



Kauno technologijos universitetas

Informatikos fakultetas

**Dirbtinio intelekto metodų tyrimas, siekiant aptikti
išsiskiriančius vairavimo atvejus iš vaizdo kamerų medžiagos**

Baigiamasis magistro studijų projektas

Paulius Melaika

Projekto autorius

Prof. Rytis Maskeliūnas

Vadovas

Kaunas, 2021



Kauno technologijos universitetas

Informatikos fakultetas

**Dirbtinio intelekto metodų tyrimas, siekiant aptikti
išsiskiriančius vairavimo atvejus iš vaizdo kamerų medžiagos**

Baigiamasis magistro studijų projektas

Programų sistemų inžinerija (6211BX011)

Paulius Melaika

Projekto autorius

Prof. Rytis Maskeliūnas

Vadovas

Prof. Robertas Damaševičius

Recenzentas

Kaunas, 2021



Kauno technologijos universitetas

Informatikos fakultetas

Paulius Melaika

Dirbtinio intelekto metodų tyrimas, siekiant aptikti išsiskiriančius vairavimo atvejus iš vaizdo kamerų medžiagos

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;
2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąrašė;
3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs (-usi);
4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Paulius Melaika

Patvirtinta elektroniniu būdu

Paulius Melaika. Dirbtinio intelekto metodų tyrimas, siekiant aptikti išsiskiriančius vairavimo atvejus iš vaizdo kamerų medžiagos. Magistro studijų baigiamasis projektas / vadovas prof. Rytis Maskeliūnas; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Informatikos mokslai, Programų sistemų inžinerija.

Reikšminiai žodžiai: dirbtinis intelektas, vaizdų atpažinimas, linijų atpažinimas, miglotoji logika, išsiskiriantys vairavimo atvejai.

Kaunas, 2021. 92 p.

Santrauka

Šiame darbe aprašomas sistemos, aptinkančios išsiskiriančius vairavimo atvejus iš vaizdo kamerų medžiagos kūrimas, pateikiamas sukurtas išsiskiriančio vairavimo atvejo aptikimo algoritmas, jo charakteristikos.

Šis darbas susideda iš analitinės, projektinės, tyrimo ir eksperimentinės dalių.

Analitinėje dalyje pateikiamas techninis pasiūlymas, atliekama literatūros ir konkurentų analizė. Projektinėje dalyje aprašomi sistemai keliami reikalavimai, pateikiama sistemos architektūra. Tyrimo dalyje aprašomas sukurtas algoritmas. Eksperimentinėje dalyje pateikiamos iširtos sukurto algoritmo charakteristikos.

Darbo pabaigoje pateikiami šio darbo rezultatai ir išvados.

Paulius Melaika. Research of Machine Learning Methods for Detection of Exceptional Driving in Video Materials. Master's Final Degree / supervisor prof. Rytis Maskeliūnas; Faculty of Informatics, Kaunas University of Technology.

Study field and area (study field group): Informatics, Software Engineering.

Keywords: artificial intelligence, image recognition, lane detection, fuzzy logic, exceptional driving
Kaunas, 2021. 92 p.

Summary

Exceptional Driving Detection System is presented in this thesis. This paper covers the exceptional driving detection from video materials algorithm and its performance metrics.

This paper consists of analysis, project, research and experiment parts.

Analysis part describes technical offer, literature and competitor analysis. Project part covers system requirements and system architecture. The creation of algorithm is described in research part. Experiment part covers created algorithm performance metrics.

This paper is wrapped up with conclusions and results.

Turinys

Lentelių sąrašas	9
Paveikslų sąrašas	11
Santrumpų ir terminų sąrašas	13
Įvadas.....	14
1. Analitinė dalis	15
1.1. Techninis pasiūlymas	15
1.1.1. Sistemos apibrėžimas	15
1.1.2. Bendras veikos tikslas	15
1.1.3. Sistemos kontekstas.....	15
1.1.4. Projekto vartotojai ir klientai.....	16
1.1.5. Rinkos tyrimas.....	17
1.1.6. Konkurencija ir alternatyvos	17
1.1.7. Projekto įgyvendinimo planai ir kokybės vertinimas	18
1.1.8. Bendri apribojimai.....	19
1.2. Projektavimo metodologijos ir technologijų analizė	19
1.2.1. Egzistuojantys pritaikymai	19
1.2.2. Eismo juostos aptikimas	19
1.2.3. Automobilio sekimas pagal šviesas.....	20
1.2.4. Automobilio numerio nuskaitymas	20
1.2.5. Automobilio greičio nustatymas.....	21
1.2.6. Susidūrimų prognozavimas	21
1.2.7. Posūkio signalų rodymo sekimas	21
1.2.8. Trajektorijų klasterizavimas	22
1.2.9. Planuojami naudoti skaičiavimo metodai.....	23
1.2.10. Konvoliucinis neuroninis tinklas	23
1.2.11. Miglotoji logika	24
1.2.12. Linijos sekimo algoritmas	24
1.2.13. Planuojamos taikyti technologijos.....	25
1.2.14. Atrinktų technologijų pagrindimas.....	25
1.2.15. Atrinktų matematinių metodų pasirinkimo pagrindimas.....	26
1.2.16. Įgyvendinimo problemos.....	27
2. Projektinė dalis	28
2.1. Reikalavimų specifikavimas.....	28
2.1.1. Projekto kūrimo pagrindas	28
2.1.2. Sistemos tikslai	28
2.1.3. Užsakovai, pirkėjai ir kiti sistema suinteresuoti asmenys	28
2.1.4. Vartotojai.....	28
2.1.5. Apribojimai sprendimui.....	29
2.1.6. Diegimo aplinka	29
2.1.7. Bendradarbiaujančios sistemos	30
2.1.8. Komerciniai specializuoti programų paketai.....	30
2.1.9. Numatoma darbo vietos aplinka.....	30
2.1.10. Sistemos kūrimo terminai.....	30
2.1.11. Sistemos kūrimo biudžetas	30

2.1.12. Svarbūs faktai ir prielaidos	30
2.1.13. Veiklos kontekstas	30
2.1.14. Veiklos padalinimas	31
2.1.15. Panaudojimo atvejai	31
2.1.16. Funkciniai reikalavimai	36
2.1.17. Reikalavimai duomenims	41
2.1.18. Nefunkciniai reikalavimai	42
2.1.19. Pagamintos sistemos, kurios gali būti nupirktos	46
2.1.20. Pagaminti komponentai, kurie gali būti panaudoti	46
2.1.21. Problemos diegimo aplinkai	46
2.1.22. Įtaka jau instaliuotoms sistemoms	46
2.1.23. Neigiamas vartotojų nusiteikimas	47
2.1.24. Kliudantys diegimo aplinkos apribojimai	47
2.1.25. Galimos naujos sistemos sukeltos problemos	47
2.1.26. Vystymo etapai	47
2.1.27. Pritaikymas	47
2.1.28. Reikalavimai esamų duomenų perkėlimui	47
2.1.29. Reikalingas duomenų transformavimas perkeliant į naują sistemą	47
2.1.30. Galimos sistemos kūrimo rizikos	48
2.1.31. Atsitiktinumų (rizikų) planas	48
2.1.32. Kaina	48
2.1.33. Vartotojo dokumentacija ir apmokymas	49
2.2. Sistemos statinis vaizdas	49
2.2.1. Sistemos posistemės	49
2.2.2. Paketų detalizavimas	49
2.3. Sistemos dinaminis vaizdas	50
2.3.1. Sistemos sekų diagramos	50
2.3.2. Sistemos bendradarbiavimo diagramos	54
2.3.3. Vairavimo atvejo būsenų diagrama	57
2.3.4. Sistemos veiklos diagramos	58
2.4. Duomenų vaizdas	62
2.5. Vaizdo apdorojimo algoritmo veiksmų diagrama	63
3. Tyrimo dalis	64
3.1. Tyrimo tikslas	64
3.2. Tyrimo aprašymas	64
3.3. Automobilių aptikimas	64
3.3.1. Modelio pasirinkimas	64
3.3.2. Duomenų rinkinys	66
3.3.3. Modelio apmokymas	67
3.4. Trajektorijų skaičiavimas	68
3.4.1. Automobilių padėtis laike	68
3.4.2. Polinominės kreivės interpoliacija	68
3.5. Linijų aptikimas	69
3.6. Miglotoji logika	74
3.6.1. Parametrai	74
3.6.2. Taisyklės	76

3.6.3. Išvestis	77
4. Eksperimentinė dalis	79
4.1. Automobilių aptikimo modelio tikslumas	79
4.2. Greitaveika	81
4.3. Aptikimo tikslumas	81
Išvados	89
Literatūros sąrašas	90

Lentelių sąrašas

1.1 lentelė Konkurentų apžvalga.....	17
1.2 lentelė Sistemos vertinimo kriterijai	18
1.3 lentelė Projekto biudžetas	18
1.4 lentelė „Tensorflow“ ir „Keras“ bibliotekų palyginimas	26
2.1 lentelė Veiklos įvykių sąrašas	31
2.2 lentelė PA „Prisijungti“	32
2.3 lentelė PA „Paleisti aptikimo sistemą“	32
2.4 lentelė PA „Sustabdyti aptikimo sistemą“	33
2.5 lentelė PA „Užsiregistruoti“	33
2.6 lentelė PA „Peržiūrėti informaciją apie išsiskyrusius vairavimo atvejus“	34
2.7 lentelė PA „Peržiūrėti užfiksuoto išsiskiriančio vairavimo atvejo kadra“	34
2.8 lentelė PA „Pažymėti vairavimą kaip išsiskiriantį“	35
2.9 lentelė PA „Apdoroti vaizdą	35
2.10 lentelė PA „Išsaugoti informaciją apie galimai išsiskiriantį vairavimo atvejį“	36
2.11 lentelė PA „Koreguoti vaizdo apdorojimo parametrus“	36
2.12 lentelė Funkcinis reikalavimas 1	37
2.13 lentelė Funkcinis reikalavimas 2	37
2.14 lentelė Funkcinis reikalavimas 3	37
2.15 lentelė Funkcinis reikalavimas 4	38
2.16 lentelė Funkcinis reikalavimas 5	38
2.17 lentelė Funkcinis reikalavimas 6	38
2.18 lentelė Funkcinis reikalavimas 7	39
2.19 lentelė Funkcinis reikalavimas 8	39
2.20 lentelė Funkcinis reikalavimas 9	39
2.21 lentelė Funkcinis reikalavimas 10	40
2.22 lentelė Funkcinis reikalavimas 11	40
2.23 lentelė Funkcinis reikalavimas 12	40
2.24 lentelė Funkcinis reikalavimas 13	41
2.25 lentelė Funkcinis reikalavimas 14	41
2.26 lentelė Funkcinis reikalavimas 15	41
2.27 lentelė Nefunkcinis reikalavimas 16	42
2.28 lentelė Nefunkcinis reikalavimas 17	42
2.29 lentelė Nefunkcinis reikalavimas 18	43
2.30 lentelė Nefunkcinis reikalavimas 19	43
2.31 lentelė Nefunkcinis reikalavimas 20	43
2.32 lentelė Nefunkcinis reikalavimas 21	44
2.33 lentelė Nefunkcinis reikalavimas 22	44
2.34 lentelė Nefunkcinis reikalavimas 23	44
2.35 lentelė Nefunkcinis reikalavimas 24	45
2.36 lentelė Nefunkcinis reikalavimas 25	45
2.37 lentelė Nefunkcinis reikalavimas 26	45
2.38 lentelė Nefunkcinis reikalavimas 27	46
2.39 lentelė Nefunkcinis reikalavimas 28	46
2.40 lentelė Kūrimo rizikos.....	48

2.41 lentelė Rizikų planas	48
3.1 lentelė Modelių savybių lentelė	65
3.2 lentelė Miglotosios logikos taisyklės	77
4.1 lentelė Pirmojo bandymo tikslumo metrikos	82
4.2 lentelė Antrojo bandymo tikslumo metrikos.....	84
4.3 lentelė Trečiojo bandymo tikslumo metrikos.....	85

Paveikslų sąrašas

1.1 pav. Sistemos konteksto diagrama	16
1.2 pav. Eismo juostų aptikimo algoritmas	20
1.3 pav. Posūkių signalų sekimo algoritmas	22
1.4 pav. Trajektorijų klasterizavimo pavyzdys	23
1.5 pav. Kelio juostos atpažinimo metodo principas	25
1.6 pav. „Keras“ (kairėje) ir „Tensorflow“ (dešinėje) bibliotekų greitaveika	26
2.1 pav. Diegimo aplinka	29
2.2 pav. Veiklos kontekstas.....	30
2.3 pav. PA diagrama	31
2.4 pav. ER modelis	42
2.5 pav. Sistemos posistemės	49
2.6 pav. Sistemos posistemės ir klasės.....	49
2.7 pav. PA „Prisijungti“ veiksmų eiliškumo diagrama	50
2.8 pav. PA „Paleisti aptikimo sistemą“ veiksmų eiliškumo diagrama	51
2.9 pav. PA „Sustabdyti aptikimo sistemą“ veiksmų eiliškumo diagrama.....	51
2.10 pav. PA „Užsiregistruoti“ veiksmų eiliškumo diagrama	52
2.11 pav. PA „Peržiūrėti informaciją apie išsiskyrusius vairavimo atvejus“ veiksmų eiliškumo diagrama	52
2.12 pav. PA „Peržiūrėti užfiksuoto išsiskiriančio vairavimo atvejo kadra“ veiksmų eiliškumo diagrama	53
2.13 pav. PA „Pažymėti vairavimą kaip išsiskiriantį“ veiksmų eiliškumo diagrama.....	53
2.14 pav. PA „Apdoroti vaizdą“ veiksmų eiliškumo diagrama	53
2.15 pav. PA „Išsaugoti informaciją apie galimai išsiskiriantį vairavimo atvejį“ veiksmų eiliškumo diagrama	54
2.16 pav. PA „Koreguoti vaizdo apdorojimo parametrus“ veiksmų eiliškumo diagrama	54
2.17 pav. PA „Prisijungti“ sąveikos diagrama	54
2.18 pav. PA „Paleisti aptikimo sistemą“ sąveikos diagrama.....	55
2.19 pav. PA „Sustabdyti aptikimo sistemą“ sąveikos diagrama.....	55
2.20 pav. PA „Užsiregistruoti“ sąveikos diagrama	55
2.21 pav. PA „Peržiūrėti informaciją apie išsiskyrusius vairavimo atvejus“ sąveikos diagrama	55
2.22 pav. PA „Peržiūrėti užfiksuoto išsiskiriančio vairavimo atvejo kadra“ sąveikos diagrama	56
2.23 pav. PA „Pažymėti vairavimą kaip išsiskiriantį“ sąveikos diagrama	56
2.24 pav. PA „Apdoroti vaizdą“ sąveikos diagrama.....	56
2.25 pav. PA „Išsaugoti informaciją apie galimai išsiskiriantį vairavimo atvejį“ sąveikos diagrama	57
2.26 pav. PA „Koreguoti vaizdo apdorojimo parametrus“ sąveikos diagrama.....	57
2.27 pav. Vairavimo atvejo būsenų diagrama	57
2.28 pav. PA „Prisijungti“ veiklos diagrama	58
2.29 pav. PA „Paleisti aptikimo sistemą“ veiklos diagrama	58
2.30 pav. PA „Sustabdyti aptikimo sistemą“ veiklos diagrama.....	59
2.31 pav. PA „Užsiregistruoti“ veiklos diagrama	59
2.32 pav. PA „Peržiūrėti informaciją apie išsiskyrusius vairavimo atvejus“ veiklos diagrama	60
2.33 pav. PA „Peržiūrėti užfiksuoto išsiskiriančio vairavimo atvejo kadra“ veiklos diagrama	60
2.34 pav. PA „Pažymėti vairavimą kaip išsiskiriantį“ veiklos diagrama.....	61

2.35 pav. PA „Apdoroti vaizdą“ veiklos diagrama	61
2.36 pav. PA „Išsaugoti informaciją apie galimai išsiskiriantį vairavimo atvejį“ veiklos diagrama	62
2.37 pav. PA „Koreguoti vaizdo apdorojimo parametrus“ veiklos diagrama.....	62
2.38 pav. Sistemos ER modelis.....	62
2.39 pav. Sistemos duomenų bazės modelis	63
2.40 pav. Vaizdo apdorojimo algoritmo veiksmų diagrama	63
3.1 pav. „ResNet 101“ architektūra	65
3.2 pav. „Faster R-CNN“ architektūra	66
3.3 pav. Trajektorijos skaičiavimas, pažymėjus visą automobilį.....	67
3.4 pav. Trajektorijos skaičiavimas, pažymėjus automobilio priekinę ir galinę dalį.....	67
3.5 pav. Automobilio padėtis laike	68
3.6 pav. Automobilio trajektorija per laiko tarpą.....	69
3.7 pav. Pilkos spalvos filtras.....	70
3.8 pav. HSV spalvų erdvė.....	70
3.9 pav. Kadre aptiktos geltonos ir baltos spalvos	71
3.10 pav. Gauso suliejimas.....	71
3.11 pav. Eismo linijų regionas.....	72
3.12 pav. Slenkstinių reikšmių atskyrimas.....	72
3.13 pav. Canny kraštų aptikimo algoritmas.....	73
3.14 pav. Kadro linijų taškai	73
3.15 pav. Aptiktos kelio linijos	74
3.16 pav. Linijos kirtimų skaičius	75
3.17 pav. Linijos krypties koeficientas	75
3.18 pav. Trajektorijos kreivumas.....	76
3.19 pav. Mažiausio ir didžiausio atstumų iki linijos skirtumas	76
3.20 pav. Vairavimo išskirtinumas	77
3.21 pav. Žemas išskirtinumas	78
3.22 pav. Aukštas išskirtinumas.....	78
4.1 pav. Klasifikacijos klaida	79
4.2 pav. Lokalizacijos klaida.....	79
4.3 pav. Bendra klaida.....	80
4.4 pav. Mokymosi greitis.....	80
4.5 pav. Apmokymo žingsniai per sekundę	80
4.6 pav. Pirmojo bandymo maišos matrica	82
4.7 pav. Pirmojo bandymo ROC kreivė	83
4.8 pav. Antrojo bandymo maišos matrica	83
4.9 pav. Antrojo bandymo ROC kreivė	84
4.10 pav. Trečiojo bandymo maišos matrica	85
4.11 pav. Trečiojo bandymo ROC kreivė	86
4.12 pav. Visų bandymų ROC kreivės.....	86
4.13 pav. Neišsiskiriantys vairuotojai	87
4.14 pav. Eismo įvykyje sudalyvavęs automobilis	87
4.15 pav. Priešprieša atvažiuojantis automobilis	87
4.16 pav. Chuliganiškai besielgiantys vairuotojai.....	88

Santrumpų ir terminų sąrašas

Santrumpos:

PA – panaudojimo atvejis.

ER diagrama – esybių ryšių diagrama.

UML – Unified Modeling Language.

SVM – (angl. *Support-vector machine*) prižiūrimo mašininio mokymosi metodas.

YOLO – (angl. *You Only Look Once*) mašininio mokymosi strategija.

ViF – (angl. *Variance inflation factor*) dispersijos infliacijos koeficientas.

CNN – (angl. *Convolutional Neural Network*) konvoliucinis neuroninis tinklas.

Terminai:

Neblaivus vairuotojas – vairuotojas, kurio kraujyje yra daugiau nei 0,4 promilės alkoholio.

Registratorius – vaizdo kamera, esanti automobilyje, fiksuojanti vaizdą kelyje.

Klasterizavimas – objektų sugrupavimas pagal panašius bruožus.

Konvoliucinis neuroninis tinklas – dirbtinio intelekto rūšis, gerai pritaikoma kompiuterinės regos srityje (vaizdams atpažinti, objektams aptikti, juos identifikuoti).

Hough transformacija – objektų savybių ištraukimo technika, naudojama kompiuterinės regos srityje.

Gradientas – diferencialinis operatorius, apibrėžiantis dydžio greičiausio augimo reikšmę ir kryptį.

ROC kreivė – grafinė diagrama, kuri iliustruoja klasifikatoriaus diagnostinius gebėjimus.

Įvadas

Automobiliai šiame pasaulyje yra pagrindinė susisiekimo priemonė. Dažnas asmuo turi asmeninį automobilį. Automobilių kiekis mūsų keliuose yra didžiulis, piko metu susidaro transporto priemonių kamščiai. Tačiau ne visi vairuotojai yra atsakingi. Dažniausiai eismo įvykiai įvyksta būtent dėl eismo dalyvių neatsakingumo. Dalis vairuotojų yra tiesiog išsiblaškę, kiti pervargę, neišsimiegoję. Dėmesio koncentracijos stoka kelyje lemia skaudžius eismo įvykius. Patys neatsakingiausi vairuotojai yra tie, kurie vairuoja chuliganiškai ar būdami neblaivūs. Neblaivaus vairuotojo reakcija yra lėtesnė negu blaivaus [1]. Šie vairuotojai sukelia skaudžiausius eismo įvykius, kuriuose žūsta žmonės. Nors įstatymai draudžia vairuoti chuliganiškai ar esant neblaiviam ir bausmės už tokius pažeidimus yra griežtos, tačiau kai kurie vairuotojai nekreipia į tai dėmesio. Prie kiekvieno vairuotojo policijos pareigūno nepastatysi. Nors dažnai būna rengiami įvairūs reidai ar prevencinės akcijos, tačiau vis tiek atsiranda asmenų, kurie nesilaiko kelių eismo taisyklių.

Šiuo metu ieškoma būdų, kaip sumažinti skaudžių eismo įvykių skaičių. Kaip jau minėta, rengiami reidai, kurių metu vairuotojai tikrinami alkoholio matuokliu ir nustatomas jų blaivumas, tikrinama ar jie nesielgia chuliganiškai. Taip pat daugėja pilietišku asmenų, kurie praneša apie įtarimą keliančius vairuotojus.

Atliekami įvairūs tyrimai, siekiant nustatyti įtarimą keliančius vairuotojus. Dažniausiai tam yra pasitelkiami vaizdų atpažinimo metodai. Yra įvairių būdų identifikuoti neblaivų / pervargusį vairuotoją. Vienas iš jų yra vairuotojo būsenos nustatymas pagal jo veidą [2]. Šis metodas vairuotojo būseną nustato pagal jo galvos padėtį, mirkčiojimą, žiovavimą. Dar vienas būdas yra aptikti įtarimą keliančius vairuotojus pagal jų daromus pažeidimus. Pavyzdžiui, važiavimas per raudoną šviesą, ištisinės linijos kirtimas [3].

Šis darbas bus orientuotas į išsiskiriančių vairavimo atvejų aptikimą iš kitų vairuotojų automobiliuose esančių ir / ar stacionarių vaizdo kamerų medžiagos. Bus siekiama, naudojantis dirbtinio intelekto metodais, atpažinti automobilį, nustatyti jo trajektoriją, aptikti eismo linijas ir pagal šiuos parametrus nustatyti išsiskiriančius vairuotojus.

Kuriamos sistemos tikslas – naudojantis dirbtinio intelekto metodais, palengvinti išsiskiriančių vairavimo atvejų aptikimą ir taip sumažinti skaudžių eismo įvykių, kuriuose žūsta ar būna sužalojami žmonės, skaičių.

Šio darbo tikslas – sukurti dirbtiniu intelektu paremtą sistemą, skirtą aptikti išsiskiriančius vairavimo atvejus, ir nustatyti jos veikimo tikslumą. Šiam tikslui pasiekti keliami uždaviniai:

- literatūros analizė;
- reikalavimų specifikavimas;
- sistemos architektūros projektavimas;
- algoritmo, skirto aptikti išsiskiriančius vairavimo atvejus, sukūrimas;
- sukurto algoritmo eksperimentinis vertinimas;
- sukurtos sistemos tikslumo nustatymas.

Šį darbą sudaro analitinė, projektinė, tyrimo ir eksperimentinė dalys. Analitinėje dalyje pateikiamas techninis pasiūlymas, atliekama literatūros ir konkurentų analizė. Projektinėje dalyje aprašomi sistemai keliami reikalavimai, pateikiama sistemos architektūra. Tyrimo dalyje aprašomas sukurtas algoritmas. Eksperimentinėje dalyje pateikiamos ištirtos sukurto algoritmo charakteristikos.

1. Analitinė dalis

1.1. Techninis pasiūlymas

1.1.1. Sistemos apibrėžimas

Produktas bus sistema, kuri bus integruojama kartu su vaizdo registratoriumi. Sistema bus skirta įtarimą keliančių vairuotojų aptikimui bendrame eismo sraute.

Ši sistema atliks tokius veiksmus:

1. nuskaitys vaizdą, gaunamą iš vaizdo registratoriaus;
2. identifikuos jame esančius automobilius;
3. seks tuos automobilius;
4. jų vairavimą suklasifikuos į kategorijas;
5. išskirs ir pažymės įtarimą keliančius vairuotojus.

Šie veiksmai bus atliekami į pagalbą pasitelkiant dirbtinio intelekto metodus. Bus naudojami konvoliuciniai neuroniniai tinklai. Būtent jie geriausiai tinka veiksams su vaizdais atlikti. Taip pat bus naudojami linijų aptikimo algoritmai, miglotoji logika.

1.1.2. Bendras veikos tikslas

Bus kuriama sistema, skirta išsiskiriančių vairavimo atvejų aptikimui vaizdo kamerų medžiagoje.

Šio darbo tikslas – naudojantis dirbtinio intelekto metodais, palengvinti išsiskiriančių vairavimo atvejų aptikimą ir taip sumažinti skaudžių eismo įvykių, kuriuose žūsta ar būna sužalojami žmonės, skaičių.

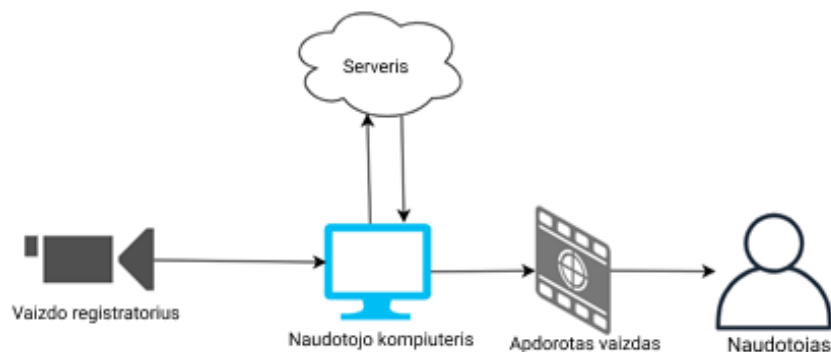
Sistema turėtų padidinti pagaunamų neblaivių vairuotojų skaičių 10 % iki 2025 metų. Taip pat ji turėtų 5 % sumažinti nelaimingų eismo įvykių, kuriuose žūsta ar būna sužalojami žmonės, skaičių.

Taip pat sistema sumažins transporto spūstis, nes ne taip dažnai bus rengiami reidai.

Šis darbas galėtų padėti policijos pareigūnams nuspręsti, kurį automobilį sustabdyti neblaivumo patikrinimui. Paprastiems vairuotojams šis darbas būtų naudingas tuo, kad parodytų, kuris automobilis kelia įtarimą ir kurio atžvilgiu reikėtų elgtis atsargiau.

1.1.3. Sistemos kontekstas

Kuriama sistema bus naudojama kartu su vaizdo registratoriumi. Sistema bus nepriklausoma nuo registratoriaus rūšies. Taip pat sistemai reikės interneto ryšio. Skaičiavimai bus atliekami debesyse. Sistema naudos iš vaizdo registratoriaus gaunamą vaizdą ir jį gražins apdorotą. Detaliau sistemos kontekstas atvaizduojamas 1.1 paveiksle.



1.1 pav. Sistemos konteksto diagrama

1.1.4. Projekto vartotojai ir klientai

Šio produkto vartotojus galima suskirstyti į dvi kategorijas:

- Policijos pareigūnai. Atsakingi už tai, kad bendrajame eisme dalyvaujantys vairuotojai laikytųsi kelių eismo taisyklių, nevairuotų neblaivūs, chuliganiškai. Dažnai rengia prevencines akcijas, patruliuoja, vykdo reidus.
- Paprasti vairuotojai. Dalyvauja bendrajame eisme. Turi būti budrūs, saugotis kitų vairuotojų, kurių veiksmai kartais būna neprognozuojami arba sukelia įtarimą dėl vairuotojo blaivumo. Dalis automobiliuose yra įsirengę vaizdo registratorius.

Policijos pareigūnai susiduria su problemomis, užtikrinant įstatymų laikymąsi kelyje. Nors baudos už kelių eismo taisyklių pažeidimus yra didelės, tačiau dalis vairuotojų jų nepaiso. Prie kiekvieno vairuotojo kelių policininko nepastatysi. Rengiami reidai, siekiant užtikrinti įstatymų laikymąsi, tačiau jų metu pareigūnai sulaukia priekaištų dėl eismo srauto stabdymo. Kuriamas produktas leistų palengvinti policijos pareigūnų darbą. Jie galėtų sustabdyti patikrinimui tik įtarimą sukėlusį vairuotoją. Taip nebeliktų problemos dėl eismo trukdymo. Taip pat sistema padėtų aptikti galimai neblaivų vairuotoją ir nerengiant jokios prevencinės akcijos. Taip policijos pareigūnai dažniau sustabdytų neblaivius vairuotojus ir sumažintų jų skaičių kelyje.

Paprasti vairuotojai taip pat nėra apsaugoti nuo neblaivių / chuliganiškai besielgiančių vairuotojų. Per tokius vairuotojus jie gali patekti į skaudžius incidentus. Jie rizikuoja savo turtu – automobiliu, sveikata. Sistema juos perspėtų apie įtartina vairuotoją ir jie to vairuotojo atžvilgiu elgtųsi atsargiau. Tai padėtų sumažinti nelaimingų atsitikimų skaičių.

Šis produktas būtų skirtas kelių policijos reikmėms. Kelių policija yra atsakinga už kelių eismo taisyklių laikymąsi viešojo naudojimo keliuose. Vienas iš pagrindinių jos tikslų – sumažinti skaudžių eismo įvykių mūsų keliuose skaičių. Didelė dalis šių įvykių įvyksta dėl to, kad vairuotojai vairuoja neblaivūs, elgiasi chuliganiškai. Kelių policija naudoja įvairias priemones, siekdama užtikrinti tvarką keliuose. Jie patruliuoja žymėtais, nežymėtais automobiliais, įrenginėja greičio matuoklius, perka alkotesterius. Tokia priemone galėtų būti ir kuriamas produktas. Kiekvienas policijos automobilis turi įrengtą vaizdo registratorių. Į tą registratorių būtų galima įdiegti ir kuriamą produktą. Kadangi policija yra viešoji įstaiga, tai produktą jie galėtų nusipirkti per viešuosius pirkimus.

Produktą galėtų naudoti ir paprasti vairuotojai. Jis jiems padėtų atkreipti dėmesį į pavojingus vairuotojus. Jie produktą galėtų nusipirkti kartu su vaizdo registratoriumi arba jį įsidiesti į jau turimą.

1.1.5. Rinkos tyrimas

Kiekvienais metais Lietuvoje nustatoma beveik 10 000 neblaivaus vairavimo atvejų [4]. Tačiau dalis vairuotojų būna nesulaikomi. Todėl skaičiai galėtų būti dar didesni. Lietuvoje yra apie 1,5 milijono važiuojančių automobilių [5], dalis jų turi vaizdo registratorius. Taip pat kelių policija pastoviai investuoja į priemones, padedančias pagerinti eismo saugumą [6].

1.1.6. Konkurencija ir alternatyvos

Konkurentais galima vadinti produktus, sistemas, kurie turi tą patį tikslą – pašalinti neblaivų vairuotoją iš eismo, įvertinti vairuotojo vairavimą. Tokie konkurentai būtų:

- Alkotesteris. Šio prietaiso privalumas yra tai, kad jis tiksliai pasako ar vairuotojas yra blaivus ar ne. Jį naudoja policijos pareigūnai. Tačiau šis prietaisas turi ir kelis trūkumus. Vienas iš jų yra tai, kad geri alkotesteriai yra brangūs (kai kurie kainuoja ir 1000 eurų). Yra ir pigesnių opcijų, tačiau jomis negalima akiai pasitikėti. Taip pat, norint alkotesteriu patikrinti vairuotoją, reikia jį sustabdyti. Tai užima laiko, todėl negalima patikrinti labai didelio skaičiaus vairuotojų.
- Alkoblokas. Šis prietaisas įrengiamas automobilyje. Jo privalumas yra tai, kad jis neleidžia neblaiviam vairuotojui užvesti automobilio. Tačiau šis prietaisas turi daug trūkumų. Vienas iš jų yra tai, kad jo įrengimas yra brangus. Taip pat šį prietaisą galima apgauti, jei šalia yra blaivus asmuo. Dar galima paminėti, kad įrengti šį prietaisą visuose automobiliuose kainuotų daug pinigų.
- Automobilyje įdiegta sistema, naudojanti kameras ir sensorius, siekdama aptikti girtą vairuotoją (pavyzdžiui, Volvo XC90 automobilyje). Ši sistema naudojasi dirbtinio intelekto metodais ir sensoriais ir neleidžia vairuoti neblaiviam vairuotojui. Ji skenuoja vairuotojo veidą ir vairavimo manieras. Šios sistemos privalumas yra tai, kad jos neapgausi taip kaip alkobloko ir ji tikrai neleis vairuoti neblaiviam vairuotojui. Tačiau ši sistema yra įdiegta tik naujuose, prabanguose automobiliuose. Dažniausiai asmenys, kurie vairuoja neblaivūs, tokių automobilių neturi.
- „Netradyne Driveri“ [7], tai sistema skirta pagerinti vairuotojo vairavimą. Ji taip pat naudoja automobilių atpažinimo, linijos sekimo, kelio ženklų aptikimo algoritmus. Tačiau ši sistema yra orientuota į šią sistemą naudojančio automobilio vairavimą, o ne į aplinkinių vairuotojų. Ji vertina vairuotojo elgesį balais, duoda patarimus, kurie padeda pagerinti vairavimą.

Kuriamos sistemos esminis pranašumas prieš anksčiau išvardintus yra tai, kad ji leistų greitai patikrinti daugybę vairuotojų. Sistemai identifikavus įtartina vairuotoją, jį galima būtų patikrinti jau su alkotesteriu, kuris patvirtintų sistemos rezultatą arba ne. Taip pat ji būtų pigesnė už visus anksčiau išvardintus konkurentus. Plačiau palyginimo rezultatai apžvelgiami 1.1 lentelėje.

1.1 lentelė Konkurentų apžvalga

Sistema	Alkotesteris	Alkoblokas	Automobilyje įdiegta sistema	Netradyne Driveri	Kuriama sistema
Tikslumas	100%	100%	50%	50%	50%
Kaina	Vidutinė	Didelė	Didelė	Didelė	Vidutinė
Sugaištas laikas	Daug	Mažai	Mažai	Mažai	Mažai
Galimybė vairuoti	Yra	Yra, labai nedidelė	Nėra	Yra	Yra

1.1.7. Projekto įgyvendinimo planai ir kokybės vertinimas

Projekto veiklos ir rezultatai atitinka magistro projekto įgyvendinimo akademinį grafiką.

Projekto veiklos:

- projekto analizė (iki 2019-11-15);
- projektavimo planas (iki 2019-12-20);
- reikalavimų specifikavimas (iki 2020-03-10);
- architektūros specifikavimas (iki 2020-04-15);
- projekto realizavimas (iki 2020-05-17);
- projekto testavimas (iki 2020-11-15);
- dokumentacija (iki 2020-12-05);
- ataskaita (iki 2020-01-15).

Produkto kokybė bus vertinama pagal 1.2 lentelėje pateiktus kriterijus.

1.2 lentelė Sistemos vertinimo kriterijai

Nr.	Kriterijus	Pagrindimas
1	Tikslumas	Koks įtarimų, kad vairavimo atvejis yra išsiskiriantis pasitvirtinimo procentas?
2	Kaina	Kiek kainuoja įdiegti sistemą?
3	Sugaištas laikas	Kiek laiko užtrunka išsiskiriančio vairavimo atvejo identifikavimas ir kaip tai paveikia eismo intensyvumą?
4	Galimybė vairuoti	Ar sistema, užfiksavusi išsiskiriantį vairavimo atvejį, padeda pašalinti vairuotoją iš eismo?

Projektą įgyvendinantis personalas:

- Prof. Rytis Maskeliūnas – projekto vadovas. Atsakingas už projekto valdymą.
- Paulius Melaika – studentas, programuotojas. Atsakingas už projekto realizavimą.

Projektui reikalinga įranga:

- 3 skirtingi vaizdo registratoriai.

Projekto biudžeto skaičiavimas pateiktas 1.3 lentelėje.

1.3 lentelė Projekto biudžetas

Išlaidos	Vienetas	Vienetų skaičius	Vieneto kaina, Eur	Viso, Eur
1. Žmonių ištekliai				
Projekto vadovas	Darbo diena	10	100	1000
Programuotojas	Darbo diena	100	80	8000
<i>Iš viso žmonių išteklių</i>				9000
2. Įranga ir prekės				
Vaizdo registratorius	Vienetas	3	100	300
<i>Iš viso įranga ir prekės</i>				300
Viso tiesioginiai projekto kaštai				9300

1.1.8. Bendri apribojimai

Sistema laikysis BDAR įstatymo reikalavimų, neskanuos automobilių numerių, jų nesaugos. Sistema rezultata turėtų gražinti greičiau nei per 0,5 sekundės, kitu atveju įtartinas automobilis jau gali būti dingęs iš akiračio. Esant sistemos sutrikimams ar nesant interneto ryšiui, sistema apie nesklandumus turėtų informuoti naudotoją.

1.2. Projektavimo metodologijos ir technologijų analizė

1.2.1. Egzistuojantys pritaikymai

Kadangi ši tyrimų sritis yra svarbi siekiant pagerinti situaciją keliuose, joje yra atlikta daug tyrimų. Mokslininkai atlieka tyrimus įvairiomis kryptimis.

Viena iš pagrindinių užduočių yra aptikti vaizdo medžiagoje esantį automobilį. Jį aptikus, galima atlikti skirtingus tyrimus, susijusius su to automobilio padėtimi, trajektorija, greičiu.

Šiame skyriuje bus apžvelgtos populiariausios metodų pritaikymo sritys:

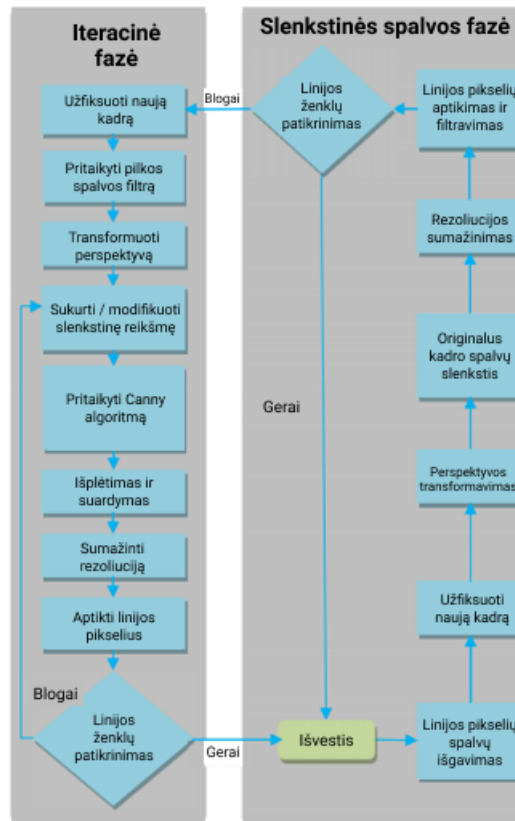
1. eismo juostos aptikimas;
2. automobilio sekimas pagal šviesas;
3. automobilio numerio nuskaitymas;
4. automobilio greičio nustatymas;
5. susidūrimų prognozavimas;
6. posūkio signalų rodymo sekimas;
7. trajektorijų klasterizavimas.

1.2.2. Eismo juostos aptikimas

Eismo juostoms aptikti naudojamas „Sumažintos raiškos“ linijos aptikimo algoritmas (R^2 algoritmas). Šis metodas susideda iš dviejų fazių [8]:

- iteracinės fazės;
- slenkstinės spalvos fazės.

Iteracinės fazės metu yra aptinkamos eismo juostos žymės. Jos aptinkamos iteruojant per vaizdo apdorojimo slenkstines reikšmes. Ši fazė aptinka eismo juostas, tačiau lėtai atlieka skaičiavimus. Slenkstinės spalvos fazė skirta pašalinio triukšmo išfiltravimui iš vaizdo. Šios fazės metu yra išryškinamos linijos. Ši fazė nėra tokia tiksli jei kelio ir eismo juostų spalvos yra panašios. Tačiau šios fazės skaičiavimai atliekami greičiau. Programa dažniausiai naudoja slenkstinės spalvos fazę, tačiau jei atsiranda staigūs spalvos pokyčiai yra pereinama į iteracinę fazę. Detalesnis veikimo algoritmas paaiškintas paveikslėlyje (1.2 pav.) [8]:



1.2 pav. Eismo juostų aptikimo algoritmas

1.2.3. Automobilio sekimas pagal šviesas

Tamsiu paros metu ar esant prastam matomumui gali būti sunku nustatyti automobilio kontūrą. Tokiais atvejais naudojamas automobilio sekimas pagal jo šviesų padėtį. Naudojamas metodas, paremtas gradientu. Metode pasitelkiamas neuroninis tinklas su atgaline sklaida. Būtent atgalinės sklaidos metodas yra vienas iš geriausiai tinkančių šablonų aptikimo metodų [9]. Sekimo pagal šviesas algoritmas gali būti išskaidomas į tris skirtingas fazes [10]:

1. Ypatybių atrinkimas ir apdirbimas:
 - a. automobilio šviesos yra izoliuojamos;
 - b. paryškunami automobilio šviesų regionai nuotraukoje;
 - c. izoliuojama kiekviena šviesa atskirai;
 - d. dvejetainė nuotrauka filtruojama siekiant sukurti kintamą šviesos gradientą.
2. Optinio srauto apskaičiavimas – iteracinis metodas, skirtas nustatyti automobilio trajektoriją.
3. Trajektorijos klasifikavimas – nustatomas automobilio judėjimas erdvėje.

1.2.4. Automobilio numerio nuskaitymas

Kartais užduotis yra nuskaityti automobilio numerį. Pritaikymas gali būti įvairus. Numeris gali būti reikalingas užfiksuoti automobilio įvažiavimo / išvažiavimo laiką, nubausti už greičio viršijimą ir panašiai.

Vienas iš metodų aptikti automobilio valstybinį numerį yra naudojantis SVM. Pirmiausia kiekviena sritis identifikuojama ar ji gali būti valstybinis numeris, ar ne. Galiausiai nustatoma valstybinio numerio lokacija [11].

1.2.5. Automobilio greičio nustatymas

Siekiant identifikuoti vairuotojo elgesį kelyje, reikia žinoti jo greitį. Neatsakingi vairuotojai dažnai jį viršija. Tam naudojamas automobilio sekimas laiko atžvilgiu. Automobilio padėtis kinta laike pagal kitus kelio objektus (stulpelius, linijas, medžius). Kai automobilis yra aptinkamas, jau galima sekti jo judėjimą laiko ir kitų objektų atžvilgiu [12]. Vienas iš būdų aptikti automobilio greitį yra šoninis konvoliucinis neuroninis tinklas. Šis tinklas naudojamas siekiant sujungti ryškius bruožus iš ankstesnių sluoksnių, siekiant gauti smulkią lokalizaciją [13].

Šiuo metu vienas iš geriausių greitaveiką turinčių ir lengviausiai apmokomų metodų yra „Connect-and-Merge“ konvoliucinis neuroninis tinklas [14].

1.2.6. Susidūrimų prognozavimas

Taip pat dirbtinio intelekto metodai naudojami ir siekiant prognozuoti būsimus susidūrimus. Tai padeda išvengti eismo įvykių. Tokius algoritmus naudoja ir autonominiai automobiliai. Šie metodai padeda įspėti vairuotoją ir taip yra sumažinama eismo įvykio tikimybė.

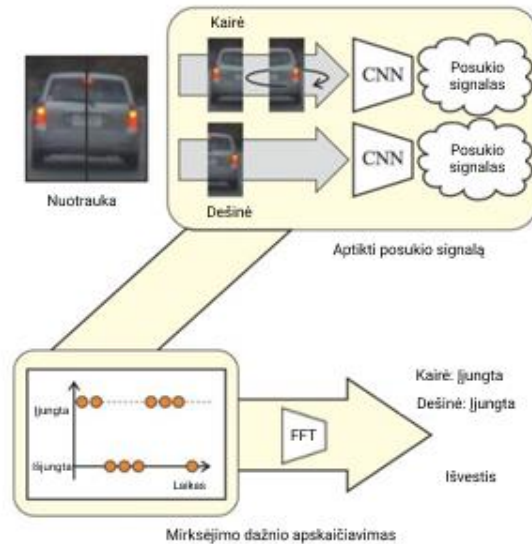
Susidūrimams prognozuoti naudojamas toks metodas [15]:

1. Vaizde atpažįstami automobiliai ir jie suklasifikuojami naudojanti YOLO strategija. Šios strategijos naujausia versija aprašyta [16]. Šios strategijos pagrindas yra konvoliucinis neuroninis tinklas, kuris yra pagrįstas vienu praėjimu, kuris sukuria SxS stačiakampių tinklą, kuriame kiekvienas stačiakampis turi savo regresijos ir klasifikacijos algoritmą. Galiausiai iš kelių gretimų stačiakampių išrenkamas stipriausias.
2. Sekamas kiekvienas automobilis. Naudojami sekikliai su koreliacijos filtro algoritmu.
3. Kiekvienam automobiliui pritaikomas ViF aprašas su SVM ir taip yra prognozuojami automobilių susidūrimai.

1.2.7. Posūkio signalų rodymo sekimas

Dar vienas konvoliucinio neuroninio tinklo pritaikymas šioje srityje skirtas aptikti ar automobilis rodo posūkį. Tai padeda prognozuoti kitų automobilių judėjimo kryptį. Taip pat tai gali padėti aptikti neatsakingus vairuotojus kelyje. Šio metodo veikimo principas aprašytas (1.3 pav.) [17]:

1. Aptinkamas automobilis.
2. Naudojantis CNN, sukuriamas klasifikatorius, kuris klasifikuoja, kuris automobilio posūkio signalas yra įjungtas. Siekiant sumažinti sudėtingumą, aptiktas automobilis perskeliamas į dvi dalis: kairę ir dešinę. Kairioji pusė yra apsukama, taip abejoms pusėms tinka tas pats klasifikatorius. Išskiriamos 4 fazės:
 - a. kairysis posūkis įjungtas;
 - b. dešinysis posūkis įjungtas;
 - c. įjungti abu posūkliai;
 - d. išjungti abu posūkliai.
3. CNN klasifikatorius geba atskirti ar posūkio signalas yra įjungtas ar ne. Mirksėjimo dažnumas apskaičiuojamas naudojantis „Fast Fourier Transform“ metodu. Kai mirksėjimas yra tam tikro dažnio, sistema supranta, kad automobilio posūkis yra įjungtas.



1.3 pav. Posūkių signalų sekimo algoritmas

1.2.8. Trajektorijų klasterizavimas

Trajektorijos nustatymas yra svarbus uždavinys, siekiant aptikti objektų bruožus. Vieną trajektoriją galima palyginti su kitomis ir iš to padaryti išvadas. Siekiant tai atlikti, pirmiausia reikia sugrupuoti skirtingas trajektorijas pagal bruožus. Automobilių sekimo atžvilgiu, trajektorija parodo vairavimo savybes, nuoseklumą.

Trajektorijos nustatomos naudojant CNN ir atliekant automobilių sekimą laiko atžvilgiu. Trajektorijos klasterizavimo metodas (1.4 pav.) susideda iš šių žingsnių:

- turimos trajektorijos paruošimo naudojimui;
- spektrinio klasterizavimo;
- hierarchinio klasterizavimo.

Kadangi trajektorijos ne visada yra nuoseklios, gali turėti vingių, kampų, gali išeiti iš vaizdo ribų, neužpildyti viso vaizdo, tai gali apsunkinti skaičiavimus. Dėl šių priežasčių reikia turimas trajektorijas paruošti naudojimui. Pirmiausia labai artimi trajektorijos taškai yra sujungiami į vieną tašką. Toliau trajektorijos yra pratęsimos iki tam tikrų vaizdo ribų. Šis veiksmas atliekamas pagal automobilio judėjimo kryptį ir greitį. Galiausiai trajektorijos taškai išdėstomi taip, kad būtų vienodai nutolę vienas nuo kito. Tai atlikę ir pritaikę linijinę interpoliaciją, mes galime atkurti pradinę trajektoriją. Ją atkūrę mes jau galime palyginti dvi trajektorijas. Trajektorijų panašumą atspindi vidutinis atstumas tarp dviejų jų taškų. Taip yra suformuojama panašumų matrica.

Spektrinis klasterizavimas atlieka veiksmus su šia matrica. Jis vyksta tokiu principu:

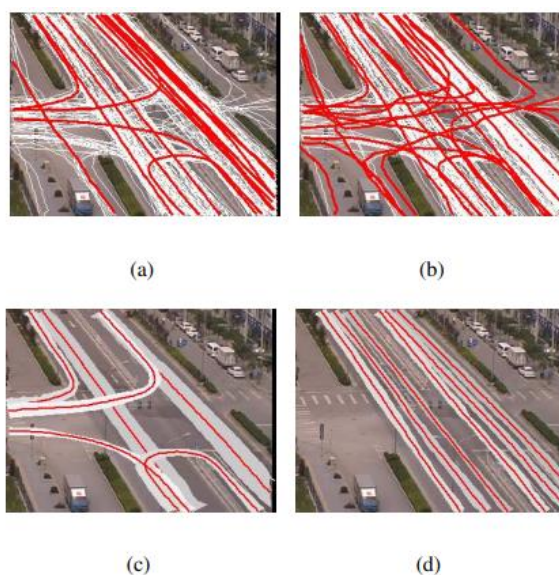
Tarkime turime duomenų rinkinį $X = x_1, x_2, \dots, x_n$. Mes jį galime padalinti į k dalių atlikdami šiuos veiksmus:

1. Apskaičiuojame duomenų rinkinio X panašumo matricą A , kur $A_{ij} = \exp(-dist(x_i, x_j)/2\sigma^2)$.
2. Sukuriame matricą $L = D^{-1/2}AD^{-1/2}$, kur D yra įstrižinė matrica, kurios i -tasis įstrižainės elementas yra A matricos i -tosios eilės suma.

3. L matricai pritaikome tikrinės vertės suskaidymą ir randame jos k didžiausių tikrinių verčių q_1, q_2, \dots, q_k .
4. Suformuojame naują matricą $Q = [q_1, q_2, \dots, q_k]$, sudėdami k tikrines vertes į stulpelius, ir kiekvieną eilutę normalizuojame iki vienetų.
5. Kiekvieną Q vektorių eilę klasterizuojame į K klasterius. Klasterizavimas atliekamas naudojantis spektrinio klasterizavimo metodu. Kiekviena eilė yra laikoma naujos savybės vektoriumi.

σ parametras reikalingas norint valdyti skirtumo didėjimą, kai didėja atstumai.

Hierarchinis klasterizavimas yra taikomas todėl, kad atlikus tik vieno lygio klasterizavimą vis tiek lieka daug skirtingų trajektorijų. Šias gautas trajektorijas klasterizuojame antrą kartą. Taip gauname norimas trajektorijas. Trajektorija, kuri yra mažiausiai nutolusi nuo kitų trajektorijų, parenkama kaip pagrindinė [18].



1.4 pav. Trajektorijų klasterizavimo pavyzdys

1.2.9. Planuojami naudoti skaičiavimo metodai

Šiame skyriuje bus išvardinti planuojami pritaikyti dirbtinio intelekto metodai, skirti vaizdams su automobiliais atpažinti ir apdoroti.

Metodų sąrašas:

- konvoliucinis neuroninis tinklas;
- miglotoji logika;
- eismo juostų aptikimas.

1.2.10. Konvoliucinis neuroninis tinklas

Šis metodas yra vienas labiausiai paplitusių, kai reikia atpažinti, klasifikuoti ir lokalizuoti nuotraukose, vaizdo medžiagoje esančius objektus, šiuo atveju – automobilius. Automobilių atpažinimas yra svarbiausia šios tiriamosios srities dalis. Viskas prasideda nuo atpažinimo, klasifikavimo, lokalizacijos. Paskui galima atlikti reikiamus skaičiavimus – greičio, padėties, trajektorijos ir t. t.

Konvoliucinio neuroninio tinklo architektūra yra įkvėpta žinduolių regos veikimo principo [19]. Konvoliucinis neuroninis tinklas susideda iš:

- konvoliucinio sluoksnio;
- „Pooling“ sluoksnio;
- „Fully-connected“ sluoksnio.

Konvoliucinis sluoksnis yra esminė konvoliucinio neuroninio tinklo dalis. Neuronai jame yra sujungti jungtimis, kurios turi bruožų svorius. Konvoliucinis sluoksnis susideda iš savybių žemėlapių. Kiekvienas neuronas tų pačių savybių žemėlapyje yra atsakingas už savybių atradimą prieš tai buvusiam sluoksnyje. Siekiant išgauti naujas savybes, įvesties savybių žemėlapiai yra konvoluojami ir jiems pritaikoma aktyvacijos funkcija. Labiausiai paplitusios aktyvacijos funkcijos yra sigmodinė, tanh ir Relu [20].

„Pooling“ sluoksnis yra atsakingas už antrinių savybių aptikimą. Jis padeda sumažinti savybių žemėlapių matmenis ir taip padidina savybių aptikimo galimybę. Dažniausiai „Pooling“ sluoksnis yra tarp dviejų konvoliucinių sluoksnių. Sujungdami kelis „Pooling“ ir konvoliucinius sluoksnius galime išgauti geriausias tiriamo objekto charakteristikas.

„Fully-connected“ sluoksnis – kiekvienas prieš tai buvusio sluoksnio neuronas yra sujungiamas su kiekvienu esamo sluoksnio neuronu. Šiame sluoksnyje nesaugoma jokia erdvinė informacija. Po paskutinio „Fully-connected“ sluoksnio seka išvesties sluoksnis. Klasifikavimo uždaviniams atlikti dažniausiai naudojama „Softmax“ regresija. Ji labiausiai tinka išvesčių tikimybėms nustatyti [21].

1.2.11. Miglotoji logika

Siekiant aptikti išsiskiriančius vairavimo atvejus reikia vairavimą įvertinti pagal tam tikrus parametrus. Vairavimo vaizdo įrašai gali būti patys įvairiausi. Vieni gali būti prastos kokybės, kiti aukštos, taip pat gali skirtis filmavimo kampas, atstumas, oro sąlygos. Dėl skirtingų vaizdo įrašų parametrų vertės toje pačioje situacijoje gali skirtis.

Miglotoji logika gali padėti išvengti priklausomybės nuo vaizdo įrašų savybių, skirtingų vairavimo atvejo parametrų. Miglotoji logika suteikia didelį lankstumą priimant sprendimus ir leidžia išvengti netikslumų ir neapibrėžtumų [22]. Vienas iš miglotosios logikos privalumų – taisyklės rašomos žmonių kalba ir taip yra formalizuojamas samprotavimas.

Miglotosios logikos architektūra:

1. Taisyklių bazė – taisyklių rinkinys su sąlygomis, reikalingas sprendimų priėmimui.
2. „Fuzifikatorius“ – skirtas paversti įvestis į miglotosios logikos rinkinius.
3. Išvadų priėmimo logika – paskaičiuoja kiekvienos įvesties atitikimą taisyklėms ir nusprendžia kurios taisyklės turi būti pritaikomos.
4. „Defuzifikatorius“ – skirtas paversti iš išvadų priėmimo logikos gautus miglotosios logikos rinkinius į konkrečią reikšmę.

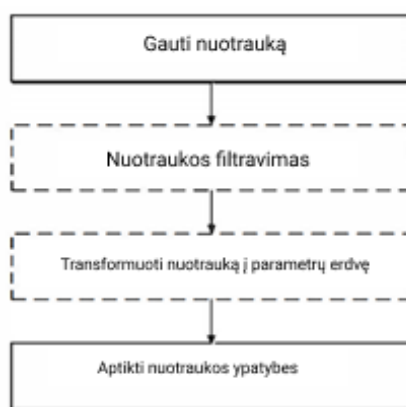
1.2.12. Linijos sekimo algoritmas

Viena iš svarbiausių užduočių šioje srityje – automobilio padėties, kitų kelio objektų atžvilgiu, nustatymas. Vienas iš kelio objektų yra linija. Linijos yra didžiojoje dalyje kelių. Vienur jos būna tik per vidurį, kitur ir iš šonų. Taip pat linijos gali būti ištisos ar su pertraukimais. Dar linijos gali nutrūkti,

būti nublukusios, neryškios. Taip pat asfaltas gali būti įvairių spalvų, būklės. Šios savybės linijos atpažinimą padaro sunkiu uždaviniu.

Vienas iš metodų atpažinti kelio liniją yra Hough transformacija. Šis metodas yra skirtas linijų aptikimui nuotraukoje. Šis metodas yra paremtas objektų atpažinimu pagal jų ribinius taškus. Kelio juostos atpažinimo metodo principas (1.5 pav.) [23]:

1. Kintamųjų erdvė yra kiekybiškai įvertinama intervalu nuo ρ iki θ .
2. Suformuojamas masyvas su pradinėmis nulinėmis reikšmėmis.
3. Kiekvienas taškas nuotraukoje sugrupuojamas taip, kad jo vertė viršytų slenkstinę reikšmę. Šie taškai yra padidinami ir išsaugomi masyve.
4. Masyve esantys lokalūs maksimumai atspindi nuotraukos linijoje esančius taškus. Iš šių taškų matricos galima nustatyti nuotraukoje esančias linijas.



1.5 pav. Kelio juostos atpažinimo metodo principas

1.2.13. Planuojamos taikyti technologijos

Šį projektą planuojama įgyvendinti pasitelkus „Python“ programavimo kalbą ir „Tensorflow“ biblioteką.

„Python“ kalba pasirinkta todėl, kad ji yra plačiai paplitusi mašininio mokymosi pritaikymo uždaviniuose. Ši kalba turi begalę bibliotekų („Scikit-learn“, „pandas“, „Tensorflow“, „Keras“ ir t.t.), kurios palengvina skaičiavimus. Taip pat su ja patogiu dirbti, kai yra dideli duomenų kiekiai. Dar vienas šios kalbos privalumas yra plati bendruomenė. Taip pat ši kalba yra mėgstama pradedančiųjų programuotojų, kadangi ją gana paprasta pramokti, ji yra lengvai skaitoma. Nors „Python“ yra gana lėtai veikianti programavimo kalba [24], jos privalumai atsveria trūkumus.

Mašininio mokymosi metodams realizuoti bus naudojama „Tensorflow“ biblioteka. Ši biblioteka yra labai galinga, kai reikia atlikti veiksmus su matricomis. Ji padeda lengvai išspręsti mašininio mokymosi problemas. Taip pat ji yra plačiai paplitusi tarp šios srities specialistų. Taip pat ši biblioteka yra lanksti ir skaičiavimai gali būti atliekami įvairiuose įrenginiuose su minimaliais kodo pakeitimais.

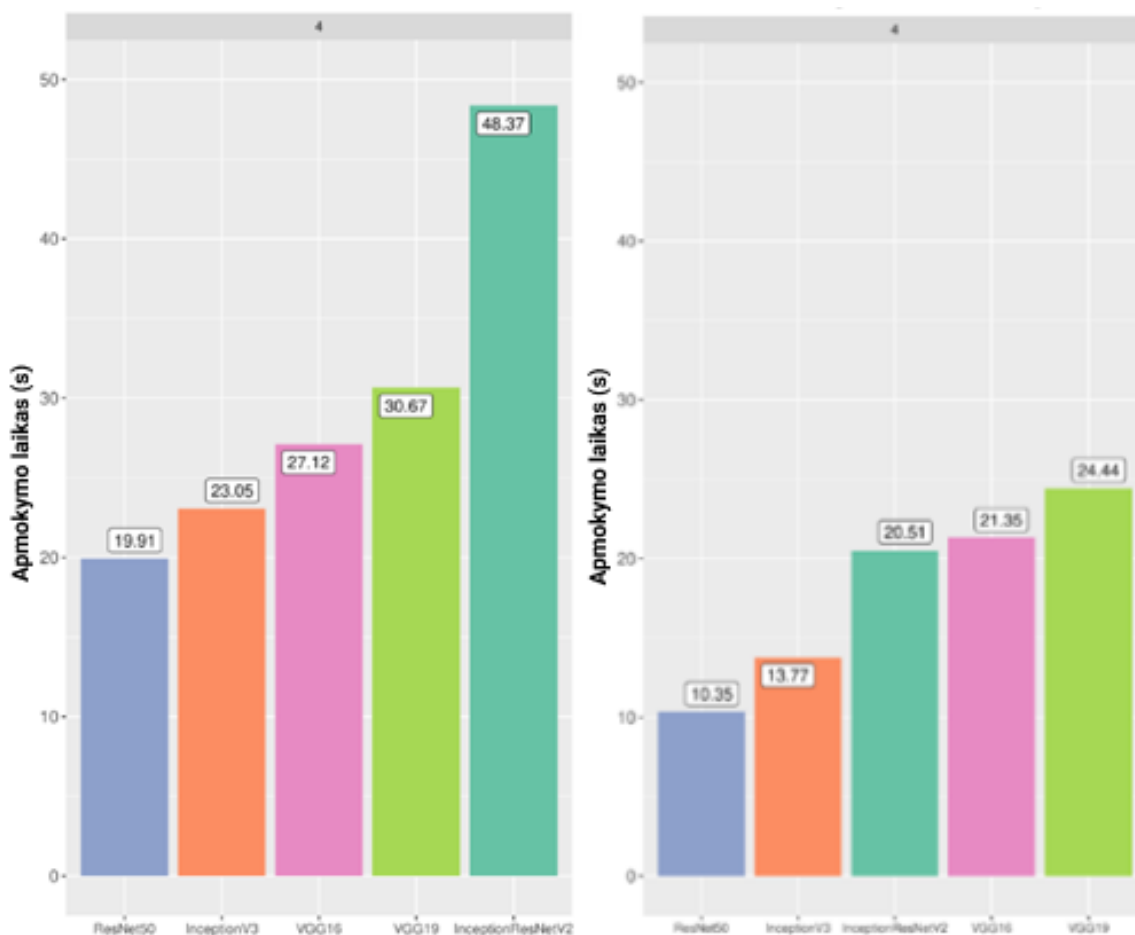
1.2.14. Atrinktų technologijų pagrindimas

Kaip jau aprašyta anksčiau, pasirinkta „Python“ programavimo kalba. Nors ši kalba yra prastesnė greitaveikos atžvilgiu lyginant su C šeimos kalbomis [25], tačiau jos turimas platus bibliotekų pasirinkimas, plati bendruomenė, lengvas skaitomumas, galimybė pritaikyti įvairių problemų sprendimui, plataus platformų diapazono palaikymas [26] lėmė jos pasirinkimą.

Mašininio mokymosi metodams realizuoti planuojama naudoti „Tensorflow“ biblioteką. Į šią biblioteką panaši „Keras“ biblioteka. Abi šios bibliotekos turi visą bazinį, mašininiam mokymuisi reikalingą funkcionalumą. Jos abi turi pakankamai išsamiai parašytą dokumentaciją. „Tensorflow“ biblioteka yra paremta C++ kalba, o „Keras“ biblioteka gali naudotis „Tensorflow“ kaip *back-end* dalimi [27]. Kadangi, reikalui esant, bus galima į „Tensorflow“ integruoti „Keras“ biblioteką, buvo nuspręsta pasirinkti būtent „Tensorflow“. Taip pat „Tensorflow“ biblioteka yra geresnė greیتaveikos atžvilgiu (1.4 lentelė, 1.6 pav.) [28]:

1.4 lentelė „Tensorflow“ ir „Keras“ bibliotekų palyginimas

Biblioteka	„Keras“	„Tensorflow“
<i>ResNet50</i>	19,91 s	10,35 s
<i>InceptionV3</i>	23,05 s	13,77 s
<i>VGG16</i>	27,12 s	20,51 s
<i>VGG19</i>	30,67 s	21,35 s
<i>InceptionResNetV2</i>	48,37 s	24,44 s



1.6 pav. „Keras“ (kairėje) ir „Tensorflow“ (dešinėje) bibliotekų greیتaveika

1.2.15. Atrinktų matematinių metodų pasirinkimo pagrindimas

Automobiliams atpažinti, klasifikuoti, nustatyti jų trajektorijoms buvo pasirinktas konvoliucinis neuroninis tinklas. Šis metodas pasirinktas, nes jis yra plačiai taikomas užduotims susijusioms su

vaizdų apdorojimu. Konvoliuciniai neuroniniai tinklai demonstruoja gerą greitaveiką ir aukštą tikslumą [29]. Taip pat pasirinkta „Tensorflow“ biblioteka implementuoja šį metodą.

Linijoms kadruose aptikti bus naudojamas Hough transformacijos algoritmas. Šis algoritmas pasirinktas, nes yra įgyvendintas „Python“ kalbos bibliotekose.

Išskirtinių vairavimo atvejų identifikavimui bus naudojama miglotoji logika. Kaip jau minėta, vaizdo įrašai gali būti patys įvairiausi, vairavimo parametrų reikšmės gali būti skirtingos, priklausomos nuo vaizdo įrašo. Miglotoji logika leis apibendrinti skirtingus parametrus ir pagal juos nustatyti ar vairavimo atvejis yra išskirtinis, išvengiant netikslumų ir neapibrėžtumų.

1.2.16. Įgyvendinimo problemos

Realizuojant šią sistemą bus susiduriama su begale sunkumų.

Vienas iš jų – duomenų rinkinių apmokymui surinkimas. Norint tai atlikti, reikės išsitraukti vaizdo įrašus iš įvairių vaizdo įrašų platformų. Iš čia kyla ir antras sunkumas – vaizdo įrašai bus optimizuoti, jų kokybė nebus pati geriausia.

Taip pat gali kilti sunkumų ir su pačių vaizdų turiniu. Nelygūs keliai, duobės gali iškraipyti automobilio trajektoriją [30]. Taip pat eismo juostos gali būti blyškios, netinkamai nupieštos, nusitrynusios.

Taip pat gali iškilti sunkumų su skaičiavimų greitaveika. Šiai problemai spręsti veikiausiai bus naudojama „OpenGL“ biblioteka.

Taip pat niekaip nebus galima 100 % nustatyti ar kaip išsiskiriantis identifikuotas vairavimo atvejis iš tikrųjų toks buvo.

2. Projektinė dalis

2.1. Reikalavimų specifikavimas

2.1.1. Projekto kūrimo pagrindas

Vairuoti automobilį chuliganiškai ar esant neblaiviam stovyje draudžia įstatymai. Neblaivūs, chuliganiškai besielgiantys vairuotojai sukelia begalę autoįvykių, kurie baigiasi mirtimis ar sunkiais sužalojimais. Neblaivus asmuo neturi reikiamos koncentracijos, reakcijos, todėl yra nepajėgus vairuoti. Nors bausmės už chuliganišką vairavimą ar vairavimą esant neblaiviam yra didelės, vis tiek atsiranda tai darančių žmonių. Vairuotojus, kurie pažeidžia šiuos įstatymus, baudžia policijos pareigūnai. Nors dažnai yra rengiami reidai, neatsakingų vairuotojų problema išlieka. Siekiant padėti pareigūnams ir sumažinti neblaivių / chuliganiškai besielgiančių vairuotojų skaičių, nuspręsta sukurti šią sistemą, skirtą išsiskiriančių vairavimo atvejų aptikimui vaizdo medžiagoje.

2.1.2. Sistemos tikslai

Sistemai keliami tikslai:

- Padėti pareigūnams nustatyti neblaivius ar chuliganiškai besielgiančius vairuotojus. Sistema palengvintų pareigūnų darbą, nes leistų įvertinti daugiau automobilių per tą patį laiką. Įdiegus sistemą turėtų būti patikrinama du kartus daugiau vairuotojų.
- Sumažinti neblaivių vairuotojų skaičių. Sistema padės identifikuoti neblaivų vartotoją kelyje. Įdiegus sistemą pagautų neblaivių vairuotojų skaičius turėtų padidėti 20 %.
- Sumažinti avarių, kilusių dėl neblaivių vairuotojų kaltės, skaičių. Sistema padės nubausti neblaivius vairuotojus ir sustabdys jų dalyvavimą eisme. Įdiegus sistemą įvykių, įvykusių dėl neblaivių vairuotojų kaltės, skaičius turėtų sumažėti 40 %.

2.1.3. Užsakovai, pirkėjai ir kiti sistema suinteresuoti asmenys

Užsakovas: Rytis Maskeliūnas, KTU profesorius. Priims projektą, teiks reikalavimus, patars, seks progresą.

Pirkėjai: policijos departamentas. Sistema galės būti parduodama policijos departamentui. Jie sistemą diegtų į savo automobilius, naudodami juose esančius vaizdo registratorius. Galės teikti pasiūlymus kaip tobulinti galutinį produktą.

Kiti asmenys: Paulius Melaika. Projektuos sistemą, rinks reikalavimus, ją įgyvendins, testuos ir diegs.

2.1.4. Vartotojai

Sistemos vartotojai skirstomi į dvi kategorijas:

Policijos pareigūnai:

- Vartotojo kategorija: naudotojas.
- Vartotojo sprendžiami uždaviniai: patruliavimas keliuose, neblaivių / chuliganiškai besielgiančių vairuotojų paieška, identifikavimas, pašalinimas iš eismo, nubaudimas.
- Patirtis dalykinėje srityje: patyręs.
- Patirtis informacinėje technologijoje: vidutiniokas.
- Papildomos vartotojo charakteristikos: dirba stresinėse situacijose.
- Vartotojų prioritetai: aukščiausias.

Paprasti vairotojai:

- Vartotojo kategorija: naudotojas.
- Vartotojo sprendžiami uždaviniai: dalyvauja eisme, siekia pasiekti kelionės tikslą saugiai.
- Patirtis dalykinėje srityje: maža.
- Patirtis informacinėse technologijose: vidutinė.
- Papildomos vartotojo charakteristikos: labai plati visuomenės grupė.
- Vartotojų prioritetai: žemas.

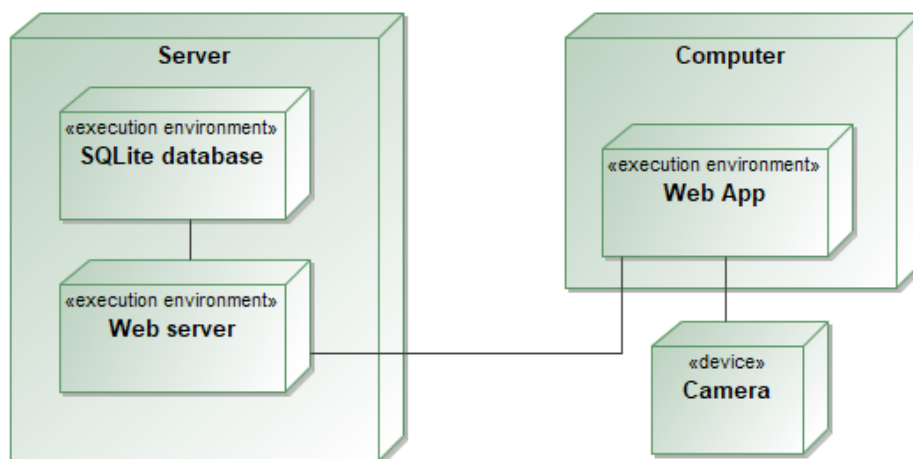
2.1.5. Apribojimai sprendimui

Sistemos serverinė dalis turi būti realizuota „Python“ kalba. Saityno aplikacija turi būti sukurta naudojantis „Vue.js“ karkasu. Šios technologijos yra plačiai paplitusios ir turi didelę bendruomenę, yra reguliariai atnaujinamos.

Darbams su vaizdų apdorojimu turi būti naudojama „Tensorflow“ biblioteka. Ši biblioteka taip pat yra plačiai naudojama, turi išsamia dokumentaciją.

2.1.6. Diegimo aplinka

Diegimo aplinka vaizduojama 2.1 paveiksle.



2.1 pav. Diegimo aplinka

Serveris:

- *Python Venv*, *Python 3.7* arba aukštesnė versija;
- *SQLite 3* versija;
- *NVIDIA RTX2060* vaizdo plokštė.

Kliento kompiuteris:

- saityno naršyklė;
- interneto ryšys.

Kamera (prijungta prie kompiuterio).

2.1.7. Bendradarbiaujančios sistemos

Bendradarbiaujančių sistemų nenustatyta.

2.1.8. Komerciniai specializuoti programų paketai

Komercinių specializuotų paketų nenustatyta.

2.1.9. Numatoma darbo vietos aplinka

Sistema bus naudojama automobilyje, todėl sistema turi būti pasiekama iš bet kurios lokacijos.

Sistema bus naudojama automobilyje, todėl ši sistema neturėtų blaškyti vairavimo.

Sistema galės būti naudojama bet kuriuo paros metu, taip pat ir naktį, neapšviestose vietose. Todėl sistema turi prisitaikyti prie paros meto - tekstai turi būti įskaitomi dieną, sistema neturi būti ryški nakties metu.

2.1.10. Sistemos kūrimo terminai

Sistema turi būti užbaigta iki baigiamojo magistro darbo gynimo datos. Jei bus nesilaikoma šio termino sistemos įgyvendinimas nusikels dar metams.

2.1.11. Sistemos kūrimo biudžetas

Kūrimo metu techninei įrangai, skirtai testavimui, pirkti negali būti išleista daugiau nei 300 eurų.

Daugiau sistemos kūrimo biudžeto apribojimų nenumatyta.

2.1.12. Svarbūs faktai ir prielaidos

Faktai:

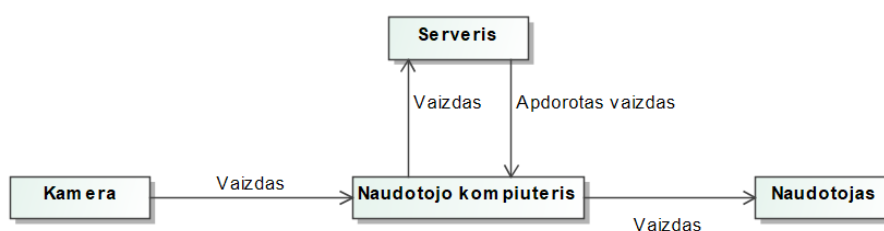
- Neblaivus vairuotojas – vairuotojas, kurio kraujyje yra daugiau nei 0,4 promilės alkoholio.

Prielaidos:

- Vaizdo perdavimas iš kameros į kompiuterį bus greitas ir lengvai įdiegiamas, nepriklausomai nuo vaizdo kameros gamintojo.

2.1.13. Veiklos kontekstas

Veiklos kontekstas vaizduojamas 2.2 paveiksle. Naudotojo kompiuteris iš kameros gaus vaizdą, kurį siųs į serverį. Serveris gražins apdorotą vaizdą, kurį kompiuteris gražins vartotojui.



2.2 pav. Veiklos kontekstas

2.1.14. Veiklos padalinimas

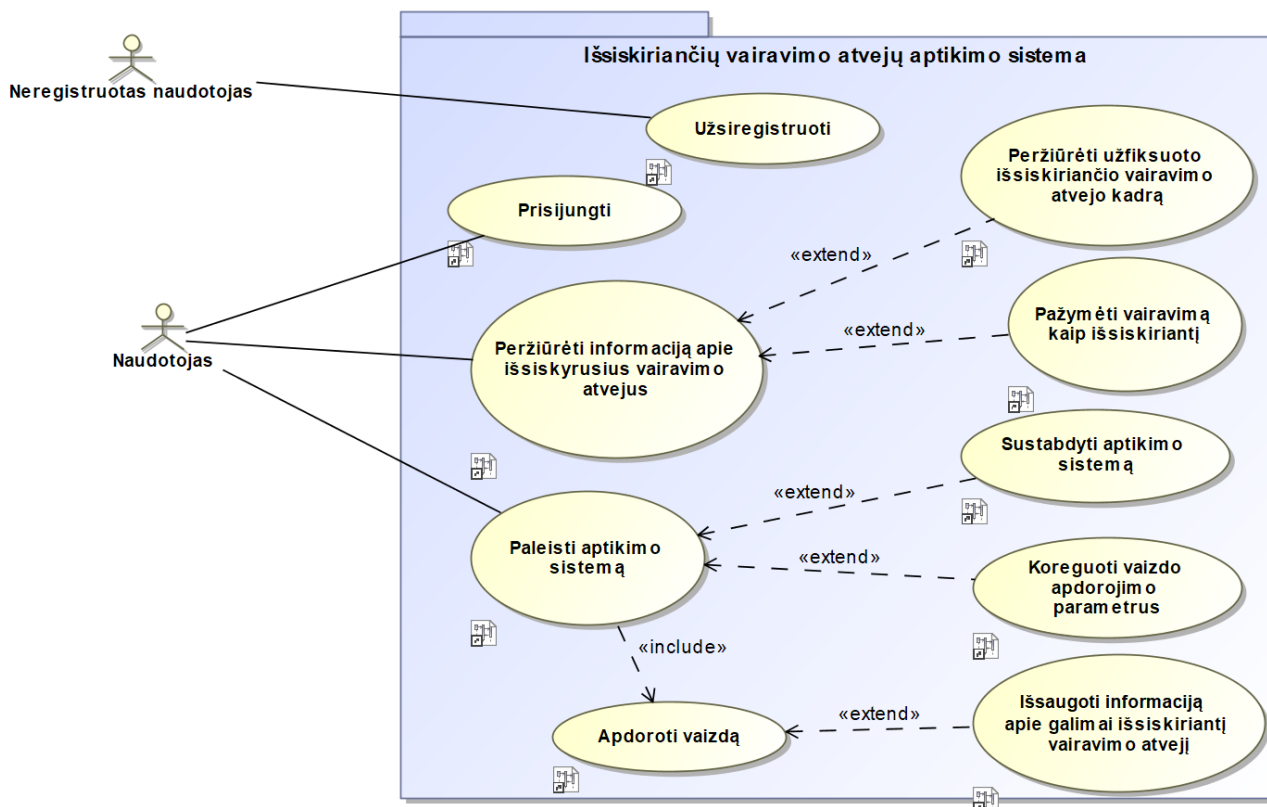
Veiklos įvykiai aprašomi 2.1 lentelėje.

2.1 lentelė Veiklos įvykių sąrašas

Eil. Nr.	Įvykio pavadinimas	Įeinantys / Išeinantys informacijos srautai
1	Kamera perduoda vaizdą	Kelio vaizdas realiu metu (<i>in</i>)
2	Vaizdo apdorojimas	Kelio vaizdas realiu metu (<i>in</i>), Apdorotas kelio vaizdas (<i>out</i>)
3	Išsiskiriančio vairavimo atvejo aptikimo momentas	Apdorotas vaizdas, su pažymėtu išsiskiriančiu vairuotoju (<i>out</i>)

2.1.15. Panaudojimo atvejai

Panaudojimo atvejai vaizduojami 2.3 paveiksle. Sistema naudotis galės tik registruotas naudotojas. Neregistruotiems naudotojams bus suteikiama galimybė užsiregistruoti.



2.3 pav. PA diagrama

PA plačiau aprašomi 2.2-2.11 lentelėse.

2.2 lentelė PA „Prisijungti“

PA 1: „Prisijungti“		
Tikslas. Prisijungti prie sistemos.		
Aprašymas. Naudotojas prisijungimo lange įveda el. paštą ir slaptažodį ir pateikia duomenis.		
Prieš sąlyga		Naudotojas yra sistemos prisijungimo lange.
Aktorius		Naudotojas
Susiję panaudojimo atvejai	Išplečiantys PA	
	Apimami PA	
	Specializuojami PA	
Pagrindinis scenarijus		
Naudotojo veiksmai		Sistemos veiksmai
1. Naudotojas į prisijungimo langą įveda savo prisijungimo duomenis (el. paštą ir slaptažodį).		2. Sistema patikrina ar naudotojo duomenys teisingi.
		3. Naudotojas prijungiamas prie sistemos. 4. Baigiamas PA.
Alternatyvus scenarijus		
		3a. Naudotojui gražinamas klaidos pranešimas, jei jo prisijungimo duomenys neteisingi.
Po sąlyga		Naudotojas prijungiamas prie sistemos.

2.3 lentelė PA „Paleisti aptikimo sistemą“

PA 2: „Paleisti aptikimo sistemą“		
Tikslas. Įjungti sistemą, tam, kad ji atliktų savo funkcijas.		
Aprašymas. Naudotojas įjungia vaizdo skaitymą.		
Prieš sąlyga		Naudotojas yra prisijungęs prie sistemos.
Aktorius		Naudotojas
Susiję panaudojimo atvejai	Išplečiantys PA	Sustabdyti sistemą Koreguoti vaizdo apdorojimo parametrus
	Apimami PA	Apdoroti vaizdą
	Specializuojami PA	
Pagrindinis scenarijus		
Naudotojo veiksmai		Sistemos veiksmai
1. Naudotojas pagrindiniame lange pasirenka sistemos paleidimo funkcionalumą.		2. Paleidžiama sistema, pradedamas skaityti vaizdas, jis apdorojamas PA „Apdoroti vaizdą“ ir gražinamas naudotojui.
3. Naudotojas nori sustabdyti sistemą.		4. Vykdomas PA „Sustabdyti sistemą“.
5. Naudotojas nori pakeisti apdorojimo parametrus.		6. Vykdomas PA „Koreguoti vaizdo apdorojimo parametrus“.
		6. Apdorotas vaizdas gražinamas naudotojui. 7. Baigiamas PA.
Po sąlyga		Aptikimo sistema pradeda veikti.

2.4 lentelė PA „Sustabdyti aptikimo sistemą“

PA 3: „Sustabdyti aptikimo sistemą“	
Tikslas. Sustabdyti aptikimo sistemos veikimą.	
Aprašymas. Naudotojas išjungia aptikimo sistemą.	
Prieš sąlyga	Naudotojas yra prisijungęs prie sistemos. Aptikimo sistema yra paleista.
Aktorius	Naudotojas
Susiję panaudojimo atvejai	Išplečiantys PA
	Apimami PA
	Specializuojami PA
Pagrindinis scenarijus	
Naudotojo veiksmai	Sistemos veiksmai
1. Naudotojas pagrindiniame lange pasirenka sistemos sustabdymo funkcionalumą.	2. Sistema sustabdoma. 3. Baigiamas PA.
Po sąlyga	Sistemos veikimas yra sustabdomas.

2.5 lentelė PA „Užsiregistruoti“

PA 4: „Užsiregistruoti“	
Tikslas. Naudotojui susikurti paskyrą.	
Aprašymas. Naudotojas registracijos lange įveda naudotojo duomenis (el. pašta, slaptažodį) ir pateikia formą.	
Prieš sąlyga	Naudotojas nėra prisijungęs prie sistemos ir yra registracijos lange.
Aktorius	Neregistruotas naudotojas
Susiję panaudojimo atvejai	Išplečiantys PA
	Apimami PA
	Specializuojami PA
Pagrindinis scenarijus	
Naudotojo veiksmai	Sistemos veiksmai
1. Naudotojas naudotojų įvedimo lange įveda naudotojo duomenis (el. pašta, slaptažodį) ir juos pateikia.	2. Sistema išsaugo naujo naudotojo duomenis. 3. Baigiamas PA.
Alternatyvus scenarijus	
	2a. Jei naudotojo įvesti duomenys yra neteisingi sistema gražina naudotoją į prieš tai buvusį duomenų įvedimo langą.
Nefunkcinis reikalavimas	Sistema, neteisingai įvedus duomenis, turi apie tai informuoti.
Po sąlyga	Sukuriama naudotojo paskyra, duomenys išsaugomi duomenų bazėje.

2.6 lentelė PA „Peržiūrėti informaciją apie išsiskyrusius vairavimo atvejus“

PA 5: „Peržiūrėti informaciją apie išsiskyrusius vairavimo atvejus“		
Tikslas. Peržiūrėti išsiskyrusių vairavimo atvejų sąrašą.		
Aprašymas. Naudotojui atvaizduojama informacija apie išsiskyrusius vairavimo atvejus (data, kadras, patvirtinimas).		
Prieš sąlyga		Naudotojas yra prisijungęs prie sistemos.
Aktorius		Naudotojas
Susiję panaudojimo atvejai	Išplečiantys PA	Peržiūrėti užfiksuoto išsiskiriančio vairavimo atvejo kadrą. Pažymėti vairavimą kaip išsiskiriantį.
	Apimami PA	
	Specializuojami PA	
Pagrindinis scenarijus		
Naudotojo veiksmai		Sistemos veiksmai
1. Naudotojas sistemoje pasirenka išsiskyrusių vairavimo atvejų peržiūros funkcionalumą, pasirenką aktualią datą.		2. Sistema išveda išsiskyrusių vairavimo atvejų sąrašą, kurie užfiksuoti tą dieną.
4. Naudotojas nori peržiūrėti konkretų išsiskyrusį vairavimo atvejį.		5. Vykdomas PA „Peržiūrėti užfiksuoto išsiskiriančio vairavimo atvejo kadrą“.
6. Naudotojas nori pažymėti vairavimo atvejį kaip išsiskyrusį.		7. Vykdomas PA „Pažymėti vairavimą kaip išsiskiriantį“.
		8. Baigiamas PA.
Po sąlyga		Naudotojui atvaizduojamas išsiskyrusių vairavimo atvejų sąrašas.

2.7 lentelė PA „Peržiūrėti užfiksuoto išsiskiriančio vairavimo atvejo kadrą“

PA 6: „Peržiūrėti užfiksuoto išsiskiriančio vairavimo atvejo kadrą“		
Tikslas. Peržiūrėti užfiksuoto išsiskiriančio vairavimo atvejo kadrą.		
Aprašymas. Naudotojas iš sąrašo pasirenka vieną konkretų išsiskyrusį vairavimo atvejį. Užfiksuotas kadras atidaromas per visą ekraną.		
Prieš sąlyga		Naudotojas yra išsiskyrusių vairavimo atvejų sąrašo peržiūros lange.
Aktorius		Naudotojas
Susiję panaudojimo atvejai	Išplečiantys PA	
	Apimami PA	
	Specializuojami PA	
Pagrindinis scenarijus		
Naudotojo veiksmai		Sistemos veiksmai
1. Naudotojas pasirenka konkretaus išsiskyrusio vairavimo atvejo peržiūrą.		2. Atvaizduojamas konkretaus išsiskyrusio vairavimo atvejo užfiksuotas kadras per visą ekraną.
		5. Baigiamas PA.
Po sąlyga		Naudotojui atvaizduojamas užfiksuoto išsiskyrusio vairavimo atvejo kadras.

2.8 lentelė PA „Pažymėti vairavimą kaip išsiskiriantį“

PA 7: „Pažymėti vairavimą kaip išsiskiriantį“	
Tikslas. Pažymėti įtarimą sukėlusį vairavimo atvejį kaip išsiskiriantį, jei įtarimai pasitvirtino.	
Aprašymas. Naudotojas, galimai išsiskiriančių vairavimo atvejų peržiūros lange, konkretų vairavimo atvejį pažymi kaip išsiskiriantį.	
Prieš sąlyga	Naudotojas yra galimai išsiskiriančių vairavimo atvejų peržiūros lange.
Aktorius	Naudotojas
Susiję panaudojimo atvejai	Išplečiantys PA
	Apimami PA
	Specializuojami PA
Pagrindinis scenarijus	
1. Naudotojas pasirenka išsiskiriančio vairavimo atvejo pažymėjimo funkcionalumą.	2. Sistema atnaujina išsiskiriančio vairavimo atvejo įrašą. 3. Baigiamas PA.
Po sąlyga	Galimai išsiskiriantis vairavimo atvejis išsaugomas kaip tikrai išsiskiriantis.

2.9 lentelė PA „Apdoroti vaizdą

PA 8: „Apdoroti vaizdą	
Tikslas. Skanuoti vaizdą ir jį apdirbti.	
Aprašymas. Sistema skanuoja vaizdą iš vaizdo registratoriaus ir jį apdirba, naudodamasi dirbtinio intelekto metodais.	
Prieš sąlyga	Sistema yra įjungta.
Aktorius	Naudotojas
Susiję panaudojimo atvejai	Išplečiantys PA
	Apimami PA
	Specializuojami PA
Pagrindinis scenarijus	
Naudotojo veiksmai	Sistemos veiksmai
	1. Sistema periodiškai nuskaityva vaizdą.
	2. Vaizdas apdirbamas naudojantis dirbtinio intelekto metodais. 3. Aptinkamas galimai išsiskiriantis vairavimo atvejis. 4. Vykdomas PA „Išsaugoti informaciją apie galimai išsiskiriantį vairavimo atvejį“. 5. Vaizdas gražinamas naudotojui. 6. Baigiamas PA.
Po sąlyga	Gaunamas vaizdas yra apdorotas.

2.10 lentelė PA „Išsaugoti informaciją apie galimai išsiskiriantį vairavimo atvejį“

PA 9: „Išsaugoti informaciją apie galimai išsiskiriantį vairavimo atvejį“		
Tikslas. Identifikuoti išsiskiriantį vairavimo atvejį ir išsaugoti kadro informaciją.		
Aprašymas. Sistema aptinka galimai išsiskiriantį vairavimo atvejį ir išsaugo kadra ir laiko žymą.		
Prieš sąlyga		Sistema yra nuskaičiusi vaizdą.
Aktorius		Sistema
Susiję panaudojimo atvejai	Išplečiantys PA	
	Apimami PA	
	Specializuojami PA	
Pagrindinis scenarijus		
Naudotojo veiksmai		Sistemos veiksmai
		1. Vaizdo medžiagoje pažymimas galimai išsiskiriantis vairavimo atvejis. 2. Įrašas įrašomas į duomenų bazę.
		3. Baigiamas PA.
Po sąlyga		Identifikuojamas galimai išsiskiriantis vairavimo atvejis, informacija išsaugoma duomenų bazėje.

2.11 lentelė PA „Koreguoti vaizdo apdorojimo parametrus“

PA 10: „Koreguoti vaizdo apdorojimo parametrus“		
Tikslas. Keisti vaizdo apdorojimo parametrus.		
Aprašymas. Naudotojas gali keisti vaizdo apdorojimo parametrus.		
Prieš sąlyga		Sistema paleista.
Aktorius		Naudotojas
Susiję panaudojimo atvejai	Išplečiantys PA	
	Apimami PA	
	Specializuojami PA	
Pagrindinis scenarijus		
Naudotojo veiksmai		Sistemos veiksmai
1. Naudotojas pakeičia parametro reikšmę.		2. Parametro reikšmė pakeičiama vaizdo apdorojimo algoritmuose.
		3. Baigiamas PA.
Po sąlyga		Vaizdas apdorojamas su nauja parametro reikšme.

2.1.16. Funkciniai reikalavimai

Sistemos funkciniai reikalavimai aprašomi 2.12-2.26 lentelėse.

2.12 lentelė Funkcinis reikalavimas 1

Reikalavimas #:	1	Reikalavimo tipas:	9	Įvykis/panaudojimo atvejis #:	1
Aprašymas:	Sistema turi patikrinti ar vartotojo duomenys yra teisingi.				
Pagrindimas:	Prie sistemos turi negalėti prisijungti neautorizuoti naudotojai.				
Šaltinis:	Paulius Melaika				
Tikimo kriterijus:	Prie sistemos gali prisijungti tik registruoti naudotojai, įvedę teisingus duomenis.				
Užsakovo tenkinimas:	4	Užsakovo netenkinimas:	5		
Priklausomybės:	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2020 kovo 09 d.				

2.13 lentelė Funkcinis reikalavimas 2

Reikalavimas #:	2	Reikalavimo tipas:	9	Įvykis/panaudojimo atvejis #:	1
Aprašymas:	Sistema turi prijungti vartotoją prie sistemos, suteikiant jam unikalų raktą.				
Pagrindimas:	Naudotojo veiksmai turi būti atsekami.				
Šaltinis:	Paulius Melaika				
Tikimo kriterijus:	Naudotojo veiksmai sistemoje susiejami su konkrečiu naudotoju.				
Užsakovo tenkinimas:	2	Užsakovo netenkinimas:	4		
Priklausomybės:	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2020 kovo 09 d.				

2.14 lentelė Funkcinis reikalavimas 3

Reikalavimas #:	3	Reikalavimo tipas:	9	Įvykis/panaudojimo atvejis #:	2
Aprašymas:	Sistema turi skaityti vaizdą iš kameros.				
Pagrindimas:	Vaizdas iš kameros turi būti prieinamas sistemai.				
Šaltinis:	Paulius Melaika				
Tikimo kriterijus:	Sistema gauna vaizdą iš kameros ir su juo gali atlikti veiksmus.				
Užsakovo tenkinimas:	4	Užsakovo netenkinimas:	5		
Priklausomybės:	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2020 kovo 09 d.				

2.15 lentelė Funkcinis reikalavimas 4

Reikalavimas #:	4	Reikalavimo tipas:	9	Įvykis/panaudojimo atvejis #:	2,8
Aprašymas:	Sistema turi nuskaitytą vaizdą gražinti naudotojui.				
Pagrindimas:	Naudotojas turi matyti vaizdą realiu metu savo ekrane.				
Šaltinis:	Paulius Melaika				
Tikimo kriterijus:	Naudotojui atvaizduojamas kameros perteikiamas vaizdas.				
Užsakovo tenkinimas:	4	Užsakovo netenkinimas:	2		
Priklausomybės:	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2020 kovo 09 d.				

2.16 lentelė Funkcinis reikalavimas 5

Reikalavimas #:	5	Reikalavimo tipas:	9	Įvykis/panaudojimo atvejis #:	3
Aprašymas:	Sistemą turi būti galima sustabdyti.				
Pagrindimas:	Sistema neturi veikti visa laiką, reikia taupyti GPU resursus.				
Šaltinis:	Paulius Melaika				
Tikimo kriterijus:	Yra galimybė sustabdyti sistemą.				
Užsakovo tenkinimas:	3	Užsakovo netenkinimas:	3		
Priklausomybės:	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2020 kovo 09 d.				

2.17 lentelė Funkcinis reikalavimas 6

Reikalavimas #:	6	Reikalavimo tipas:	9	Įvykis/panaudojimo atvejis #:	4
Aprašymas:	Sistema turi išsaugoti naujo naudotojo registracijos duomenis.				
Pagrindimas:	Užsiregistravusio naudotojo duomenys bus reikalingi jam prisijungiant.				
Šaltinis:	Paulius Melaika				
Tikimo kriterijus:	Sistema įrašo registracijos duomenis į duomenų bazę.				
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:	5		
Priklausomybės:	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2020 kovo 09 d.				

2.18 lentelė Funkcinis reikalavimas 7

Reikalavimas #:	7	Reikalavimo tipas:	9	Įvykis/panaudojimo atvejis #:	4
Aprašymas:	Sistema turi patikrinti naudotojo įvedamus registracijos duomenis.				
Pagrindimas:	Naudotojui, registracijos metu įvedus klaidingus duomenis, jis negalės prisijungti prie sistemos.				
Šaltinis:	Paulius Melaika				
Tikimo kriterijus:	Sistema išsaugo tik teisingus naudotojo duomenis.				
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:	5		
Priklausomybės:	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2020 kovo 09 d.				

2.19 lentelė Funkcinis reikalavimas 8

Reikalavimas #:	8	Reikalavimo tipas:	9	Įvykis/panaudojimo atvejis #:	5
Aprašymas:	Sistema turi išvesti išsiskyrusių vairavimo atvejų sąrašą.				
Pagrindimas:	Naudotojas turi turėti galimybę peržiūrėti išsiskyrusių vairavimo atvejų sąrašą.				
Šaltinis:	Paulius Melaika				
Tikimo kriterijus:	Sistema atvaizduoja išsiskyrusių vairavimo atvejų sąrašą ekrane.				
Užsakovo tenkinimas:	3	Užsakovo netenkinimas:	3		
Priklausomybės:	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2020 kovo 09 d.				

2.20 lentelė Funkcinis reikalavimas 9

Reikalavimas #:	9	Reikalavimo tipas:	9	Įvykis/panaudojimo atvejis #:	6
Aprašymas:	Sistema turi atvaizduoti konkretaus išsiskyrusio vairavimo atvejo informaciją.				
Pagrindimas:	Naudotojas turi turėti galimybę peržiūrėti išsiskyrusio vairavimo atvejo informaciją.				
Šaltinis:	Paulius Melaika				
Tikimo kriterijus:	Sistema naudotojui atvaizduoja konkretaus išsiskyrusio vairavimo atvejo informaciją.				
Užsakovo tenkinimas:	3	Užsakovo netenkinimas:	5		
Priklausomybės:	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2020 kovo 09 d.				

2.21 lentelė Funkcinis reikalavimas 10

Reikalavimas #:	10	Reikalavimo tipas:	9	Įvykis/panaudojimo atvejis #:	7
Aprašymas:	Sistema turi atnaujinti konkretaus išsiskyrusio vairavimo atvejo informaciją.				
Pagrindimas:	Naudotojas turi turėti galimybę koreguoti konkretaus išsiskyrusio vairavimo atvejo informaciją.				
Šaltinis:	Paulius Melaika				
Tikimo kriterijus:	Sistema atnaujina išsiskyrusio vairavimo atvejo duomenis duomenų bazėje.				
Užsakovo tenkinimas:	4	Užsakovo netenkinimas:	5		
Priklausomybės:	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2020 kovo 09 d.				

2.22 lentelė Funkcinis reikalavimas 11

Reikalavimas #:	11	Reikalavimo tipas:	9	Įvykis/panaudojimo atvejis #:	8
Aprašymas:	Sistema turi periodiškai nuskaityti vaizdą (bent kas 0,2 s).				
Pagrindimas:	Vaizdas turi būti nenutrūkstamas.				
Šaltinis:	Paulius Melaika				
Tikimo kriterijus:	Sistema perteikia realaus pasaulio vaizdą be pauzių.				
Užsakovo tenkinimas:	4	Užsakovo netenkinimas:	2		
Priklausomybės:	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2020 kovo 09 d.				

2.23 lentelė Funkcinis reikalavimas 12

Reikalavimas #:	12	Reikalavimo tipas:	9	Įvykis/panaudojimo atvejis #:	8
Aprašymas:	Sistema turi identifikuoti išsiskyrusį vairavimo atvejį.				
Pagrindimas:	Sistema turi vaizde aptikti išsiskyrusį vairavimo atvejį.				
Šaltinis:	Paulius Melaika				
Tikimo kriterijus:	Sistema identifikuoja išsiskyrusį vairavimo atvejį.				
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:	5		
Priklausomybės:	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2020 kovo 09 d.				

2.24 lentelė Funkcinis reikalavimas 13

Reikalavimas #:	13	Reikalavimo tipas:	9	Įvykis/panaudojimo atvejis #:	9
Aprašymas:	Sistema turi pažymėti išsiskyrusį vairavimo atvejį vaizdo medžiagoje.				
Pagrindimas:	Sistema turi išsaugoti išsiskyrusio vairavimo atvejo fakto įrodymą.				
Šaltinis:	Paulius Melaika				
Tikimo kriterijus:	Sistema išsaugo išsiskyrusio vairavimo atvejo vaizdą ir jį jame pažymi.				
Užsakovo tenkinimas:	4	Užsakovo netenkinimas:	3		
Priklausomybės:	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2020 kovo 09 d.				

2.25 lentelė Funkcinis reikalavimas 14

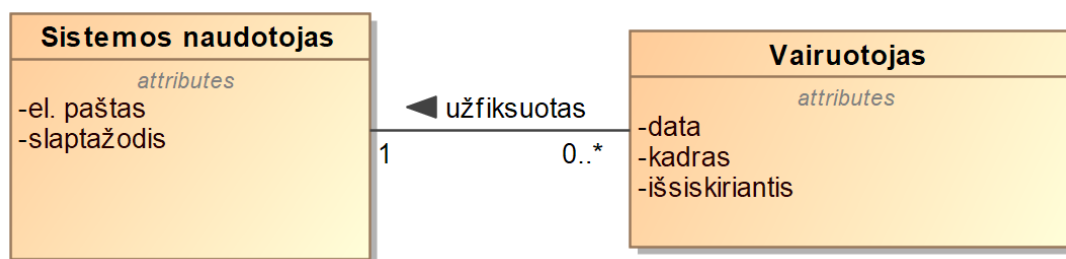
Reikalavimas #:	14	Reikalavimo tipas:	9	Įvykis/panaudojimo atvejis #:	9
Aprašymas:	Sistema turi įrašyti išsiskyrusio vairavimo atvejo duomenis į duomenų bazę.				
Pagrindimas:	Išsiskyrusio vairavimo atvejo įrašas turi būti išsaugomas.				
Šaltinis:	Paulius Melaika				
Tikimo kriterijus:	Sistema išsaugo išsiskyrusio vairavimo atvejo įrašą.				
Užsakovo tenkinimas:	4	Užsakovo netenkinimas:	5		
Priklausomybės:	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2020 kovo 09 d.				

2.26 lentelė Funkcinis reikalavimas 15

Reikalavimas #:	15	Reikalavimo tipas:	9	Įvykis/panaudojimo atvejis #:	10
Aprašymas:	Sistema naudotojui turi atvaizduoti vaizdą su išsiskyrusiu vairavimo atveju.				
Pagrindimas:	Naudotojas turi realiu metu galėti identifikuoti išsiskyrusį vairavimo atvejį.				
Šaltinis:	Paulius Melaika				
Tikimo kriterijus:	Sistema ekrane atvaizduoja realų vaizdą ir jame pažymi išsiskyrusį vairavimo atvejį.				
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:	5		
Priklausomybės:	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2020 kovo 09 d.				

2.1.17. Reikalavimai duomenims

Sistemos ER modelis pateikiamas 2.4 paveiksle. Sistemoje bus dvi esybės – sistemos naudotojas ir vairuotojas.



2.4 pav. ER modelis

2.1.18. Nefunkciniai reikalavimai

Sistemos nefunkciniai reikalavimai aprašomi 2.27-2.39 lentelėse.

Reikalavimai sistemos išvaizdai:

2.27 lentelė Nefunkcinis reikalavimas 16

Reikalavimas #:	16	Reikalavimo tipas:	10	Įvykis/panaudojimo atvejis #:	visi
Aprašymas:	Patvirtinimo mygtukas visada turi būti kairėje pusėje, o atmetimo dešinėje.				
Pagrindimas:	Vartotojas spaudžia iš įpročio neskaitydamas.				
Šaltinis:	Paulius Melaika				
Tikimo kriterijus:	Visi patvirtinimo mygtukai sistemoje yra kairėje, o atmetimo dešinėje.				
Užsakovo tenkinimas:	1			Užsakovo netenkinimas:	5
Priklausomybės:	Nėra			Konfliktai:	Nėra
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2020 kovo 09 d.				

2.28 lentelė Nefunkcinis reikalavimas 17

Reikalavimas #:	17	Reikalavimo tipas:	10	Įvykis/panaudojimo atvejis #:	visi
Aprašymas:	Sąsaja turi būti patogi naudoti.				
Pagrindimas:	Naudotojui bus lengviau perprasti sistemą.				
Šaltinis:	Paulius Melaika				
Tikimo kriterijus:	Atlikus apklausą, praėjus mėnesiui nuo sistemos diegimo, 70 % naudotojų turi patvirtinti, kad sąsaja yra patogi naudoti.				
Užsakovo tenkinimas:	2			Užsakovo netenkinimas:	4
Priklausomybės:	Nėra			Konfliktai:	Nėra
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2020 kovo 09 d.				

Reikalavimai panaudojamumui:

2.29 lentelė Nefunkcinis reikalavimas 18

Reikalavimas #:	18	Reikalavimo tipas:	11	Įvykis/panaudojimo atvejis #:	visi
Aprašymas:	Visas sistemos funkcionalumas turi būti pasiekiamas dviem mygtukų paspaudimas iš pagrindinio lango.				
Pagrindimas:	Vartotojai neturi klaidžioti po sistemą.				
Šaltinis:	Paulius Melaika				
Tikimo kriterijus:	Visas sistemos funkcionalumas yra pasiekiamas dviem mygtukų paspaudimais iš pagrindinio lango.				
Užsakovo tenkinimas:	2	Užsakovo netenkinimas:	2		
Priklausomybės:	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2020 kovo 09 d.				

2.30 lentelė Nefunkcinis reikalavimas 19

Reikalavimas #:	19	Reikalavimo tipas:	11	Įvykis/panaudojimo atvejis #:	visi
Aprašymas:	Sistema turi turėti „Dark mode“ darbui naktį.				
Pagrindimas:	Sistema tamsiu paros metu turi neblaškyti.				
Šaltinis:	Paulius Melaika				
Tikimo kriterijus:	Sistemoje yra „Dark mode“ režimas.				
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:	1		
Priklausomybės:	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2020 kovo 09 d.				

Reikalavimai vykdymo charakteristikoms:

2.31 lentelė Nefunkcinis reikalavimas 20

Reikalavimas #:	20	Reikalavimo tipas:	12	Įvykis/panaudojimo atvejis #:	8
Aprašymas:	Vaizdas ekrane turi nevēluoti daugiau nei 1 sekundę.				
Pagrindimas:	Vaizdas ekrane turi būti tiesioginis.				
Šaltinis:	Paulius Melaika				
Tikimo kriterijus:	Vaizdas ekrane nevēluoja daugiau nei 1 sekundę.				
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:	5		
Priklausomybės:	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2020 kovo 09 d.				

2.32 lentelė Nefunkcinis reikalavimas 21

Reikalavimas #:	21	Reikalavimo tipas:	12	Įvykis/panaudojimo atvejis #:	visi
Aprašymas:	Sistema turi pasileisti per 10 sekundžių.				
Pagrindimas:	Sistema turi būti greit įjungiamą, tai leis patikrinti daugiau vairuotojų.				
Šaltinis:	Paulius Melaika				
Tikimo kriterijus:	Sistema pasileidžia per 10 sekundžių.				
Užsakovo tenkinimas:	3	Užsakovo netenkinimas:	3		
Priklausomybės:	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2020 kovo 09 d.				

Reikalavimai veikimo sąlygoms:

2.33 lentelė Nefunkcinis reikalavimas 22

Reikalavimas #:	22	Reikalavimo tipas:	13	Įvykis/panaudojimo atvejis #:	visi
Aprašymas:	Sistema turi veikti ir prastomis oro sąlygomis (sningant, lyjant, esant rūkui, prastam matomumui).				
Pagrindimas:	Sistema turi veikti ištisą parą, visus metus.				
Šaltinis:	Paulius Melaika				
Tikimo kriterijus:	Sistema tinkamai veikia bet koku oru.				
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:	5		
Priklausomybės:	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2020 kovo 09 d.				

Reikalavimai sistemos priežiūrai:

2.34 lentelė Nefunkcinis reikalavimas 23

Reikalavimas #:	23	Reikalavimo tipas:	14	Įvykis/panaudojimo atvejis #:	visi
Aprašymas:	Sistemą pritaikyti bet kokiam registratoriui turi užtrukti ne daugiau nei 10 darbo valandų.				
Pagrindimas:	Sistema turi galėti veikti kartu su bet koku vaizdo registratoriumi.				
Šaltinis:	Paulius Melaika				
Tikimo kriterijus:	Prie sistemos galima prijungti bet kokią vaizdo registratorių.				
Užsakovo tenkinimas:	2	Užsakovo netenkinimas:	2		
Priklausomybės:	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2020 kovo 09 d.				

2.35 lentelė Nefunkcinis reikalavimas 24

Reikalavimas #:	24	Reikalavimo tipas:	14	Įvykis/panaudojimo atvejis #:	visi
Aprašymas:	Sistemos priežiūrai ir palaikymui per mėnesį turi būti skiriama daugiausia 40 darbo valandų.				
Pagrindimas:	Sistema turi veikti be papildomos priežiūros ir palaikymo.				
Šaltinis:	Paulius Melaika				
Tikimo kriterijus:	Metus po sistemos diegimo, kiekvieną mėnesį sistemos priežiūrai ir palaikymui skiriama maksimaliai 40 darbo valandų.				
Užsakovo tenkinimas:	1	Užsakovo netenkinimas:	4		
Priklausomybės:	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2020 kovo 09 d.				

Reikalavimai saugumui:

2.36 lentelė Nefunkcinis reikalavimas 25

Reikalavimas #:	25	Reikalavimo tipas:	15	Įvykis/panaudojimo atvejis #:	visi
Aprašymas:	Sistema gali naudotis tik autorizuotas naudotojas.				
Pagrindimas:	Sistema turi būti apsaugota nuo pašalinių asmenų veiksmų.				
Šaltinis:	Paulius Melaika				
Tikimo kriterijus:	Sistemos funkcionalumas pasiekiamas tik autorizuotam naudotojui.				
Užsakovo tenkinimas:	3	Užsakovo netenkinimas:	5		
Priklausomybės:	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2020 kovo 09 d.				

2.37 lentelė Nefunkcinis reikalavimas 26

Reikalavimas #:	26	Reikalavimo tipas:	15	Įvykis/panaudojimo atvejis #:	8
Aprašymas:	Sistemos fiksuojamas vaizdas turi būti atvaizduojamas tik naudotojui.				
Pagrindimas:	Sistemos fiksuojamas vaizdas turi būti nepasiekiamas iš išorės.				
Šaltinis:	Paulius Melaika				
Tikimo kriterijus:	Sistemos vaizdas atvaizduojamas tik sistemos naudotojui.				
Užsakovo tenkinimas:	2	Užsakovo netenkinimas:	5		
Priklausomybės:	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2020 kovo 09 d.				

Teisiniai reikalavimai

2.38 lentelė Nefunkcinis reikalavimas 27

Reikalavimas #:	27	Reikalavimo tipas:	17	Įvykis/panaudojimo atvejis #:	8
Aprašymas:	Sistemos užfiksuoti vaizdai turi būti galimi naudoti įrodymams (turi matytis laikas, vieta).				
Pagrindimas:	Sistemos užfiksuoti vaizdai bus naudojami pažeidimų įrodymui.				
Šaltinis:	Paulius Melaika				
Tikimo kriterijus:	Sistemos užfiksuotuose vaizduose matosi laikas ir vieta.				
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:	5		
Priklausomybės:	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2020 kovo 09 d.				

2.39 lentelė Nefunkcinis reikalavimas 28

Reikalavimas #:	28	Reikalavimo tipas:	17	Įvykis/panaudojimo atvejis #:	visi
Aprašymas:	Sistema turi laikytis BDAR įstatymo.				
Pagrindimas:	Sistema turi laikytis įstatymų.				
Šaltinis:	Paulius Melaika				
Tikimo kriterijus:	Sistema atitinka BDAR įstatymą.				
Užsakovo tenkinimas:	3	Užsakovo netenkinimas:	5		
Priklausomybės:	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2020 kovo 09 d.				

2.1.19. Pagamintos sistemos, kurios gali būti nupirktos

Pagamintos sistemos, kurias galima nusipirkti:

- DADDS. Veikia tik kai yra įdiegta automobilyje.
- „Netradyne“ platforma „Driveri“.
- „Sparkbit.

2.1.20. Pagaminti komponentai, kurie gali būti panaudoti

<http://cbcl.mit.edu/software-datasets/streetscenes/> - nuotraukų rinkinys neuroninio tinklo apmokymui.

2.1.21. Problemos diegimo aplinkai

Sistema gali eikvoti automobilio akumulatoriaus energiją.

2.1.22. Įtaka jau instaliuotoms sistemoms

Įtakos instaliuotoms sistemoms nenustatyta.

2.1.23. Neigiamas vartotojų nusiteikimas

Visuomenė gali būti nusiteikusi prieš kiekvieno automobilio skanavimą, gali dalintis skenuojančio naudotojo lokacija socialiniuose tinkluose.

2.1.24. Kliudantys diegimo aplinkos apribojimai

Neįmanoma užtikrinti gero interneto ryšio automobiliuose.

2.1.25. Galimos naujos sistemos sukeltos problemos

Naujos sistemos sukeltų problemų nenustatyta.

2.1.26. Vystymo etapai

Sistemos vystymo etapai:

1. 2019-11-01 – 2019-11-15
 - a. reikalavimų surinkimas;
 - b. reikalavimų analizė;
2. 2019-11-16 – 2019-12-01
 - a. projektavimas;
3. 2019-12-02 – 2020-03-15
 - a. duomenų surinkimas;
 - b. duomenų analizė;
4. 2020-03-15 – 2020-05-01
 - a. duomenų paruošimas;
5. 2020-05-02 – 2021-01-01 (iteracinis procesas)
 - a. ML modelio pasirinkimas;
 - b. ML modelio įvertinimas;
 - c. ML modelio realizavimas;
 - d. ML modelio apmokymas;
 - e. ML modelio optimizavimas;
6. 2021-01-02 – 2021-02-01
 - a. testavimas;
7. 2021-02-02 – 2021-03-01
 - a. diegimas;
8. 2021-03-02 - ...
 - a. palaikymas.

2.1.27. Pritaikymas

Sistema bus diegiama visiškai nuo nulio, jokių senų duomenų perkelti, palaikyti nereikės.

2.1.28. Reikalavimai esamų duomenų perkėlimui

Reikalavimų duomenų perkėlimui nenustatyta.

2.1.29. Reikalingas duomenų transformavimas perkeliant į naują sistemą

Duomenų transformavimas nereikalingas.

2.1.30. Galimos sistemos kūrimo rizikos

Sistemos kūrimo rizikos aprašomos 2.40 lentelėje.

2.40 lentelė Kūrimo rizikos

Rizikos faktorius	Tikimybinis įvertinimas
Reikalavimų specifikacijos pasikeitimai realizavimo fazėje	10
Netikslūs laiko paskaičiavimai	8
Sunkiai realizuojamas atpažinimo algoritmas	10
Įtemptas darbo grafikas	7
Netinkamas projekto valdymas	5
Netikslus kaštų įvertinimas	5
Augantys reikalavimai	7
Prasta kokybė	4
Prastas produktyvumas	5
Projekto atšaukimas	1

2.1.31. Atsitiktinumų (rizikų) planas

Sistemos rizikų planas pateikiamas 2.41 lentelėje.

2.41 lentelė Rizikų planas

Rizikos faktorius	Problemos sprendimas
Reikalavimų specifikacijos pasikeitimai realizavimo fazėje	Norint keisti reikalavimus tokioje fazėje, jie turi būti tikrai labai svarbūs.
Netikslūs laiko paskaičiavimai.	Naudoti Agile metodologiją.
Sunkiai realizuojamas atpažinimo algoritmas	Išmokti dalykinę sritį prieš realizaciją.
Įtemptas darbo grafikas	Planuoti laiką iš anksto.
Netinkamas projekto valdymas	Laikytis nustatytų terminų.
Netikslus kaštų įvertinimas	Sekti išlaidas ir jas riboti.
Augantys reikalavimai	Nauji reikalavimai turi būti labai svarbūs.
Prasta kokybė	Naudoti kodo peržiūros techniką, KISS principą.
Prastas produktyvumas	Gerai išsimiegoti.
Projekto atšaukimas	Realizuoti sistemą pagal reikalavimus.

2.1.32. Kaina

Vaizdo registratorius – 3x100 eurų.

Projekto vadovas – 10 dienų x 100 eurų = 1000 eurų

Programuotojas – 100 dienų x 80 eurų = 8000 eurų

Suma = 9300 eurai

2.1.33. Vartotojo dokumentacija ir apmokymas

Dokumentacija naudotojui. PDF failas. Aprašomas sistemos funkcionalumas su pavyzdžiais. Padaromi mokomieji video. Atsakingas Paulius Melaika.

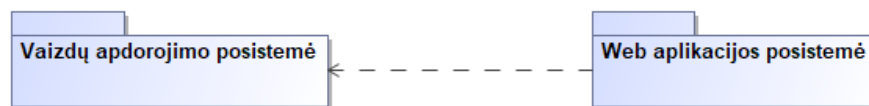
Dokumentacija programuotojams. Turės būti paruošta API dokumentacija. PDF formatu. Ji bus skirta sistemos plėtimui ateityje, jos tobulinimui ir taisymui. Atsakingas Paulius Melaika.

2.2. Sistemos statinis vaizdas

2.2.1. Sistemos posistemės

Sistemoje bus dvi posistemės (2.5 pav.):

- Vaizdų apdorojimo posistemė;
- Web aplikacijos posistemė.



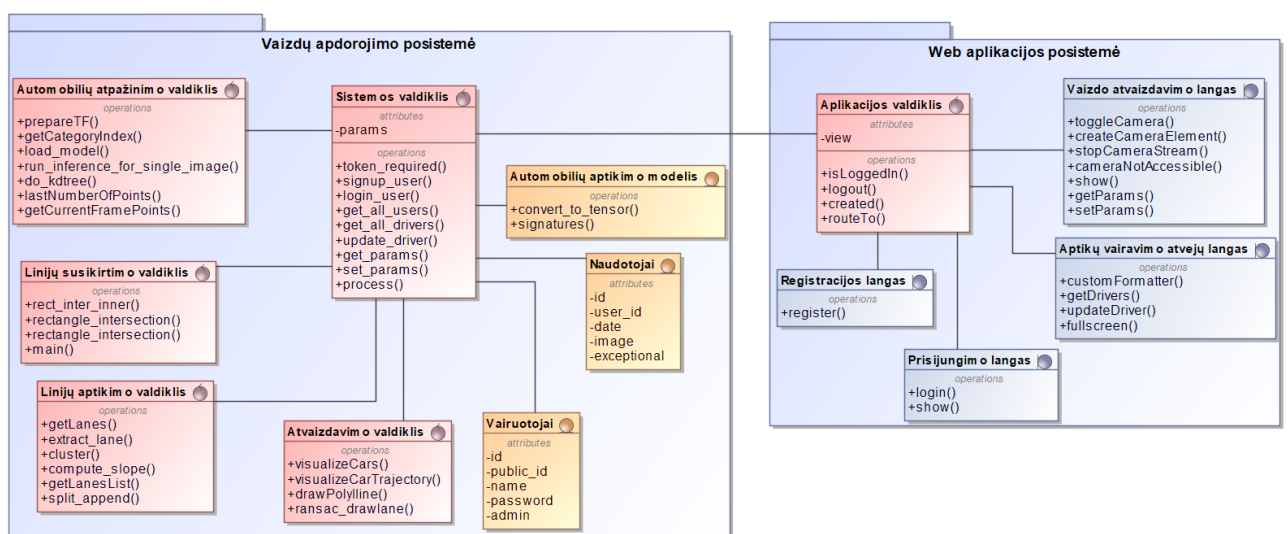
2.5 pav. Sistemos posistemės

2.2.2. Paketų detalizavimas

Vaizdų apdorojimo posistemėje atliekami veiksmai su vaizdais. Taip pat joje yra sistemos naudotojų autorizacijos prie serverio resursų logika. Šioje posistemėje yra ir dirbtinio intelekto modelis, skirtas vaizdų apdorojimui. Ši posistemė iš Web aplikacijos posistemės gaus vaizdo kadrus, atliks su jais veiksmus ir gražins apdorotą vaizdą.

Web aplikacijos posistemė skirta kliento matomam vaizdui naršyklėje atvaizduoti.

Detaliau sistemos posistemės ir klasės pavaizduotos 2.6 paveiksle.



2.6 pav. Sistemos posistemės ir klasės

2.3. Sistemos dinaminis vaizdas

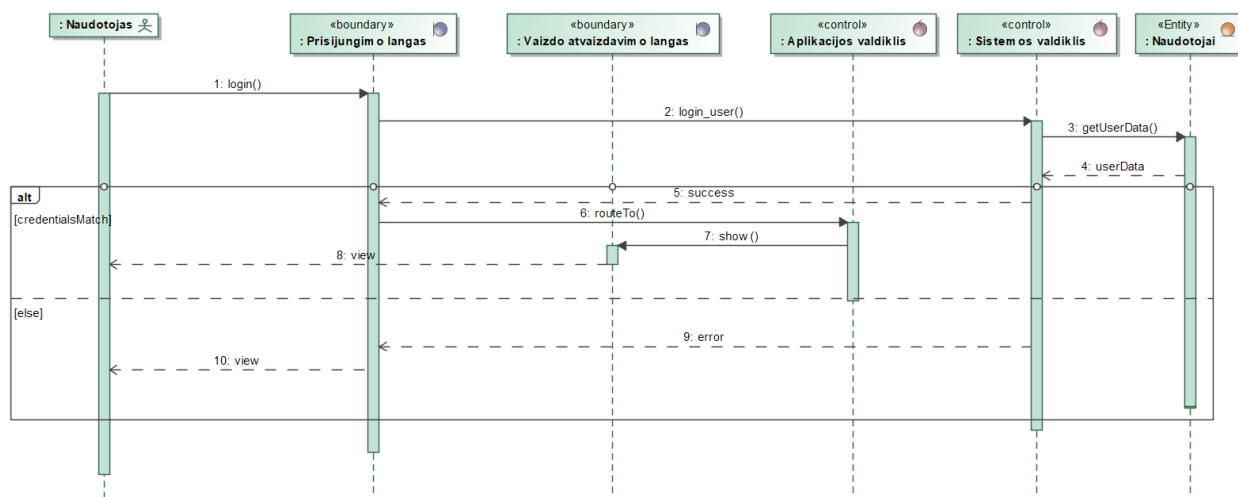
Sistemos sąveika atvaizduojama sekų diagramomis (2.7-2.16 pav.) ir bendradarbiavimo diagramomis (2.17-2.26 pav.).

Sistemoje svarbi vairotojo būsenų kaita. Ji atvaizduojama būsenų diagrama (2.27 pav.).

Sistemos ir jos naudotojų atliekami veiksmai atvaizduoti veiklos diagramomis (2.28-2.37 pav.).

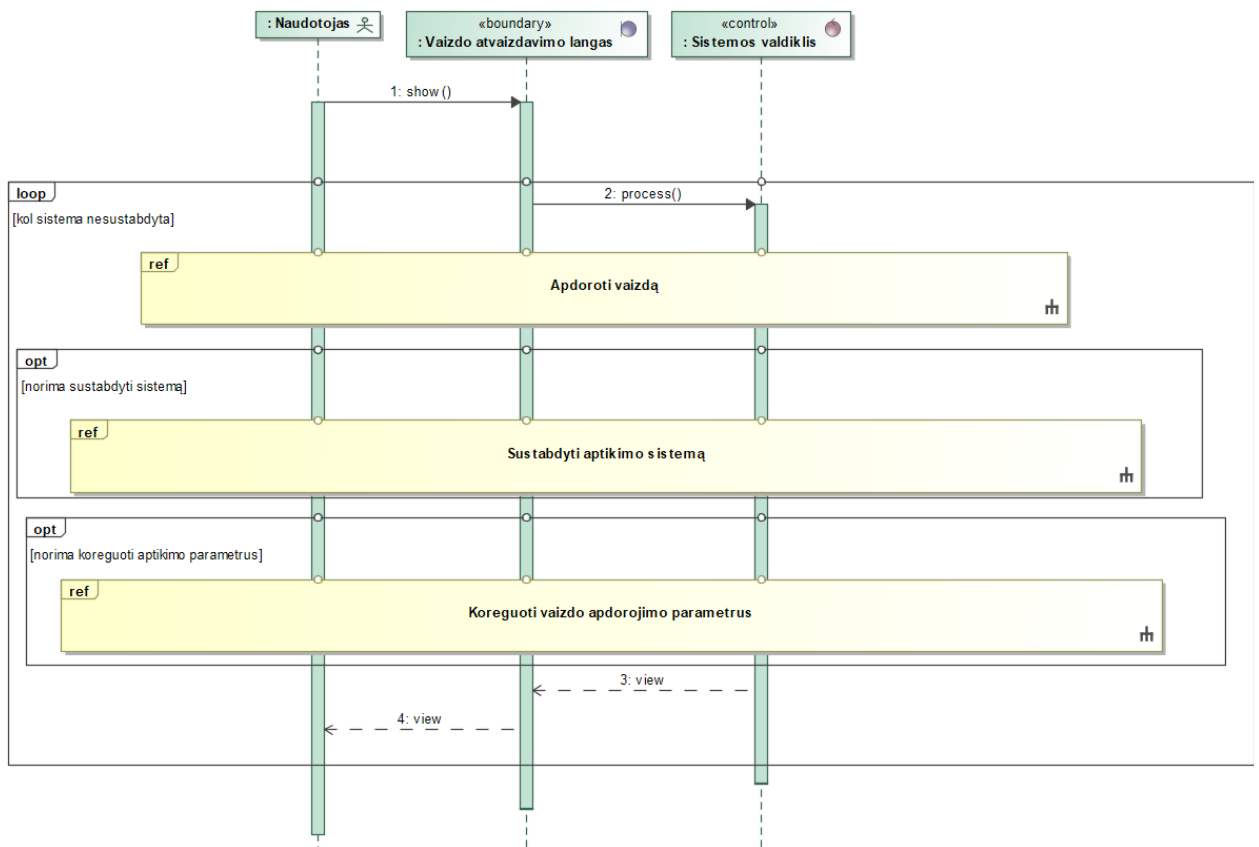
2.3.1. Sistemos sekų diagramos

Sistemos sąveika atvaizduojama sekų diagramomis (2.7-2.16 pav.).



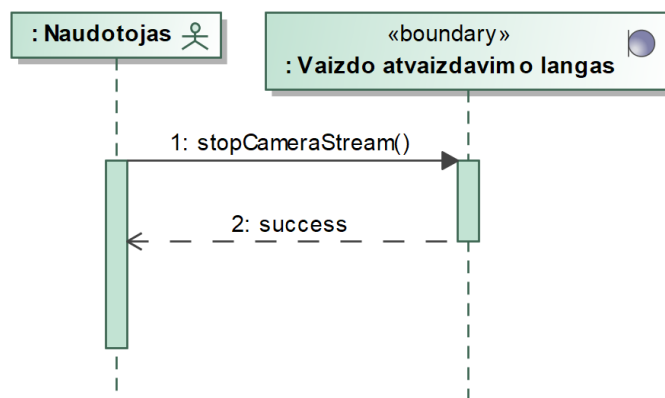
2.7 pav. PA „Prisijungti“ veiksmų eiliškumo diagrama

Naudotojas prisijungimo lange įveda prisijungimo duomenis, sistema patikrina ar duomenys teisingi. Jei duomenys teisingi, naudotojui atvaizduojamas vaizdo atvaizdavimo langas, o jei duomenys neteisingi, naudotojas gražinamas į prisijungimo langą.



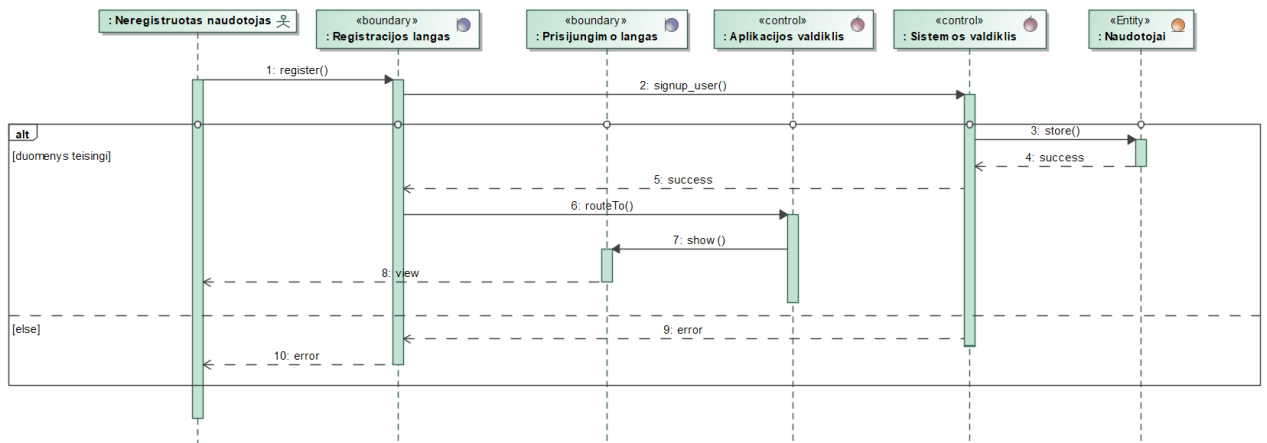
2.8 pav. PA „Paleisti aptikimo sistemą“ veiksmų eiliškumo diagrama

Naudotojas, būdamas vaizdo atvaizdavimo lange, gali paleisti aptikimo sistemą. Kol sistema nebus sustabdyta bus vykdomas PA „Apdoroti vaizdą“. Taip pat naudotojas gali sustabdyti aptikimo sistemą arba koreguoti vaizdo apdorojimo parametrus.



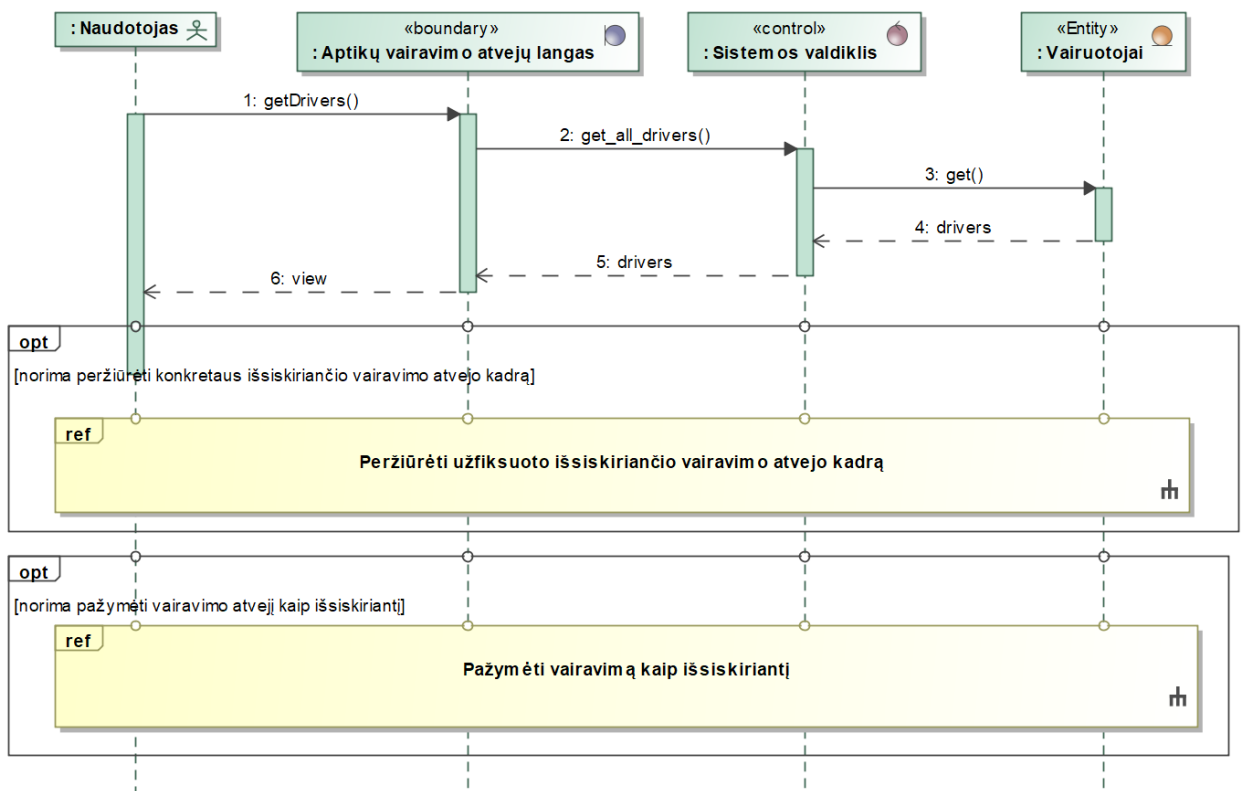
2.9 pav. PA „Sustabdyti aptikimo sistemą“ veiksmų eiliškumo diagrama

Naudotojas, būdamas vaizdo atvaizdavimo lange, gali pasirinkti aptikimo sistemos sustabdymo funkcionalumą. Tai atlikus vaizdo apdorojimas yra sustabdomas.



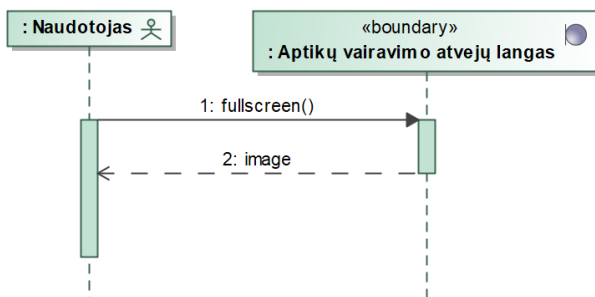
2.10 pav. PA „Užsiregistruoti“ veiksmų eiliškumo diagrama

Norėdamas užsiregistruoti, naudotojas registracijos lange įveda savo duomenis. Sistema patikrina ar duomenys yra teisingi. Jei duomenys teisingi, sistema išsaugo juos duomenų bazėje. Jei duomenys neteisingi, naudotojas gražinamas į registracijos langą ir jam yra parodoma, kurie duomenys yra neteisingi.



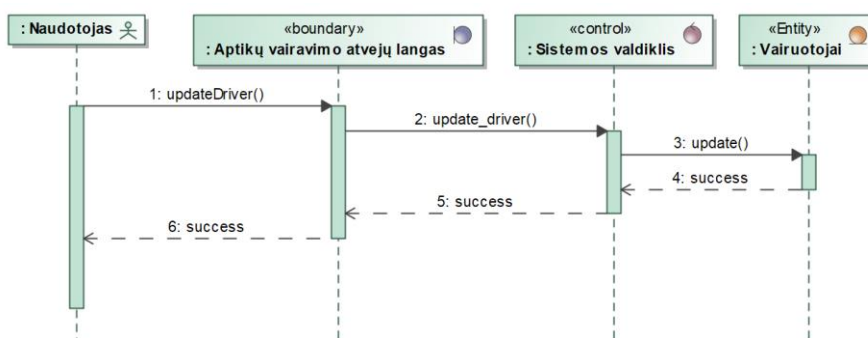
2.11 pav. PA „Peržiūrėti informaciją apie išsiskyrusius vairavimo atvejus“ veiksmų eiliškumo diagrama

Naudotojas gali peržiūrėti informaciją apie išsiskyrusius vairavimo atvejus. Tai jis gali padaryti nuėjęs į aptiktų vairavimo atvejų langą. Sistema iš duomenų bazės pasiima išsiskyrusių vairavimo atvejų informaciją ir ją gražina naudotojui. Naudotojas gali peržiūrėti konkretaus užfiksuoto vairavimo atvejo kadra arba pažymėti vairavimą kaip tikrai išsiskiriantį.



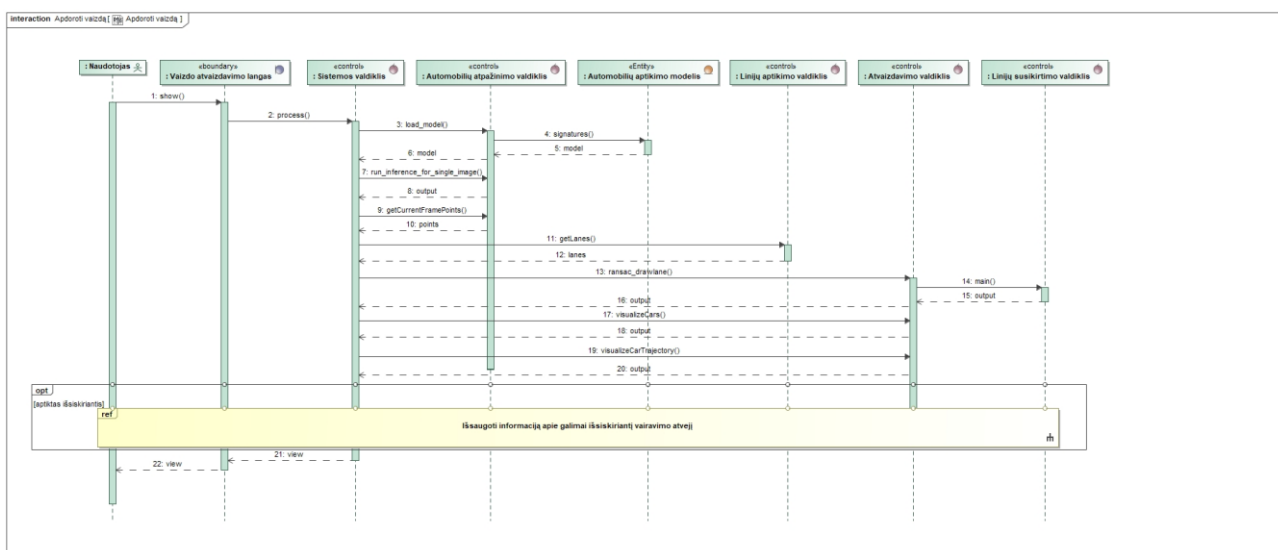
2.12 pav. PA „Peržiūrėti užfiksuoto išsiskiriančio vairavimo atvejo kadra“ veiksmų eiliškumo diagrama

Naudotojui pasirinkus išsiskiriančio vairavimo atvejo kadro peržiūrą, sistema kadra atvaizduoja pilno ekrano režimu.



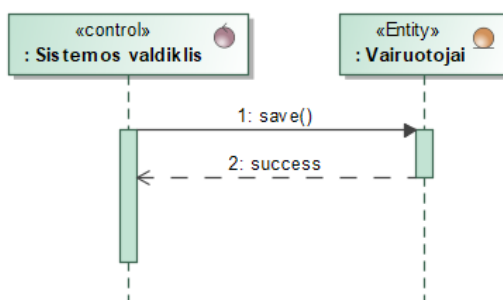
2.13 pav. PA „Pažymėti vairavimą kaip išsiskiriantį“ veiksmų eiliškumo diagrama

Naudotojas, būdamas aptiktų vairavimo atvejų lange, gali bet kurį vairavimo atvejį pažymėti kaip išsiskiriantį. Tokiu atveju, sistema atnaujina vairavimo atvejo įrašą duomenų bazėje ir gražina naudotojui sėkmės pranešimą.



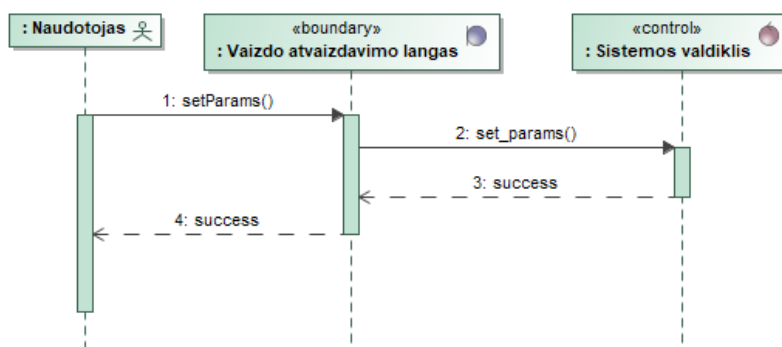
2.14 pav. PA „Apdoroti vaizdą“ veiksmų eiliškumo diagrama

Kai aptikimo sistema yra paleista, naudotojui vaizdo apdorojimo lange atvaizduojamas iš sistemos gautas vaizdas. Sistema vaizde aptinka automobilius, paskaičiuoja jų trajektoriją, aptinka eismo linijas ir pritaiko miglotąją logiką. Jei aptinkamas išsiskiriantis vairavimo atvejis vykdomas PA „Išsaugoti informaciją apie galimai išsiskiriantį vairavimo atvejį“.



2.15 pav. PA „Išsaugoti informaciją apie galimai išsiskiriantį vairavimo atvejį“ veiksmų eiliškumo diagrama

Išsiskiriančio vairavimo atvejo informacija yra išsaugoma duomenų bazėje.

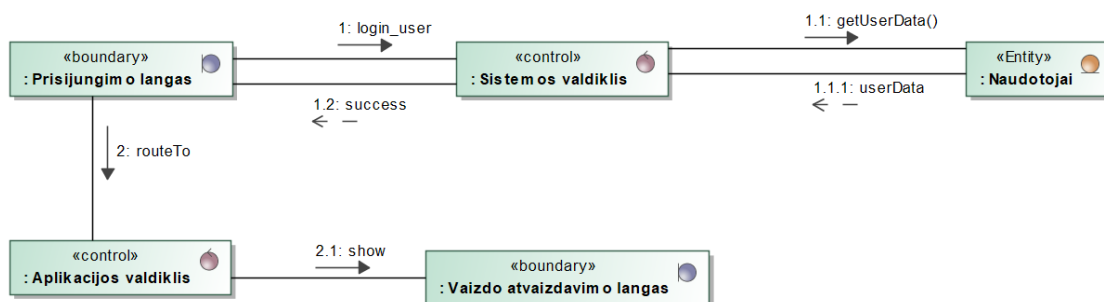


2.16 pav. PA „Koreguoti vaizdo apdorojimo parametrus“ veiksmų eiliškumo diagrama

Naudotojas, būdamas vaizdo atvaizdavimo lange, gali koreguoti vaizdo apdorojimo parametrus. Tokiu atveju, sistemos valdiklis juos išsaugo ir naudoja vaizdo apdorojimui.

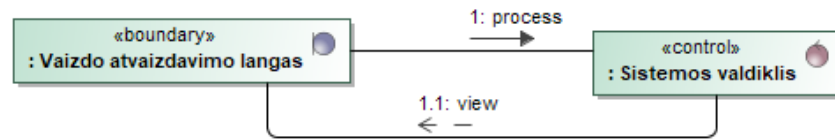
2.3.2. Sistemos bendradarbiavimo diagramos

Sistemos sąveika atvaizduojama bendradarbiavimo diagramomis (2.17-2.26 pav.).



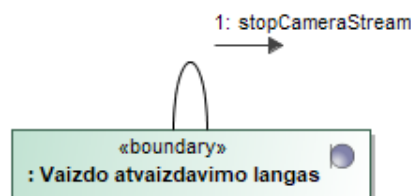
2.17 pav. PA „Prisijungti“ sąveikos diagrama

PA „Prisijungti“ prasideda prisijungimo lange. Duomenys iš prisijungimo lango siunčiami į sistemos valdiklį, kuris juos išsaugo duomenų bazėje. Aplikacijos valdiklis naudotojui gražina vaizdo atvaizdavimo langą.



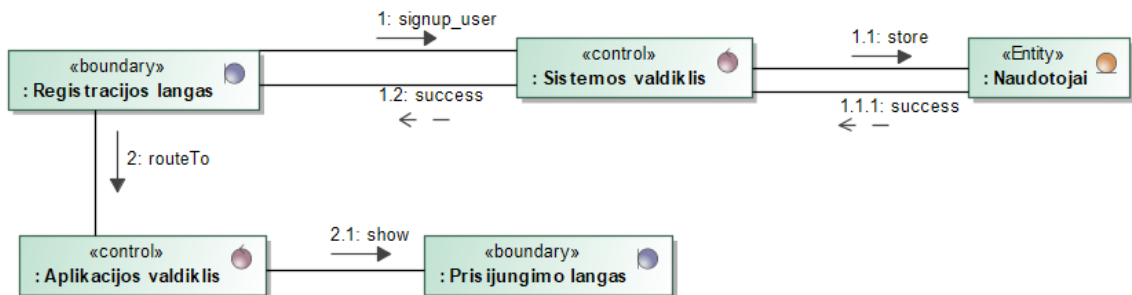
2.18 pav. PA „Paleisti aptikimo sistemą“ sąveikos diagrama

Vaizdo atvaizdavimo lange pasirinkus aptikimo sistemos paleidimo funkcionalumą, kameros vaizdas pradamas siųsti sistemos valdikliui, kuris jį apdoroja ir vaizdas yra gražinamas į vaizdo atvaizdavimo langą.



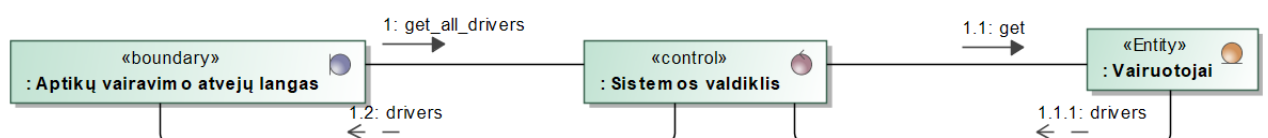
2.19 pav. PA „Sustabdyti aptikimo sistemą“ sąveikos diagrama

Sustabdant sistemą, vaizdas nustoja būti siunčiamas sistemos valdikliui.



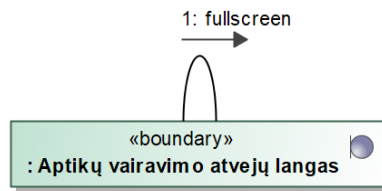
2.20 pav. PA „Užsiregistruoti“ sąveikos diagrama

Naudotojui registruojantis sistemoje, jo duomenys iš registracijos lango siunčiami į sistemos valdiklį. Sistemos valdiklis duomenis išsaugo duomenų bazėje. Aplikacijos valdiklis naudotoją nukreipia į prisijungimo langą.



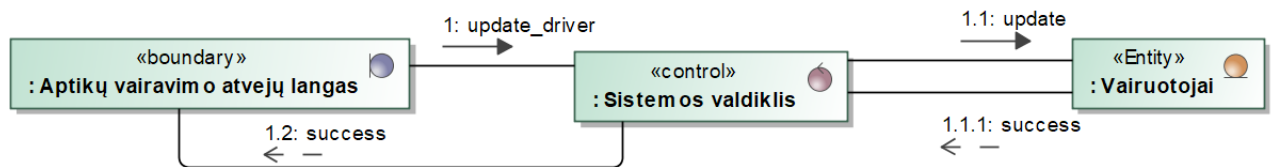
2.21 pav. PA „Peržiūrėti informaciją apie išsiskyrusius vairavimo atvejus“ sąveikos diagrama

Naudotojui pasirinkus išsiskyrusių vairavimo atvejų peržiūros funkcionalumą, aptiktų vairavimo atvejų lange atvaizduojami vairavimo atvejų duomenys, kurios sistemos valdiklis gražino iš duomenų bazės.



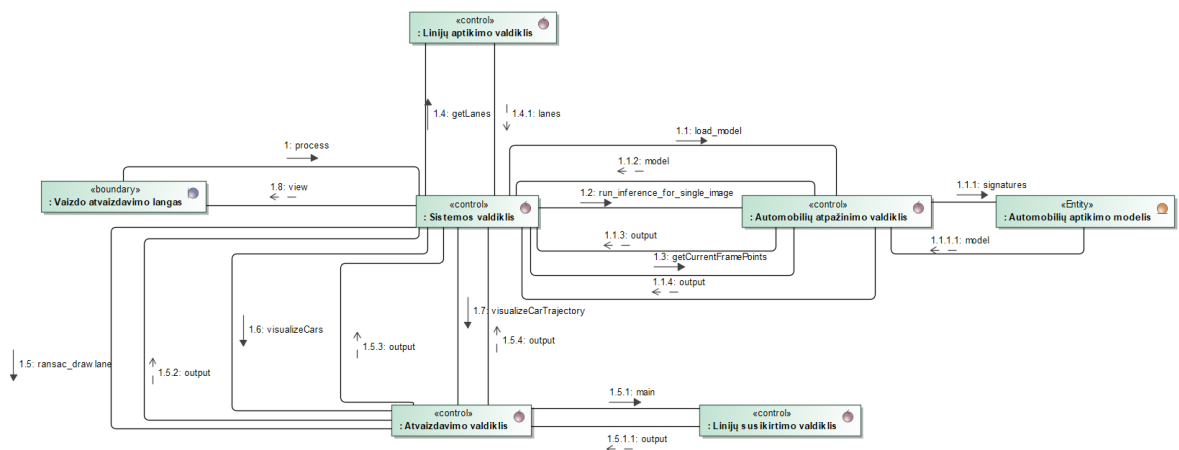
2.22 pav. PA „Peržiūrėti užfiksuoto išsiskiriančio vairavimo atvejo kadra“ sąveikos diagrama

Kai naudotojas nori peržiūrėti išsiskiriančio vairavimo atvejo kadra, jis yra atvaizduojamas pilno ekrano režimu.



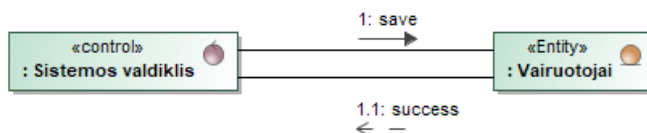
2.23 pav. PA „Pažymėti vairavimą kaip išsiskiriantį“ sąveikos diagrama

Naudotojui aptiktų vairavimo atvejų lange pažymėjus vairavimą kaip išsiskiriantį, vairavimo atvejo informacija yra siunčiama sistemos valdikliui, kuris atnaujina atvejo informaciją duomenų bazėje.



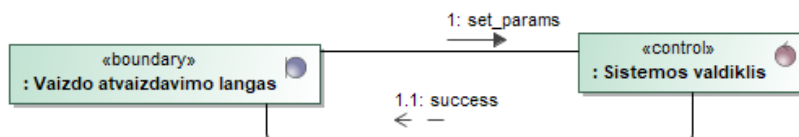
2.24 pav. PA „Apdoroti vaizdą“ sąveikos diagrama

Paleidus aptikimo sistemą, vaizdo atvaizdavimo langas siunčia kadra į sistemos valdiklį. Sistemos valdiklis iš linijų aptikimo valdiklio gauna linijų informaciją. Taip pat automobilių atpažinimo valdiklis, pasinaudojęs automobilių aptikimo modeliu, gauna automobilių lokaciją kadre ir ją gražina sistemos valdikliui. Sistemos valdiklis kreipiasi į atvaizdavimo valdiklį, kuris atlieka skaičiavimus ir gražina apdorotą vaizdą. Atlikus visus veiksmus, sistemos valdiklis gražina galutinį vaizdą į vaizdo atvaizdavimo langą.



2.25 pav. PA „Išsaugoti informaciją apie galimai išsiskiriantį vairavimo atvejį“ sąveikos diagrama

Sistemai aptikus galimai išsiskiriantį vairavimo atvejį, jo informacija yra išsaugoma duomenų bazėje.

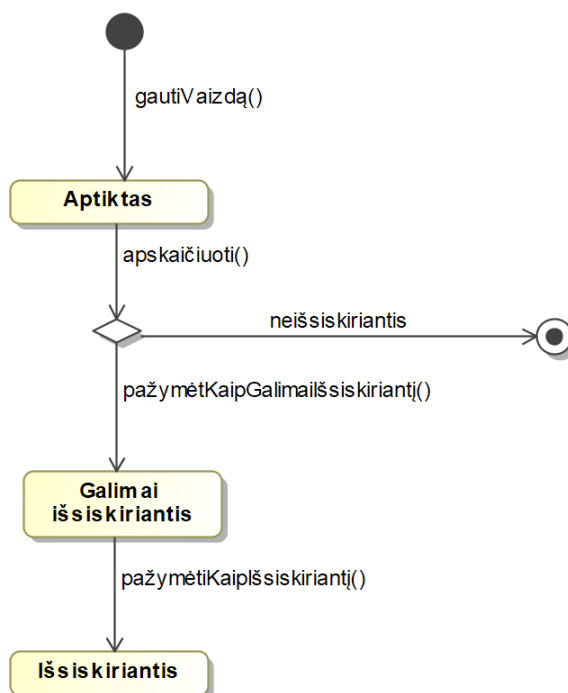


2.26 pav. PA „Koreguoti vaizdo apdorojimo parametrus“ sąveikos diagrama

Naudotojui pakoregavus vaizdo apdorojimo parametrus, vaizdo atvaizdavimo langas juos siunčia sistemos valdikliui, kuris išsaugo jų reikšmes.

2.3.3. Vairavimo atvejo būsenų diagrama

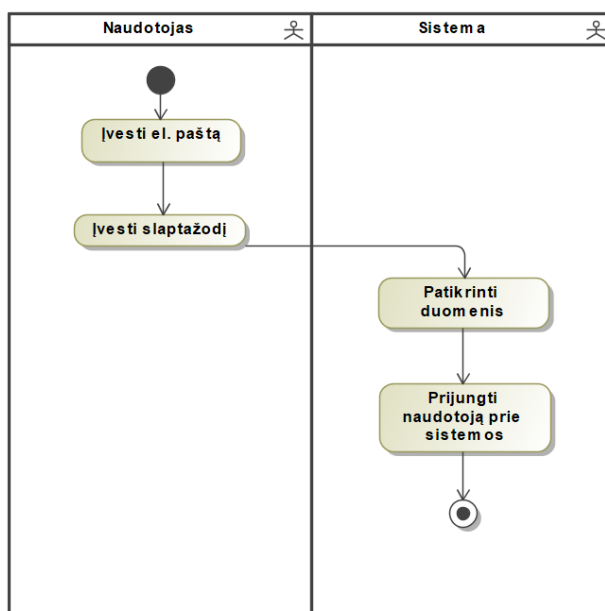
Pirmas žingsnis – vaizdo nuskaitymas. Jame yra aptinkamas vairavimo atvejis. Sistema paskaičiuoja jo išskirtinumą. Jei vairavimo atvejis yra neišsiskiriantis, jis yra ignoruojamas. Jei vairavimo atvejis yra išsiskiriantis, jis pažymimas kaip „Galimai išsiskiriantis“. Naudotojas gali pakeisti vairavimo atvejo statusą į „Išsiskiriantis“. Detaliau vairavimo atvejo būsenų diagrama vaizduojama 2.27 paveiksle.



2.27 pav. Vairavimo atvejo būsenų diagrama

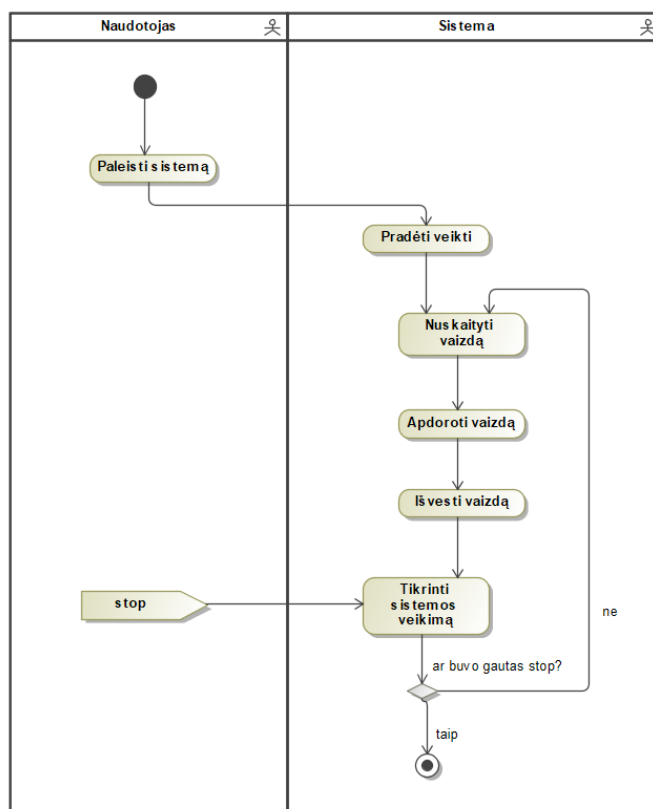
2.3.4. Sistemos veiklos diagramos

Sistemos ir jos naudotojų atliekami veiksmai atvaizduoti veiklos diagramomis (2.28-2.37 pav.).



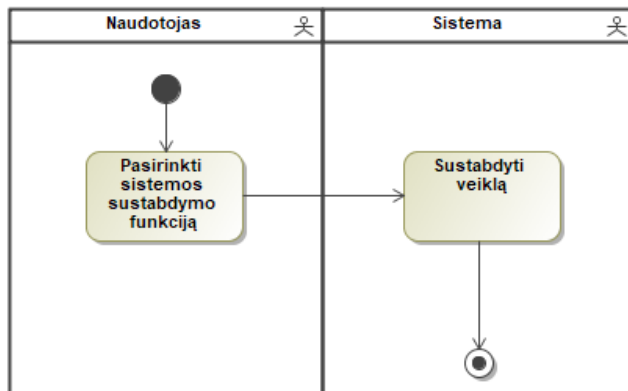
2.28 pav. PA „Prisijungti“ veiklos diagrama

Naudotojas, norėdamas prisijungti prie sistemos, įveda savo el. paštą ir slaptažodį. Sistema patikrina naudotojo įvestus duomenis ir prijungia jį prie sistemos.



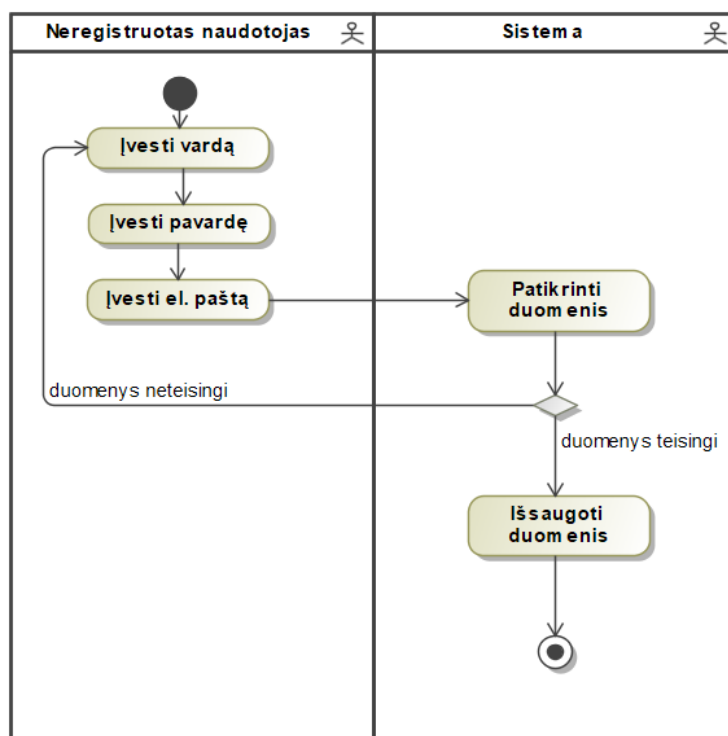
2.29 pav. PA „Paleisti aptikimo sistemą“ veiklos diagrama

Naudotojas pasirenka sistemos paleidimo funkciją. Sistema pradeda vaizdo nuskaitymą. Vaizdas yra apdorojamas ir išvedamas naudotojui. Naudotojui pasirinkus sistemos sustabdymo funkciją, jos veikimas yra sustabdomas.



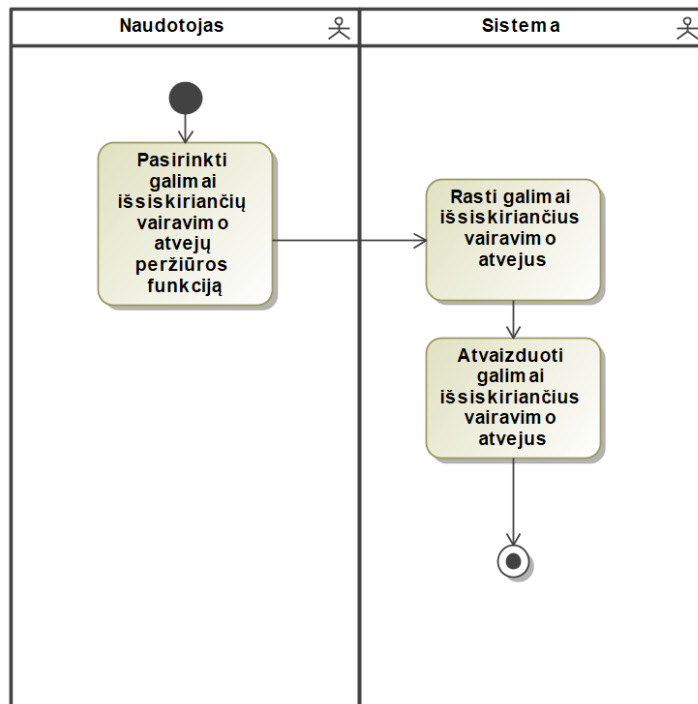
2.30 pav. PA „Sustabdyti aptikimo sistemą“ veiklos diagrama

Naudotojui pasirinkus sistemos sustabdymo funkciją, ji yra sustabdoma.



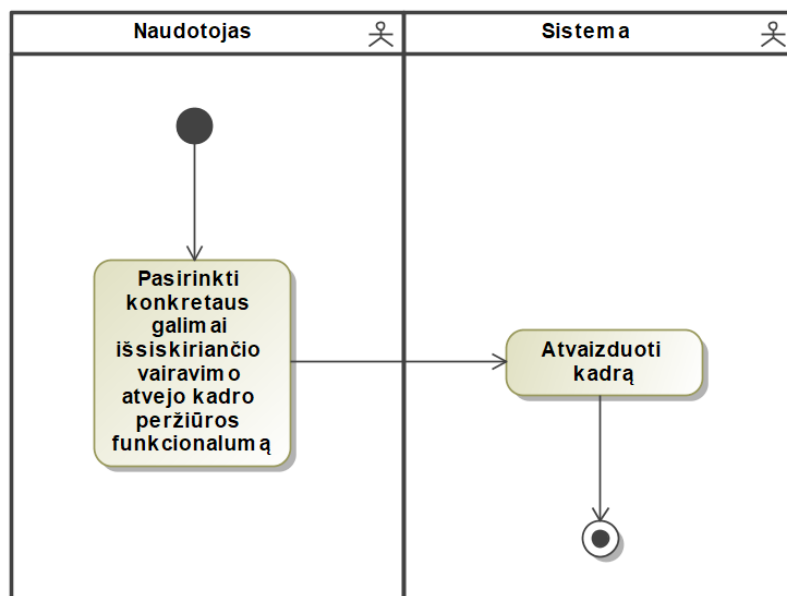
2.31 pav. PA „Užsiregistruoti“ veiklos diagrama

Neregistruotas naudotojas įveda savo vardą, pavardę ir el. pašta. Sistema patikrina naudotojo duomenis. Jei duomenys yra teisingi, sistema juos išsaugo.



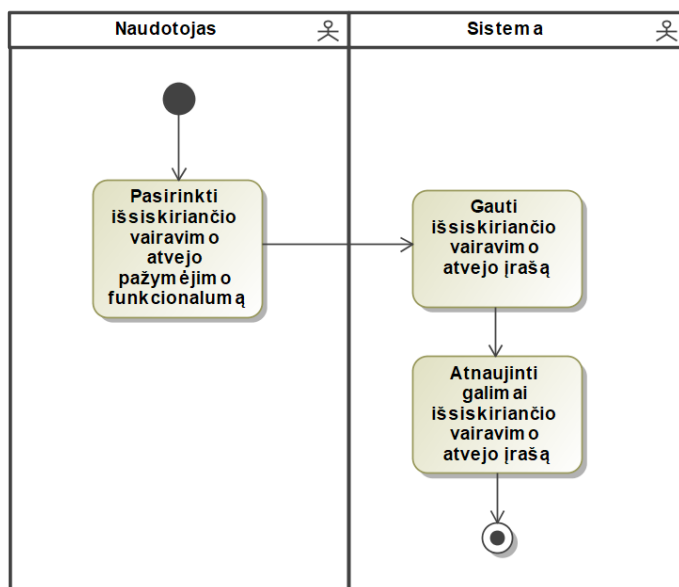
2.32 pav. PA „Peržiūrėti informaciją apie išsiskyrusius vairavimo atvejus“ veiklos diagrama

Naudotojui pasirinkus galimai išsiskiriančių vairavimo atvejų peržiūros funkciją, sistema juos randa ir atvaizduoja.



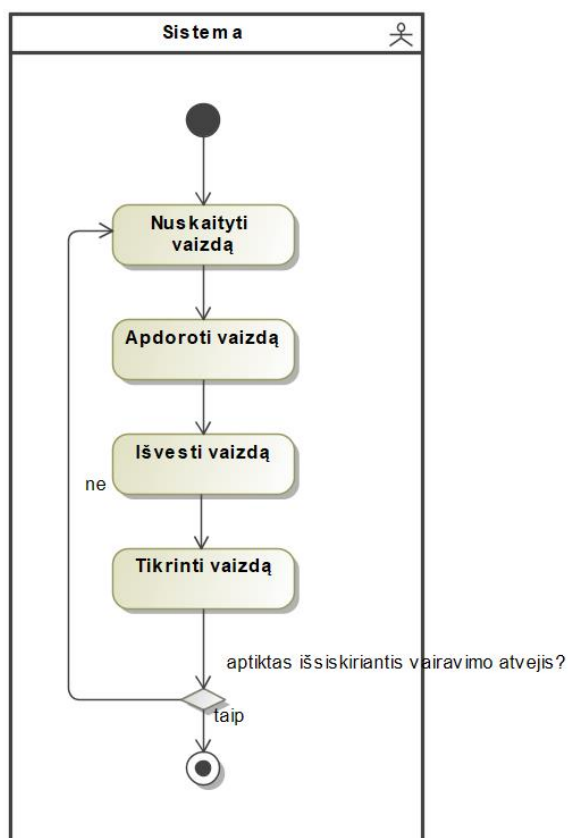
2.33 pav. PA „Peržiūrėti užfiksuoto išsiskiriančio vairavimo atvejo kadra“ veiklos diagrama

Naudotojui pasirinkus konkretaus galimai išsiskiriančio vairavimo atvejo kadro peržiūros funkcionalumą, kadras atvaizduojamas pilno ekrano režime.



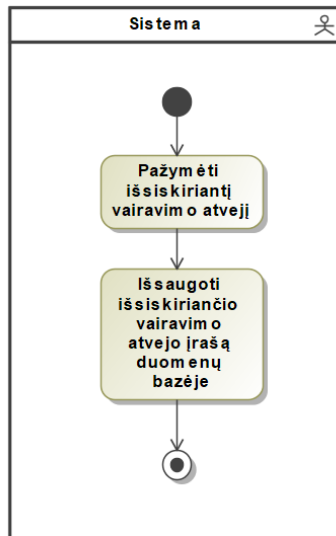
2.34 pav. PA „Pažymėti vairavimą kaip išsiskiriantį“ veiklos diagrama

Kai naudotojas pažymi vairavimo atvejį kaip išsiskiriantį, sistema gauna to atvejo įrašą ir jį atnaujina.



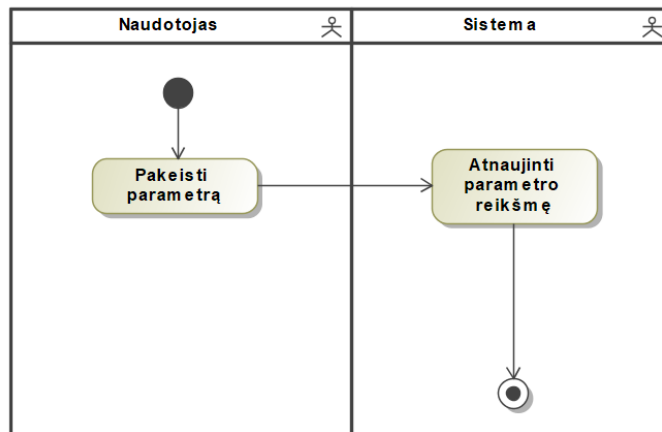
2.35 pav. PA „Apdoroti vaizdą“ veiklos diagrama

Kai aptikimo sistema yra paleista, sistema nuskaityto vaizdą, jį apdoroja ir ieško išsiskiriančių vairavimo atvejų.



2.36 pav. PA „Išsaugoti informaciją apie galimai išsiskiriantį vairavimo atvejį“ veiklos diagrama

Aptikus išsiskiriantį vairavimo atvejį, sistema pažymi jį vaizde ir išsaugo jo informaciją duomenų bazėje.

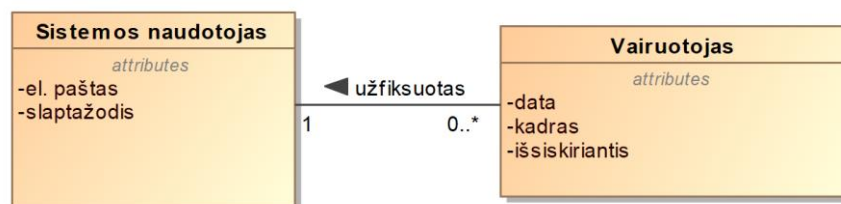


2.37 pav. PA „Koreguoti vaizdo apdorojimo parametrus“ veiklos diagrama

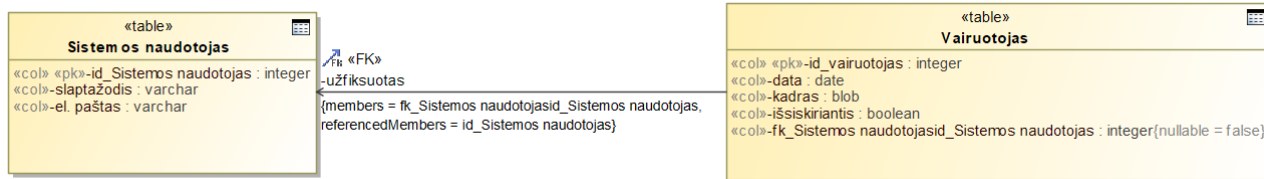
Naudotojui pakeitus vaizdo apdorojimo parametą, sistema atnaujina jo reikšmę.

2.4. Duomenų vaizdas

Duomenų vaizdas pateikiamas ER modelio (2.38 pav.) ir duomenų bazės modelio (2.39 pav.) diagramomis. Sistemoje yra dvi esybės – sistemos naudotojas ir vairuotojas.



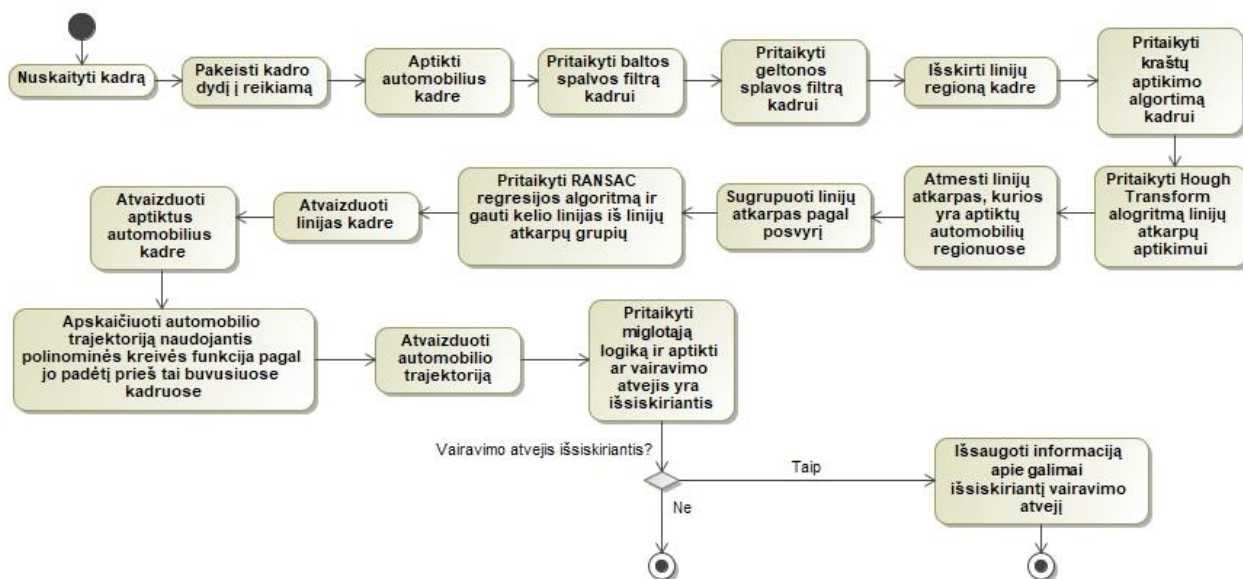
2.38 pav. Sistemos ER modelis



2.39 pav. Sistemos duomenų bazės modelis

2.5. Vaizdo apdorojimo algoritmo veiksmų diagrama

Vaizdo apdorojimo algoritmo veiksmų diagrama pateikiama 2.40 paveiksle. Sistema nuskaityto kadra ir pakeičia jo dydį į reikiamą. Sistema aptinka kadre esančius automobilius. Siekiant išgauti eismo linijas, sistema kadrai pritaiko baltos ir geltonos spalvų filtrus, išskiria linijų regioną, pritaiko kraštų aptikimo algoritmą ir Hough transformaciją. Atmetamos tos linijos, kurios yra aptiktų automobilių regionuose. Linijos yra sugrupuojamos ir grupėms pritaikomas RANSAC regresijos algoritmas. Gautos linijos atvaizduojamos kadre kartu su aptiktais automobiliais. Naudojantis polinominės kreivės funkcija, apskaičiuojama automobilio trajektorija. Pritaikius miglotąją logiką, aptinkami išsiskiriantys vairavimo atvejai ir jų informacija išsaugoma duomenų bazėje.



2.40 pav. Vaizdo apdorojimo algoritmo veiksmų diagrama

3. Tyrimo dalis

3.1. Tyrimo tikslas

Tyrimo tikslas – nustatyti, kokie vaizdo įrašo parametrai gali būti naudojami, siekiant aptikti išsiskiriančius vairavimo atvejus, rasti metodus, skirtus juos išgauti, ir sukurti algoritmą, aptinkantį išsiskiriančius vairavimo atvejus pagal parametrus, gautus iš vaizdo įrašo.

3.2. Tyrimo aprašymas

Siekiant nustatyti ar vairavimo atvejis yra išsiskiriantis, reikia išgauti vairavimo atvejo parametrus. Vėliau, panaudojant miglotąją logiką, pagal tuos parametrus bus klasifikuojamas kiekvienas vairavimo atvejis atskirai.

Šiame darbe, vairavimo atvejui įvertinti, bus naudojami šie parametrai:

- automobilio trajektorija;
- automobilio padėtis eismo juostų atžvilgiu.

Norint gauti šiuos parametrus reikia atlikti šiuos veiksmus:

- aptikti automobilį;
- apskaičiuoti automobilio trajektoriją;
- aptikti eismo juostas.

3.3. Automobilių aptikimas

Norint apskaičiuoti automobilio trajektoriją, reikia ją atpažinti ir nustatyti jo padėtį vaizdo kadre. Turint informaciją apie kadre aptiktus automobilius ir ją susiejus su prieš tai apdorotų kadru informacija, bus galima apskaičiuoti automobilių trajektoriją.

3.3.1. Modelio pasirinkimas

Automobiliams atpažinti pasirinkta „Tensorflow“ biblioteka. Kaip jau minėta, toks sprendimas buvo priimtas todėl, kad ji pasižymi geresne greitimeika nei „Keras“ biblioteka.

Apmokymui naudojamas modelis buvo pasirinktas iš „TensorFlow 2 Object Detection API Model Zoo“ modelių rinkinio. Šiame rinkinyje pateikiami iš anksto „COCO 2017“ duomenų rinkiniu apmokyti modeliai. Buvo nuspręsta rinktis modelį, kuris yra iš anksto dalinai apmokytas, o ne kurti savo modelį nuo nulio. Kuriant modelį nuo nulio, jo konfigūravimas ir apmokymas gali užtrukti labai daug laiko. Taip pat šiame rinkinyje pateikiami modeliai yra pilnai konfigūruojami, juos galima naudoti kaip šabloną ir mokymą pradėti be jokio pradinio taško.

Renkantis modelį buvo atsižvelgiama į šias savybes:

- Ar modelį galima apmokyti su skirtingo dydžio paveikslėliais?
- Kokia yra apmokyto modelio greitimeika?
- Koks yra apmokyto modelio tikslumas?

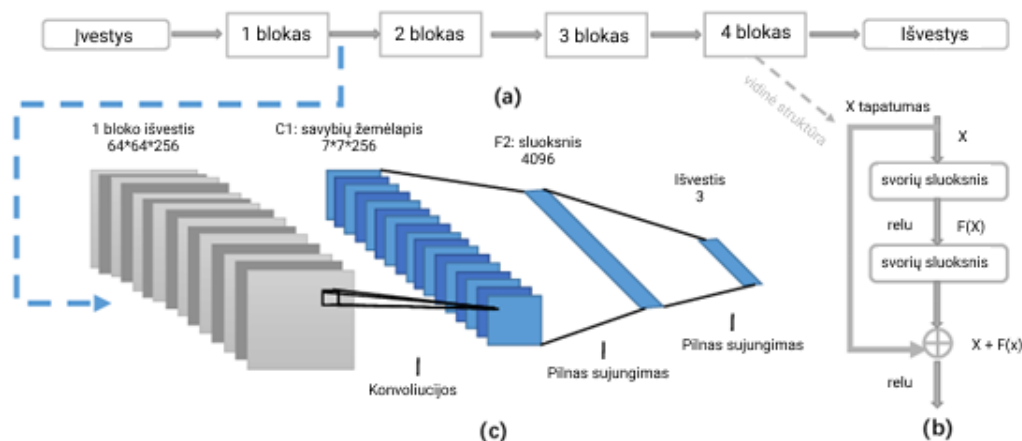
Modelių savybės aprašomos 3.1 lentelėje [31]:

3.1 lentelė Modelių savybių lentelė

Modelis	Greitaveika (ms)	Skirtingo dydžio paveikslėliai	Tikslumas (COCO mAP)
SSD MobileNet V1 FPN 640x640	48	Ne	29,1
SSD MobileNet V2 FPNLite 640x640	38	Ne	28,2
SSD ResNet50 V1 FPN 640x640 (RetinaNet50)	46	Ne	34,3
SSD ResNet101 V1 FPN 640x640 (RetinaNet101)	57	Ne	35,6
Faster R-CNN ResNet50 V1 640x640	53	Taip	29,3
Faster R-CNN ResNet101 V1 640x640	55	Taip	31,8
Faster R-CNN ResNet152 V1 640x640	64	Taip	32,4
Faster R-CNN Inception ResNet V2 640x640	206	Taip	206

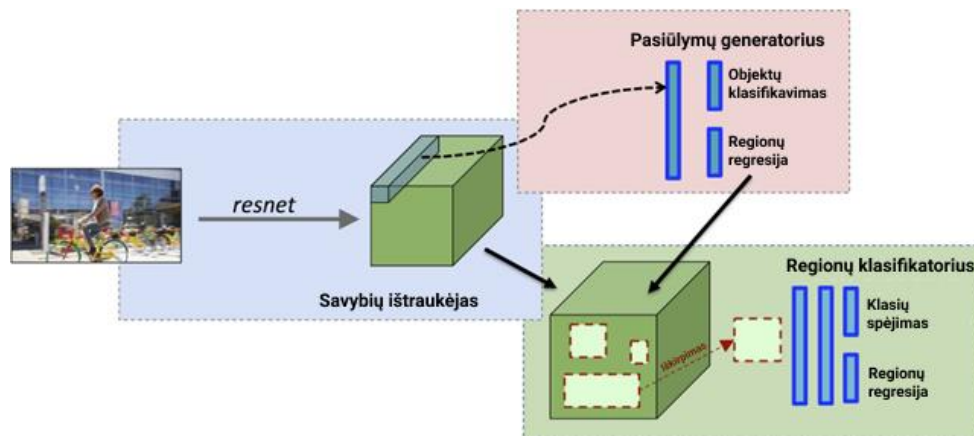
Atsižvelgus į parametrus buvo nuspręsta naudoti „Tensorflow Object Detection API Faster R-CNN ResNet101“ modelį.

Jis susideda iš „ResNet101“ savybių ištraukimo modelio ir „Faster R-CNN“ tinklo architektūros. „ResNet 101“ architektūra vaizduojama 3.1 paveiksle. [32].



3.1 pav. „ResNet 101“ architektūra

Iš „ResNet“ gautos savybės perduodamos pasiūlymų generatoriui (*Proposal generator*), kuris sugeneruoja regionus. Regionai kartu su savybėmis perduodami regionų klasifikatoriui (*Box classifier*), kuris atlieką klasių spėjimą ir regionų regresiją. „Faster R-CNN“ architektūra vaizduojama 3.2 paveiksle. [33].



3.2 pav. „Faster R-CNN“ architektūra

3.3.2. Duomenų rinkinys

Norint apmokyti modelį, reikėjo paruošti duomenų rinkinį. Duomenyse reikėjo sužymėti automobilius.

Pirmiausia reikėjo susirasti daugybę automobilių nuotraukų. Buvo rasti keli automobilių nuotraukų rinkiniai:

- „Stanford Cars Dataset“ [34];
- „GTI Vehicle Image Database“ [35];
- „BDD100K dataset“ [36].

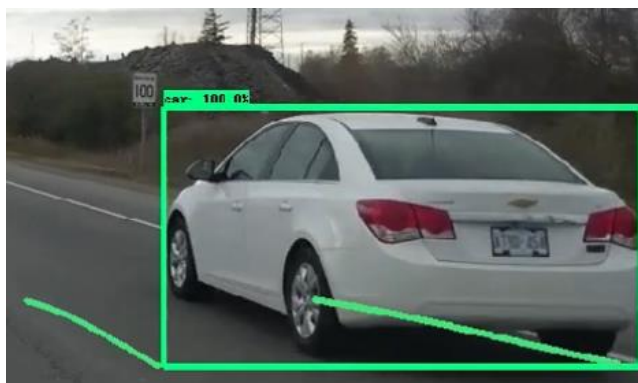
„Stanford Cars Dataset“ duomenų rinkinys turėjo daugybę automobilių nuotraukų, tačiau jos buvo iš įvairiausių kampų, dažniausiai automobiliai nufotografuoti stovintys ar baltame fone.

„GTI Vehicle Image Database“ duomenų rinkinys susideda iš daugybės automobilių galinių dalių nuotraukų, tačiau nuotraukos yra mažos, jose nesimato aplinkos.

„BDD100K dataset“ duomenų rinkinys susideda iš daugybės eismo nuotraukų. Jame automobiliai nufotografuoti dalyvaujantys eisme, matosi visa aplinka. Būtent šis rinkinys ir buvo pasirinktas, nes labiausiai tiko norimai išspręsti problemai.

Sekantis etapas – automobilių sužymėjimas nuotraukose, skirtose modeliui apmokyti. Žymėjimas vykdytas pasitelkiant pusiau automatizuotą įrankį „CVAT“.

Pirmo bandymo metu nuotraukose buvo apvedamas visas automobilis. Apmokius modelį ir realizavus trajektorijos apskaičiavimą pastebėta, kad šis būdas turi trūkumų. Trajektorijos apskaičiavimas veikė tik tiems automobiliams, kurie vaizde yra nutolę, o tiems kurie yra arti trajektorijos skaičiavimas veikė netiksliai. Taip atsitiko dėl to, kad arti esantis automobilis yra pasisukęs šonu. Gautas rezultatas vaizduojamas 3.3 paveiksle.



3.3 pav. Trajektorijos skaičiavimas, pažymėjus visą automobilį

Siekiant sumažinti trajektorijos netikslumus, nuspręsta apmokant žymėti tik priekines ir galines automobilių dalis. Toks metodas leido pagerinti trajektorijos apskaičiavimo tikslumą. Gautas rezultatas vaizduojamas 3.4 paveiksle.



3.4 pav. Trajektorijos skaičiavimas, pažymėjus automobilio priekinę ir galinę dalį

3.3.3. Modelio apmokymas

Modeliui apmokyti sužymėta 2000 nuotraukų. Testavimui paruošta dar 300 nuotraukų.

Tinklo konfigūracija naudota tokia pati kaip modelyje, gautame iš „TensorFlow 2 Object Detection Model Zoo“. Atlikti tik keli nežymūs pakeitimai. Pakeista aptinkamų klasių vertė į vienetą. Taip pat pakeistas nuotraukų matmenų santykis. Minimalus matmuo – 240 pikselių, maksimalus – 1280 pikselių. Taip pat išjungtas nuotraukų išplėtimas iki maksimalių matmenų.

Modelio apmokymas užtruko 500 epochų. Pasirinktas kintamas mokymosi greitis. Mokymosi greičio bazė – 0,001. Siekiant išvengti persimokymo pirmosiose epochose, pirmos dvi epochos buvo skirtos apšilimui. Pradinė apšilimo mokymosi greičio vertė – 0,000013333. Aktyvacijai naudota „Softmax“ funkcija. Taip pat buvo atliekama L2 reguliarizacija.

Modelio tikslumo įvertinimui naudotos „COCO“ aptikimo metrikos.

Apmokius modelį, pastebėta, kad jis persimoko. Siekiant išvengti persimokymo, panaudotos duomenų augmentacijos:

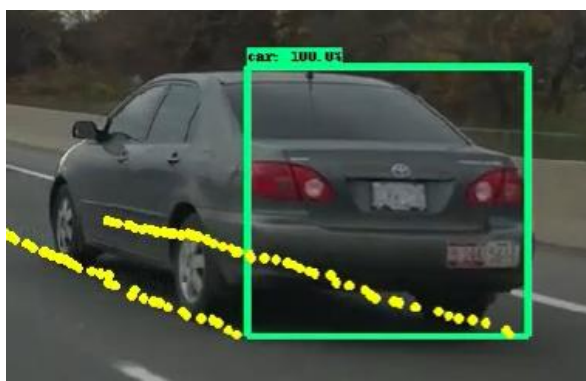
- atsitiktinis horizontalus nuotraukos apvertimas;
- atsitiktinis nuotraukos dydžio pakeitimas;
- atsitiktinis nuotraukos ryškumo pakeitimas;
- atsitiktinis nuotraukos perstūmimas.

3.4. Trajektorijų skaičiavimas

Apmokytas modelis gražina vaizdo kadre aptiktus automobilius, jų vietą kadre, aptikimo tikslumą. Aptiktu laikomas automobilis, kurio aptikimo tikslumas didesnis nei 90 %. Kiekvieno kadro automobilių padėties taškai saugomi masyve.

3.4.1. Automobilių padėtis laike

Kiekvieno automobilio trajektorijai identifikuoti naudojami jo aptikimo kvadrato apatiniai, kraštiniai taškai. Kadre aptikus automobilį, naudojantis $k-d$ medžio algoritmu, aptinkami artimiausi, prieš tai buvusiuose kadruose užfiksuoti taškai. Jei taškas yra arčiau nei per 30 pikselių, jis laikomas to automobilio tašku, prieš tai buvusiuose kadruose. Taip gaunamas automobilio padėties kitimas laiko atžvilgiu. Kartais automobilis gali būti neužfiksuojamas kai kuriuose kadruose. Todėl neradus artimų taškų prieš tai buvusiame kadre, nagrinėjamas dar ankstesnis. Jei paskutiniuose 5 kadruose nerandamas joks artimas taškas, automobilis laikomas naujai užfiksuotu. Rezultatas vaizduojamas 3.5 paveiksle.



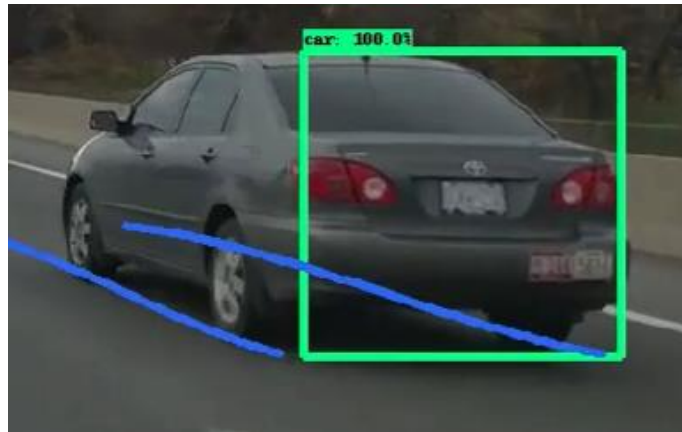
3.5 pav. Automobilio padėtis laike

3.4.2. Polinominės kreivės interpoliacija

Kiekvienam aptiktam automobiliui turimi du apatinių taškų rinkiniai – kairės pusės ir dešinės pusės.

Kiekvienam rinkiniui atskirai pritaikoma 3 laipsnio polinominės kreivės funkcija. Interpoliacijai naudojamas mažiausių kvadratų metodas. Pirmojo laipsnio polinominė funkcija yra paprasta tiesė. Antrojo laipsnio – kreivė, turinti tik vieną ekstremumą. Automobilio trajektorija gali būti įvairių formų, todėl buvo pasirinkta trečiojo laipsnio kreivė, kuri gali turėti kelis ekstremumus.

Gavus polinominės kreivės funkciją, ji pritaikoma kiekvienai x reikšmei ir taip gaunama automobilio trajektorija per tam tikrą laiko tarpą. Rezultatas vaizduojamas 3.6 paveiksle.



3.6 pav. Automobilio trajektorija per laiko tarpą

3.5. Linijų aptikimas

Norint įvertinti automobilio trajektoriją, reikia atskaitos taško. Tokiu atskaitos tašku pasirinkta naudoti eismo linijas. Jos egzistuoja didžiojoje dalyje kelių. Eismo linijos gali būti įvairiausios. Jos gali būti geltonos arba baltos, ištiesos arba su pertraukimais.

Linijų aptikimo algoritmas pateikiamas pseudo-kodu:

Įvestys: kadras, baltos spalvos vertė, geltonos spalvos vertė, slenkstinė reikšmė

Išvestys: aptiktos linijos

Linijų aptikimas (kadras, baltos spalvos vertė, geltonos spalvos vertė, slenkstinė reikšmė):

Aptiktos linijos = []

Pilkas kadras = Pritaikyti pilkos spalvos filtrą (Kadras)

HSV kadras = Konvertuoti į HSV erdvę (Kadras)

Baltos spalvos kadras = Baltos spalvos kaukė (Pilkas kadras, baltos spalvos vertė)

Geltonos spalvos kadras = Geltonos spalvos kaukė (HSV kadras, geltonos spalvos vertė)

Apdorotas kadras = Sujungti (Baltos spalvos kadras, Geltonos spalvos kadras)

Gauso suliejimas (Apdorotas kadras)

Išskirti eismo linijų regioną (Apdorotas kadras)

Slenkstinės reikšmės išskyrimas (Apdorotas kadras, slenkstinė reikšmė)

Canny kraštų aptikimo algoritmas (Apdorotas kadras)

Linijos = Hough transformacija (Apdorotas kadras)

Sugrupuotos linijų atkarpos = Grupuoti pagal krypties koeficientą (Linijos)

for Linijų atkarpų grupė **in** Sugrupuotos linijų atkarpos:

 Tiesinė linijos funkcija = Ransac regresija (Linijų atkarpų grupė)

 Linija = Interpoliuoti (Tiesinė linijos funkcija, x reikšmės)

 Aptiktos linijos.Pridėti (Linija)

return Aptiktos linijos

Toliau detaliam aprašomas linijų aptikimo algoritmas.

Norint kadre aptikti eismo linijas, reikia atlikti šiuos veiksmus:

- apdoroti kadra;
- atlikti Hough transformaciją;
- atlikti Ransac regresiją.

Pirmasis linijų aptikimo žingsnis – kadro paruošimas.

Linijos gali būti geltonos arba baltos. Kadre reikia aptikti šių spalvų regionus.

Siekiant aptikti baltą spalvą, kadrui pritaikytas pilkos spalvos filtras (3.7 pav.)



3.7 pav. Pilkos spalvos filtras

Siekiant aptikti geltonos spalvos regionus, kadras konvertuojamas į HSV spalvų erdvę (3.8 pav.).

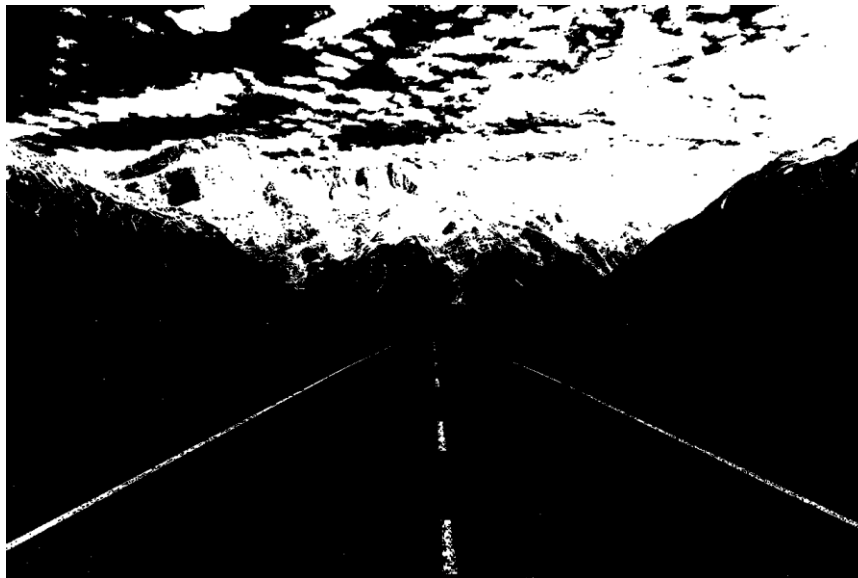


3.8 pav. HSV spalvų erdvė

Pilkos spalvos kadru pritaikome baltos spalvos kaukę, o HSV kadru geltonos spalvos kaukę.

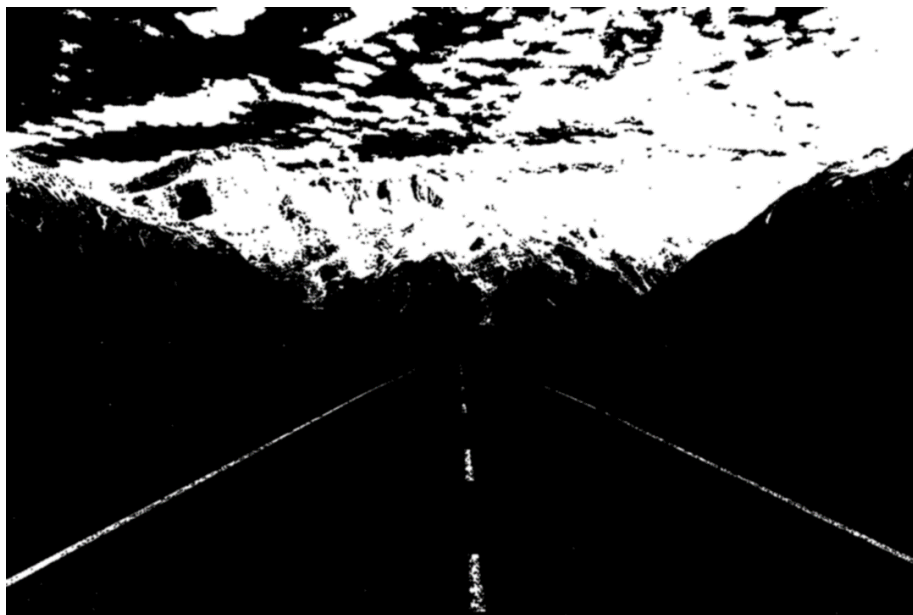
Kadangi skirtinguose vaizdo įrašuose geltonos ir baltos spalvų reikšmės gali skirtis, apatines spalvų vertės ribas paliekame konfigūruojamas.

Gautus geltonos ir baltos spalvos kadrus sujungiame (3.9 pav.).



3.9 pav. Kadre aptiktos geltonos ir baltos spalvos

Siekdami pašalinti pašalinį triukšmą, nuotraukai pritaikome Gauso suliejimo algoritmą (3.10 pav.).



3.10 pav. Gauso suliejimas

Kadangi eismo linijos būna tik vaizdo perspektyvos apačioje, išskiriame kadre linijų regioną, kitus kadro elementus panaikiname (3.11 pav.).



3.11 pav. Eismo linijų regionas

Sekantis žingsnis – slenkstinės reikšmės nustatymas. Visus taškus, kurie viršija slenkstinę reikšmę, padarome baltus, o kurie jos neviršija – juodus (3.12 pav.).



3.12 pav. Slenkstinių reikšmių atskyrimas

Gautam kadrai pritaikome Canny kraštų aptikimo algoritmą (3.13 pav.). Tai atliekama siekiant kadre aptikti esminius taškus ir sumažinti duomenų kiekį, kurį reikės apdoroti.



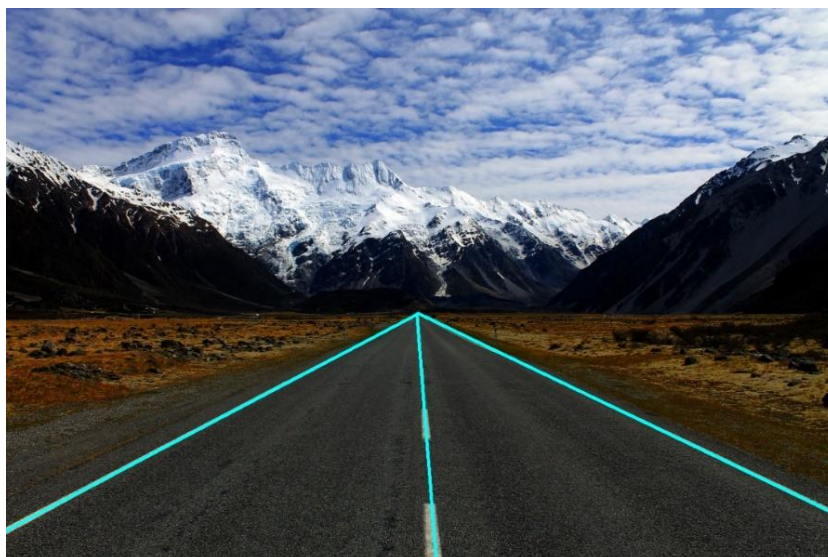
3.13 pav. Canny kraštų aptikimo algoritmas

Kadru pritaikome Hough transformacijos algoritmą ir gauname kadre esančias tiesias linijas. Aptiktų linijų taškus išsaugome masyve (3.14 pav.).



3.14 pav. Kadro linijų taškai

Kadro linijas sugrupuojame pagal krypties koeficientą. Taip gauname skirtingų linijų atkarpų masyvus. Kiekvienam masyvui pritaikome Ransac regresijos algoritmą. Gaunama tiesinė linijos funkcija. Kiekvienai x vertei paskaičiuojame reikšmę ir taip gauname kelio liniją (3.15 pav.)



3.15 pav. Aptiktos kelio linijos

Realizavus linijų aptikimo algoritmą, pastebėta, kad prastos kokybės vaizdo įrašuose dažnai pasitaiko daug pašalinio triukšmo. Dėl triukšmo kartais aptinkamos linijos, kurios neegzistuoja. Balti regionai pasitaiko ant automobilių, kelio ženkluose ar kituose kelio objektuose. Todėl buvo imtasi papildomų veiksmų, siekiant pašalinti linijos aptikimų netikslumus.

Vienas iš veiksmų – automobilių zonoje aptiktų baltos spalvos reikšmių pašalinimas. Apmokytas modelis gražina automobilio lokaciją kadre. Šios vietos ignoruojamos aptinkant eismo linijas.

Siekiant išvengti dėl pašalinio triukšmo atsirandančių linijų, nuspręsta saugoti aptiktas linijas, susiejant jas su kadro indeksu. Aptikus naują liniją, jos padėtis ir krypties koeficientas buvo lyginami su anksčiau aptiktomis linijomis. Jei ankstesniuose kadruose aptinkamos bent 5 panašios linijos, linija laikoma tikra.

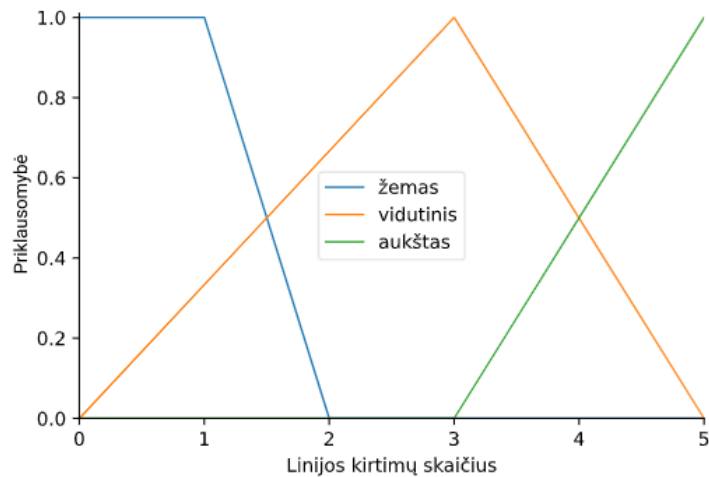
3.6. Miglotoji logika

Kaip jau minėta, vaizdo įrašai gali būti įvairiausios kokybės, tie patys parametrai juose gali turėti skirtingas reikšmes. Siekiant išvengti priklausomybės nuo vaizdo įrašų, išskirtinumui aptikti nuspręsta naudoti miglotąją logiką. Taip pat miglotoji logika padeda pašalinti sprendimus, kurie yra priimti dėl neteisingų parametrų.

3.6.1. Parametrai

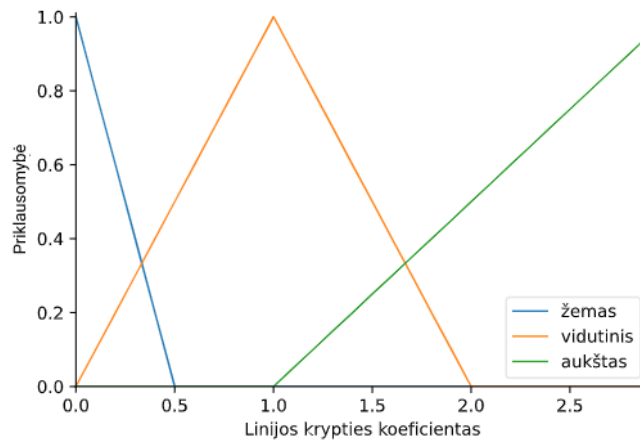
Iš vaizdo įrašo išgauti du elementai – automobilio trajektorija ir eismo linijos. Iš šių elementų buvo išskirti parametrai, kurie panaudoti miglotoje logikoje.

Linijos kirtimų skaičius – parametras, kuris nurodo kiek kartų automobilis kirto eismo juostą. Automobiliai gali persirikiuoti iš vienos juostos į kitą. Tokiu atveju jie kerta eismo liniją, tačiau tik vieną ar du kartus. Toks linijos kirtimų skaičius laikomas žemu. Jei vairuotojas yra neblaivus, pervargęs ar važiuoja chuliganiškai, jis gali nepaisyti eismo juostų, važiuoti ant jų, rikiuotis iš juostos į juostą be perstojo. Taip jis liniją kirs daug kartų. Plačiau linijos kirtimų skaičiaus priklausomybė pavaizduota 3.16 paveiksle.



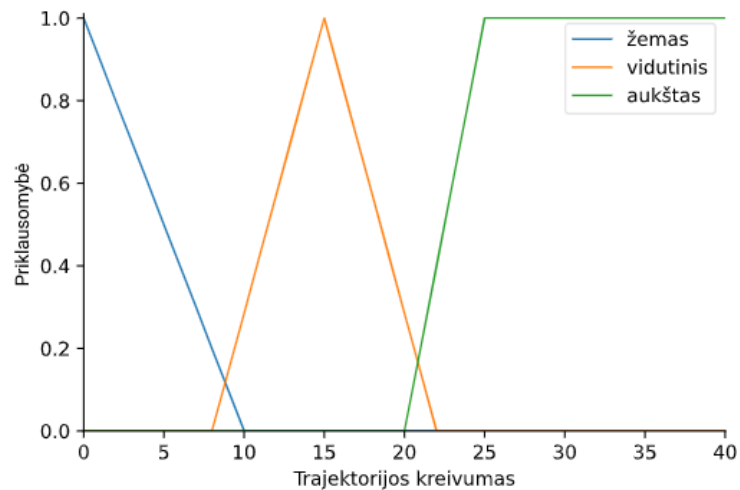
3.16 pav. Linijos kirtimų skaičius

Linijos krypties koeficientas – parametras, skirtas įvertinti aptiktas kelio linijas. Kaip jau minėta, dėl prastos vaizdo įrašų kokybės ir juose esančio triukšmo, eismo juostų aptikimas gali gražinti klaidingus rezultatus. Šis parametras padeda ignoruoti neteisingai aptiktas linijas. Iš vairuotojo perspektyvos eismo linijos yra labiau vertikalios nei horizontalios ir šiek tiek pakrypusios į vairuotojo perspektyvos centrą. Horizontalių linijų krypties koeficientas yra žemas (iki 0,5). Tokios linijos yra ignoruojamos. Miglotajai logikai naudojama absoliuti krypties koeficiento reikšmė. Plačiau linijos krypties koeficiento priklausomybė pavaizduota 3.17 paveiksle.



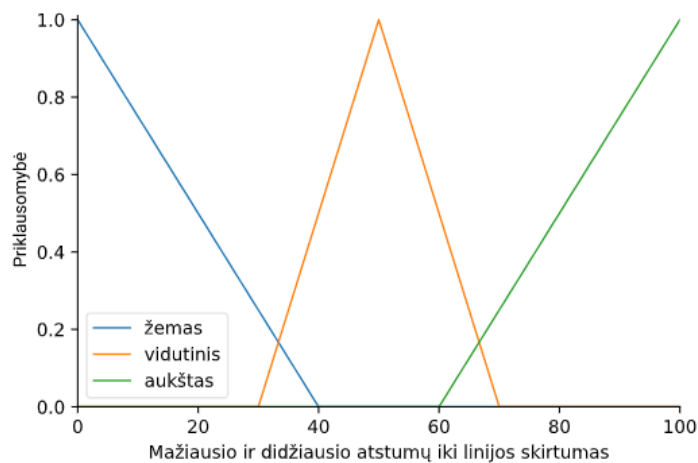
3.17 pav. Linijos krypties koeficientas

Trajektorijos kreivumas – parametras, skirtas įvertinti ar vairuotojas važiuoja tiesiai. Jis apskaičiuojamas vertinant trajektorijos ekstremumų atstumą iki tiesės, nubrėžtos per trajektorijos pradžią ir pabaigą. Imami labiausiai nutolę ekstremumai į abi kryptis. Jų atstumo iki linijos reikšmės susumuojamos. Trajektorijos kreivumo priklausomybė vaizduojama 3.18 paveiksle.



3.18 pav. Trajektorijos kreivumas

Mažiausio ir didžiausio atstumų iki linijos skirtumas – parametras, skirtas įvertinti, kaip automobilis važiuoja eismo juostoje. Šis parametras įvertina ar automobilis važiuoja pastoviu atstumu nuo kelio linijos ar po ją vinguriuoja. Linijos kirtimų skaičius skirtas įvertinti neblaivaus ar pervargusio vairuotoją vairavimą, kai vairuotojas kerta kelio linijas. Tačiau vairuotojas gali vinguriuoti po kelią, tačiau nekirsti eismo linijų. Tokiam vairavimui įvertinti ir yra skirtas mažiausio ir didžiausio atstumų iki linijos skirtumo parametras. Randami tolimiausias ir artimiausias trajektorijos taškai nuo per visą kadrą pratęstos linijos. Apskaičiuojamas šių taškų nuotolis nuo eismo linijos ir randamas jų skirtumas. Tai ir yra šio parametro vertė. Plačiau parametro priklausomybė pavaizduota 3.19 paveiksle.



3.19 pav. Mažiausio ir didžiausio atstumų iki linijos skirtumas

3.6.2. Taisyklės

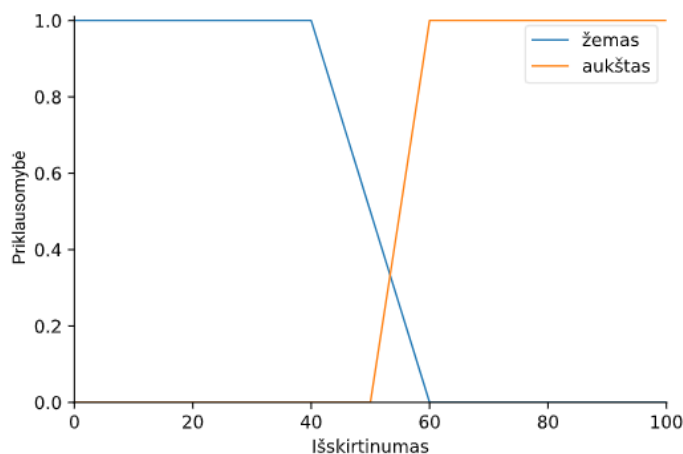
Vairavimo atvejais gali būti išsiskiriantis arba neišsiskiriantis. Siekiant tai nustatyti, buvo apibrėžtos taisyklės. Taip pat taisyklės naudotos ir siekiant pašalinti neteisingus parametrus, kai linijos yra aptiktos neteisingai. Taisyklės aprašomos 3.2 lentelėje.

3.2 lentelė Miglotosios logikos taisyklės

Linijos kirtimų skaičius	Linijos krypties koeficientas	Trajektorijos kreivumas	Mažiausio ir didžiausio atstumų iki linijos skirtumas	Išskirtinumas
-	-	Aukštas	-	Aukštas
-	Ne žemas	-	Aukštas	Aukštas
Aukštas	Ne žemas	-	-	Aukštas
Vidutinis	Ne žemas	-	Vidutinis	Aukštas
Vidutinis	Ne žemas	Vidutinis	-	Aukštas
-	Žemas	-	Aukštas	Žemas
Žemas	-	Žemas	Žemas	Žemas
Vidutinis	Ne žemas	Žemas	Ne žemas	Žemas
Aukštas	Žemas	-	-	Žemas
-	Žemas	-	Aukštas	Žemas
-	-	Žemas	Vidutinis	Žemas
Žemas	-	Vidutinis	-	Žemas
Žemas	-	Žemas	Aukštas	Žemas

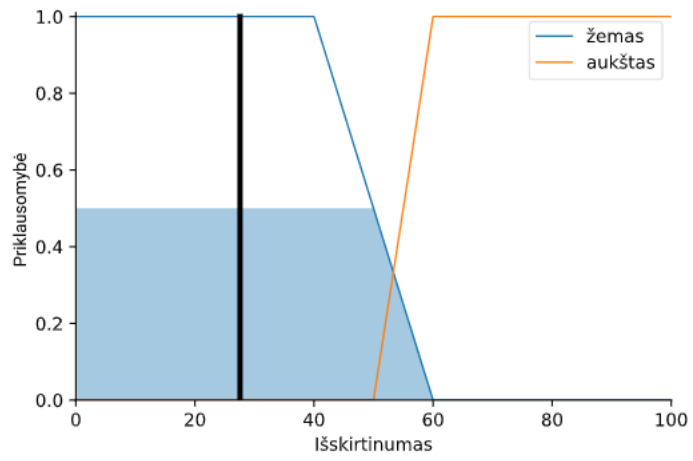
3.6.3. Išvestis

Miglotoji logika, apdorojusi įvestis pagal taisykles, įvertina vairavimo išskirtinumą. Vairavimo išskirtinumas gali būti aukštas arba žemas (3.20 pav.).

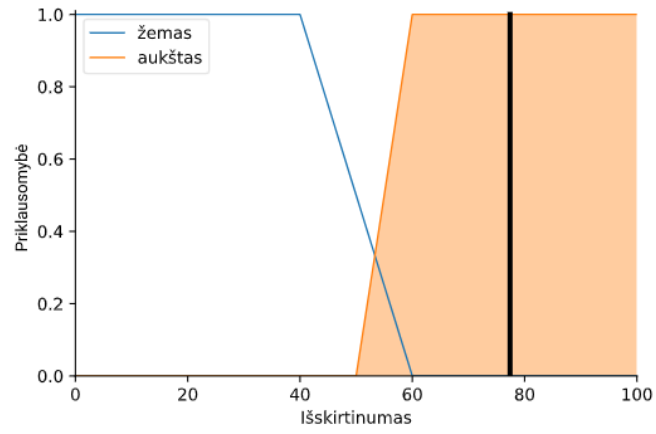


3.20 pav. Vairavimo išskirtinumas

Miglotoji logika gražina procentinę išskirtinumo reikšmę. Jei išskirtinumas viršija 60 procentų, vairavimo atvejis laikomas išskirtiniu (3.22 pav.), jei neviršija laikomas neišskirtiniu (3.21 pav.).



3.21 pav. Žemas išskirtinumas



3.22 pav. Aukštas išskirtinumas

4. Eksperimentinė dalis

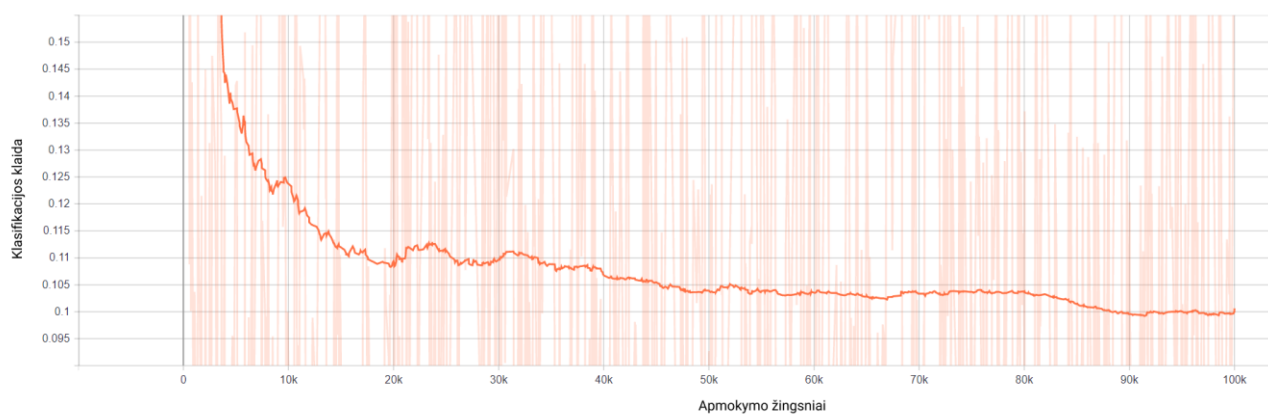
Šioje dalyje nagrinėjamos sukurto algoritmo techninės charakteristikos.

4.1. Automobilių aptikimo modelio tikslumas

Modelio apmokymo parametrai:

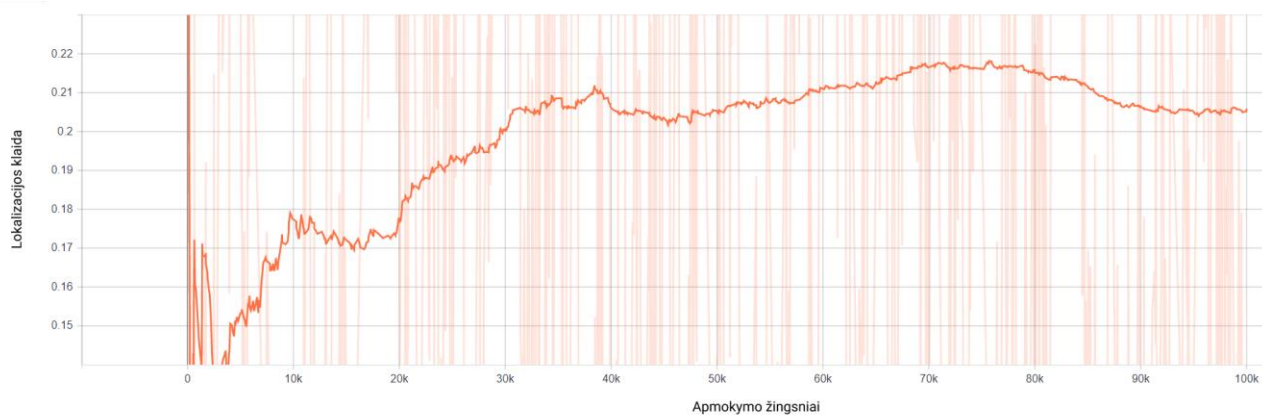
- nuotraukų kiekis: 2000;
- epochų kiekis: 500;
- paveikslėlio plotis: 1280;
- paveikslėlio aukštis 720;
- paketo dydis: 1;
- aktyvacijos funkcija: Softmax.

Apmokius modelį gauta klasifikacijos klaida: 0,103 (4.1 pav.). Ji parodo, kaip tiksliai modelis aptinka automobilius.



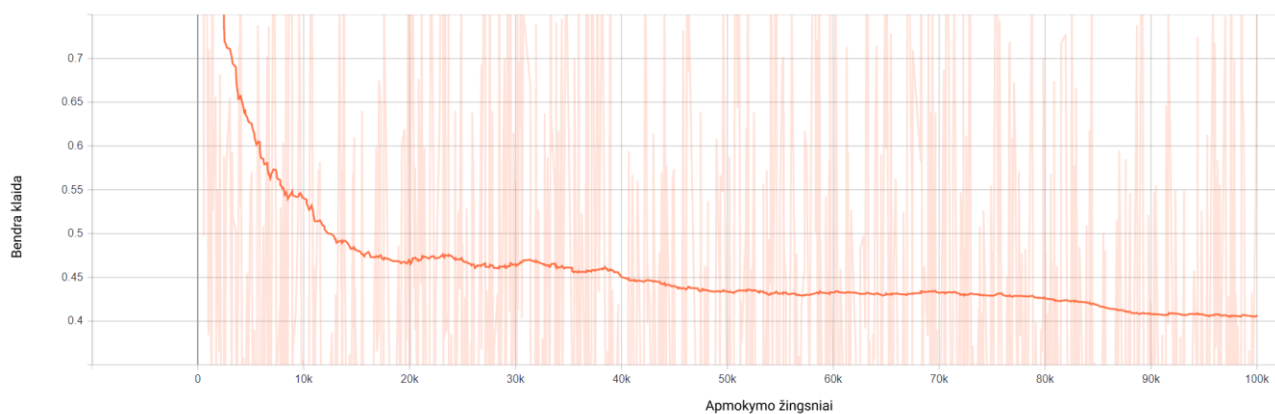
4.1 pav. Klasifikacijos klaida

Apmokius modelį gauta lokalizacijos klaida: 0,206 (4.2 pav.). Ji parodo, kaip tiksliai yra atpažįstama aptikto automobilio vieta kadre.



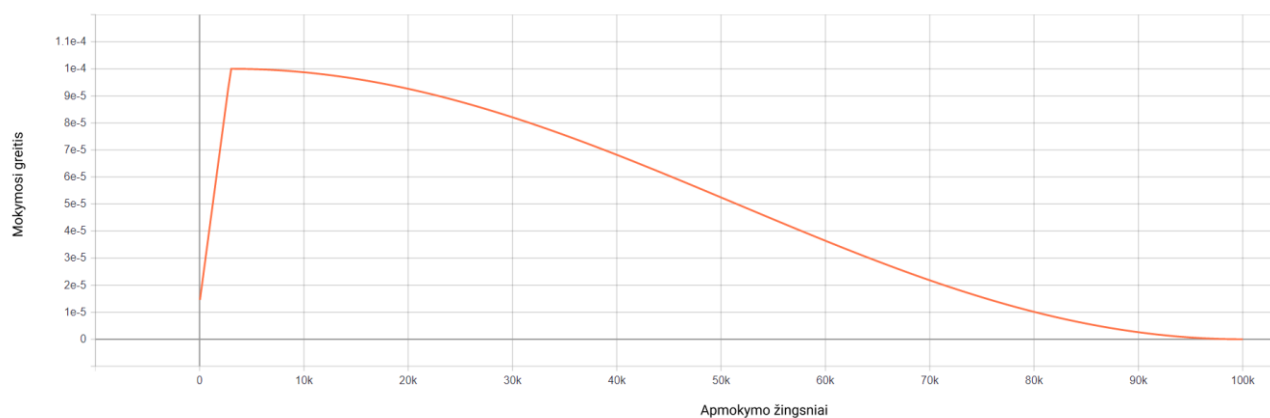
4.2 pav. Lokalizacijos klaida

Bendra apmokymo klaida – 0,406 (4.3 pav.). Ji parodo, kaip tiksliai yra atpažįstamas ir lokalizuojamas automobilis.



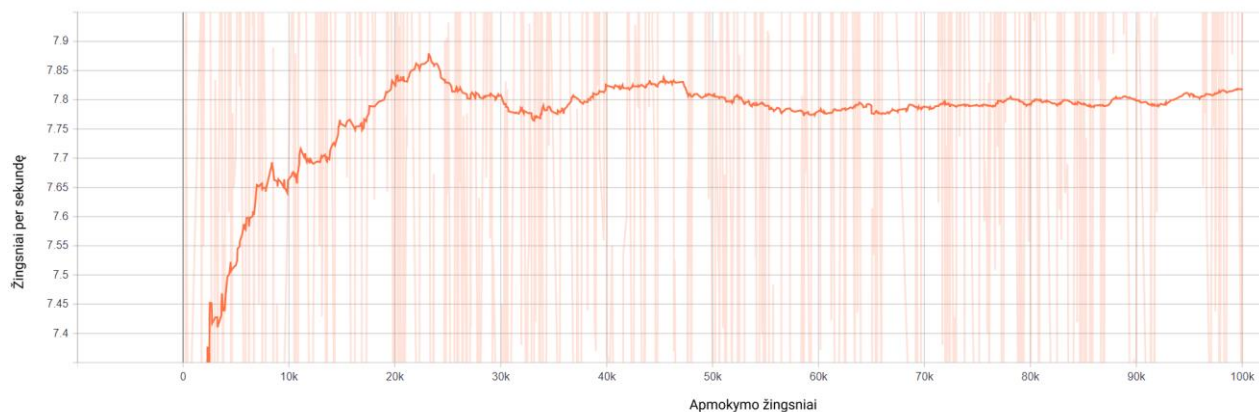
4.3 pav. Bendra klaida

Mokymosi greitis buvo pasirinktas kintantis. Pradinė vertė – 0,004 (4.4 pav.).



4.4 pav. Mokymosi greitis

Apmokymo žingsniai per sekundę (4.5 pav.).



4.5 pav. Apmokymo žingsniai per sekundę

Apmokius modelį, atliktas modelio testavimas su 300 nuotraukų, kurios nebuvo naudotos apmokymo procese.

Apskaičiuotas apmokyto modelio vidutinis tikslumas (mAP) - 0.3469. Taip pat paskaičiuoti vidutiniai tikslumai pagal automobilio dydį kadre. Dideliems automobiliams vidutinis tikslumas – 0,679, vidutiniams – 0,326, mažiems – 0,019.

Iš rezultatų matome, kad kadre geriausiai aptinkami labiausiai nutolę automobiliai, o prasčiausiai – arti esantys.

4.2. Greitaveika

Šiame poskyryje aprašomi algoritmo greitaveikos tyrimo rezultatai.

Rezultatai gauti ištyrus algoritmo veikimą su 500 kadru. Kadro plotis – 1280 pikselių, aukštis – 720 pikselių.

Gauti rezultatai:

- automobilių aptikimo pirmajame kadre greitaveika: 2,15 sekundės;
- automobilio aptikimo kadruose greitaveikos vidurkis: 0,0665 sekundės;
- miglotosios logikos greitaveikos vidurkis (1 automobiliui): 0,0060 sekundės;
- linijų aptikimo kadre greitaveikos vidurkis: 0,0067 sekundės;
- trajektorijos apskaičiavimo 1 automobiliui greitaveikos vidurkis: 0,0078 sekundės;
- miglotosios logikos inicializavimo greitaveika: 0,0174 sekundės;
- automobilių aptikimo modelio inicializavimo greitaveika: 6,862 sekundės;
- „Tensorflow“ bibliotekos inicializavimo greitaveika: 0,348 sekundės;
- algoritmo greitaveika vienam kadru (su skaičiavimų atvaizdavimu): 0,1036 sekundės;
- algoritmo greitaveika vienam kadru (be skaičiavimų atvaizdavimo): 0,1004 sekundės.

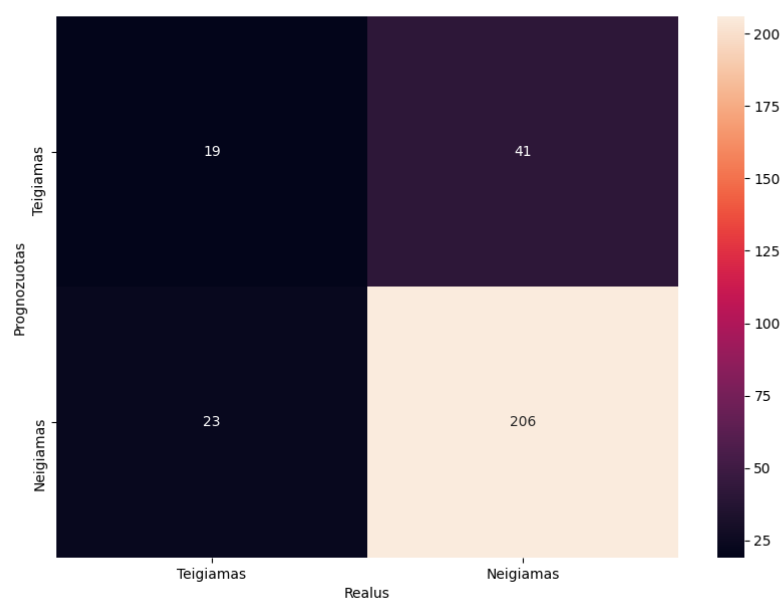
Iš gautų rezultatų matome, kad vieno kadro apdorojimas užtrunka 0,1 sekundės. Taigi per sekundę galima apdoroti 10 kadru. Norint naudoti sistemą realiu laiku, reikia vaizdo medžiagą sistemai paduoti 10 kadru per sekundę dažniu. Taip bus galima realiu metu aptikti išsiskiriančius automobilius.

Taip pat sistemos paleidimas užtrunka iki 8 sekundžių, o pirmojo kadro apdorojimas iki 3 sekundžių.

4.3. Aptikimo tikslumas

Sistemos veikimas išbandytas su realiais eismo įvykių vaizdo įrašais. Tikrinta ar sistema geba aptikti išsiskiriančius vairavimo atvejus.

Pirmuoju bandymu gauti rezultatai aprašomi maišos matrica (ang. *Confucion matrix*) (4.6 pav.) ir tikslumo metrikomis (4.1 lentelė).

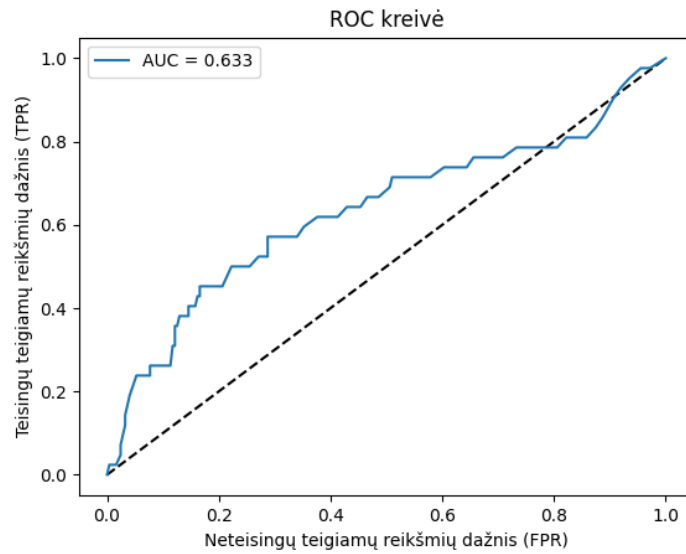


4.6 pav. Pirmojo bandymo maišos matrica

4.1 lentelė Pirmojo bandymo tikslumo metrikos

Metrika	Reikšmė
Teisingų teigiamų reikšmių dažnis (TPR)	0,452
Teisingų neigiamų reikšmių dažnis (TNR)	0,834
Teigiama nuspėjamoji vertė (PPV)	0,317
Neigiama nuspėjamoji vertė (NPV)	0,900
Neteisingų neigiamų reikšmių dažnis (FNR)	0,548
Neteisingų teigiamų reikšmių dažnis (FPR)	0,166
Klaidingų aptikimų dažnis (FDR)	0,683
Klaidingų praleidimų dažnis (FOR)	0,101
Subalansuotas tikslumas (BA)	0,643
F1 vertė	0,373

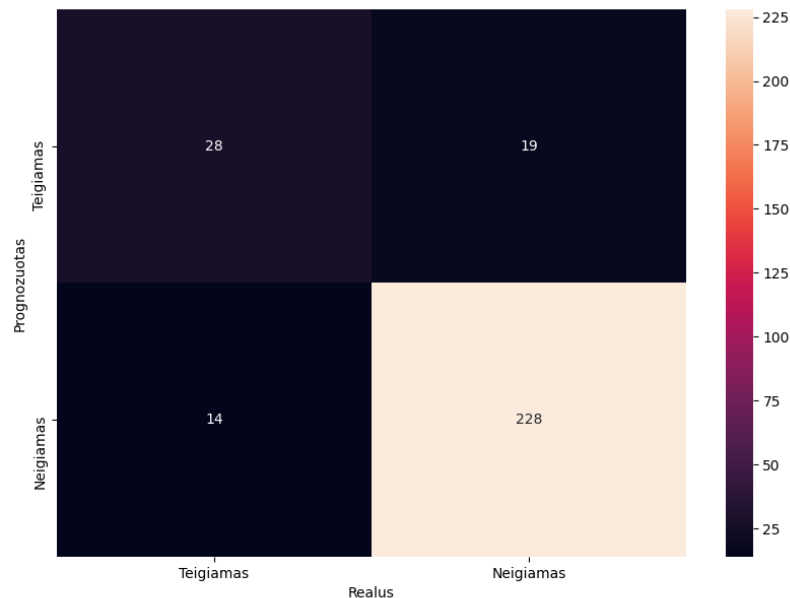
Iš gautų rezultatų matosi, kad sistemos tikslumas siekia vos 64,3 %. Sistema aptinka daug neteisingų teigiamų reikšmių. Taip pat ji neaptinka daugiau nei pusės išsiskiriančių atvejų. Prastą sistemos tikslumą patvirtina ir ROC kreivė (4.7 pav.), bei AUC vertė. ROC kreivė parodo kaip sistema geba atskirti išsiskiriančius vairuotojus. ROC kreivė yra arti neutralios kreivės, taip pat AUC vertė lygi 0,633. Sistema turi tik 63,3 % šansą tinkamai atskirti išsiskiriantį vairuotoją nuo neišsiskiriančio.



4.7 pav. Pirmojo bandymo ROC kreivė

Bandymo metu pastebėta, kad, išgaunant linijas, kadruose yra daug pašalinio baltos spalvos triukšmo. Tai lemia netikslų linijų aptikimą, dėl kurio sistemos tikslumas yra žemas. Siekiant sumažinti triukšmą, minimali baltos spalvos vertė padidinta nuo 150 iki 170 ir atliktas antrasis bandymas.

Antro bandymo metu gauti rezultatai aprašomi maišos matrica (ang. *Confusion matrix*) (4.8 pav.) ir tikslumo metrikomis (4.2 lentelė).

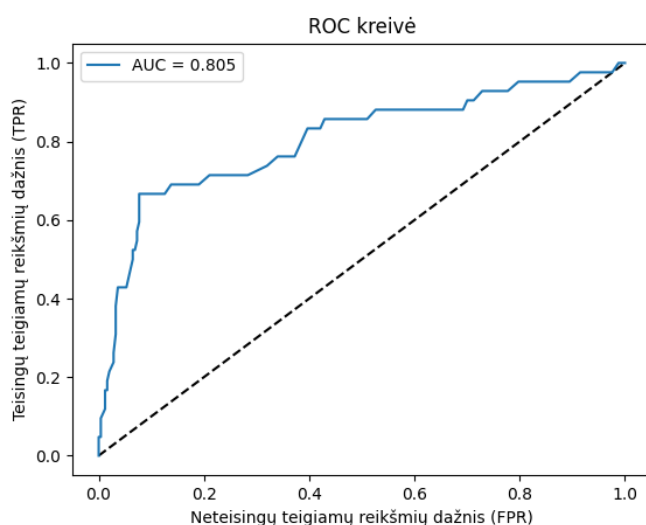


4.8 pav. Antrojo bandymo maišos matrica

4.2 lentelė Antrojo bandymo tikslumo metrikos

Metrika	Reikšmė
Teisingų teigiamų reikšmių dažnis (TPR)	0,667
Teisingų neigiamų reikšmių dažnis (TNR)	0,923
Teigiama nuspėjamoji vertė (PPV)	0,595
Neigiama nuspėjamoji vertė (NPV)	0,942
Neteisingų neigiamų reikšmių dažnis (FNR)	0,333
Neteisingų teigiamų reikšmių dažnis (FPR)	0,077
Klaidingų aptikimų dažnis (FDR)	0,404
Klaidingų praleidimų dažnis (FOR)	0,058
Subalansuotas tikslumas (BA)	0,795
F1 vertė	0,629

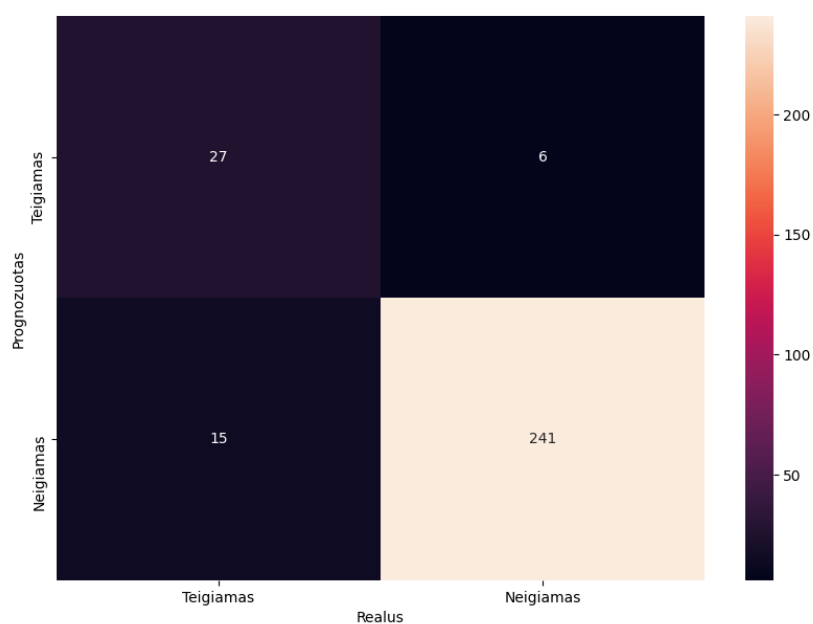
Antrojo bandymo metu, sistemos tikslumas siekė 79,5 %. Sistemos aptinkamų neteisingų teigiamų reikšmių skaičius sumažėjo, tačiau jis vis tiek išliko gana didelis. Tačiau bendrai sistemos tikslumas žymiai pagerėjo. Tai parodo ir ROC kreivė (4.9 pav.) bei AUC vertė. ROC kreivė nutolo nuo neutralios vertės. AUC parametras parodo, kad sistema gali atskirti išsiskiriantį vairuotoją nuo neišsiskiriančio su 80,5 % tikimybe.



4.9 pav. Antrojo bandymo ROC kreivė

Antrojo bandymo metu pastebėta, kad sistema kartais gražina neteisingą teigiamą reikšmę, kai automobilio trajektorija yra trumpa. Taip atsitinka dėl polinominės kreivės funkcijos. Kai trajektorijos pradžios ir pabaigos taškai yra šalia vienas kito, polinominė kreivė gali gautis su labai dideliu ekstremumu. Tokiu atveju trajektorijos kreivumo parametro vertė yra labai didelė, todėl miglotoji logika gražina neteisingą išvestį. Bandymo metu nustatyta, kad automobilio trajektorija gaunasi neteisinga, kai trajektorijos pradžios ir pabaigos taškai yra vienas nuo kito nutolę per mažiau nei 40 pikselių. Nuspręsta tokių trajektorijų nevertinti.

Trečiojo bandymo metu gauti sistemos spėjimų rezultatai aprašyti maišos matrica (ang. *Confusion matrix*) (4.10 pav.) ir tikslumo metrikomis (4.3 lentelė).

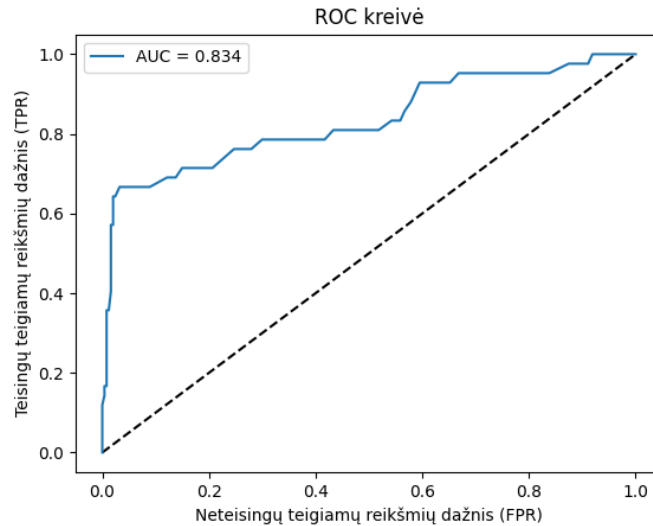


4.10 pav. Trečiojo bandymo maišos matrica

4.3 lentelė Trečiojo bandymo tikslumo metrikos

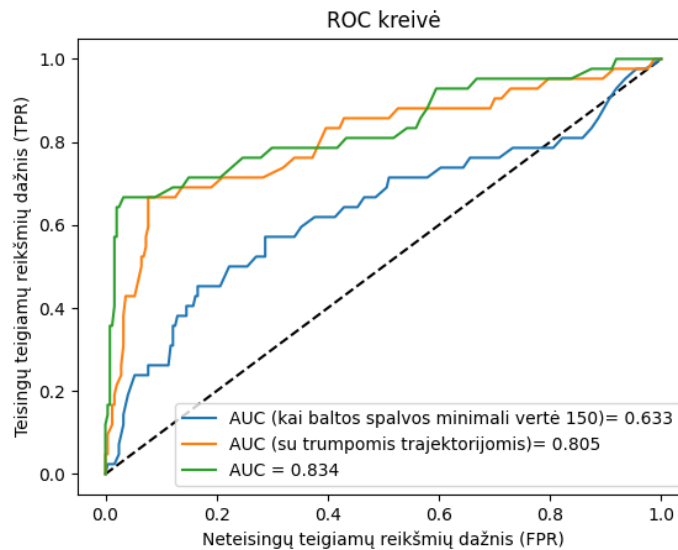
Metrika	Reikšmė
Teisingų teigiamų reikšmių dažnis (TPR)	0,643
Teisingų neigiamų reikšmių dažnis (TNR)	0,976
Teigiama nuspėjamoji vertė (PPV)	0,818
Neigiama nuspėjamoji vertė (NPV)	0,941
Neteisingų neigiamų reikšmių dažnis (FNR)	0,357
Neteisingų teigiamų reikšmių dažnis (FPR)	0,024
Klaidingų aptikimų dažnis (FDR)	0,181
Klaidingų praleidimų dažnis (FOR)	0,058
Subalansuotas tikslumas (BA)	0,809
F1 vertė	0,72

Dėl atliktų pakeitimų sistemos tikslumas išaugo iki 80,9 %. Iš gautų rezultatų matome, kad sistema gražina labai mažai neteisingų teigiamų spėjimų. Nors sistema kartais neaptinka išsiskiriančių vairuotojų, tačiau ji beveik visada tinkamai identifikuoja neišsiskiriančius vairuotojus. ROC kreivė (4.11 pav.) dar labiau nutolo nuo neutralios vertės. AUC parametras parodo, kad sistema gali atskirti išsiskiriantį vairuotoją nuo neišsiskiriančio su 83,4 % tikimybe.



4.11 pav. Trečiojo bandymo ROC kreivė

4.12 paveiksle pavaizduotos visų bandymų ROC kreivės. Iš grafiko matome, kad sistemos tikslumas su kiekvienu bandymu pagerėdavo, o geriausias gavosi paskutinio bandymo metu.

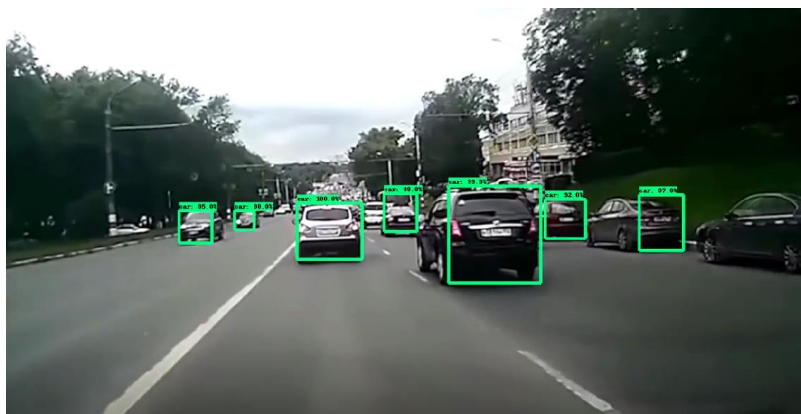


4.12 pav. Visų bandymų ROC kreivės

Sistema dažniausiai neaptikdavo tų išsiskiriančių vairuotojų, kurie į kelią įvažiuodavo iš šalutinio kelio ar kitų teritorijų. Vaizdo įrašo kadre buvo matomas tik tų automobilių šonas, o sistemoje automobilių aptikimas paremtas jų priekinių ir galinių dalių atpažinimu.

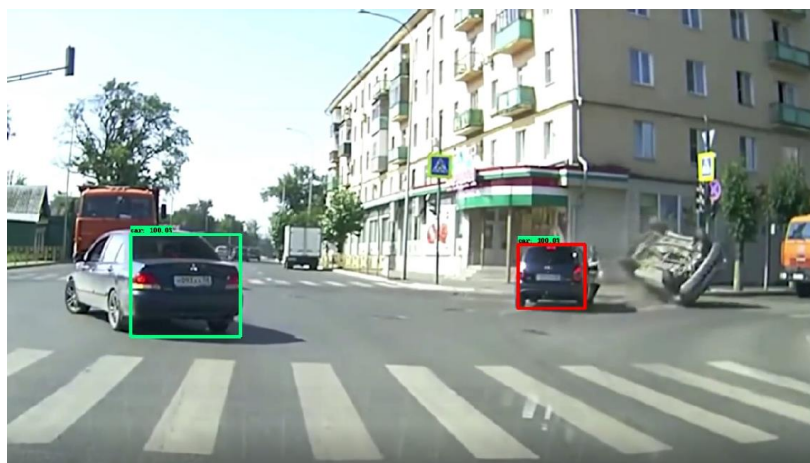
Keletas sistemos veikimo pavyzdžių pateikiama žemiau. Žalias kvadratas rodo neišsiskiriantį vairavimo atvejį, o raudonas – išsiskiriantį.

4.13 paveiksle matome, kaip sistema reaguoja į tvarkingai besielgiančius, neišsiskiriančius vairuotojus.



4.13 pav. Neišsiskiriantys vairuotojai

4.14 paveiksle pavaizduota, kaip sistema reaguoja į automobilį, ką tik sudalyvavusį eismo įvykyje.



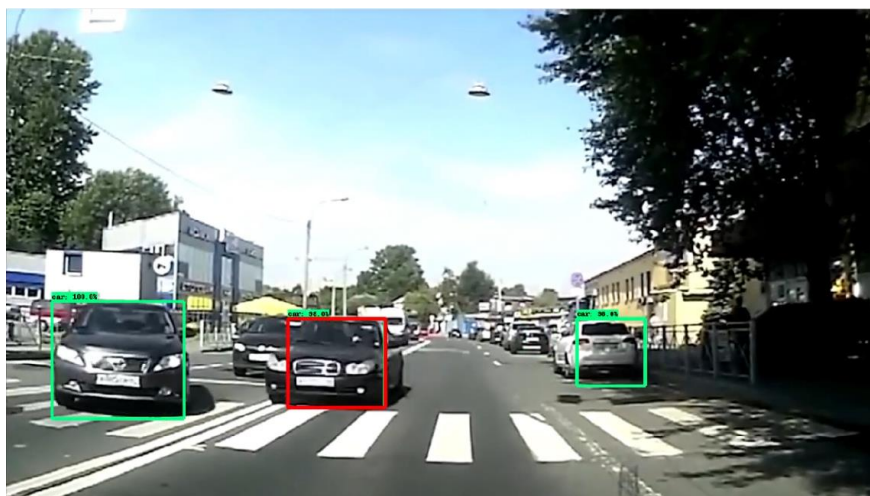
4.14 pav. Eismo įvykyje sudalyvavęs automobilis

4.15 paveiksle pavaizduota sistemos reakcija į priešprieša atvažiuojantį automobilį.



4.15 pav. Priešprieša atvažiuojantis automobilis

4.16 paveiksle vaizduojami chuliganiškai besielgiantys vairuotojai. Sistema aptinka tik vieną iš jų. Vairuotojas, kuris chuliganiškai lenkia kitus automobilius buvo aptiktas, o kitas, kuris rikiuojasi perėjoje, ne.



4.16 pav. Chuliganiškai besielgiantys vairuotojai

Išvados

1. Atliekant analizę, palygintos „Tensorflow“ ir „Keras“ bibliotekos. Pasirinkta „Tensorflow“ biblioteka, nes ji pasižymi geresne greitimeika ir „Keras“ integravimo galimybe.
2. Analizės metu nustatyta, kad konvoliuciniai neuroniniai tinklai geriausiai tinka vaizdų apdorojimo uždaviniams atlikti, nes jie demonstruoja gerą greitimeiką ir aukštą tikslumą.
3. Atliekant analizę pastebėta, kad vairavimo parametrų reikšmės, gautos iš skirtingos kokybės vaizdo įrašų, gali turėti skirtingas vertes. Siekiant apibendrinti skirtingus parametrus ir išvengti netikslumų ir neapibrėžtumų, nuspręsta naudoti miglotąją logiką.
4. Projektuojant sistemą, jai išskelti tikslai, surinkti funkciniai ir nefunkciniai reikalavimai, sudarytos sistemos panaudos atvejų, veiklos konteksto, paketų, sekų ir bendradarbiavimo diagramos, apibrėžtos vairavimo atvejo būsenos.
5. Tyrimo metu sukurti automobilių trajektorijos, linijų aptikimo, vairavimo atvejo išskirtinumo nustatymo algoritmai.
6. Atliekant automobilių aptikimo modelio apmokymą, nustatyta, kad trajektorijai skaičiuoti labiau tinka apmokymas su automobilių priekinėmis ir galinėmis dalimis nei, kad su visu automobiliu.
7. Kuriant linijų aptikimo algoritmą, nustatyta, kad vaizdo kadruose yra daug pašalinio triukšmo. Siekiant jį pašalinti naudotas Gauso suliejimo algoritmas. Taip pat, siekiant išvengti klaidingai aptiktų linijų, jos buvo lyginamos su prieš tai aptiktomis.
8. Išskirtinumo identifikavimui naudota miglotoji logika padėjo išvengti priklausomybės nuo vaizdo įrašų kokybės, pašalinti neteisingai apskaičiuotus parametrus.
9. Sukurta sistema atitinka jai keliamus funkcinius ir nefunkcinius reikalavimus.
10. Ištestavus automobilių aptikimo modelį, nustatyta, kad jis geriausiai aptinka toli kadre esančius automobilius, o prasčiausiai – arti esančius.
11. Ištyrus sukurto algoritmo greitimeiką, nustatyta, kad vieno kadro apdorojimas užtrunka 0,1 sekundės. Per 1 sekundę algoritmas gali apdoroti 10 kadru. Taip pat nustatyta, kad sistemos paleidimas gali užtrukti iki 8 sekundžių.
12. Eksperimento būdu buvo įvertintas sukurto algoritmo tikslumas. Pradinis sistemos tikslumas siekė 64,3 %. Atlikus sistemos parametrų modifikacijas, sistemos tikslumas buvo pagerintas iki 80,9 %. Sistema gražina labai mažai neteisingų teigiamų spėjimų. Nors sistema kartais neaptinka išsiskiriančių vairuotojų, tačiau ji beveik visada tinkamai identifikuoja neišsiskiriančius vairuotojus.
13. Eksperimento metu nustatyta, kad sistema neaptinka išsiskiriančių vairavimo atvejų, kai kadre yra matoma šoninė automobilio dalis. Taip yra todėl, kad automobilių aptikimo modelis buvo apmokytas tik su priekinėmis ir galinėmis automobilių dalimis.

Literatūros sąrašas

- [1] X. Zhang, X. Zhao, H. Du and J. Rong, "A Study on the Effects of Fatigue Driving and Drunk Driving on Drivers' Physical Characteristics," *Traffic injury prevention*, vol. 15, 01 2014.
- [2] N. N. Charniya and V. R. Nair, "Drunk driving and drowsiness detection," in *2013 4th International Conference on Intelligent Systems, Modelling and Simulation*, Bangkok, Thailand, 2017.
- [3] K. Klubsuwan, W. Koodtalang and S. Mungsing, "Traffic Violation Detection Using Multiple Trajectories Evaluation of Vehicles," in *2013 4th International Conference on Intelligent Systems, Modelling and Simulation*, Bangkok, Thailand, 2013.
- [4] „2018 metais neblaivūs vairuotojai sukėlė mažiau eismo įvykių, pareigūnai išaiškino daugiau tokių vairuotojų,“ Lietuvos kelių policijos tarnyba, [Tinkle]. Prieiga per: <http://lkpt.policija.lrv.lt/lt/naujienos/2018-metais-neblaivus-vairuotojai-sukele-maziau-eismo-ivykiu-pareigunai-isaiskino-daugiau-tokiu-vairuotoju>. [Kreiptasi 2019 lapkričio 27].
- [5] „Paskelbta, kiek Lietuvoje yra važiuojančių automobilių,“ LRT, 14 02 2015. [Tinkle]. Prieiga per: <https://www.lrt.lt/naujienos/verslas/4/70177/paskelbta-kiek-lietuvoje-yra-vaziuojanciu-automobiliu>. [Kreiptasi 2019 lapkričio 27].
- [6] V. Radžiūnas, „Perspėja: į gatves išriedės krūva nežymėtų policijos automobilių, o baudas gausite akimirksniu,“ LRYtas, [Tinkle]. Prieiga per: <https://www.lrytas.lt/auto/saugus-eismas/2019/11/04/news/perspeja-i-gatves-isriedes-kruva-nezymetu-policijos-automobiliu-o-baudas-gausite-akimirksniu-12419794>. [Kreiptasi 2019 lapkričio 27].
- [7] „Netradyne Driveri,“ [Tinkle]. Prieiga per: <https://www.netradyne.com/driveri/>.
- [8] L. Dang, G. Tewolde, X. Zhang and J. Kwon, "Reduced resolution lane detection algorithm," in *2017 IEEE AFRICON*, Cape Town, South Africa, 2017.
- [9] C. J. Thornton, *Techniques in computational learning : an introduction / C.J. Thornton.*, London; New York: Chapman & Hall Computing, 1992.
- [10] B. Carswell and V. Chandran, "Automated recognition of drunk driving on highways from video sequences," in *Proceedings of 1st International Conference on Image Processing*, Austin, TX, USA, 1994.
- [11] S. Miyata and K. Oka, "Automated license plate detection using a support vector machine," in *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Phuket, Thailand, 2016.
- [12] P. Chanawangsa, J. Wan, C. Wu and C. W. Chen, "A novel 2D-3D hybrid approach to vehicle trajectory and speed estimation," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Qingdao, China, 2014.
- [13] C.-H. He and K.-M. Lam, "Fast Vehicle Detection with Lateral Convolutional Neural Network," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, Canada, 2018.

- [14] F. Zhang, F. Yang, C. Li and G. Yuan, "CMNet: A Connect-and-Merge Convolutional Neural Network for Fast Vehicle Detection in Urban Traffic Surveillance," *IEEE Access*, no. 7, pp. 72660-72671, 2019.
- [15] V. M. Arceda and E. L. Riveros, "Fast car Crash Detection in Video," in *2018 XLIV Latin American Computer Conference (CLEI)*, São Paulo, Brazil, 2018.
- [16] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018.
- [17] K. Yoneda, A. Kuramoto and N. Sukanuma, "Convolutional neural network based vehicle turn signal recognition," in *2017 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, Okinawa, Japan, 2017.
- [18] Z. Fu, W. Hu and T. Tan, "Similarity based vehicle trajectory clustering and anomaly detection," in *IEEE International Conference on Image Processing 2005*, Genova, Italy, 2005.
- [19] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, no. 86, pp. 2278-2323, 1998.
- [20] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of ICML*, Haifa, Israel, 2010.
- [21] T. Guo, J. Dong, H. Li and Y. Gao, "Simple convolutional neural network on image classification," in *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, Beijing, China, 2017.
- [22] F. Deroncourt, „Introduction to fuzzy logic,“ 2013.
- [23] Y. Katyal, S. Alur and S. Dwivedi, "Safe driving by detecting lane discipline and driver drowsiness," in *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*, Ramanathapuram, India, 2014.
- [24] A. Nagpal and G. Gabrani, "Python for Data Analytics, Scientific and Technical Applications," in *2019 Amity International Conference on Artificial Intelligence (AICAI)*, Dubai, United Arab Emirates, United Arab Emirates, 2019.
- [25] H. Zhang and J. Nie, "Program performance test based on different computing environment," in *2016 IEEE International Conference of Online Analysis and Computing Science (ICOACS)*, Chongqing, China, 2016.
- [26] A. Kumar and S. Panda, "A Survey: How Python Pitches in IT-World," in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, Faridabad, India, 2019.
- [27] R. Airola and K. Hager, "Image Classification, Deep Learning and Convolutional Neural Networks: A Comparative Study of Machine Learning Frameworks," Karlstad University, 2017.
- [28] W. Rosinski, „Deep Learning Frameworks Speed Comparison,“ GitHub, 2019. [Tinkle]. Prieiga per: <https://wrosinski.github.io/deep-learning-frameworks/>. [Kreiptasi 2019 lapkričio 18].

- [29] Z. Xiong, X. Tao, N. Zhao and B. Lin, "SINGLE IMAGE SUPER-RESOLUTION USING A NON-LOCAL 3D CONVOLUTIONAL NEURAL NETWORK," in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Anaheim, CA, USA, 2018.
- [30] S. Barnwal, "Vehicle Behavior Analysis for Uneven Road Surface Detection," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Las Palmas, Spain, 2015.
- [31] „TensorFlow 2 Detection Model Zoo,“ Github, [Tinkle]. Prieiga per: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md. [Kreiptasi 2020 spalio 5].
- [32] "Highly Efficient Follicular Segmentation in Thyroid Cytopathological Whole Slide Image - Scientific Figure on ResearchGate," [Tinkle]. Prieiga per: https://www.researchgate.net/figure/a-The-structure-of-ResNet-101-b-The-basic-structure-of-each-block-in-ResNet-101-c_fig1_331110894. [Kreiptasi 2020 lapkričio 12].
- [33] J. Huang, „CSEP 576: Object Detection with Convolutional Networks,“ [Tinkle]. Prieiga per: https://courses.cs.washington.edu/courses/csep576/18sp/lectures/8_1_detection.pdf. [Kreiptasi 2020 lapkričio 12].
- [34] J. Krause, M. Stark, J. Deng and L. Fei-Fei, "3D Object Representations for Fine-Grained Categorization," in *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.
- [35] J. Arróspide, L. Salgado and M. Nieto, "Video analysis based vehicle detection and tracking using an MCMC sampling framework," *EURASIP Journal on Advances in Signal Processing*, vol. 2012, 2012.
- [36] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan and T. Darrell, "BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2020.