis veikti nekorektiškai. Tinklo apmokymas yra vykdomas giliojo mokymo metu keičiant tarp neuronų esančių jungčių svorius.

Gilusis mokymasis yra vis labiau aptarinėjama tema kalbant apie dirbtinį intelektą. Kaip subkategorija mašininio mokymo, gilusis mokymasis susijęs su neuroninių tinklų naudojimu, siekiant pagerinti tokias sritis kaip, pavyzdžiui, kalbos atpažinimas, kompiuterinė rega ir natūralios kalbos apdorojimas. Jis greitai tampa viena iš labiausiai geidžiamų kompiuterinių mokslo sričių.

Kuriama sistema pasižymės vaizdo atpažinimo sistema, kuomet ant roboto sumontuota kamera fiksuos vaizdą ir siųs jį dirbtiniam neuroniniam tinklui. Tuo tarpu dirbtinis neuroninis tinklas, sudarytas iš daugelio dirbtinių neuronų, apdoros vaizdą pagal konvoliucinį principą, kuomet vaizdas nagrinėjamas pagal tam tikrus principus ir taip galiausiai gaus koeficiento reikšmę, kuri nurodys, ar gautame vaizde yra laukiamas objektas (šiuo atveju į žmogaus ranką), ar ne.

Taigi apibendrinus, šio projekto pagrindinis tikslas buvo sukurti programinę įrangą, kuri leistų pademonstruoti apmokyto neuroninio tinklo vaizdo atpažinimą realiuoju laiku atitinkamai perduodant komandą „ištiesti ranką“ robotui.

2. ANALITINĖ DALIS

2.1. Projekto tikslas ir adresatas

Projekto tikslai:

- Sukurti sistemą magistriniam darbui
- Išmokti pritaikyti mašininio mokymosi žinias praktikoje
- Suprasti konvoliucinio neuroninio tinklo vaizdų apdorojimo principus

Potencialūs sistemos vartotojai:

- Žiūrovai

2.2. Įranga

2.2.1. Aparatinė įranga

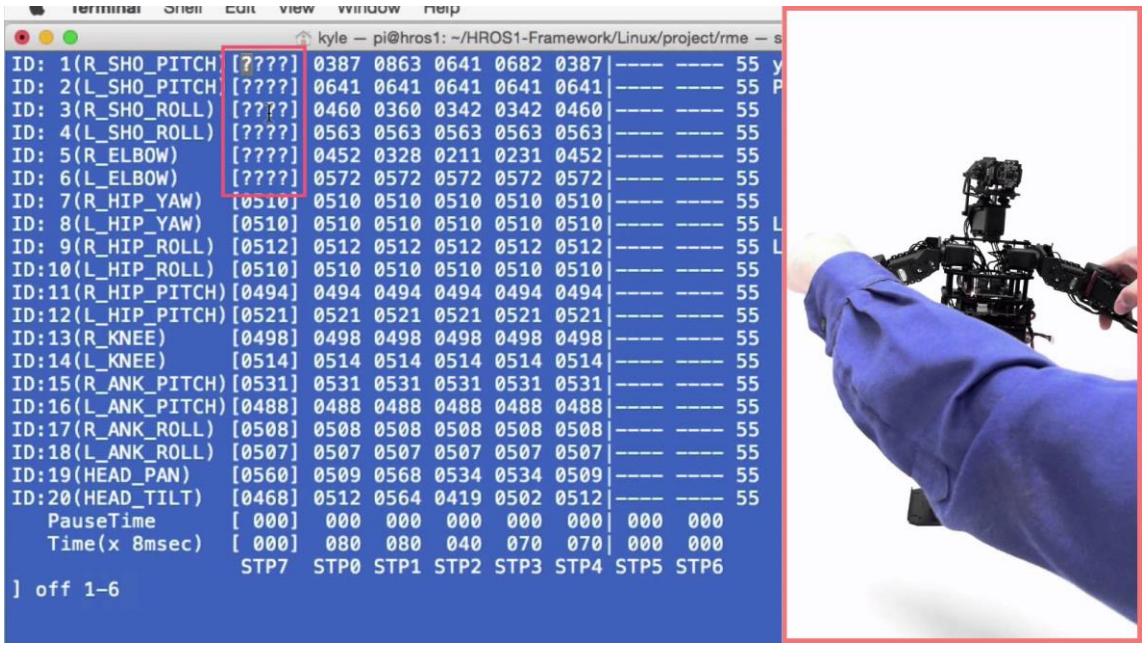
Šiam darbui atlikti reikalinga aparatinė įranga pateikta 2.1 lentelėje.

Lentelė 2.1 Reikalinga programinė įranga

Nr.	Pavadinimas	Paskirtis
1	Robotas: Interbotix Labs OS1 Humanoid Endoskeleton Robot	Fiziniam sistemos atvaizdavimui.
2	Kamera: Pixy R1.3A	Aplinkos fiksavimui.
3	Personalinis kompiuteris su viena iš platformų (pasirinktinai): Linux, Raspberry Pi, Intel Edison, Windows	Platforma kurioje veiks sistema.
4	MagicDraw ir kita įranga	Reikalinga projektavimo darbams atlikti.
5	Klasteris	Lygiagretiems skaičiavimas (jeigu įvertinus skaičiavimų sudėtingumą bus reikalingas)

Pav. 2.1 Robotas ir jo panaudojimo pavyzdys

Šiame projekte buvo naudojamas HR-OS1 Humanoid Endoskeleton robotas [2]. Jis pavaizduotas 2.1 paveiksle. Jis turi integruota kompiuterį su Linux operacine sistema. Kompiuteris pasižymi "Intel Atom" procesoriumi, kuris atsakinga už roboto komandų paleidimą. HR-OS1 yra, keičiama, modulinė, humanoidinė robotų kūrimo platforma. Ji turi įmontuotą programinę įrangą, kuri iškviečia roboto veiksmus. Roboto programinės įrangos sąsają pavaizduota 2.2 paveiksle.



Pav. 2.2 Roboto sąnarių valdymo sąsaja

Roboto sąsaja naudojama, kai modelis pradeda prognozuoti naujas nuotraukas. Po to, kai CNN grąžina tikimybę, atitinkama komanda siunčiama į roboto sąsają. Tuomet roboto sąsaja nuskaito įvesties reikšmę ir jeigu reikšmė teigiama, paleidžia komandą, kad robotas pakeltų ranką.

2.2.2. Programinė įranga

Reikalingą programinę įrangą sudaro:

- Pixymon – vaizdo kameros vartotojo sąsaja
- Spyder IDE, ar kita teksto redagavimo programa
- Anaconda – atvirojo kodo mokslo platforma paremta Python programavimo kalba.
- PuTTY - SSH ir telnet klientas

2.3. Bibliotekos pasirinkimas

Projektui įgyvendinti pasirinkau egzistuojančią giliojo mokymosi bilioteką Keras.

Keras [3] yra "Theano" ir "TensorFlow" giliojo mokymosi biblioteka. Tai aukšto lygio neuronų tinklų biblioteka, sukurta su "Python" programavimo kalba ir galinti veikti kartu su TensorFlow arba Theano giliojo mokymosi bibliotekomis. Ji buvo sukurta sutelkiant dėmesį į greitą eksperimentavimą. Galimybė pereiti nuo idėjos prie rezultatų su kuo mažesniu vėlavimu yra labai svarbu veiksmams atlikti. Keras giliojo mokymosi biblioteka leidžia lengvai ir greitai kurti prototipus. Ji palaiko tiek konvoliucinius tinklus, tiek rekurentinius tinklus, tiek jų derinius. Keras taip pat palaiko sutartines ryšių schemas (įskaitant daugelio įvesties ir daugialypės išvesties mokymą) ir nuosekliai veikia su CPU ir GPU. Pagrindinė Keras duomenų struktūra yra modelis, kuris apibrėžia tinklo sluoksnius. Pagrindinis modelio tipas yra nuoseklusis (ang. Sequential) modelis. Keras pagrindinis principas yra moduliacija. Modelis yra suprantamas kaip atskirų, visiškai konfigūruojamų modulių seka arba grafika, kurią galima sujungti su kuo mažiau apribojimų. Visų pirma, neuroniniai sluoksniai, sąnaudų funkcijos, optimizatoriai, inicializavimo schemos, aktyvavimo funkcijos, sureguliavimo schemos yra savarankiški moduliai, kuriuos vartotojai gali sujungti, kad būtų sukurti nauji modeliai. Kiekvienas modulis turi būti kuo mažesnis ir paprastesnis. Kiekviena kodo eilutė turėtų būti suprantama iš pirmo žvilgsnio. Nauji moduliai kuriami paprasčiausiai pridedant naujas klases ar funkcijas, o esami moduliai imami kaip pavyzdžiai.

2.4. Egzistuojantys sprendimai

Kadangi darbo sritis pasirinkta gana specifinė, konkrečių identiškų sprendimų rasti yra sunku, todėl iš pradžių pabandysiu bendrai apžvelgti, kur vaizdų atpažinimo technologijos gali būti panaudotos šiuolaikinėse sistemose, o po to pateiksiu apžvalgą poros panašių projektų į manąjį.

2.4.1. The Wolfram Language Image Identification Project

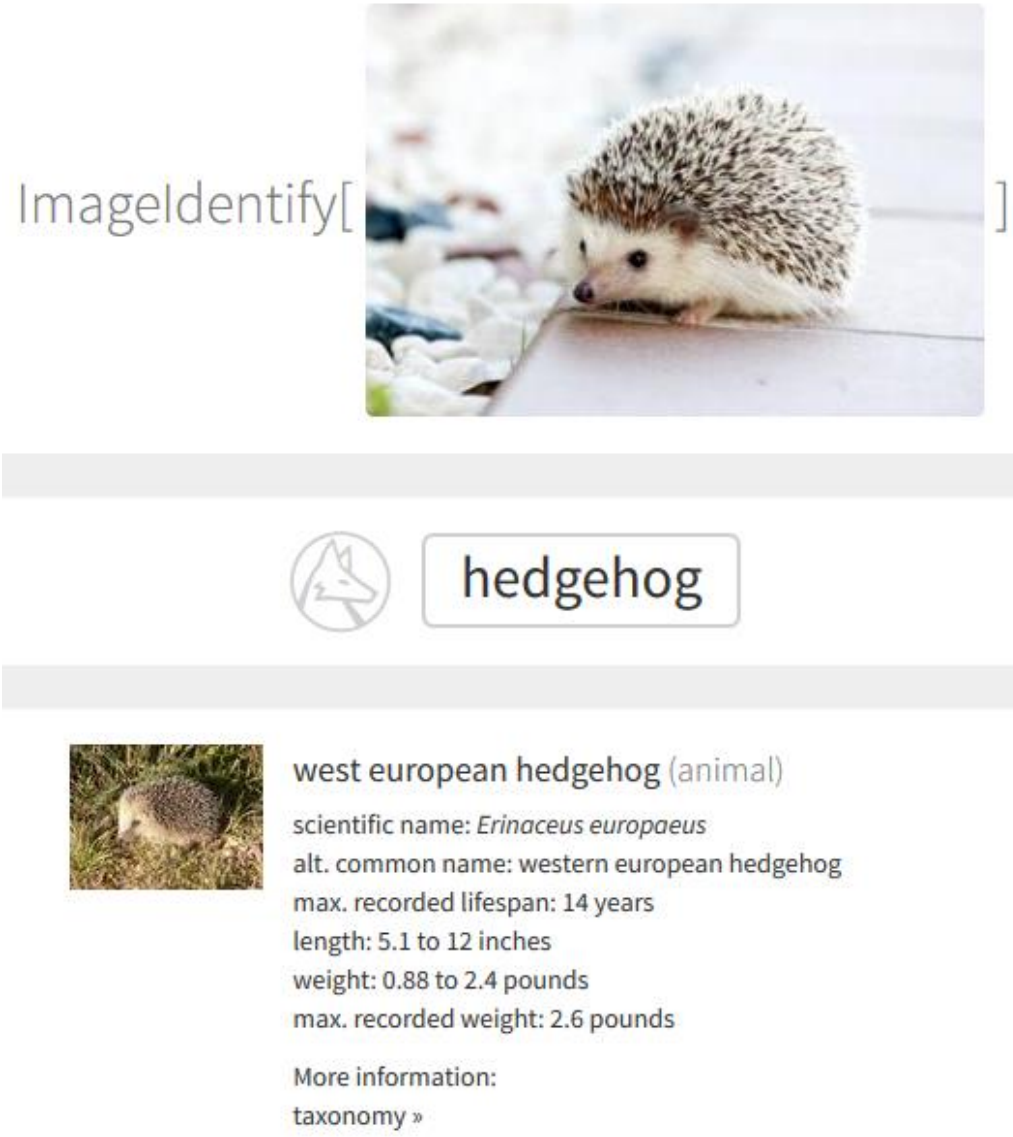
Paveikslėlių identifikavimo projektas, kuris pristato vieną „Wolfram Language“ programavimo kalbos „ImageIdentify“ funkcijos veikimą [4].

„ImageIdentify“ yra tipinė automatizuota „superfunkcija“, viena iš daugelio, kurios gerai žinomos „Wolfram Language“ kalboje [5]. Vartotojui tai tik viena lengvo naudojimo funkcija, kuri gali būti įterpta ir naudojama rašant kodą, tačiau viduje, ji remiasi sudėtingais algoritmais.

„ImageIdentity“ yra viena iš mašininio mokymosi funkcijų „Wolfram Language“ kalboje. Iš esmės tai vidinis klasifikatorius, apmokytas didelio kiekio vaizdų rinkiniu.

Pagrindinė dalis dabartinės „ImageIdentify“ klasifikatoriaus remiasi giliaisiais neuroniniais tinklais.

„ImageIdentify“ yra laipsniškai tobulinamas vaizdais, kuriuos svetainėje įkelia jos vartotojai.



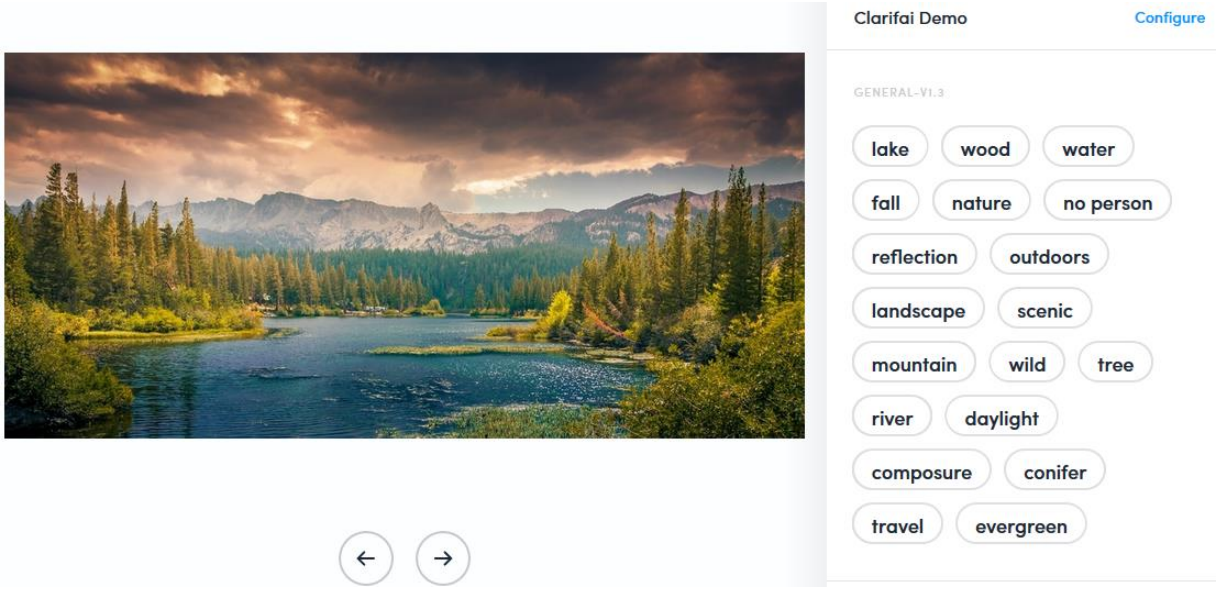
Pav. 2.3 “The Wolfram Language Image Identification Project” vartotojo sąsaja

2.4.2. Clarifai

Clarifai [6] – nuotraukų ir vaizdo įrašų atpažinimo API. Clarifai automatiškai pažymi visas nuotraukas ir vaizdo įrašus, tam kad palengvintų organizuotai valdyti ir ieškoti duomenų tarp viso įkelto turinio.

Clarifai leidžia pačiam vartotojui apmokyti platformą, kad ji atpažintų naujus objektus ir idėjas, naudojant įkeltus duomenis.

Taip pat platforma leidžia ieškoti turinio pagal vizualinį panašumą, žodžio žymą, ar abiejų kombinaciją, tam kad gauti rekomendacijas ir susijusį turinį realiu laiku.



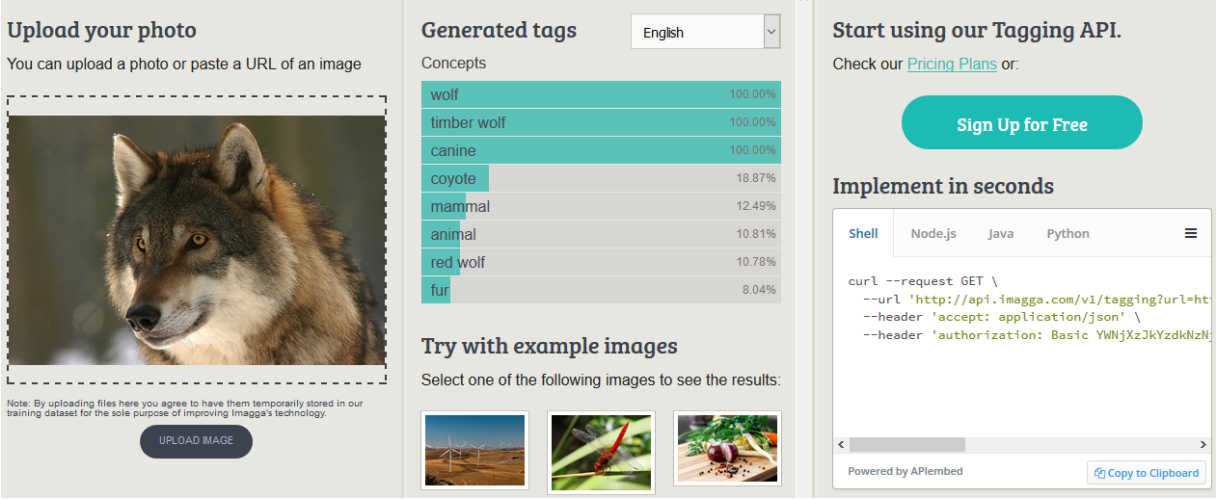
Pav. 2.4 "Clarifai" vartotojo sąsaja

2.4.3. Imagga

„Immaga" [7] – vaizdų atpažinimo platforma-paslauga siūlanti vaizdų žymėjimo API programuotojams ir verslo klientams, kaip priemonė kurti kintamo mastelio, vaizdo intensyvumo programas debesyse.

„Immaga" siūlo išskirtinio našumo automatizuotą vaizdų atpažinimą [8], kuris gali preciziškai atpažinti net didelių matmenų ir apimties vaizdus.

„Imagga" taip pat siūlo savaiminio mokymosi sprendimą, kuomet algoritmas apsimoka vartotojo įkeltų duomenų dėka.

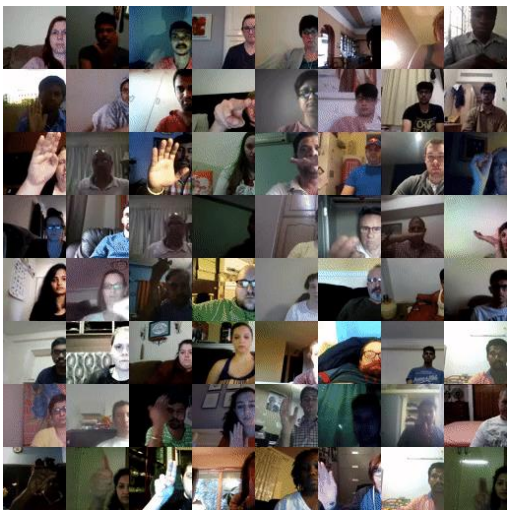


Pav. 2.5 „Imagga“ vartotojo sąsaja

2.4.4. Gestų atpažinimo sistema, naudojant gilų mokymąsi

Šis darbas buvo pristatytas "PyData" konferencijoje Varšuvoje 2017 m. [9]. Autorė pristatė "Python" pagrįstą giliojo mokymosi gestų atpažinimo modelį. Ji teigė, kad modelis yra įdiegtas įterptinėje sistemoje, veikia realiuoju laiku ir gali atpažinti 25 skirtingus rankos gestus iš paprasto kameros srauto. Priešingai nei įprastiniai vizualiniai žaidimų valdikliai, tokie kaip "Microsoft Kinect" [10], jų sistemoje nereikia giluminės informacijos. Tokios architektūros kūrimas yra sudėtingas procesas, kurį reikia atidžiai apsvarstyti kiekviename žingsnyje. Jie atliko tokius procesus: surinko daugiau kaip 150 000 trumpų žmogaus gestus atvaizdavusių vaizdo klipų, pasirinko kokį giliojo mokymosi karkasą naudoti, sukūrė tinklo architektūrą, leidžiančią klasifikuoti vaizdo įrašus su RGB įėjimo duomenimis realiu laiku įterptinėse sistemose ir galiausiai sukūrė projekto aplikaciją.

Šių autorių būdas šiek tiek skiriasi nuo mano pasirinkto būdo, nes jie naudojo vaizdo įrašų pavyzdžius kaip duomenų rinkinio elementus ir bandė atpažinti judantį gestą (kelis kadrus vienu metu).



Pav. 2.6 Pavyzdinės sistemos duomenų pavyzdžiai

Toliau gilinantis į gestų atpažinimą galima pasidomėti Maryam Asadi-Aghbolaghi ir Albert Clapes atliktu giliojo mokymosi metodų tyrimu [11]. Autoriai šiame darbe pateikia dabartinių giliojo mokymosi metodologijų, skirtų veiksmams ir gestams atpažinti vaizdų sekose, tyrimą. Jų darbas supažindina su taksonomija, kurioje apibendrinami svarbūs giliojo mokymosi aspektai, siekiant abiejų užduočių. Taip pat peržiūri siūlomas architektūras, sintezės strategijas, pagrindinius duomenų rinkinių detales. Apibendrina ir aptaria pagrindinius iki šiol pasiūlytus darbus, kaip jie apdoroja turimus duomenis, aptaria jų pagrindines savybes ir nustato būsimų mokslinių tyrimų galimybes bei iššūkius.

2.5. Situacijos Lietuvoje įvertinimas

Šiuo metu Lietuvoje pavyko rasti keletą įmonių ir organizacijų atliekančią tyrimus dirbtinio intelekto, kompiuterinės regos ir mobilių autonominių robotų srityse, bei kuriančią tam tikrus 3D kompiuterinės regos sprendimus:

- Neurotechnology - siūlo plataus masto multi-biometrinį AFIS SDK, kompiuterizuotą, įterptinį sumaniosios kortelės pirštų atspaudą, veido, akių diafragmos, balso ir delnų atpažinimo SDK. Taip pat užsiima AI ir robotų technikos moksliniais tyrimais ir plėtra.
- Oxipit.ai - sprendžia medicinines problemas naudodami gilų mokymąsi.
- True Insight – tobulina reklamą, naudodami gilųjį mokymąsi, numatydami, kur žmonės žiūri į vaizdus ir vaizdo įrašus.
- Pixevia – dirbtinis intelektas sumaniems miestams ir skraidyklėms.
- Gradient Insight – užsiima skaitmeniniu patologijos vaizdų segmentavimu.

3. PROJEKTINĖ DALIS

3.1. Projekto paskirtis

Šio projekto tikslas yra sukurti robotą, kuris gali vizualiai atpažinti siūlomą ranką ir jį priimti. Kai robotas mato žmogų, siūlančią rankos judesį, jis atsako ištiesdamas ranką. Tai yra vizuali ir interaktyvi demonstracija, kurios dėka studentai labiau suinteresuotos mašininio mokymosi ir robotų.

Šiam tikslui pasiekti panaudotas mažas humanoidinis robotas su sumontuota paprasta kamera ant jo galvos ir giliuosius konvoliucinius neuroninius tinklus vaizdų atpažinimo sistemai. Atpažinimas kaip ir apmokymas buvo atliekami kompiuteriu, o už komandą pakelti ranką buvo atsakingas robote integruotas kompiuteris.

Taigi, pagrindinė sistemos problema yra teisingas vaizdo atpažinimas, kuri susideda iš:

- Objekto aptikimo
- Objekto atpažinimo
- Identifikacijos
- Neuroninio tinklo apmokymo

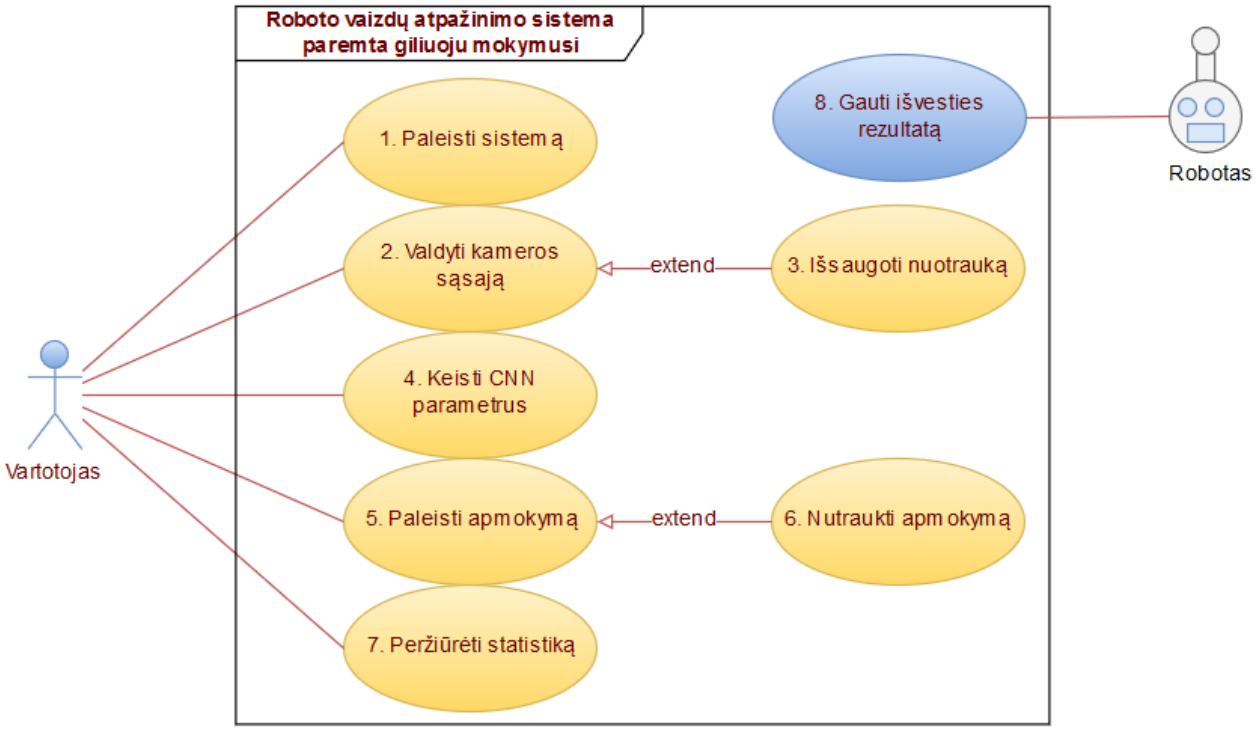
Išsprendus šią problemą gausime sistemą, kuri sugeba atpažinti vaizdus ir siųsti atitinkamą komandą robotui.

3.2. Sistemos paskirtis ir panaudos atvejai

Kuriama sistema veiks Windows operacinėje sistemoje. Jos paskirtis realaus laiko vaizdų atpažinimo demonstracija.

Sistema gebės:

- Būti paleidžiama
- Išsaugoti nuotrauką valdant kameros sąsają
- Keisti CNN parametrus
- Paleisti / nutraukti apmokymą
- Atvaizduoti apmokymų statistiką
- Perduoti išvesties rezultatą robotui



Pav. 3.1 Sistemos panaudos atvejai

3.3. Bendrieji apribojimai

Bendrieji reikalavimai, arba kitaip, nefunkciniai reikalavimai pagrinde nurodo ir apibrėžia programinės įrangos veikimą ir išvaizdą, dėl šios priežasties buvo apibrėžti 13 svarbiausių ir didžiausią naudą turinčių reikalavimų. Visi šie reikalavimai yra išvardinti žemiau esančiose lentelėse.

3.3.1. Reikalavimai sistemos išvaizdai

Lentelė 3.1 Nefunkcinis reikalavimas išvaizdai

Reikalavimas #: 1	**Reikalavimo tipas: 10**	**Įvykis/panaudojimo atvejis#: 1, 2, 3, 4, 5, 6, 7, 8**
Aprašymas:	Sistemos vartotojo sąsaja turi būti minimalistinė, su esminėmis funkcijomis.	
Pagrindimas:	Vengti nereikalingų stiliaus komponentų.	
Šaltinis:	Mantas Lukoševičius – užsakovas	
Tikimo kriterijus:	Langas su veikiančia vartotojo sąsaja	
Užsakovo tenkinimas: 4		**Užsakovo netenkinimas: 5**
Priklausomybės:	nėra	**Konfliktai: nėra**
Papildoma medžiaga	-	
Istorija:	Užregistruotas 2016 lapkričio 3 d.	

Lentelė 3.2 Nefunkcinis reikalavimas išvaizdai

Reikalavimas #: 2	**Reikalavimo tipas: 10**	**Įvykis/panaudojimo atvejis#: 1, 2, 3, 4, 5, 6, 7, 8**
Aprašymas:	Mygtukai išdėstyti lango apačioje.	
Pagrindimas:	Mygtukai atskirti nuo pradinio lango	

Šaltinis:	Mantas Lukoševičius – užsakovas	
Tikimo kriterijus:	Langas su veikiančiais mygtukais	
Užsakovo tenkinimas: 3		**Užsakovo netenkinimas: 5**
Priklausomybės:	nėra	**Konfliktai: nėra**
Papildoma medžiaga	-	
Istorija:	Užregistruotas 2016 lapkričio 3 d.	

3.3.2. Reikalavimai panaudojamumui

Lentelė 3.3 Nefunkcinis reikalavimas panaudojamumui

Reikalavimas #: 4	**Reikalavimo tipas: 10**	**Įvykis/panaudojimo atvejis#: 1, 2, 3, 4, 5, 6, 7, 8**
Aprašymas:	Sistema turėtų padėti išvengti klaidų vartotojams.	
Pagrindimas:	Vartotojui atlikus veiksmą ir gavus klaidą, sistema turi pasiūlyti paaiškinimą ir problemos sprendimą	
Šaltinis:	Mantas Lukoševičius – užsakovas	
Tikimo kriterijus:	Pagalbiniai pranešimai kiekvienam veiklos scenarijui	
Užsakovo tenkinimas: 3		**Užsakovo netenkinimas: 3**
Priklausomybės:	nėra	**Konfliktai: nėra**
Papildoma medžiaga	-	
Istorija:	Užregistruotas 2016 lapkričio 4 d.	

Lentelė 3.4 Nefunkcinis reikalavimas panaudojamumui

Reikalavimas #: 5	**Reikalavimo tipas: 10**	**Įvykis/panaudojimo atvejis#: 1, 2, 3, 4, 5, 6, 7, 8**
Aprašymas:	Sistemos vartotojo sąsaja turėtų būti parašyta taisyklinga lietuvių kalba.	
Pagrindimas:	Lietuvis vartotojas turi suprasti sistemoje naudojamą lietuvių kalbą.	
Šaltinis:	Mantas Lukoševičius – užsakovas	
Tikimo kriterijus:	Neturi būti nei gramatinių, nei sintaksės klaidų.	
Užsakovo tenkinimas: 2		**Užsakovo netenkinimas: 3**
Priklausomybės:	nėra	**Konfliktai: nėra**
Papildoma medžiaga	-	
Istorija:	Užregistruotas 2016 lapkričio 4 d.	

Lentelė 3.5 Nefunkcinis reikalavimas panaudojamumui

Reikalavimas #: 6	**Reikalavimo tipas: 10**	**Įvykis/panaudojimo atvejis#: 1, 2, 3, 4, 5, 6, 7, 8**
Aprašymas:	Sistema turėtų būti naudojama švietimo tikslais ir kaip IT ir robotikos mokymosi priemonė.	
Pagrindimas:	Sistema turi turėti mokslinę vertę.	
Šaltinis:	Mantas Lukoševičius – užsakovas	
Tikimo kriterijus:	Sistema tinkama mokyti Lietuvos mokyklose.	
Užsakovo tenkinimas: 2		**Užsakovo netenkinimas: 2**
Priklausomybės:	nėra	**Konfliktai: nėra**

Papildoma medžiaga	-
Istorija:	Užregistruotas 2016 lapkričio 5 d.

Lentelė 3.6 Nefunkcinis reikalavimas panaudojamumui

Reikalavimas #: 7	**Reikalavimo tipas: 10**	**Įvykis/panaudojimo atvejis#: 1, 2, 3, 8**
Aprašymas:	Laikas nuo vaizdo gavimo iki apdorojimo ir atvaizdavimo ne didesnis nei 1 sekundė.	
Pagrindimas:	Pagrindinė sistemos funkcija turi veikti greičiau nei per 1 sekundę.	
Šaltinis:	Mantas Lukoševičius – užsakovas	
Tikimo kriterijus:	Sistema apdoroja, atpažįsta ir atvaizduoja greičiau nei per 1 sekundę.	
Užsakovo tenkinimas: 5		**Užsakovo netenkinimas: 4**
Priklausomybės:	nėra	**Konfliktai: nėra**
Papildoma medžiaga	-	
Istorija:	Užregistruotas 2016 lapkričio 5 d.	

Lentelė 3.7 Nefunkcinis reikalavimas panaudojamumui

Reikalavimas #: 8	**Reikalavimo tipas: 10**	**Įvykis/panaudojimo atvejis#: 1, 4, 5, 6, 7**
Aprašymas:	Rankos atpažinimo tikslumas ne mažesnis kaip 90%.	
Pagrindimas:	Gauti nemažesnį vaizdo atpažinimo tikslumą.	
Šaltinis:	Mantas Lukoševičius – užsakovas	
Tikimo kriterijus:	Tikslumas 90%.	
Užsakovo tenkinimas: 5		**Užsakovo netenkinimas: 5**
Priklausomybės:	nėra	**Konfliktai: nėra**
Papildoma medžiaga	-	
Istorija:	Užregistruotas 2016 lapkričio 5 d.	

Lentelė 3.8 Nefunkcinis reikalavimas panaudojamumui

Reikalavimas #: 9	**Reikalavimo tipas: 10**	**Įvykis/panaudojimo atvejis#: 1, 2, 3**
Aprašymas:	Sistema neturėtų fiksuoti rankų esančių toliau kaip per 1 metrą.	
Pagrindimas:	Toliau esančios rankos nėra aktualios sistemai, nes pasisveikinimas iš taip toli nevyksta.	
Šaltinis:	Mantas Lukoševičius – užsakovas	
Tikimo kriterijus:	Nedidesnis kaip 1 metro atstumas.	
Užsakovo tenkinimas: 3		**Užsakovo netenkinimas: 4**
Priklausomybės:	nėra	**Konfliktai: nėra**
Papildoma medžiaga	-	
Istorija:	Užregistruotas 2016 lapkričio 11 d.	

Lentelė 3.9 Nefunkcinis reikalavimas panaudojamumui

Reikalavimas #: 10	**Reikalavimo tipas: 10**	**Įvykis/panaudojimo atvejis#: 1, 2, 3, 8**
Aprašymas:	Gali veikti tiek lauke, tiek uždaroje patalpoje.	
Pagrindimas:	Sistema turi veikti tiek uždarose patalpose, tiek lauke.	
Šaltinis:	Mantas Lukoševičius – užsakovas	
Tikimo kriterijus:	Įvairi aplinka	
Užsakovo tenkinimas: 3		**Užsakovo netenkinimas: 2**
Priklausomybės:	nėra	**Konfliktai: nėra**
Papildoma medžiaga	-	
Istorija:	Užregistruotas 2016 lapkričio 11 d.	

Lentelė 3.10 Nefunkcinis reikalavimas panaudojamumui

Reikalavimas #: 11	**Reikalavimo tipas: 10**	**Įvykis/panaudojimo atvejis#: 1, 2, 3**
Aprašymas:	Privalomas bent vidutinis apšvietimo lygis.	
Pagrindimas:	Nuotraukoje turi skirtis aplinkos spalva nuo rankos spalvos	
Šaltinis:	Mantas Lukoševičius – užsakovas	
Tikimo kriterijus:	Saulėta, arba apsiniaukusi diena, dirbtinis apšvietimas	
Užsakovo tenkinimas: 2		**Užsakovo netenkinimas: 3**
Priklausomybės:	nėra	**Konfliktai: nėra**
Papildoma medžiaga	-	
Istorija:	Užregistruotas 2016 lapkričio 11 d.	

Lentelė 3.11 Nefunkcinis reikalavimas panaudojamumui

Reikalavimas #: 12	**Reikalavimo tipas: 10**	**Įvykis/panaudojimo atvejis#: 1, 2, 3, 4, 5, 6, 7, 8**
Aprašymas:	Sistema turi veikti bent vienoje iš platformų: PC su (Linux OS, macOS, arba Windows), Raspberry Pi, Intel Edison.	
Pagrindimas:	Prioritetas Raspberry pi.	
Šaltinis:	Mantas Lukoševičius – užsakovas	
Tikimo kriterijus:	Veikimas bent vienoje platformoje.	
Užsakovo tenkinimas: 4		**Užsakovo netenkinimas: 5**
Priklausomybės:	nėra	**Konfliktai: nėra**
Papildoma medžiaga	-	
Istorija:	Užregistruotas 2016 spalio 2 d.	

Lentelė 3.12 Nefunkcinis reikalavimas panaudojamumui

Reikalavimas #: 13	**Reikalavimo tipas: 10**	**Įvykis/panaudojimo atvejis#: 1, 8**
Aprašymas:	Sistema neturėtų būti priešiška jokioms religinėms ir etninėms grupėms.	
Pagrindimas:	Nekurstyti religinės ir etninės nesantaikos.	
Šaltinis:	Mantas Lukoševičius – užsakovas	
Tikimo kriterijus:	Jokių įžeidžiančių ar provokuojančių užuominų sistemoje.	

Užsakovo tenkinimas: 1		**Užsakovo netenkinimas: 1**
Priklausomybės:	nėra	**Konfliktai: nėra**
Papildoma medžiaga	-	
Istorija:	Užregistruotas 2016 lapkričio 20 d.	

3.4. Reikalavimai sistemai

Reikalavimai sistemai, arba kitaip, funkciniai reikalavimai išvardinti žemiau esančiose lentelėse.

Lentelė 3.13 Funkcinis reikalavimas

Reikalavimas #: 14	**Reikalavimo tipas: 9**	**Įvykis/panaudojimo atvejis#: 2, 3**
Aprašymas:	Kamera registruoja vaizdą	
Pagrindimas:	Reikalinga siekiant nesukelti didelio sistemos vėlinimo	
Šaltinis:	Mantas Lukoševičius – užsakovas	
Tikimo kriterijus:	Registruojamas vaizdas kas 1 sekundę	
Užsakovo tenkinimas: 3		**Užsakovo netenkinimas: 5**
Priklausomybės:	CNN apmokymo	**Konfliktai: nėra**
Papildoma medžiaga	Veiklos konteksto diagrama (2 pav.), terminų žodynas	
Istorija:	Užregistruotas 2016 rugsėjo 27 d.	

Lentelė 3.14 Funkcinis reikalavimas

Reikalavimas #: 15	**Reikalavimo tipas: 9**	**Įvykis/panaudojimo atvejis#: 4, 5, 6**
Aprašymas:	Sistema apmoko tinklą pagal įkeltus duomenis	
Pagrindimas:	Reikalinga, norint kad sistema veiktų gebėtų atpažinti objektą	
Šaltinis:	Mantas Lukoševičius – užsakovas	
Tikimo kriterijus:	Apmokyti neuroninį tinklą trunka ne ilgiau nei 24h, neuroninis tinklas apmokytas ir geba atpažinti objektus	
Užsakovo tenkinimas: 5		**Užsakovo netenkinimas: 5**
Priklausomybės:	Visi su duomenimis ir apmokymu susiję reikalavimai	**Konfliktai: 5**
Papildoma medžiaga	-	
Istorija:	Užregistruotas 2016 spalio 4 d.	

Lentelė 3.15 Funkcinis reikalavimas

Reikalavimas #: 16	**Reikalavimo tipas: 9**	**Įvykis/panaudojimo atvejis#: 1, 2, 3, 4, 5, 6**
Aprašymas:	Sistema turi atpažinti ištiestą ranką	
Pagrindimas:	Pagrindinis sistemos funkcionalumas	
Šaltinis:	Mantas Lukoševičius – užsakovas	
Tikimo kriterijus:	Sistema vaizdus turi atpažinti 90% tikslumu	
Užsakovo tenkinimas: 5		**Užsakovo netenkinimas: 5**

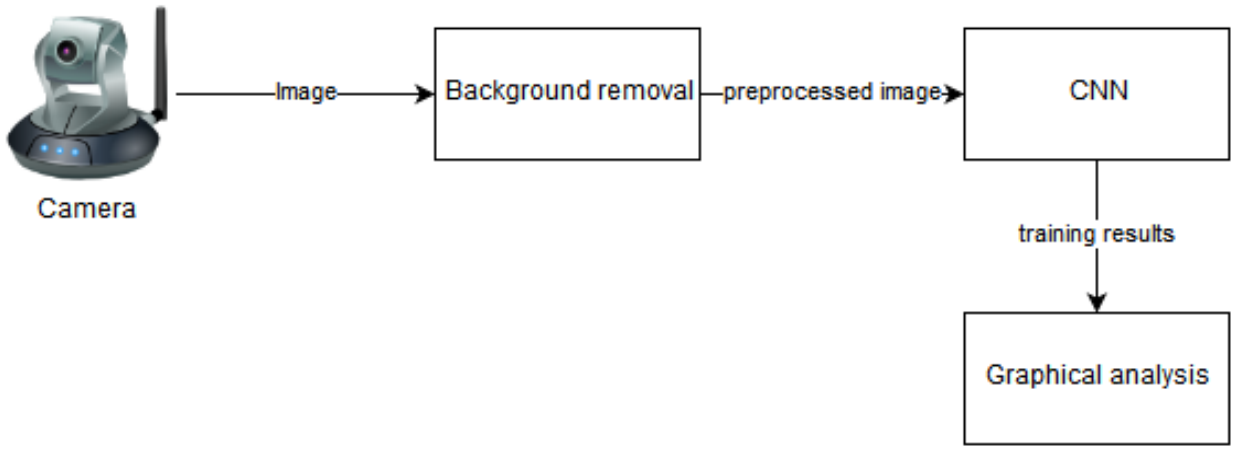
Priklausomybės:	Visi su duomenimis, apmokymu ir atpažinimu susiję reikalavimai	**Konfliktai: 2**
Papildoma medžiaga	-	
Istorija:	Užregistruotas 2016 spalio 4 d.	

Lentelė 3.16 Funkcinis reikalavimas

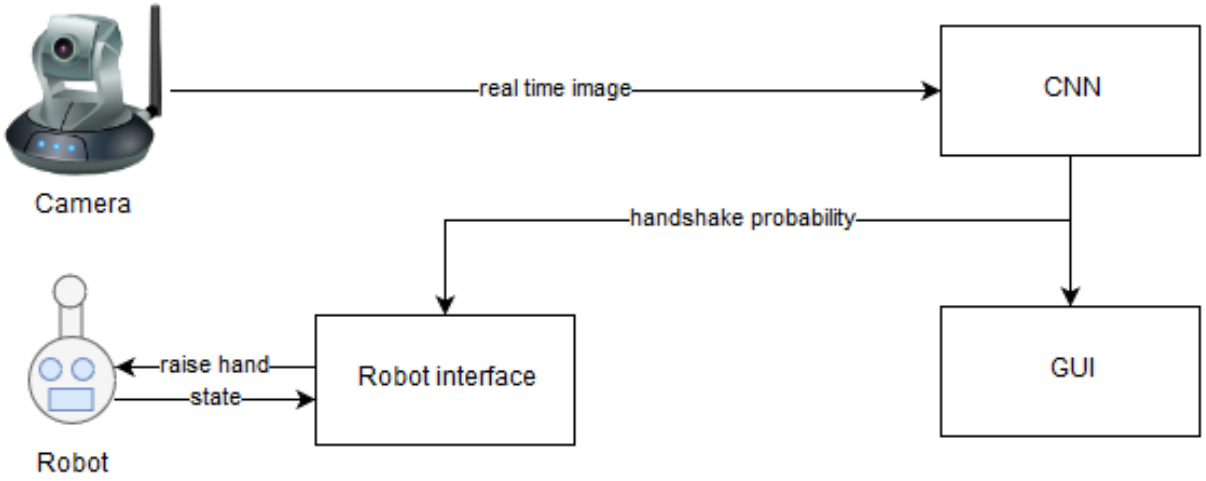
Reikalavimas #: 17	**Reikalavimo tipas: 9**	**Įvykis/panaudojimo atvejis#: 4, 5, 6, 7**
Aprašymas:	Sistema registruoja visus apmokymo metu gaunamus pranešimus registre	
Pagrindimas:	Reikalinga saugoti pranešimus registre, kad įvykus klaidai, būtų žinoma priežastis	
Šaltinis:	Mantas Lukoševičius – užsakovas	
Tikimo kriterijus:	Registre užregistruojamos klaidos, tikslumas po kiekvienos epochos.	
Užsakovo tenkinimas: 3		**Užsakovo netenkinimas: 2**
Priklausomybės:	Nėra	**Konfliktai: nėra**
Papildoma medžiaga	-	
Istorija:	Užregistruotas 2016 rugsėjo 27 d.	

3.5. Sistemos architektūra

Šio poskyrio tikslas yra atskleisti pagrindinius sistemos architektūrinius sprendimus. Sistemos modelis susideda iš kelių dalių, parodytų 3.2 ir 3.3 paveikslėliuose, įskaitant kameros, kamerų vaizdų išankstinį apdorojimą, konvoliucinio neuroninio tinklo mokymą naudojant gilų mokymąsi, grafinę vartotojo sąsają, robotų sąsają ir patį robotą.



Pav. 3.2 Sistemos apmokymo modelis

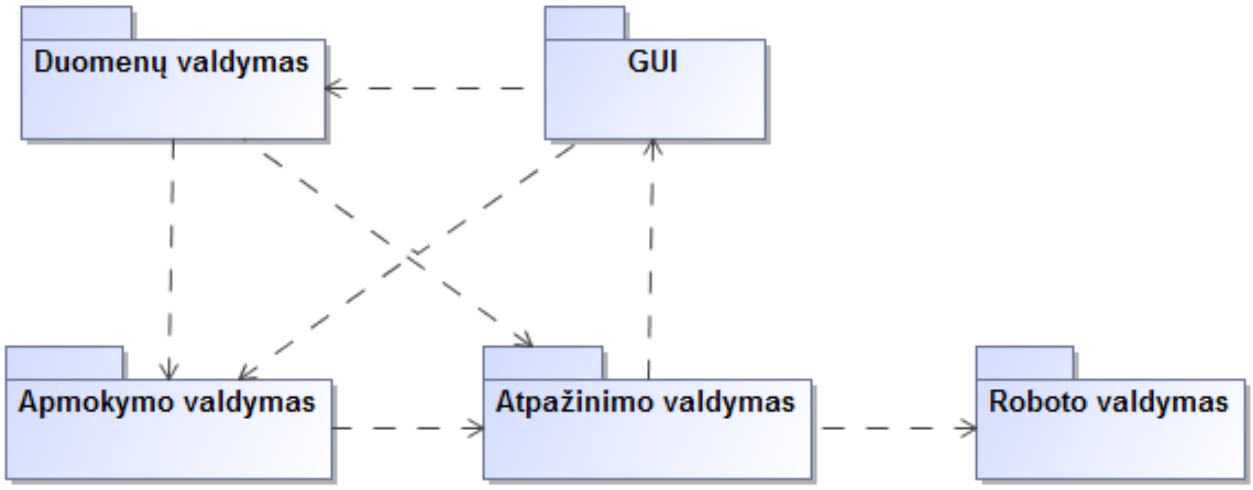


Pav. 3.3 Sistemos paleidimo modelis

Iš pradžių, kamera buvo naudojama, kad surinkti vaizdų rinkinį. Po tolesnių tyrimų, kurie aprašyti 4 skyriuje, duomenų rinkinyje esantys vaizdai turėjo būti iš anksto apdoroti, tam, kad būtų galima tobulinti modelį, kuris yra kita sistemos dalis. Naudojant Keras biblioteką modelis buvo sukurtas, sukompiliuotas ir pagaliau apmokytas. Paskutinė dalis buvo paleisti modelį, skirtą atpažinti naujus gyvus vaizdus. Dėl šios priežasties fotoaparato sąsaja buvo užprogramuota fotografuoti kas 0,5 sek., Modelis nuskaito tuos vaizdus kaip įvestį ir grąžina pasiūlyto pasisveikinimo tikimybę kaip rezultatą. Jei šis rezultatas viršija tam tikrą slenkstį, robotų sąsaja siunčia robotui komandą atlikti atitinkamą užduotį.

3.6. Sistemos statinis vaizdas

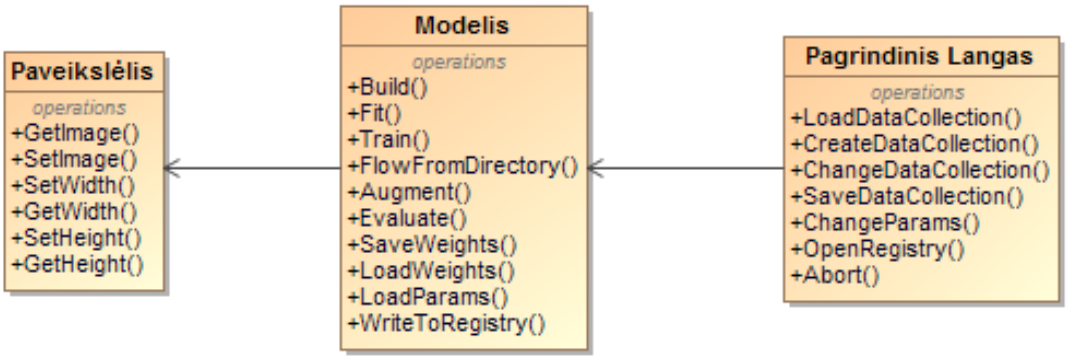
Sistema suskaidyta į penkis paketus, kurie pateikti 3.4 paveikslėlyje:



Pav. 3.4 Abstraktus sistemos skaidymas į paketus

3.6.1. Duomenų valdymo paketas

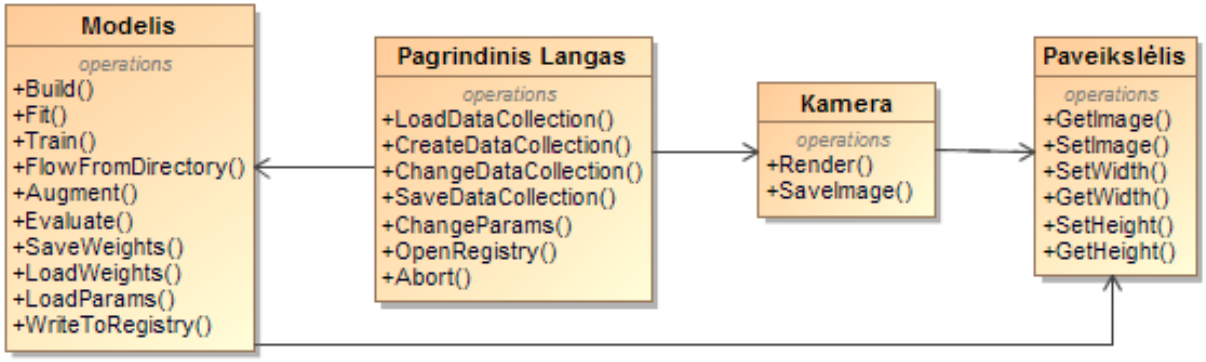
Paketas Duomenų valdymas skirtas paveikslėlių saugojimui, duomenų rinkinio sudarymui, jo redagavimui.



Pav. 3.5 Paketo Duomenų valdymas klasių diagrama

3.6.2. GUI paketas

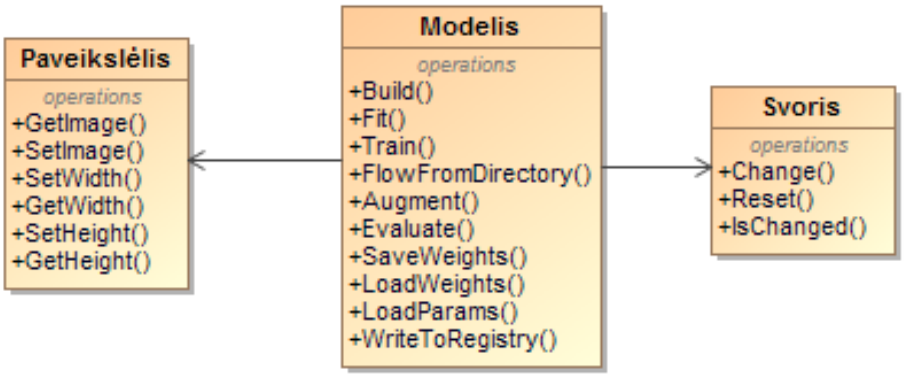
Paketas GUI skirtas sistemos atvaizdavimui ir valdymui (apmokymui, paleidimui, duomenų saugojimui, parametrų keitimui).



Pav. 3.6 Paketo GUI klasių diagrama

3.6.3. Apmokymo valdymo paketas

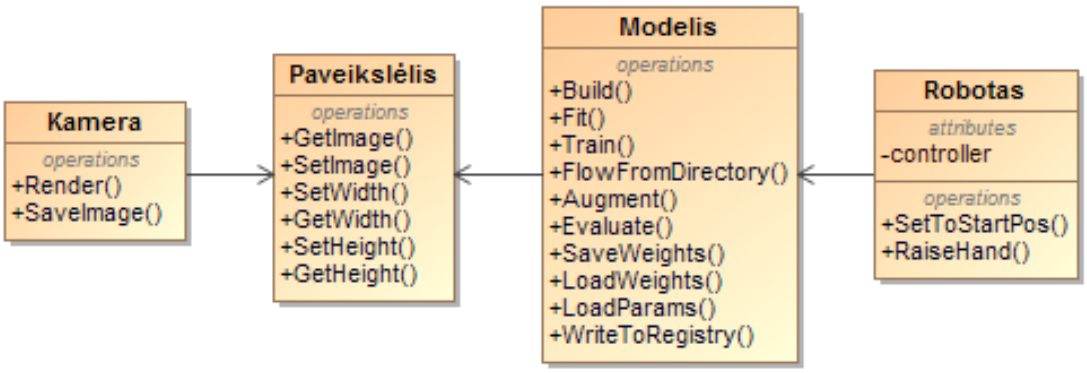
Paketas Apmokymo valdymas skirtas CNN apmokymo paleidimui, svorių saugojimui.



Pav. 3.7 Paketo Apmokymo valdymas klasių diagrama

3.6.4. Atpažinimo valdymo paketas

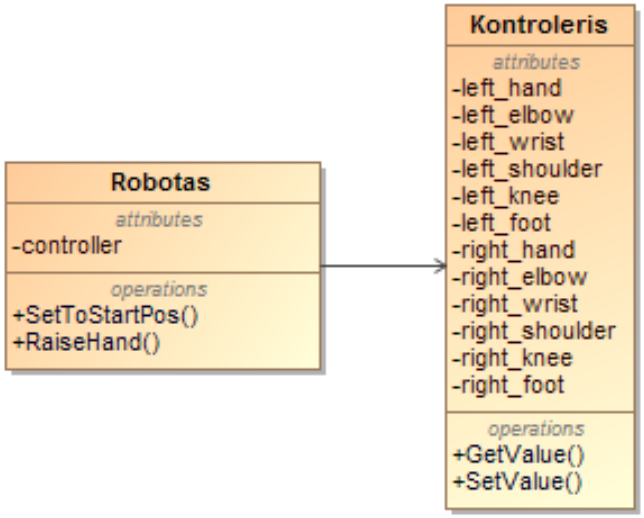
Paketas Atpažinimo valdymas skirtas sistemos paleidimui, kurią sudaro sistema, kamera ir robotas. Kamera fiksuoja vaizdą, sistema jį apdoroja ir siunčią komandą robotui.



Pav. 3.8 Paketo Atpažinimo valdymas klasių diagrama

3.6.5. Roboto valdymo paketas

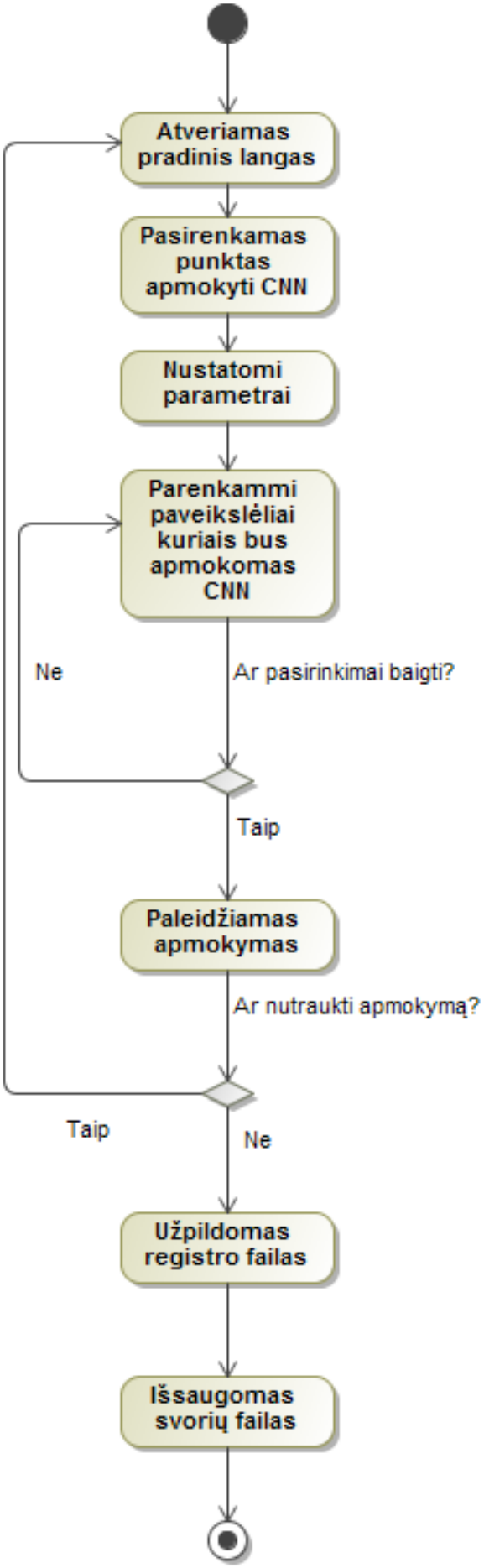
Paketas Roboto valdymas skirtas roboto kontrolerių valdymui gavus komandą sveikintis ir nesisveikinti.



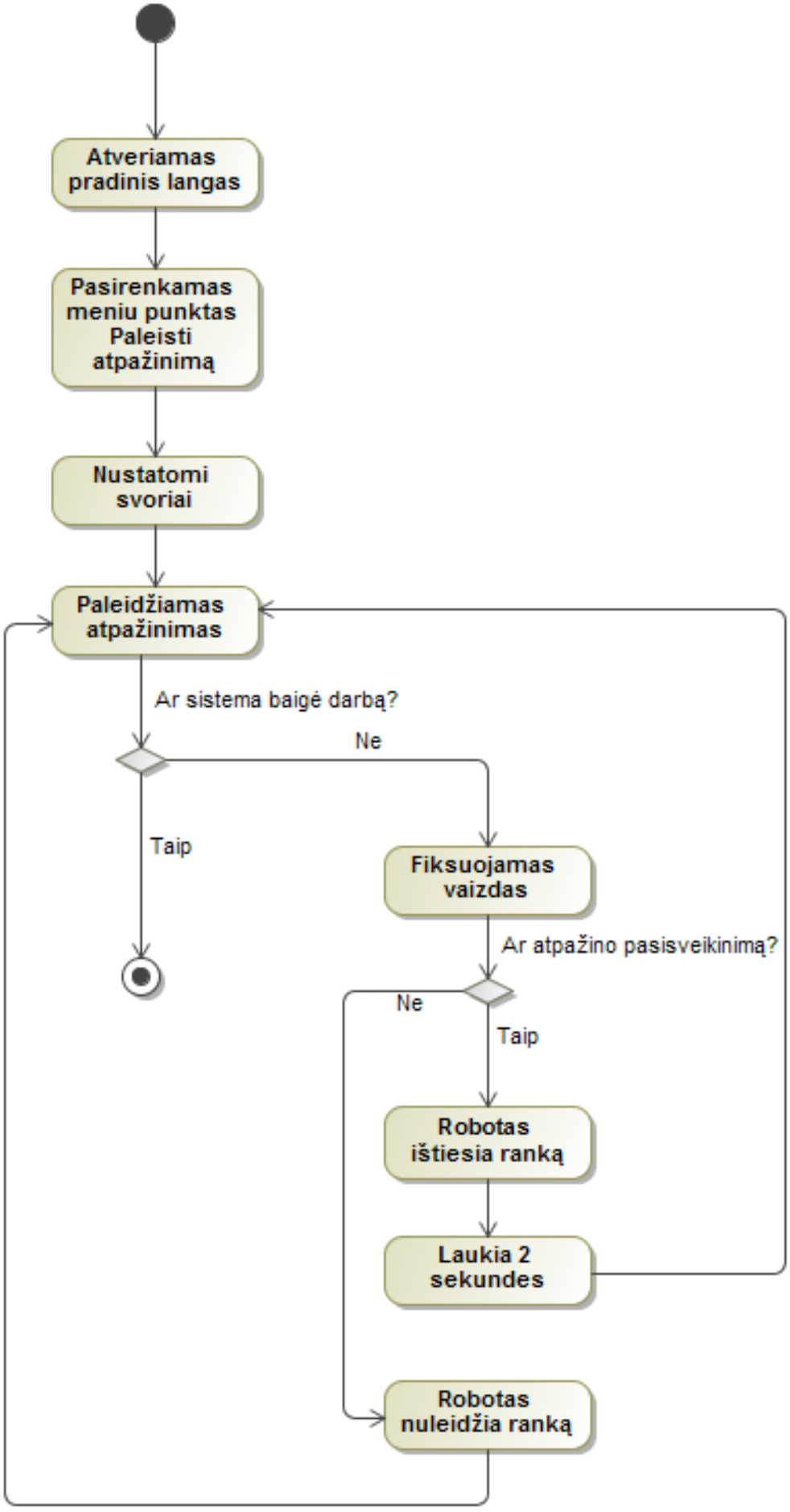
Pav. 3.9 Paketo Roboto valdymas klasių diagrama

3.7. Veiklos diagrama

Žemiau pateikiamos duomenų valdymo, apmokymo ir atpažinimo veiklos diagramos (žr. Pav. 3.10-3.11):

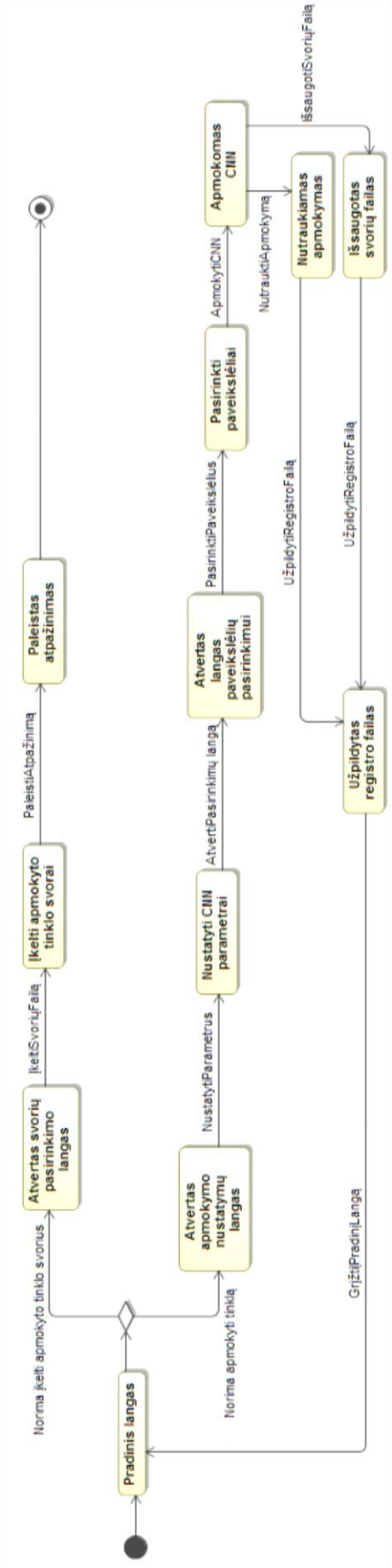


Pav. 3.10 Apmokymo veiklos diagrama

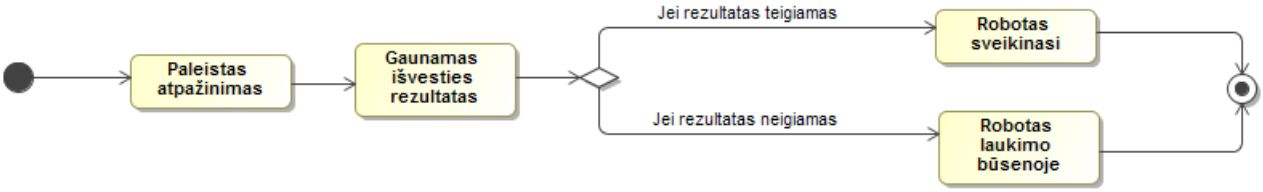


Pav. 3.11 Atpažinimo veiklos diagrama

3.8. Būsenų diagramos

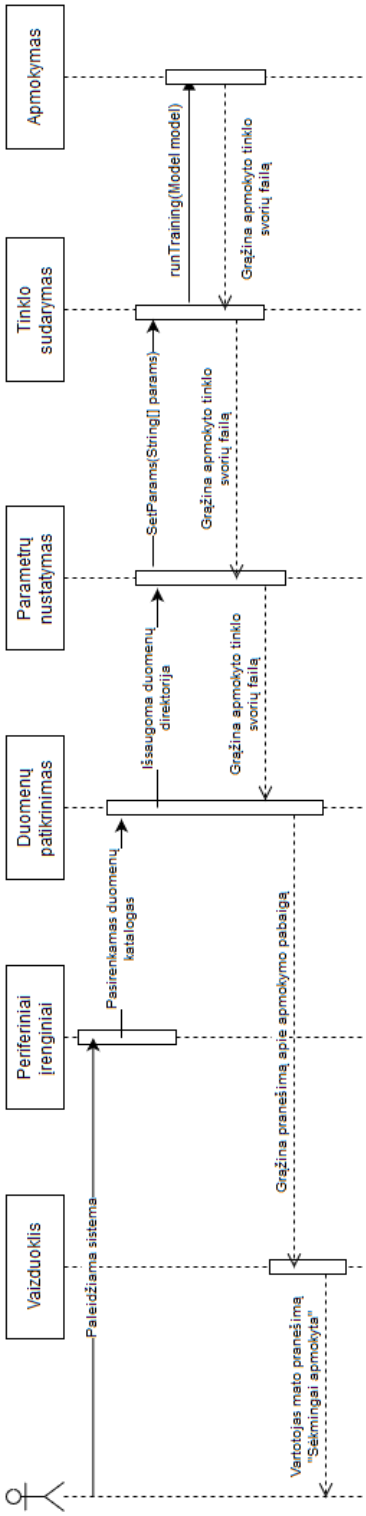


Pav. 3.12 Apmokymo ir paleidimo būsenų diagrama

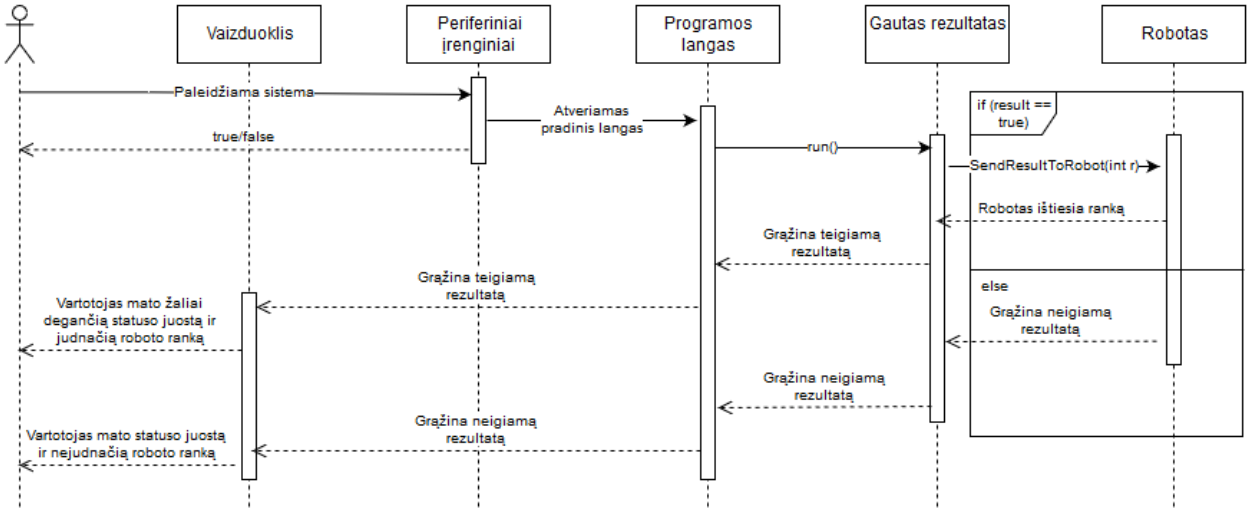


Pav. 3.13 Roboto valdymo būsenų diagrama

3.9. Sekų diagramos



Pav. 3.14 Apmokymo sekų diagrama



Pav. 3.15 Sistemos paleidimo sekų diagrama

4. TYRIMO IR EKSPERIMENTINĖ DALYS

4.1. Esamas funkcionalumas

Šio projekto metu sukurta programinė įranga, skirta robotų regos naudojant gilųjį mokymąsi demonstracijai pavaizduoti. Sistemos kūrimą galima suskirstyti į kelis pagrindinius etapus:

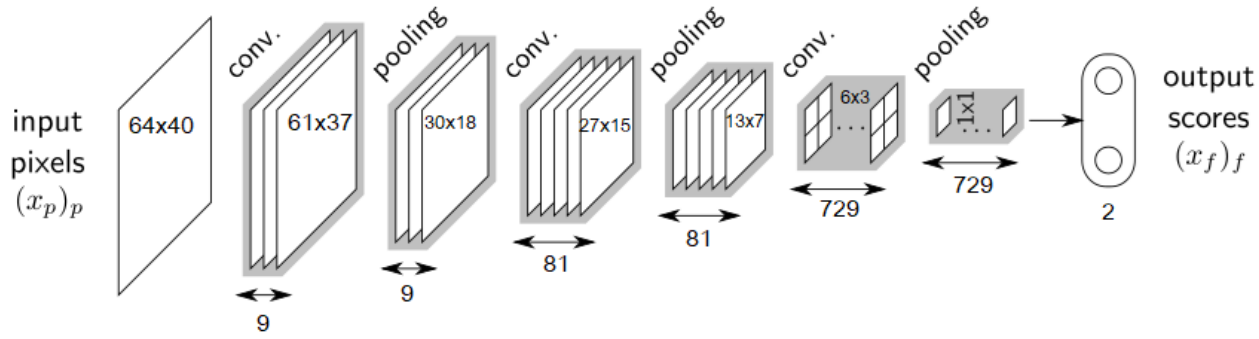
1. Iš pradžių bandyta perprasti egzistuojančius atvirojo kodo sprendimus ir tiesiog juos išsibandyti.
2. Tuomet buvo pradėtas rinkti duomenų rinkinys. Šiam tikslui buvo sukurta programinė įranga. Iš viso buvo surinkta daugiau nei 4000 skirtingų paveikslėlių, kuriais buvo apmokomas neuroninis tinklas.
3. Tuomet pasirinkus įrankius ir technologijas kuriamai sistemai buvo parašytas pirmas programinis kodas, kurio paskirtis sumodeliuoti neuroninio tinklo architektūrą ir apmokyti neuroninį tinklą su surinktais duomenimis.
4. Sekantis etapas buvo sukurti programinę įrangą kuri realiu laiku gebėtų fiksuoti vaizdą ir grąžinti atpažinimo rezultatą.
5. Taip pat reikėjo integruoti egzistuojantį kameros sąsają realaus laiko atpažinimo programinėje įrangoje.
6. Galiausiai paskutinis etapas buvo parašyti programinį kodą robotui, kad šis ištiestų ranką.

4.2. Neuroninio tinklo sukūrimas

Kuriant CNN [12] susiduriama su problema, kaip pasiekti optimalų tinklo dydį ir gauti kuo tikslesnį atpažinimo algoritmą. Neuroninio tinklo dydis yra aktualus apmokymo metu, kuo didesnis tinklas – tuo ilgiau trunka apmokymas, tačiau galimai pasiekiamas aukštesnis tikslumas. Ir atvirkščiai, kuo mažesnis tinklas, tuo jis greičiau apmokomas, tačiau galimai tikslumas bus mažesnis. Taigi, problema rasti optimalų tinklo dydį, kuris vykdytų apmokymą neilgiau kaip 24h ir pasiektų bent 90% tikslumą.

Žemiau pavaizduotas asmeniškai mūsų sukurto konvoliucinio neuroninio tinklo schema (Pav. 4.1), kuriame galima matyti kaip jis veikia. Pradžioje, kaip įėjimą turime 64x40 pikselių dydžio paveikslėlį, jį perfiltruojame su pasirinktais filtrais (šiuo atveju 3x3 dydžio paveikslėliais), tuomet gauname tiek svorių žemėlapių kiek turėjome filtrų. Šiuo atveju gauname, kad konvoliucinis sluoksnis turi devynis naujus filtruotus 61x37 dydžio paveikslėlius, tuomet vykdome sujungimą (ang. pooling). Sujungimo metu įprastai imami 2x2 dydžio paveikslėlių fragmentai ir juose paimama didžiausia svorio reikšmė. Taigi po šio etapo gauname devynis 30x18 paveikslėlius, tuomet vėl vyksta paveikslėlių filtravimas, po jo gauname jau aštuoniasdešimt vieną 27x15 paveikslėlius. Sujungimo metu gaunami aštuoniasdešimt vienas 5x5 paveikslėlis. Paskutinį kartą

vykdant filtravimą gauname septynis šimtus dvidešimt devynis 6x3 paveikslėlius su svoriais ir įvykdę galutinį sujungimą – lieka septyni šimtai dvidešimt devyni 1x1 (paskutinio sujungimo metu paimti 4x3 dydžio paveikslėlių fragmentai) pikselių rinkiniai su svoriais, tuomet paduoda vieną iš galimų variantų, CNN mums grąžina tikimybę, kiek ši reikšmė panaši į atsakymą.



Pav. 4.1 Konvoliucinio neuroninio tinklo architektūra

Smulkiai apie klasifikacijos būdus aprašo A. Krizhevsky, I. Sutskever ir G. E. Hinton straipsnyje ImageNet Classification with Deep Convolutional Neural Networks [13].

4.3. Neuroninio tinklo apmokymo procesas

Prieš pradedant apmokyti modelį, yra keletas parametrų, apibūdinančių treniruočių detales. Pirmasis parametras yra epochų skaičius. Pati epocha yra sutartinis etapas, paprastai apibrėžiamas kaip "vienas paleidimas per visą duomenų rinkinį", naudojamas atskirti mokymą skirtingais etapais, kuris yra naudingas renkant informaciją ir periodiškai vertinant. Apskritai tai reiškia, kiek kartų procesas bus vykdomas su apmokymo duomenų rinkiniu. Antrasis parametras yra paketo dydis. Paketo dydis apibrėžia mėginių, kurie bus platinami per tinklą, skaičių. Pavyzdžiui, yra 200 mokymų pavyzdžių ir mes norime nustatyti paketo dydį, lygų 30. Algoritmas paima pirmuosius 30 apmokymo duomenų rinkinio pavyzdžių ir apmoko tinklą. Tada dar kartą paimamas antrasis 30 pavyzdžių paketas ir tinklas apmokomas vėl. Šią procedūrą galima atlikti tol, kol tinklą apmokysime su visais pavyzdžiais. Tačiau problema paprastai būna su paskutiniu pavyzdžių rinkiniu. Šiame pavyzdyje lieka paskutiniai 20 mėginių, kaip dalybos iš 30 liekana. Paprasčiausias sprendimas yra tiesiog apmokyti tinklą iš naujo su šiais likusiais 20 pavyzdžių. Paskutiniai du parametrai yra tikslinis dydis ir klasės būsena. Tikslinis dydis priima vaizdų aukščio ir pločio reikšmes iš duomenų rinkinio, o klasių būsenai galima priskirti vieną iš šių verčių: ["kategoriškas", "dvejetainis", "retas", "nėra"] (ang. Categorical, binary, sparse, none). Šiuo atveju šio argumento vertė yra "dvejetainis", todėl, kad mes norime pripažinti tik teisingą ar klaidingą rezultatą. Tik nustačius visus šiuos parametrus apmokymas gali būti pradėtas. Pradėję apmokymą konsolėje galima pamatyti šiek tiek informacijos apie mokymo eigą. Yra dvi pagrindinės reikšmės, kurios keičiasi kiekvieną epochą, tai tikslumas ir praradimas. Tikslumo reikšmė matuojama procentais, o praradimas yra konstanta.

Apmokymo tikslumas ir praradimas apskaičiuojami apmokymo metu. Šie skaičiai rodo, kaip gerai tinklas apsimoko duomenis, su parinktais duomenimis. Mokymo tikslumas įprastai nuolatos didėja.

4.4. Validacijos procesas

Norėdami validuoti modelį, turime turėti naują duomenų rinkinį, kad nauji vaizdai nebūtų buvę naudojami mokymo procese. Validacija paprastai atliekama kartu su mokymu. Po kiekvienos epochos modelis yra testuojamas validacijos rinkinio duomenimis, apskaičiuojamas validacijos praradimas ir tikslumas. Šie skaičiai tiksliausiai parodo, kaip gerai modelis gebės pažinti naujus vaizdus, kurių dar niekada anksčiau nebuvo matęs. Validacijos tikslumas iš pradžių didėja ir galiausiai pradeda mažėti, tuomet įvyksta persimokymas. Persimokymas įvyksta tada, kuomet modelis pernelyg gerai išmoksta apmokymo rinkinio pavyzdžius. Tada modeliui tampa sunku apibendrinti naujus pavyzdžius, kurie nebuvo parengti. Pavyzdžiui, mano modelis atpažįsta konkrečius mokymų rinkinio vaizdus, o ne bendrus šablonus. Mokymo tikslumas įprastai būna didesnis už validacijos tikslumą.

4.5. Testavimo procesas

Norint išbandyti modelį, mums reikia kito naujo duomenų rinkinio. Paprastai bandymas atliekamas rankiniu būdu, perduodant vaizdą iš duomenų rinkinio apmokytam modeliui, kad būtų gautas rezultatas. Gautas rezultatas yra procentinė reikšmė, parodanti kokiu tikslumu spėjama kad rezultatas yra viena ar kita reikšmė.

4.6. Duomenų rinkinio paruošimas

Kaip buvo paminėta ankstesniame skyriuje, šio projekto įgyvendinimui reikėjo rinkti vaizdinius duomenis. Kadangi sistema atpažįsta tik pasisveikinimus, galimi tik du rezultatai: sveikinimas yra atpažintas arba ne. Projekto metu buvo surinkta daugiau nei 4 000 skirtingų vaizdų, skirtų neuroninio tinklo mokymui. Maždaug 2000 kiekvienos kategorijos. Vienos nuotraukos raiška siekia 318x198. Žemiau pavaizduotame paveikslėlyje (Pav. 4.2) galima matyti, kad kadre įprastai yra vienas žmogus ištiesęs ranką arbe ne. Taip pat buvo bandoma užfiksuoti vaizdus su kiek įmanoma daugiau skirtingų aplinkų. Žmonių drabužiai taip pat buvo įvairūs, norėdami užfiksuoti kuo įvairesnes spalvas. Tai svarbu siekiant užtikrinti, kad atpažinimas nebūtų ribojamas tam tikra konkrečia situacija.

Pav. 4.2 Nuotraukų pavyzdžiai: teigiami viršuje, neigiami apačioje

Nuotraukos buvo suskirstytos į tris rinkinius: mokymo, validacijos ir testavimo. Neuroninis tinklas mokomas su mokymo duomenimis. Tada jis validuojamas su validacijos duomenimis, siekiant patikrinti, ar gerai apmokytas neuroninis tinklas atlieka atpažinimą naujais pavyzdžiais. Testavimo duomenys skirti validuoti galutinį neuroninio tinklo gebėjimą išgauti galutinį tikrąjį atpažinimą. Be to, apmokymo metu buvo naudojamas duomenų padidinimas [14], kuriame prieš įvairaus pobūdžio vaizdų apmokymą atliekamos tam tikros nuotraukos transformacijos (rotacija, vertimas, mastelio didinimas/mažinimas).

4.7. Fono šalinimas

Iš pradžių bandyta apmokyti neuroninį tinklą su duomenimis, gaunamais tiesiai iš kameros, be jų išankstinio apdorojimo. Tačiau buvo pastebėta, kad modelis su geriausiu bandymu pasiekė 78 procentų apmokymo tikslumą ir apie 64 procentus validacijos tikslumą ir tuomet sekė persimokymas, kurio metu klaidos lygis labai padidėjo. Dėl šios priežasties reikėjo ieškoti sprendimų, kaip išvengti persimokymo ir kaip padidinti modelio validacijos tikslumą. Siekiant šio

tikslo buvo bandoma keisti modelio parametrus, tačiau tai nepagerino rezultatų tiek, kiek tikėtasi. Tada buvo nuspręsta apdoroti pačius duomenis. Iš ankstesnių eksperimentų pavyko susidaryti įspūdį, kad persimokymas įvyksta dėl pernelyg didelės spalvų gamos ir pačios vaizdų spalvos. Dėl šios priežasties nusprendėme pašalinti nuotraukos foną ir mokyti neuroninį tinklą su nuotraukomis be fono. Tačiau tai sukėlė naują problemą. Kaip nustatyti, kur yra fonas ir kur yra objektas (šiuo atveju žmogus)? Dėl šios problemos, buvo nuspręsta pirmąją nuotrauką padaryti be žmogaus ir laikyti, kad tai yra fonas, o visi kiti vaizdai yra objektai su fonais. Tai šiuo atveju šiek tiek apribojo sistemą, nes kamera turėjo būti tik fiksuotoje padėtyje ir negalėjo būti judinama duomenų rinkimo metu. Tada mes galėjome atimti du vaizdus ir gauti vaizdą be fono. Įprastai po atimties kai kurie triukšmai visada likdavo nuotraukose. Kad sumažintume tai, mes nustatėme leistiną pikselių RGB verčių klaidą. Gautos nuotraukos po fono nuėmimo pavaizduotos žemiau (Pav. 4.3).

Pav. 4.3 Tos pačios nuotraukos prieš ir po fono nuėmimo

Rezultatas buvo akivaizdus. Galima pasiekti 92 proc. apmokymo tikslumą naudojant vienspalvį natūralų foną. Tai reiškia, kad modelis yra pakankamai tikslus, kad atpažintų ištiestą ranką, kai žmogaus fonas yra lygus ir jo nereikia pašalinti.

Norint gauti šį rezultatą, pirmiausia turėjome parengti bandymų planą, kuriame būtų aišku, kaip mokymasis yra tinkamiausias. Nustatėme tris metodus (reguliarius mokymus, mokymasis su pašalintu fonu, mokymus su prisegtais fonais) ir penkių tipų duomenis (kai fono spalva yra specifinė, kai fonas yra lygus ir natūralus, kai fonas yra statinis, kai fonas keičiasi ir kai fonas yra su keletu pašalinių asmenų). Visi bandymų rezultatai pateikti žemiau.

4.8. Eksperimentų rezultatai ir jų analizė

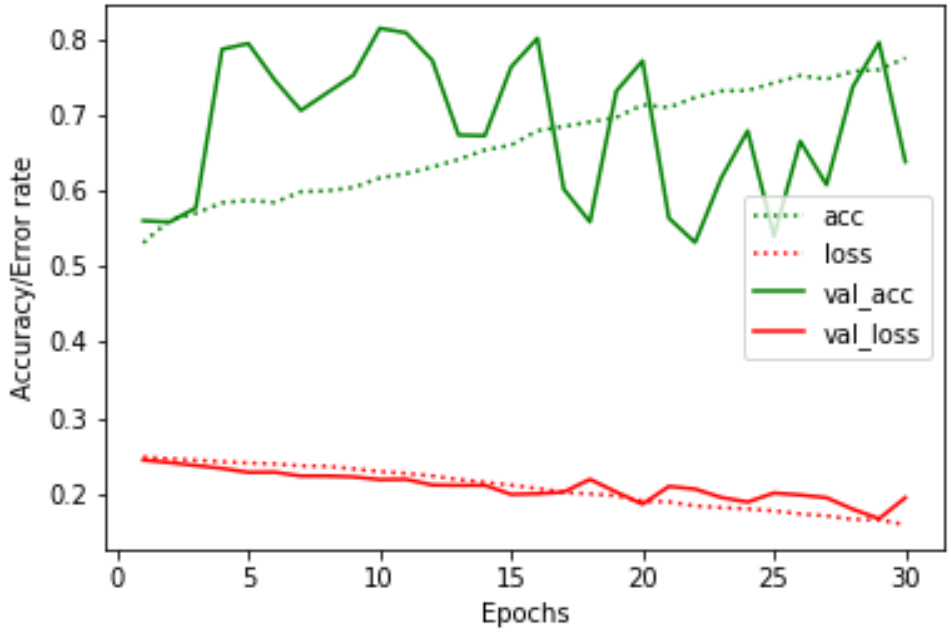
Šiame skyriuje išsamiai paaiškinta, kokie eksperimentai buvo atlikti ir kokie rezultatai buvo pasiekti.

Kaip buvo minėta anksčiau, pirmieji eksperimentai buvo atlikti naudojant duomenų rinkinį su nepašalintais fono vaizdais mokymui ir validavimui. Kiti parametrai buvo:

- Paveikslėlio plotis: 64
- Paveikslėlio aukštis: 40
- Apmokymo duomenų rinkinio imtis: 421
- Validacijos duomenų rinkinio imtis: 122
- Epochos: 30
- Paketo dydis: 32
- Modelio klaidos funkcija: vidurkio kvadratinė klaida
- Modelio optimizavimo funkcija: sgd
- Modelio metrikos: tikslumas

Po mokymo mes turime rezultatų, kurie parodyta 4.4 paveiksle, ir jie po paskutinio iteracijos buvo:

- Mokymo tikslumas: 78%
- Mokymo klaida: 0,16
- Validacijos tikslumas: 64%
- Validacijos klaida: 0,19



Pav. 4.4 Apmokymo duomenimis, be fono pašalinimo, rezultatų grafikas

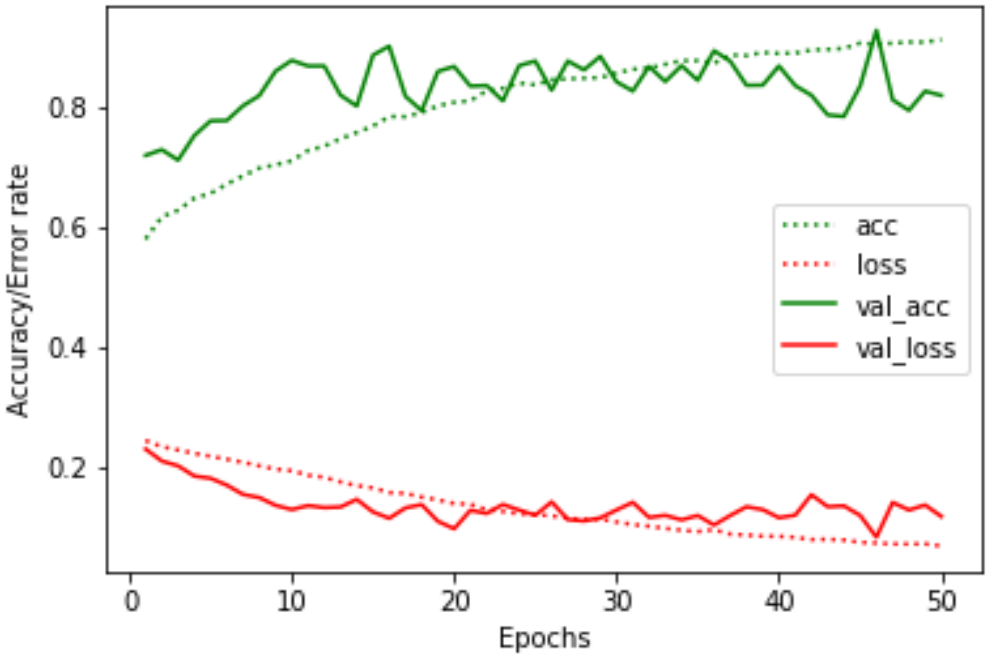
Rezultatas rodo, kad modelis validuoja naujus vaizdus 64% tikslumu. Tačiau po to, kai rankiniu būdu išbandėte šį modelį su vaizdais, kurie labai skirtingoje aplinkoje, rezultatas buvo dar blogesnis.

Kitas eksperimentas buvo atliktas mokymo modeliu su pašalintu fonu. Modelio parametrai buvo:

- Paveikslėlio plotis: 64
- Paveikslėlio aukštis: 40
- Apmokymo duomenų rinkinio imtis: 421
- Validacijos duomenų rinkinio imtis: 121
- Epochos: 50
- Paketo dydis: 32
- Modelio klaidos funkcija: vidurkio kvadratinė klaida
- Modelio optimizavimo funkcija: sgd
- Modelio metrikos: tikslumas

Po mokymo mes turime rezultatų, kurie parodyta 4.5 paveiksle, ir jie po paskutinio iteracijos buvo:

- Mokymo tikslumas: 91%
- Mokymo klaida: 0,07
- Validacijos tikslumas: 82%
- Validacijos klaida: 0,12



Pav. 4.5 Apmokymo duomenimis, be fono pašalinimo, rezultatų grafikas

Šiuo metu rezultatas rodo, kad modelis validuoja su naujais vaizdais 82% tikslumu ir po to, kai šis modelis buvo išbandytas su labai skirtingų aplinkų vaizdais, rezultatas rodomas tuo pačiu tikslumu.

Visi atlikti testai ir jų rezultatai pateikti lentelėje žemiau. Kai kurie iš sudėtingiausių mokymų eksperimentų nebuvo atlikti, nes nebuvo prasminga juos atlikti dėl prastai atliktų paprastesnių mokymų rezultatų.

Lentelė 4.1 Skirtingų apmokymų validacijos lentelė

30 epochų	***A***	***B***	***C***
1. *Konkrečios spalvos fonas*	-	91/82	86/77
2. *Lygus natūralus fonas*	81/77	92/80	81/70
3. *Statinis fonas(margas)*	52/54	62/55	58/52
4. *Kintantis fonas*	40/50	53/50	Nebandyta

Lentelės stulpeliai rodo modelio mokymą, eilutes - kaip buvo patvirtintas modelis.

- A - paprastas mokymasis
- B - parengtas pašalinus foną
- C - apmokyti su fono priedais, patvirtinti su nekeičiamaisiais
- x / y - mokymo tikslumas / patvirtinimo tikslumas

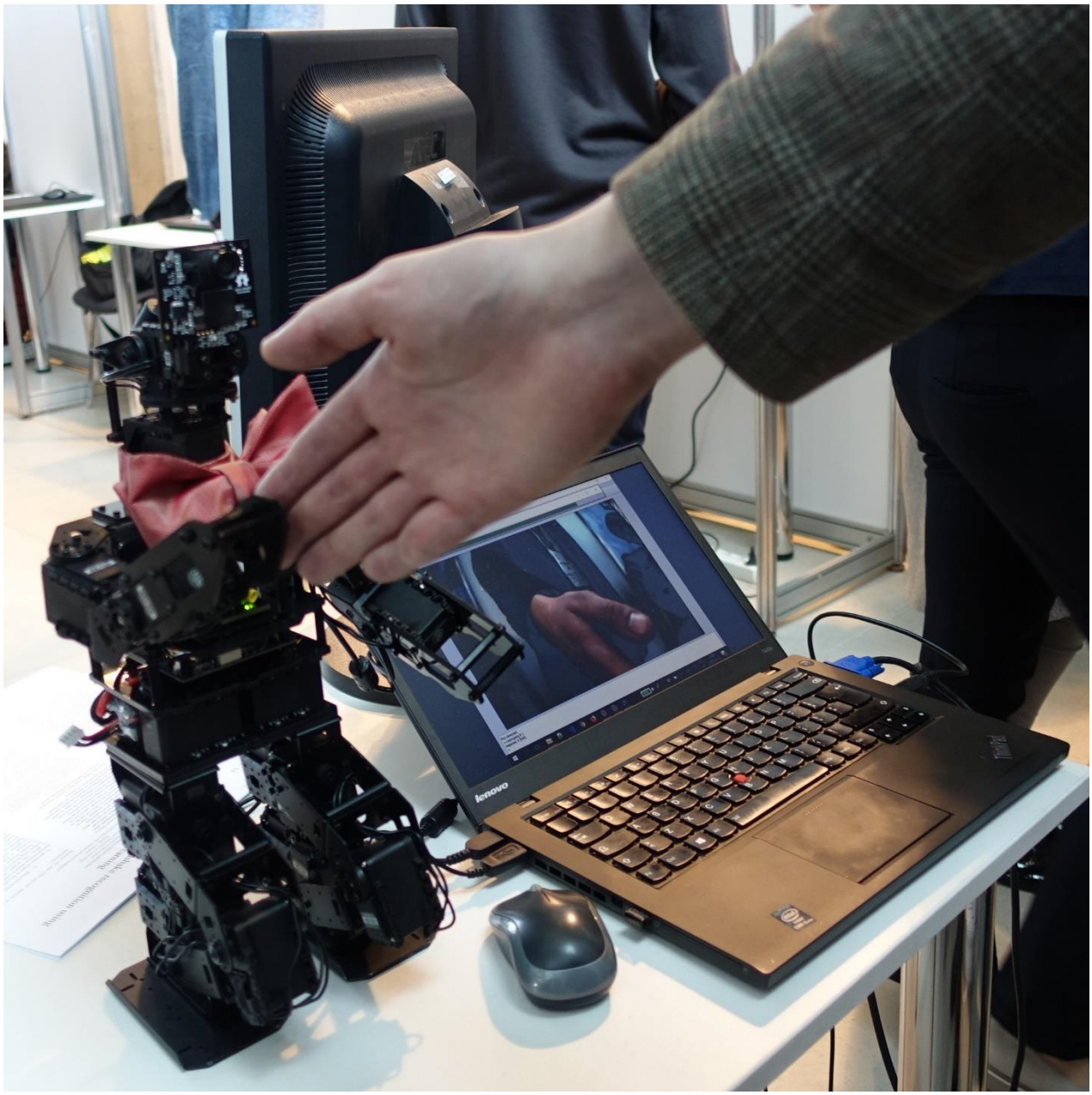
Rezultatai parodė, kad fono pašalinimas gerokai padidina modelio atpažinimo tikslumą. Geriausi rezultatai buvo iš eksperimentų, kuriuose dalyvavo mokymai su pašalintomis fono nuotraukomis ir validuojami taip pat pašalinus fono vaizdus arba su lygaus fono vaizdais.

Rezultatus galima paaiškinti taip, kad fono pašalinimas sumažina duomenų svyravimus ir mašininio mokymosi modelis yra sutelkiamas į asmenį esantį nuotraukoje. Be fono pašalinimo, modeliai yra linkę persimokyti, tikriausiai dėl to, jog priimant sprendimą remiasi netinkamais vaizdo ypatumais. Gali būti, kad panašų tikslumą galima pasiekti be fono pašalinimo, tačiau tam galimai reiktų daug daugiau duomenų, ilgesnių mokymų ir tikriausiai galingesnių modelių. Tokiu atveju modeliai turi daryti išvadą, kad asmuo iš pirmo žvilgsnio yra svarbiausias vaizdų objektas ir išmokti atskirti jį atskirai. Judėjimo arba gylio informacija, kuri naudojama daugelyje gestų atpažinimo sistemų, taip pat padėtų atskirti priekį esantį asmenį nuo paprasto fono.

4.9. Realus sistemos išbandymas

Sukurta sistema buvo pristatyta „KTU Technorama 2018“ technologijų parodoje. Jos metu visi savanoriai galėjo išbandyti kaip veikia sistemos dabartinė versija. Renginio metu buvo

pastebėta, kad sistema šiek tiek priklausoma nuo nuotraukos spalvų tono, nustačius kad kamera automatiškai keistų baltos spalvos balansą pasisveikinimo atpažinimo tikslumas padidėjo.



Pav. 4.6 Sistemos pristatymas "KTU Technorama 2018" technologijų parodoje

Šis darbas laimėjo įmonės „UAB Intermedix Lietuva“ įsteigtą prizą.

5. IŠVADOS

1) Iš pradžių buvo atlikta rinkoje egzistuojančių sprendimų analizė, buvo išbandytos kelios skirtingos giliojo mokymosi bibliotekos ir galiausiai pasirinkta „Keras“ dėl lengvai ir greitai kuriamų prototipų, konvoliucinių tinklų palaikymo ir sklandaus veikimo vykdant apmokymus tiek CPU, tiek GPU.
2) Rankiniu būdu buvo surinktas paveikslėlių duomenų rinkinys. Duomenys buvo su skirtingomis aplinkomis, žmonėmis, aprangomis. Ši įvairovė pravertė geram sistemos veikimui. Duomenų rinkinys buvo padalintas į tris kategorijas: apmokymo, validacijos ir testavimo.
3) Atlikus keletas skirtingų mokymų eksperimentų, stebint ir tiriant mokytų modelių tikslumą. Eksperimentai parodė, kad rezultatai labiau priklausė ne nuo naudojamo modelio ir jo parametrų, o nuo vaizdų transformacijos. Norint pasiekti geriausių rezultatų, vaizdas šiame procese vaidino pagrindinį vaidmenį. Geriausias rezultatas pasiektas nuimant foną prieš apmokymą.
4) Geriausi validacijos rezultatai gauti su duomenimis, turinčiais vienspalvį foną. Validuojant šiais duomenimis tikslumas siekė 82%. Validuojant su natūralaus vienspalvio fono duomenimis gautas 80% tikslumas.
5) Sistema sėkmingai realizuota įgyvendinant iškeltus funkcinius ir nefunkcinius reikalavimus. Šis projektas yra tinkama priemonė įgauti naujų žinių ir sudominti aplinkinius.
6) Įsitikinau, kad gilusis mokymasis tinkamas spręsti vaizdų atpažinimo problemoms.
7) Veikianti sistema pademonstruota „KTU Technorama 2018“ technologijų parodoje.
8) Sistema gali būti toliau tobulinama, gerinant vaizdo atpažinimą įvairesnėse aplinkose, išmokant naujų gestų.

6. LITERATŪRA

[1] Nežinomas autorius. Nervų sistema [Žiūrėta 2016 09 21]. Prieiga internete <https://lt.wikipedia.org/wiki/Nerv%C5%B3_sistema>

[2] HR-OS1 Humanoid Endoskeleton specifications, [Žiūrėta 2018 03 28]. Prieiga internete <http://www.trossenrobotics.com/HR-OS1>

[3] "Why use Keras?" [Žiūrėta 2017 11 12]. Prieiga internete <https://keras.io/why-use-keras/>

[4] How the Wolfram Language Image Identification Project Works [Žiūrėta 2016 11 08]. Prieiga internete <https://www.imageidentify.com/about/how-it-works>

[5] The Wolfram Language Image Identification Project [Žiūrėta 2016 11 08]. Prieiga internete <https://www.imageidentify.com/result/0uo3d169br3e0/>

[6] Clarifai Demo [Žiūrėta 2016 11 08]. Prieiga internete <https://www.clarifai.com/demo>

[7] Build your apps on top of an advanced image tagging technology! [Žiūrėta 2016 11 08]. Prieiga internete <https://imagga.com/>

[8] Auto-Tagging demo [Žiūrėta 2016 11 08]. Prieiga internete <https://imagga.com/auto-tagging-demo?url=https://imagga-com-assets.azureedge.net/static/images/tagging/wolf-725380_640.jpg>

[9] Joanna Materzynska, "Building a Gesture Recognition System using Deep Learning", [Žiūrėta 2018 04 12]. Prieiga internete <https://medium.com/twentybn/building-a-gesture-recognition-system-using-deep-learning-video-d24f13053a1>

[10] Micheal Beyeler, "Hand Gesture Recognition Using a Kinect Depth Sensor", [Žiūrėta 2018 04 15]. Prieiga internete <https://hub.packtpub.com/hand-gesture-recognition-using-kinect-depth-sensor/>

[11] Maryam Asadi-Aghbolaghi, Albert Clapes: A survey on deep learning based approaches for action and gesture recognition in image sequences [Žiūrėta 2018 04 17]. Prieiga internete <http://sunai.uoc.edu/~vponcel/doc/survey-deep-learning_fg2017.pdf>

[12] Feng J., Darrell T.: Learning The Structure of Deep Convolutional Networks [Žiūrėta 2016 11 16]. Prieiga internete <http://www.cv-

foundation.org/openaccess/content_iccv_2015/papers/Feng_Learning_The_Structure_ICCV_2015_paper.pdf>

[13] Krizhevsky A., Sutskever I., Hinton G. E.: ImageNet Classification with Deep Convolutional Neural Networks [Žiūrėta 2016 10 18]. Prieiga internete <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>

[14] Prasad Pai, “Data Augmentation Techniques in CNN using Tensorflow” [Žiūrėta 2018 04 12]. Prieiga internete <https://medium.com/ymedialabs-innovation/data-augmentation-techniques-in-cnn-using-tensorflow-371ae43d5be9>

7. PRIEDAI

Straipsnis „A polite robot: visual handshake recognition using deep learning“ anglų kalba, kuris buvo priimtas į konferenciją SYSTEM2018 ir bus pristatytas iki magistrinio darbo gynimo.

A polite robot: visual handshake recognition using deep learning

Liutauras Butkus, Mantas Lukoševičius
Faculty of Informatics
Kaunas University of Technology
Kaunas, Lithuania
liutauras.butkus@ktu.edu, mantas.lukosevicius@ktu.lt

***Abstract*—Our project was to create a demo system where a small humanoid robot accepts an offered handshake when it sees it. The visual handshake recognition, which is the main part of the system proved to be not an easy task. Here we describe how and how well we solved it using deep learning. In contrast to most gesture recognition research we did not use depth information or videos, but did this on static images. We wanted to use a simple camera and our gesture is rather static. We have collected a special dataset for this task. Different configurations and learning algorithms of convolutional neural networks were tried. However, the biggest breakthrough came when we could eliminate the background and make the model concentrate on the person in front. In addition to our experiment results we can also share our dataset.**

Keywords—image recognition, computer vision, deep learning, convolutional neural networks, robotics

Introduction

The goal of this project is to create a robot that can visually recognize an offered handshake and accept it. When the robot sees a man offering a handshake, it responds by stretching its arm too. This serves as a visual and interactive demonstration, which would get students more interested in machine learning and robotics.

For this purpose we used a small humanoid robot, a simple camera mounted on it, and deep convolutional neural networks for image recognition. The recognition, as well as training of it, were done on a PC and the command to raise the arm was sent back to the robot.

This article mainly shares our experience in developing and training the visual handshake recognition system, which proved to not be trivial. In particular, we will discuss how images were collected, preprocessed, what architecture of convolutional neural networks was used, how it was trained and tested; what gave good and what not so good results.

This document is divided into several sections. Section II reviews existing solution to similar problem. Section III introduces our method for this project. Section IV describes the data set used in this study. Section V emphasize importance of data preprocessing before training. Section VI describes robot interface. Sections VII and VIII provide analysis of results and conclusions.

Related work

To get more deeply to this subject, we will review an existing solution which is similar with the one this article is about. The project called “Gesture Recognition System using Deep Learning”. This work was presented in PyData Warsaw 2017 conference [1]. The author introduced a Python-based, deep learning gesture recognition model. She claimed, that the model is deployed on an embedded system, works in real-time and can recognize 25 different hand gestures from simple webcam stream. In contrast to traditional vision-based gesture controllers like the Microsoft Kinect [2], their system requires no depth information. The development of such an architecture is a complex process that requires careful consideration during each step. They had such processes: large-scale crowd-acting operation to collect over 150,000 short video clips, a process to decide which deep learning framework to use, the development of a network architecture that allows for classifications of video clips solely with RGB input frames, the iterations necessary to make the neural network run in real-time on embedding devices, and lastly, the discovery and development of playful gesture-based applications.

Their approach is slightly different from my approach that is because they used video samples as their dataset items and tried to recognize moving gesture (several frames at a time). More about this approach variants is described in

Maryam Asadi-Aghbolaghi work “A survey on deep learning based approached for action and gesture recognition in image squences” [3]. While my task was to recognize gesture from image (single frame). This approach was chosen because offering a handshape is quite a static gesture: holding a stretched arm still. Furthermore, our used camera sensor wasn’t expensive, which made the task more difficult.

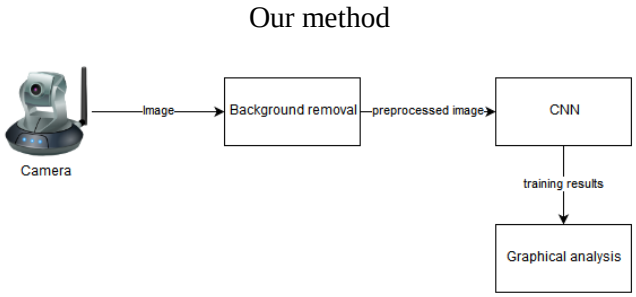


Fig. 1. System training model.

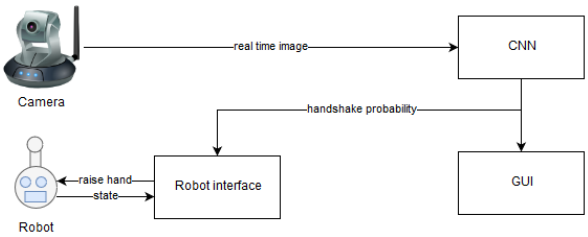


Fig. 2. System running model.

The system model consists of several parts showed in Figure 1 and 2, including camera, camera images pre-processing, convolutional neural network’s training using deep learning, graphical user interface, robot interface and robot itself.

At first camera was used to collect the image dataset. This is described in section IV. After later research, which is described in section VI, the images in the dataset had to be pre-processed to be able to up-train the model, which is the next part of our system. Using Keras library the model was created, compiled and finally trained with the images (this process explained in section VI). The final part is to run the model to recognize new live images. For this reason camera’s interface was programmed to take photos at every 0.5 second, the model gets those images as an input and returns probability of seeing an offered handshake as an output result. If this result is above a certain threshold, a robot interface sends a command to robot to perform a corresponding task. This part more deeply described in section VI.

A. Choice of using deep learning libraries

Deep learning [4] (also known as deep structured learning or hierarchical learning) is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Learning can be supervised, semi-supervised or unsupervised.

Deep learning models are loosely related to information processing and communication patterns in a biological nervous system, such as neural coding that attempts to define a relationship between various stimuli and associated neuronal responses in the brain.

Deep learning architectures such as deep neural networks, deep belief networks and recurrent neural networks [5] have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics and drug design, where they have produced results comparable to and in some cases superior to human experts.

Convolutional networks [4], also known as convolutional neural networks, or CNNs, are a specialized kind of neural network for processing data that has a known grid-like topology. Examples include time-series data, which can be thought of as 1-D grid taking samples at regular time intervals, and image data, which can be thought of as a 2-D grid of pixels. Convolutional networks have been tremendously successful in practical applications. The name “convolutional neural network” indicates that the network employs a mathematical operation called convolution. Convolution is a specialized kind of linear operation. Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers.

Keras [6] is a deep learning library for Theano and TensorFlow. It is a high-level neural networks library, written in Python and capable of running on top of either TensorFlow or Theano. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research.

Keras deep learning library allows for easy and fast prototyping (through total modularity, minimalism, and extensibility). It supports both convolutional networks (we used in my solution) and recurrent networks, as well as combinations of the two. Keras also supports arbitrary connectivity schemes (including multi-input and multi-output training) and runs seamlessly on CPU and GPU. The core data structure of Keras is a model, a way to organize layers. The main type of model is the Sequential model, a linear stack of layers. Keras' Guiding principles include Modularity. A model is understood as a sequence or a graph of standalone, fully-configurable modules that can be plugged together with as little restrictions as possible. In particular, neural layers, cost functions, optimizers, initialization schemes, activation functions, regularization schemes are all standalone modules that users can combine to create new models. Each module should be kept short and simple. Every piece of code should be transparent upon first reading. No black magic: it hurts iteration speed and ability to innovate. New modules are dead simple to add (as new classes and functions), and existing modules provide ample examples. To be able to easily create new modules allows for total expressiveness, making Keras suitable for advanced research.

B. Training process

Before starting to up-train the model there are several parameters which describe training details. The first parameter is epochs count. Epoch itself is an arbitrary milestone, generally defined as "one pass over the entire dataset", used to separate training into distinct phases, which is useful for logging and periodic evaluation. In general it means how many times the process will go through the training set. Second parameter is batch size. Batch size defines number of samples that going to be propagated through the network. For instance, there are 200 training samples and we want to set up batch size equal to 30. Algorithm takes first 30 samples from the training dataset and trains network. Next it takes second 30 samples and trains network again. The procedure can be done until we propagate through the networks all samples. However, the problem usually happens with the last set of samples. In this example the last 20 samples which is not divisible by 30 without remainder. The simplest solution is just to get final 20 samples and train the network. The last two parameters are target size and class mode. Target size accepts image height and width values from dataset and class mode can be assign one of these values: ["categorical", "binary", "sparse", "None"]. In this case this argument gets "binary" as a value, just because we want to recognize only true or false result. Now the training can be started. After start in the console we can see some information about training. There are two main values which are changing each epoch, its accuracy and loss. Accuracy value is percent, while loss is not. Train accuracy and train loss are calculated on the go, during training. These figures show how well our network is doing on the data it is being trained. Training accuracy usually keeps increasing throughout training.

C. Validation process

To validate the model we need to have new dataset this new images, which has not been used in training process. Validation is usually carried out together with training. After every epoch, the model is tested against a validation set, and validation loss and accuracy are calculated. These numbers tell you how good your model is at predicting outputs for inputs it has never seen before. Validation accuracy increases initially and drops as you over fit. Overfitting happens when our model fits too well to the training set. It then becomes difficult for the model to generalize to new examples that were not in the training set. For example, our model recognizes specific images in your training set instead of general patterns. Our training accuracy will be higher than the accuracy on the validation/test set.

D. Testing process

To test the model we need another new dataset. Testing usually is run manually by giving an image from dataset for up-trained model to get a result. And the result is a percent value that shows probability on each output option.

Data Collection Preparation

As we mentioned in the previous section a collection of image data was needed to implement this project. As the system only recognizes greetings, only two results are possible: greetings are recognized or not. During the development of the whole project, more than 4,000 different images were collected for the training of the neural network. Approximately 2000 for each category. Single-image resolution is 318x198.

We can see in Figure 3, that in the image, one person was usually with his hand stuck or not. It was also tried to capture images in as many different environments as possible. Human clothing was also varied trying to capture as diverse as possible colors. This is important in order to ensure that recognition is not restricted to a particular specific situation.

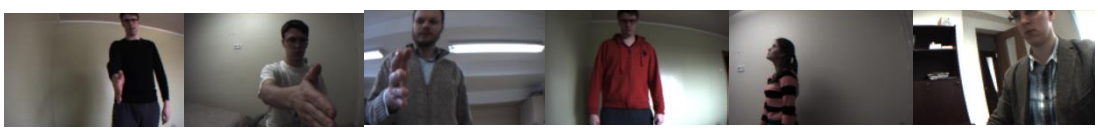


Fig. 3. Image samples: top positive, bottom negative.

The pictures were divided into three sets: training, validation and testing. The neural network is taught with training data. It is then validated with validation data to verify that a well-trained neural network performs recognition with new examples. The test data is intended to validate the final neural network's capability to obtain the final true recognition. In addition, data augmentation [7] was used during training, in which various small transformations were made to the images before training on them (rotation, translation, up- /down-scaling).

If there are people who are interested in this task, we could share the data with everyone who wants it.

Background Removal

Initially, we tried to train the neural network with the data obtained directly from the camera without preprocessing them. However, it has been noticed that the model with the best attempt reached 78 percent training accuracy and about 64 percent validation accuracy followed by overfitting, during which the error rate increased significantly. For this reason, it was necessary to look for solutions on how to avoid overfitting and how to increase the validation accuracy of the model. To achieve this, attempts were made to change the model's parameters, but this did not improve result as much as it was expected. Then it was decided to process the data itself. From previous experiments, we were able to get the impression that overfitting appears due to the excessive color gamut and color of the images. For this reason, we have decided to try removing background images and training a neural network with pictures without background. However, that causes a new problem. How to detect where the background is and where is an object (in this case a human)? For this problem, we decided to take the first image without a human and claim that it is a background and all other images are objects with backgrounds. Though, in this case camera had to be in fixed position. Then we were able to subtract two images and get image without a background. Usually, after subtraction some noise always had left in images. To reduce it, we set a permissible error for pixel RGB values. You can see those images in Figure 4.

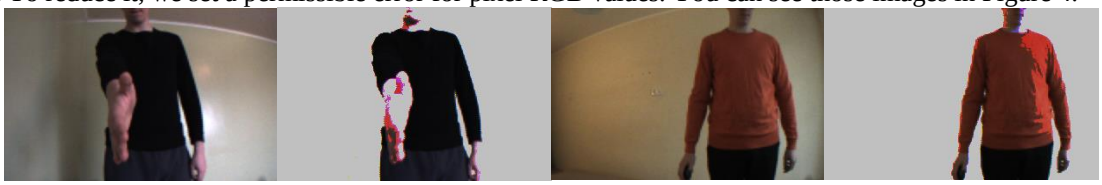


Fig. 4. Same image with and without background.

The result was obvious. It's possible to achieve 91 percent accuracy with a smooth natural background. This means that the model is quite precise enough to recognize the extended hand when the background behind the human is equal and does not need to be removed.

In order to obtain this result, first we needed to draw up a test plan, which would make it clear how the training is most appropriate. We've identified three methods (regular training, training with removed background, training with attached backgrounds), and five types of data (when the background is a specific color, when the background is smooth and natural, when the background is static color, when the background is changing and when the background is with a few outsiders). All test results are presented in the results section (VII).

Interfacing Robot

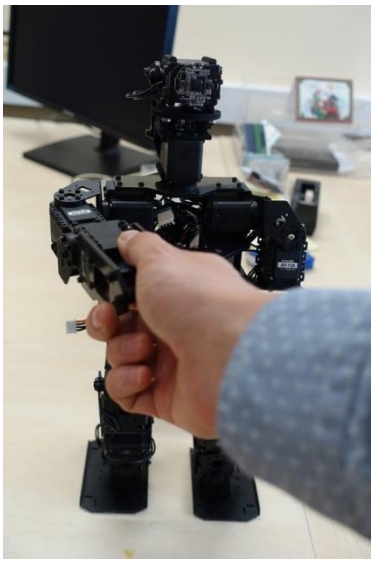


Fig. 5. Photo of our robot and use case.

For this project HR-OS1 Humanoid Endoskeleton robot [8] was used. It is showed in Figure 5. It has integrated onboard Linux computer with Intel Atom processor, which gives all the processing power to run robot. The HR-OS1 is a hackable, modular, humanoid robot development platform designed from the ground up with customization and modification in mind. It has built in software which invokes robot actions. You can see robot's software interface in Figure 6.

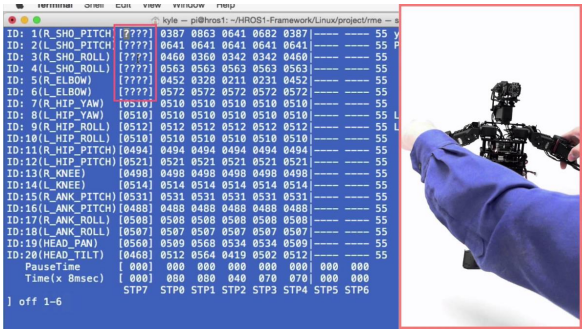


Fig. 6. Robot joints management interface.

Robot interface is used when the model is started to predict new images. After CNN return probability of the image it is sent to robot interface. Then robot interface reads the input value and if it is true interface runs command for a robot to raise its hand.

Experiment results and Analysis

In this section we will explain in detail what experiments were done and what results were achieved.

As we mentioned in the previous section the first experiments were carried out using dataset with non-removed background images for training and validation. The other parameters were:

- Image width: 64
- Image height: 40
- Training dataset samples: 421
- Validation dataset samples: 122
- Epochs: 30
- Batch size: 32
- Model loss function: mean squared error
- Model optimizer: sgd
- Model metrics: accuracy

After training we have got the results, which are shown in Figure 7, and they after final iteration were:

- Training accuracy: 78%
- Training loss: 0.16
- Validation accuracy: 64%
- Validation loss: 0.19

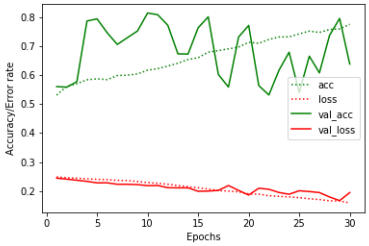


Fig. 7. Training without removing background images results graph.

The result shows that model validates new images by 64% accuracy. However, after testing manually this model with images which very different environments, the result have been even worse.

The next experiment were held by training model with removed background. The model parameters were:

- Image width: 64
- Image height: 40
- Training dataset samples: 421
- Validation dataset samples: 121
- Epochs: 50
- Batch size: 32
- Model loss function: mean squared error
- Model optimizer: sgd
- Model metrics: accuracy

After training we got the results, which are shown in Figure 8, and they after final iteration were:

- Training accuracy: 91%
- Training loss: 0.07
- Validation accuracy: 82%
- Validation loss: 0.12

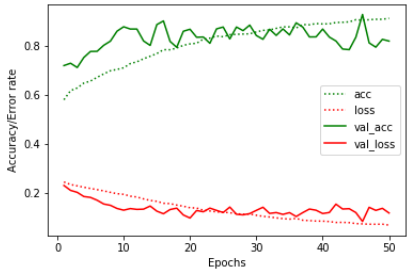


Fig. 8. Training with removed background iamges results graph.

This time the result shows that model validates new images by 82% accuracy and after testing manually this model with images which very different environments, the result shown about the same accuracy.

All my tests and their results are shown in the table below. Some of the most sophisticated training experiments have not been completed, as it was not immediately meaningful to perform them due to poor results from simple training.

TABLE I. DIFFERENT TRAINING VALIDATION TABLE

30 epochs	**A**	**B**	**C**
5. Background of a specific color	-	91/82	86/77
6. A smooth natural background	81/77	92/80	81/70
7. Static background (colorful)	52/54	62/55	58/52

8. *Changing background*	40/50	53/50	Not tried
9. *Background with similar people*	Not tried	Not tried	Not tried

Columns of the table represents training of the model, rows – how model was validated.

A - Simple training
B - Trained with removed background
C - Trained with background attachments, validates with non-removable ones
x/y – training accuracy / validation accuracy

The results showed that the removal of the background significantly improves the accuracy of the model recognition. The best results were from experiments where training took place with removed background pictures and validating with also removed background images or smooth background images.

Discussion and future work

In this work several different training experiments were performed, watching and studying accuracy of the trained models. The experiments showed that the results depended more not on the model used and its parameters, but on the transformation of the images. To achieve the best results image preprocessing played a key role in this experiment. The best result was reached by removing background before training.

Our interpretation of the results is that removing the background reduces the variation in the data and makes the machine learning model focus on the person in the image. Without the background removal the models are prone to overfitting, probably basing their decision on wrong features of the image. Might be that similar accuracy can be achieved without background removal, but with much more data, training, and probably more powerful models. In that case the models have to infer that the person in the foreground is the most important object in the images and learn how to distinguish it on its own. Motion or depth information, which is used in many gesture recognition systems would also make separation of the person in front form the background easier.

Best validation results are on data with smooth natural backgrounds. The accuracy of this validation data reached 92%. For a near-future work we will attempt to create a model that can better recognize an offered handshakes in a wider range of environments.

References

[1] Joanna Materzynska, “Building a Gesture Recognition System using Deep Learning”, https://medium.com/twentybn/building-a-gesture-recognition-system-using-deep-learning-video-d24f13053a1

[2] Micheal Beyeler, “Hand Gesture Recognition Using a Kinect Depth Sensor”, https://hub.packtpub.com/hand-gesture-recognition-using-kinect-depth-sensor/

[3] Maryam Asadi-Aghbolaghi, Albert Clapes, Marco Bellantonio, Hugo Jair Escalante, “A survey on deep learning based approaches for action and gesture recognition in image sequences”, http://sunai.uoc.edu/~vponcel/doc/survey-deep-learning_fg2017.pdf, pp. 2

[4] Ian Goodfellow, Yoshua Bengio, Aaron Courville, “Deep Learning”, pp. 1-8, 326

[5] Denny Britz, „Recurrent Neural Networks Tutorial, Part 1 – Introduction to RNNs“, http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/

[6] “Why use Keras?”, https://keras.io/why-use-keras/

[7] Prasad Pai, “Data Augmentation Techniques in CNN using Tensorflow”, https://medium.com/ymedialabs-innovation/data-augmentation-techniques-in-cnn-using-tensorflow-371ae43d5be9

[8] HR-OS1 Humanoid Endoskeleton spescifications, http://www.trossenrobotics.com/HR-OS1