



Kauno technologijos universitetas

Informatikos fakultetas

**Dirbtinio intelekto metodų tyrimas siekiant nustatyti ar
vairuotojas kalba telefonu**

Baigiamasis magistro projektas

Deividas Šetkus

Projekto autorius

Prof. Rytis Maskeliūnas

Vadovas

Kaunas, 2020



Kauno technologijos universitetas

Informatikos fakultetas

Dirbtinio intelekto metodų tyrimas siekiant nustatyti ar vairuotojas kalba telefonu

Baigiamasis magistro projektas

Informatika (6211BX007)

Deividas Šetkus

Projekto autorius

Prof. Rytis Maskeliūnas

Vadovas

Doc. Liudas Motiejūnas

Recenzentas

Kaunas, 2020



Kauno technologijos universitetas

Informatikos fakultetas

Deividas Šetkus

Dirbtinio intelekto metodų tyrimas siekiant nustatyti ar vairuotojas kalba telefonu

Akademinio sąžiningumo deklaracija

Patvirtinu, kad mano, Deivido Šetkaus, baigiamasis projektas tema „Dirbtinio intelekto metodų tyrimas siekiant nustatyti ar vairuotojas kalba telefonu“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

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

(vardą ir pavardę įrašyti ranka)

(parašas)

Šetkus, Deividas. Dirbtinio intelekto metodų tyrimas siekiant nustatyti ar vairuotojas kalba telefonu. Magistro baigiamasis projektas / vadovas Prof. Rytis Maskeliūnas; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Gamtos mokslai, Informatika (09P)

Reikšminiai žodžiai: išsiblaškęs vairuotojas; vairuotojas, kalbantis telefonu; asmens aptikimas;

Kaunas, 2020. 50 p.

Santrauka

Vairuotojai, kalbantys telefonu, yra problema visame pasaulyje. Šiame projekte yra tiriama galimybė nustatyti vairuotoją, kalbantį telefonu, naudojant dirbtinį intelektą. Ši problema nebuvo sprendžiama naudojant duomenų rinkinį, kuris nėra gautas transporto priemonės viduje, dėl sunkumų jį gauti. Su surinktu iš interneto duomenų rinkiniu ir nufotografuotu stovinčioje transporto priemonėje duomenų rinkiniu buvo atliekami konvoliucinių neuroninių tinklų tyrimai, siekiant išgauti kuo geresnius rezultatus, aptinkant vairuotoją, kalbantį telefonu.

Setkus, Deividas. Research of artificial intelligence methods for driver talking on the phone detection. Master's Final Degree Project / supervisor Prof. Rytis Maskeliunas; Faculty of Informatics, Kaunas University of Technology.

Study field and area (study field group): Natural sciences, Informatics (09P)

Keywords: distracted driver; driver talking on the phone; person detection

Kaunas, 2020. 50 pages.

Summary

Drivers talking on the phone is a problem in the World. In this project, we use artificial intelligence and try to predict if the driver is talking on the phone. This problem was not solved before with a dataset from outside the vehicle because there are no publicly available datasets and this information is hard to acquire. After acquiring the dataset from the Internet with web crawler and photographed training dataset in a standing vehicle, four convolutional neural networks were researched to get the best generalization with training dataset.

Turinys

Lentelių sąrašas	7
Paveikslų sąrašas	8
Santrumpų ir terminų sąrašas	9
Įvadas.....	10
1. Literatūros apžvalga	11
1.1. Mokymasis su ir be mokytojo	11
1.2. Konvoliuciniai neuroniniai tinklai.....	11
1.3. Mokymosi su mokytoju modeliai	12
1.3.1. „AlexNet“	12
1.3.2. „VGGNet-16“	13
1.3.3. „GoogLeNet“.....	14
1.3.4. „ResNet-50“	15
1.3.5. Modelių palyginimas.....	15
1.4. Objektų lokalizacijos modeliai	16
1.5. Optimizatoriai.....	16
1.6. Panašūs darbai	18
2. Vairuotojo, kalbančio telefonu, nustatymo projektavimas	20
2.1. Projekto planas	20
2.2. Nefunkciniai reikalavimai	20
2.3. Funkciniai reikalavimai	20
2.4. Tyrimo aplinkos projektavimas.....	21
2.4.1. Objektų aptikimo programos projektavimas	21
2.4.2. Modelių mokymo programos projektavimas.....	23
2.4.3. Modelio svorių vertinimo programos projektavimas	25
2.5. Tyrimo planas	26
2.6. Sprendimo programos projektavimas.....	27
3. Realizacija	30
3.1. Aparatūrinė ir programinė įranga	30
3.2. Duomenų rinkinio kūrimas.....	30
3.3. Duomenų rinkinio paruošimas	32
3.4. Modelių tyrimas su nepakeistais duomenų rinkiniais	33
3.5. Modelių tyrimas su keičiamais duomenų rinkiniais.....	37
3.5.1. Mokymo duomenų rinkinio raiškos mažinimo tyrimas.....	37
3.5.2. Mokymo duomenų rinkinio papildymo tyrimas.....	38
3.5.3. Veiksmingumo tyrimas, naudojant duomenų rinkinio kraštines.....	39
3.5.4. Naujo modelio kūrimo tyrimas.....	40
3.5.5. Duomenų rinkinio papildymas naudojant duomenų papildymo grupę	40
3.5.6. Tyrimas, kai mokymo metu naudojamas transporto priemonės nuotraukos ketvirtis.....	44
3.5.7. Sumažintos įvesties modelių tyrimas	45
Išvados	47
Literatūros sąrašas	48
Priedai.....	51

Lentelių sąrašas

1.1 lentelė. Metodų kiekybinis palyginimas [18].....	15
3.1 lentelė. Naudoto kompiuterio techniniai ir įrangos parametrai.....	30
3.2 lentelė. Naudotoje „Anaconda Navigator“ aplinkoje parsųstos programinių įrangų versijos ...	30
3.3 lentelė. „YOLO“ modelio objektų aptikimo tyrimo rezultatai.....	32
3.4 lentelė. Paruoštų duomenų rinkinių nuotraukų kiekiai	33
3.5 lentelė. Modelių mokymo rezultatai, naudojant transporto priemonių duomenų rinkinį	34
3.6 lentelė. Modelių mokymo rezultatai, naudojant vairuotojų duomenų rinkinį.....	34
3.7 lentelė. Greitaveikos tyrimo rezultatai, naudojant transporto priemonių duomenų rinkinį	34
3.8 lentelė. Greitaveikos tyrimo rezultatai, naudojant vairuotojų duomenų rinkinį	34
3.9 lentelė. Modelių mokymo rezultatai, naudojant virš 96x96 raiškos transporto priemonių duomenų rinkinį	35
3.10 lentelė. Modelių mokymo rezultatai, naudojant virš 128x128 raiškos transporto priemonių duomenų rinkinį.....	35
3.11 lentelė. Modelių mokymo rezultatai, naudojant virš 48x48 raiškos vairuotojų duomenų rinkinį	36
3.12 lentelė. Modelių mokymo rezultatai, naudojant virš 64x64 raiškos vairuotojų duomenų rinkinį	36
3.13 lentelė. Modelių mokymo tikslumai, naudojant virš 64x64 raiškos žmonių ir virš 96x96 raiškos transporto priemonių duomenų rinkinius kartu	36
3.14 lentelė. Modelių mokymo tikslumai, naudojant sumažintą iki 48x48 vairuotojų duomenų rinkinį	37
3.15 lentelė. Greitaveikos tyrimo rezultatai, naudojant sumažintą iki 48x48 raiškos vairuotojų duomenų rinkinį.....	38
3.16 lentelė. Duomenų papildymo tyrimo rezultatai.....	38
3.17 lentelė. Modelių mokymo rezultatai, naudojant iki 48x48 raiškos sumažintą vairuotojų duomenų rinkinį ir grupę treniravimo generatorių	42
3.18 lentelė. Greitaveikos tyrimo rezultatai, naudojant iki 48x48 raiškos sumažintą vairuotojų duomenų rinkinį ir grupę treniravimo generatorių	43
3.19 lentelė. Modelių svorių saugojimo ir užkrovimo trukmės tyrimo rezultatai.....	43
3.20 lentelė. Modelių mokymo rezultatai, naudojant iki 64x64 raiškos sumažintą vairuotojų duomenų rinkinį ir grupę treniravimo generatorių	44
3.21 lentelė. Vairuotojo, kalbančio telefonu, nustatymas, naudojant transporto priemonės nuotraukos ketvirtį.....	45
3.22 lentelė. Sumažintos įvesties modelių tyrimo rezultatai.....	45

Paveikslų sąrašas

1.1 pav. Aktyvacijos funkcijų pavyzdžiai [20]	12
1.2 pav. „AlexNet“ modelio architektūra [22]	13
1.3 pav. „VGG-16“ modelio architektūra [25].....	14
1.4 pav. „GoogLeNet“ modulis [26]	14
1.5 pav. Chaojie Ou surinktų nuotraukų iš interneto pavyzdžiai [45]	18
2.1 pav. Objektų aptikimo veiklos diagrama	23
2.2 pav. Modelių mokymo veiklos diagrama.....	24
2.3 pav. Modelio svorių vertinimo veiklos diagrama	25
2.4 pav. Sprendimo algoritmas, nustatant vairuotoją, kalbantį telefonu, iš transporto priemonės nuotraukos	27
2.5 pav. Sprendimo algoritmas, nustatant vairuotoją, kalbantį telefonu, iš asmens nuotraukos.....	28
3.1 pav. Atsitiktinės duomenų rinkinio nuotraukos	31
3.2 pav. Paruoštų duomenų aplankai	32
3.3 pav. Mokymo rinkinio kraštinių nuotraukos (kairėje pusėje) ir validacijos rinkinio kraštinių nuotraukos	39
3.4 pav. Modelio mokymo veiklos diagrama.....	41

Santrumpų ir terminų sąrašas

Santrumpos:

ReLU – (angl. *Rectified linear unit*) Funkcija, kurios vertė lygi teigiamos ašies vertei.

MAC – (angl. *multiply and accumulate*) Dauginimo ir kaupimo funkcija.

Terminai:

F₁ vertė – (angl. *F₁ score, F-score, F-measure*) Modelio tikslumo vertinimo sistema, kurioje atsižvelgiama į tikslumą (angl. *precision*) ir atgaminimą (angl. *recall*).

Aktyvacijos funkcija – (angl. *activation function*) neurono išvestis, priklausanti nuo įvesties [1].

Klaidų matrica – (angl. *confusion matrix*) lentelė, kurioje yra apibendrinta informacija apie modelio klasifikavimą, duotam duomenų rinkiniui [2].

Įvadas

Dirbtinio intelekto srities pradžia yra laikoma 1956 metais surengta „Dirbtinio intelekto tyrimo projektas Dartmouth“ [3]. Nuo šio projekto pradžios, praėjus daugiau nei pusei amžiaus, įvyko dauguma mokslinių laimėjimų, tokių kaip dirbtinio intelekto laimėjimas šachmatais prieš čempioną Garį Kasparovą [4], autonominių mašinų greitkeliuose prototipai [5], objektų atskirimas geriau nei žmogus [6]. Dirbtinis intelektas gali būti panaudotas problemoms spręsti ir jų sprendimams optimizuoti.

Vairavimas išsiblaškius yra paplitęs po visą pasaulį. Išsiblaškę vairuotojai kelia pavojų ne tik sau, bet ir aplinkiniams. Vien per 2016 metus Jungtinės Amerikos Valstijose dėl išsiblaškiusio vairavimo žuvo 3450 žmonės, 391000 sužeisti [7]. Pagrindinė išsiblaškyimo priežastis - kalbėjimas telefonu. Jungtinėse Amerikos Valstijose kasdien vairuojant telefonu kalba 3,3 % vairuotojų, t.y. 481000 vairuotojų per dieną [8], o Lietuvoje atsižvelgiant į „Regitros“ ir „ERGO Insurance“ duomenis, galime teigti, kad apie 530000 taip yra pažeidę taisykles, o apie 153000 tai daro dažnai [9,10].

Šiame darbe nagrinėsime kaip, panaudojant dirbtinį intelektą, galėtumėme atpažinti vairuotojus, kalbančius telefonu, siekdami sumažinti išsiblaškiusių vairuotojų skaičių.

Darbo tikslas - sukurti mašininio mokymosi programą, kuri nustatytų ar vaizdinėje medžiagoje yra transporto priemonės vairuotojas, kalbantis telefonu.

Darbo uždaviniai:

1. Atlikti konvoliucinių neuroninių tinklų analizę;
2. sukurti vaizdinės medžiagos duomenų rinkinį, skirtą modeliui apmokyti ir testuoti;
3. atlikti tyrimus su projekte tiriamais modeliais, modifikuojant įvesties nuotraukas ir keičiant modelių įvesties raišką;

1. Literatūros apžvalga

1.1. Mokymasis su ir be mokytojo

Mokymosi su mokytoju modeliui apmokyti yra suteikiama informacija su įvesties reikšmėmis ir norimomis gauti rezultatų reikšmėmis. Tam, kad mašininio mokymosi modelis išmoktų teisingai atlikti sprendimus, duomenys yra laikomi teisingais.

Mokymosi be mokytojo modeliui pateikiame informaciją, tačiau neduodame informacijos kaip šią informaciją sugrupuoti. Mašininio mokymosi modelis turi pats atrasti informacijos struktūrą, pagal kurią būtų galima sugrupuoti gautą informaciją. Dažniausiai informacija yra sugrupuojama į pasirinktą skaičių spiečių [11].

Mokymasis su mokytoju yra pranašesnis tuo, kad galime turimus duomenis suklasifikuoti į vaizdinę medžiagą, kurioje egzistuoja vairuotojas, kalbantis telefonu ir be vairuotojo, kalbančio telefonu. Atitinkamai, galime lengvai modifikuoti modelio parametrus, kad gautumėme didesnę F_1 vertę. F_1 vertė yra bedimensinis dydis, kuris kinta $[0;1]$ režiuose. F_1 vertė yra apskaičiuojama [12]:

$$F_1 = 2 * \frac{\frac{\text{teisingi teigiami}}{\text{visi spėti teigiami}} * \frac{\text{teisingi teigiami}}{\text{visi teigiami}}}{\frac{\text{teisingi teigiami}}{\text{visi spėti teigiami}} + \frac{\text{teisingi teigiami}}{\text{visi teigiami}}}; \quad (1)$$

Mokymasis be mokytojo yra pranašesnis tuo, kad būtų sutaupytas laikas, neskirstant vaizdinės medžiagos į grupes, tačiau yra rizika, kad modelis gali nesugrupuoti vaizdinės medžiagos į norimas dvi (vairuotojo, kalbančio telefonu ir vairuotojo, nekalbančio telefonu) sritis [13].

1.2. Konvoliuciniai neuroniniai tinklai

Konvoliucinių neuroninių tinklų veikimo principas yra simuliuoti žmogaus smegenų veiklą. Žmogaus smegenys yra sudarytos iš nervinių ląstelių, vadinamų neuronais. Neuronai chemiškai perduoda signalus tarpusavyje neuromediatoriais (angl. *neurotransmitter*) [14]. Priklausomai nuo signalų, smegenys priima sprendimus. Šis biologinis principas yra konvoliucinių neuroninių tinklų pagrindas. Konvoliuciniai neuroniniai tinklai yra sudaryti iš keliuose sluoksniuose išsidėsčiusių dirbtinių neuronų. Pirmasis sluoksnis yra sudarytas iš įvesties (angl. *input*) neuronų. Įvesties neuronais yra perduodamas signalas paslėptuose sluoksniuose esantiems neuronams. Tarp neuronų sluoksnių, neuronai yra sujungti vienas su kitu jų signalo stiprumu, dar kitaip vadinamais svoriais. Priklausomai nuo signalų stiprumo, gaunamos neuronų „aktyvacijos“ vertės. Iš paslėptų neuronų signalas yra perduodamas išvesties signalams ir pagal jų „aktyvacijos“ vertes priimamas neuroninio tinklo sprendimas [15].

Konvoliucija – matematinė operacija (2), kurios rezultatas yra požymių žemėlapis [16].

$$s = x * w; \quad (2)$$

čia s – požymių žemėlapis, x – įvestis, w – filtras.

Dvidimensėje erdvėje atitinkamai pritaikome „slenkančio lango“ (angl. *sliding window*) principą ir (2) formulė tampa išreiškiamą [16]:

$$S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i - m, j - n)K(m, n); \quad (3)$$

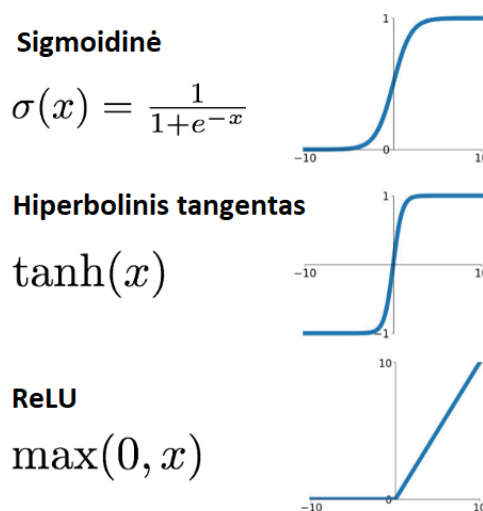
čia S – dvimatis požymių žemėlapis, I – dvimatė įvestis, K – dvimatis filtras, i, j – įvesties aukštis ir plotis, m, n – filtro aukštis ir plotis.

1.3. Mokymosi su mokytoju modeliai

1.3.1. „AlexNet“

„AlexNet“ yra laikomas vienu iš modelių, padariusių didžiausią įtaką kompiuterinei regai [17,18]. Šis modelis 2012 metais nugalėjo vieną iš sunkiausių kompiuterinės regos konkursų „Image Net Large Scale Visual Recognition Challenge“, suklysdamas tik 15 %, o artimiausias varžovas atsiliko 10 % [19]. Šis neuroninių tinklų modelis buvo sukurtas Alekso Križevskio, Iljos Sutskeverio ir Geofrio Hintono.

Šis modelis nuo pirmtakų išsiskyrė mokymo spartumu dėl 2 priežasčių: 1) aktyvacijos funkcijai apskaičiuoti buvo naudojama ReLU tiesė, o ne įprasta sigmoidinė ar hiperbolinis tangentas 1.1. pav.; 2) konvoliucijos metu buvo ieškomas lokalus maksimumas, o ne lokalus vidurkis [18].

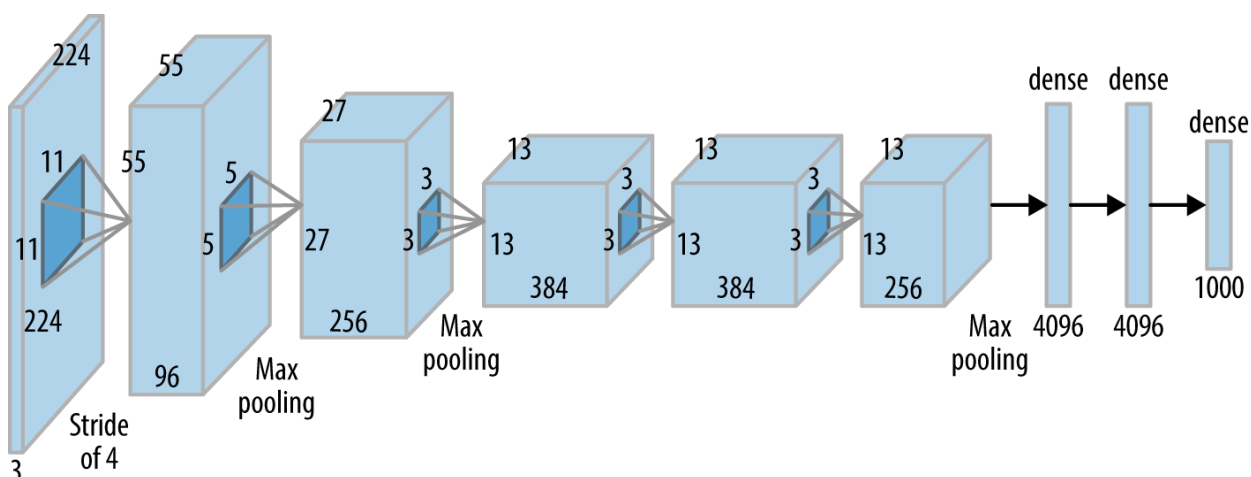


1.1 pav. Aktyvacijos funkcijų pavyzdžiai [20]

Šis modelis pasižymi penkiais skirtingų matmenų neuroninių tinklų sluoksniais. Tarp sujungtų sluoksnių yra apskaičiuojamos ReLU aktyvacijos funkcijos. Modelio įvestis – RGB 227x227 raiškos vaizdas.

Pirmame sluoksnyje atliekamos 96-ių 11x11 raiškos filtrų konvoliucijos per visą įvesties vaizdą. Konvoliucijos yra atliekamos „slenkančio lango“ principu, paslenkant filtrą 4 pikseliais. Gautoje išvestyje ieškomos maksimalios vertės 3x3 zonoje, „slenkančio lango“ principu kas 2 pikselius. Rezultatai yra perduodami į antrą sluoksnį 55x55x96 matmenimis. Kituose sluoksniuose yra atkartojamas procesas su skirtingais matmenimis. Antrame sluoksnyje yra naudojamos 256-ių 5x5 filtrų konvoliucijos, kurios yra paslenkamos 1 pikseliu. Išvestyje su vienodais parametrais, kaip ir pirmame sluoksnyje, ieškomos maksimalios vertės. Po šių veiksmų matmenys pakinta į 27x27x256. Trečiame ir ketvirtame sluoksniuose yra atliekamos 384-ių 3x3 filtrų konvoliucijos, kurios yra paslenkamos 1 pikseliu. Po trečio sluoksnio matmenys pakinta į 13x13x384. Penktame sluoksnyje atliekamos 256-ių 3x3 filtrų konvoliucijos, paslenkamos 1 pikselio žingsniu. Išvestyje ieškoma maksimalių verčių kaip ir pirmame sluoksnyje. Dimensijos pakinta į 13x13x256. Atlikus šiuos

veiksmus, visos gautos neuronų vertės yra sujungiamos į vieną eilę ir sukuriami 3 visiškai sujungti sluoksniai [21]. Visa modelio architektūra pavaizduota 1.2 pav.



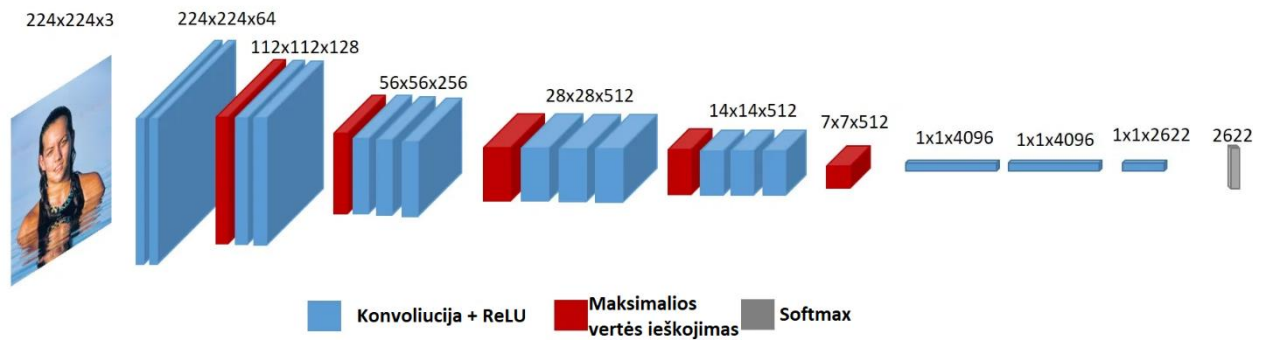
1.2 pav. „AlexNet“ modelio architektūra [22]

1.3.2. „VGGNet-16“

„VGG“ (angl. *Virtual Geometry Group*) yra grupė tyrėjų, dirbančių Oksfordo universitete. Šios grupės tyrėjai Karen Simonyan ir Andrew Zissermanas 2014 metais pasiūlė patobulintą „AlexNet“ versiją. Jie ištyrė, kaip „AlexNet“ veiktų esant daugiau sluoksnių ir didesniai gyliui, jei būtų naudojamas smulkus filtras [23]. Pagrindinė modelio idėja, kad su dideliu kiekiu 3x3 filtrų, galima gauti tokius pat gerus rezultatus kaip ir su 5x5 ar 7x7 filtrais [24].

„VGG“ modelis, apskaičiuodamas vaizdo konvoliucijas, visada naudoja 3x3 dydžio filtras ir „slenkančio lango“ principu pasislenka 1 pikselio žingsniu. Maksimali vertė yra ieškoma 2x2 regionuose, paslenkant filtrą 2 pikseliais po maksimalios vertės radimo. Dėl šios priežasties, kiekvieną kartą atlikus maksimalios vertės ieškojimus, pirmieji 2 matmenys sumažėja perpus. Tarp paslėptų sluoksnių ir tarp visiškai sujungtų sluoksnių yra atliekamos aktyvacijos funkcijos, panaudojant ReLU tiesę.

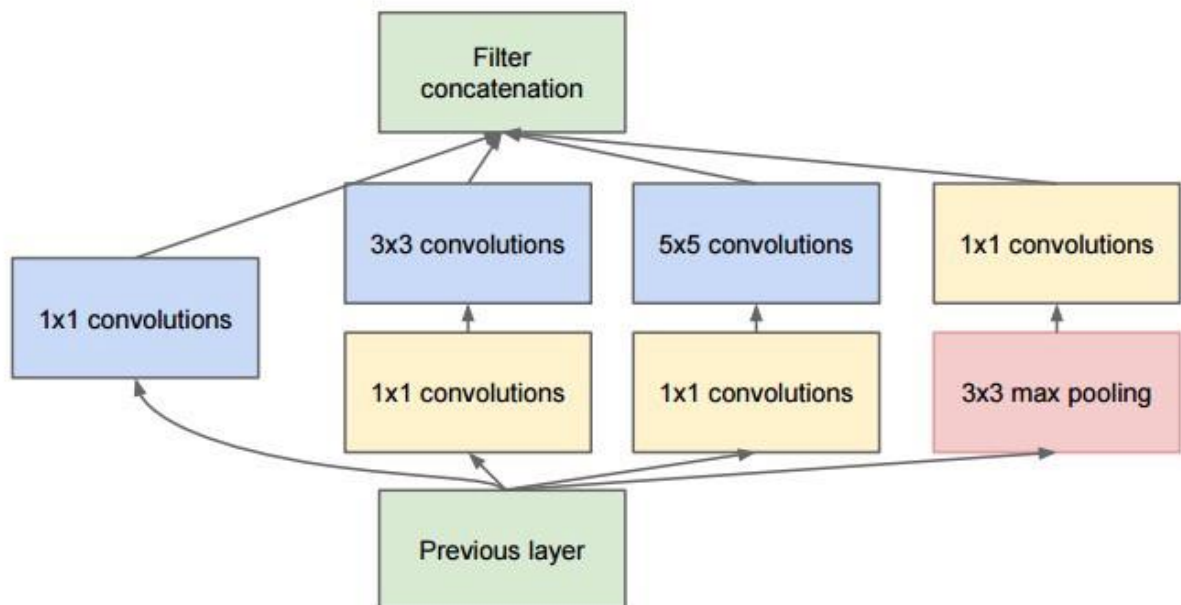
Šio modelio įvestis – RGB 224x224 raiškos vaizdas. Visas modelis gali būti aprašytas konvoliucijų filtrų kiekiu ir kada buvo panaudota maksimalių verčių paieška. Pirmame ir antrame sluoksnyje yra naudojami 64-i filtras, su kuriais yra apskaičiuojamos konvoliucijos. Pagal filtrų skaičių, šiuose sluoksniuose matmenys pakinta į 224x224x64. Po antro sluoksnio apskaičiuojamos maksimalios vertės srityse. Trečiame ir ketvirtame sluoksniuose naudojami 128-i filtras. Matmenys pakinta į 112x112x128. Po ketvirto sluoksnio apskaičiuojamos maksimalios vertės srityse. Penktame, šeštame ir septintame sluoksniuose naudojami 256-i filtras. Matmenys pakinta į 56x56x256. Po septinto sluoksnio apskaičiuojamos maksimalios vertės srityse. Aštuntame, devintame ir dešimtame sluoksniuose naudojama 512-ika filtras. Matmenys pakinta į 28x28x512. Po dešimto sluoksnio apskaičiuojamos maksimalios vertės srityse. Vienuoliktame, dvyliktame ir tryliktame sluoksniuose naudojama 512-ika filtrų. Matmenys pakinta į 14x14x512. Po trylikto sluoksnio apskaičiuojamos maksimalios vertės srityse. Apskaičiavus trylikto sluoksnio maksimalias vertes srityse, vertės yra sujungiamos į 1x1x4096 eilę ir sukuriami 3 visiškai sujungti sluoksniai [23]. Visa modelio architektūra pavaizduota 1.3 pav.



1.3 pav. „VGG-16“ modelio architektūra [25]

1.3.3. „GoogLeNet“

2014 metais „Google“ tyrėjų komanda dalyvavo konvoliucinių neuroninių tinklų projekte, kurio vienas iš tikslų – optimizuoti konvoliucinius neuroninius tinklus, sumažinant parametrų kiekį. Komanda šį tikslą pasiekė naudodama 1x1 konvoliucijas. Šios 1x1 konvoliucijos yra naudojamos redukuoti parametrų skaičių. Siekiant efektyvesnio skaičiavimo, 1x1 konvoliucijos yra atliekamos prieš 3x3 ir 5x5 konvoliucijas, taip sumažinamas kiekis parametrų, pateikiamų sudėtingesniems skaičiavimams [26]. Lyginant su tais pačiais metais sukurtu modeliu „VGGNet-16“, „GoogLeNet“ turi daugiau kaip 10 kartų mažiau parametrų, o gaunami rezultatai yra panašaus tikslumo [18].



1.4 pav. „GoogLeNet“ modulis [26]

Šiame modelyje yra naudojami moduliai, iš kurių yra sudedama architektūra. Modulį sudaro 1x1, 3x3, 5x5 ir maksimalios vertės paieškos konvoliucijos. Šių konvoliucijų išsidėstymas modulyje pavaizduotas 1.4. pav. Visuose sluoksniuose, išskyrus pirmą, apskaičiuojant konvoliucijas „slenkančio lango“ principu, paslinkimas atliekamas 1 pikselio žingsniu. Įvestis – RGB 224x224 raiškos vaizdas. Pirmame sluoksnyje yra atliekamos konvoliucijos su 7x7 filtru, paslenkant filtrą 2 pikselių žingsniu. Prieš antrą sluoksnį yra atliekamas lokalaus maksimumo ieškojimas, naudojant 3x3

filtrą, slenkant 2 pikselių žingsniu. Antrame sluoksnyje yra atliekama 1x1 filtro konvoliucijos, išvestis apskaičiuojama su 3x3 filtro konvoliucijomis. Prieš trečią sluoksnį yra atliekamas lokalaus maksimalios vertės ieškojimas, naudojant 3x3 filtrą, slenkant 2 pikselių žingsniu. Nuo trečio iki vienuolikto sluoksnio (imtinai) yra naudojami „GoogLeNet“ moduliai [26]. Šeštame ir devintame sluoksniuose yra panaudojamas papildomas minkštojo maksimumo (angl. *softmax*) išsišakojimas, kurio funkcinė prasmė – sumažinti nykstančio gradiento (angl. *vanishing gradient*) daromą įtaką gilesnių sluoksnių neuronų mokymui [27].

1.3.4. „ResNet-50“

Nuo „AlexNet“ sukūrimo, naujesni modeliai buvo kuriami su vis daugiau paslėptų sluoksnių. Esant daugiau paslėptų sluoksnių, galime pastebėti daugiau vaizdo požymių. Naudojant įprastinę modelių architektūrą, didinant paslėptų sluoksnių skaičių, taip pat didiname riziką, kad turėsime nykstančio gradiento problemą. Esant nykstančio gradiento problemai, gilesni sluoksniai gali būti nebemokomi, o tai sumažina modelio veiksmingumą [28]. Šiai problemai spręsti buvo sukurtas „ResNet“ modelis. Jis yra sudarytas iš atminties (angl. *residual*) blokų. Šio bloko funkcinė prasmė – išsaugoti ankstesnes aktyvacijos vertes ir jas vėliau panaudoti kitai aktyvacijai skaičiuoti. „ResNet“ aktyvacijos funkcija yra apskaičiuojama [29]:

$$a_{l+2} = g(z_{l+2} + a_l); \quad (4)$$

čia a – aktyvacijos, l – sluoksnis, g – aktyvacijos funkcija, z – aktyvacijų ir svorių sandauga.

„ResNet“ modelyje įvestis – RGB 224x224 raiškos vaizdas. Konvoliucijų skaičiavimo metu pirmame sluoksnyje yra naudojamas 7x7 filtras, vėlesniuose sluoksniuose naudojami 3x3 filtrai. Tarp atminties blokų yra sluoksniai, kuriuose filtrai atliekami su 2 pikselių žingsniu ir filtrų skaičius yra padvigubinamas (toliau vadinamas sumažinimo (angl. *downsampling*) sluoksniu). Pirmame sluoksnyje yra atliekamos konvoliucijos su 64-iais filtrais, 2 pikselių žingsniu. Išvestyje ieškomos maksimalios vertės zonoje kas 2 pikselių žingsniu. Visas likęs neuroninis tinklas yra sudarytas iš atminties blokų. Visuose juose yra atliekama konvoliucija. 10, 22, 40 sluoksniai yra sumažinimo sluoksniai. Po 49 sluoksnio atliekamas vidutinės 3x3 zonoje vertės ieškojimas ir yra išskleidžiamas į eilutę [30].

1.3.5. Modelių palyginimas

Šiame projekte siekiama, kad pasirinktas modelis būtų kuo tikslesnis, su mažesnėmis resursų sąnaudomis. „AlexNet“ modelio privalumas yra mažiausias MAC funkcijų kiekis. Dėl šios priežasties šis modelis, iš keturių aprašytų, yra greičiausiai vykdomas. Dėl savo tiesiškumo, jis yra lengvai realizuojamas. „VGG“ modelio svorių ir MAC skaičius nusileidžia kitiems metodams, tačiau, atliktame modelių palyginime [31] buvo pastebėta, kad „VGG“ veikia geriau nei „AlexNet“ ar „GoogLeNet“, triukšmui esant vaizde. „GoogLeNet“ architektūros didžiausias privalumas – mažiausias saugomų svorių skaičius bei aukštas tikslumo lygis. „ResNet“ privalumas yra aukščiausia tikslumo vertė, lyginant su kitais 3 modeliais [18].

1.1 lentelė. Metodų kiekybinis palyginimas [18]

Parametrai	„AlexNet“	„VGG“	„GoogLeNet“	„ResNet“
Klydimas, %	16,4	7,4	6,7	5,3
Svorių skaičius	61 milijonas	138 milijonai	7 milijonai	25,5 milijonai

MAC skaičius	724 milijonai	15,5 milijardai	1,43 milijardai	3,9 milijardai
--------------	---------------	-----------------	-----------------	----------------

1.4. Objektų lokalizacijos modeliai

Vieni iš labiausiai naudojamų objektų atpažinimo modelių yra „YOLO“ („You Only Look Once“) ir „SSD“ („Single Shot Detection“) [32]. Šie modeliai yra populiarūs dėl jų spartos. „YOLO“ ir „SSD“ didžiausias skirtumas – kaip naudojami apgaubtieji tūriai (angl. *bounding box*). Abu modeliai padalija vaizdinę medžiagą į regionus. „YOLO“ modelyje kiekviename regione yra įvertinama apgaubtųjų tūrių vieta, kuriuose gali būti objektas. Kiekviename įvertintame apgaubtajame tūryje yra apskaičiuojamas pasitikėjimo koeficientas, kuris nurodo tikimybę, kad egzistuoja objektas apgaubtajame tūryje [33]. „SSD“ atveju apgaubtieji tūriai yra nustatyti iš anksto [34]. „YOLO“ modelyje mažo pasitikėjimo koeficiento plotai yra pašalinami, o likę yra naudojami aptikti galimą objektą. Jei gretimų apgaubtųjų tūrių spėjamos klasės sutampa, apgaubtieji tūriai yra sujungiami. „SSD“ modelyje, panaudojant požymių žemėlapius, tikrinama kokia tikimybė, kad tame tūryje yra objektas bei jo tikslesnė pozicija [34].

„YOLO“ modelio trūkumai yra, kad modeliui sunku atskirti šalia esančius smulkius objektus, naudojant neįprastus kraštinių santykius (angl. *aspect ratio*) sumažėja modelio tikslumas. „SSD“ pagrindinis trūkumas - fiksuoti apgaubtieji plotai [35].

1.5. Optimizatoriai

Optimizatoriai yra naudojami atrasti tikimiausias reikšmes nuostolių funkcijos (angl. *cost function*) vertės minimizavimui.

Gradientinis nusileidimas yra optimizatorių pagrindas. Funkcijos gradientas parodo sparčiausią vertės kitimo kryptį. Gradiento skaičiavimo algoritmas baigiamas, kai pasiekiamas lokalus ar globalus funkcijos minimumas. Norint sumažinti nuostolių funkcijos vertę, turime imti neigiamą gradiento kryptį. Kompiuterinėse programose gradientinis nusileidimas gali būti apskaičiuojamas 2 variantais: iteraciniu ir matriciniu. Iteraciniame variante, kintamojo priskyrimo funkcija [36]:

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) = \theta_j - \alpha \nabla_j J(\theta); \quad (5)$$

čia θ – funkcijos parametrų rinkinys, j – j -tasis parametras, α – mokymosi greičio koeficientas, $J(\theta)$ – nuostolių funkcija, ∇ - gradientas.

Matricinė gradientinio nusileidimo forma yra vadinama normaline lygtimi (angl. *normal equation*). Ji apskaičiuoja galutines θ vertes per vieną iteraciją [36]:

$$\theta = X^T (X X^T)^{-1} y; \quad (6)$$

čia θ – funkcijos parametrų rinkinys, X – parametrų verčių matrica, y – išvesties vertė.

Palyginus matricinę ir iteracinę formas, galime pastebėti šiuos skirtumus: normalinėje lygtyje nėra mokymosi greičio koeficiento, normalinėje lygtyje koeficientai apskaičiuojami be iteracijų, normalinėje lygtyje visada randamas globalus minimumas, o iteracinėje gali būti rastas lokalus minimumas. Normalinės funkcijos pagrindiniai trūkumai – aukštesni atminties reikalavimai, lėtesnis skaičiavimas, nei iteraciniu variantu, kai yra naudojamas didesnis duomenų kiekis. Normalinę funkciją rekomenduojama naudoti, esant mažesniai, nei 10000 verčių duomenų kiekiui. Atliekant skaičiavimus su daugiau nei 10000 verčių duomenų kiekiais, galime sparčiau apskaičiuoti minimumą, naudodami iteracinį gradientinio nusileidimo variantą [37].

Kaip optimizatorius, gradientinis nusileidimas turi svarbų trūkumą – mokymosi greičio koeficientas yra konstanta. Kad gradientinis nusileidimas veiktų optimaliai, reikia ištestuoti modelį su skirtingomis mokymosi greičio vertėmis ir pasirinkti tinkamiausią vertę. Pasirinkus per aukštą mokymosi greičio koeficiento vertę, gali įvykti divergencija ir lokalus minimumas nebus atrastas, o naudojant per žemos vertės koeficientą, modelio vertės nekinta pakankamai sparčiai, kad būtų atrastas lokalus minimumas arba jis atrandamas neefektyviomis laiko sąnaudomis [38]. Kitas svarbus trūkumas – naudojamas tokios pat vertės mokymosi greičio koeficientas visiems parametrams. Jeigu bent vienas iš parametru išsiskiria savo pasiskirstymu, modelis gali būti apmokomas neoptimaliai, nes išsiskiriantiems parametrams mokymosi greičio koeficientas bus netinkamas apmokymui (parametre susiduriama su per aukštos/žemos vertės koeficiento problema) [39].

„AdaGrad“ (Adaptyvaus gradiento, angl. *Adaptive Gradient*) yra patobulintas gradientinio nusileidimo optimizatorius, kuriame yra atsižvelgiama į minėtus trūkumus. Šiame optimizatoriuje naudojama vienoda mokymosi greičio vertė, tačiau skaičiavimų metu ji yra naudojama taip, kad būtų atsižvelgiama į parametru pasiskirstymą kiekvienos iteracijos metu [40]. „AdaGrad“ algoritmas yra apskaičiuojamas [39]:

$$x_{j+1} = \prod_x^{G_j^{\frac{1}{2}}} x_j - \alpha_0 G_j^{-\frac{1}{2}} g_j; \quad (7)$$

čia x – parametru rinkinys, j – j -tasis parametras, g – gradientas, G – kovariacijos matrica.

Šioje funkcijoje kovariacijos matricos įstrižinės vertės kontroliuoja mokymosi greičio koeficientu vertes parametrams. Norint paspartinti optimizatoriaus skaičiavimo laiką, kovariacijos matricą galime pakeisti įstrižine kovariacijos matrica. Kovariacijos matricoje kaupiamos buvusių gradientu vertės. Kiekvienoje iteracijoje kovariacijos matrica atnaujinama pagal šią funkciją [41]:

$$G = G + g * g^T; \quad (8)$$

čia G – kovariacijos matrica, g – gradientas.

Šis verčių atnaujinimas iteracijų pabaigoje gali sukelti problemų, nes kovariacijos matricos vertės vis auga, atitinkamai, efektyvi mokymosi greičio koeficiento vertė mažėja. Modelio mokymo pabaigoje vertės beveik nekinta [41].

Šiuo metu vienas iš populiariausių optimizatorių yra „Adam“ (angl. *Adaptive Moment Estimation*) [42]. Šiame optimizatoriuje parametru vertės yra apskaičiuojamos naudojant judesio kiekį. Judesio kiekis yra naudojamas sandaugoje su mokymosi greičio koeficientu, pakeičiant efektingą jo vertę. Judesio kiekis funkcijoje paspartina judėjimą link minimumo ir slopina pašalinius virpėjimus [40]. „Adam“ optimizatorius, lyginant su „AdaGrad“, yra pranašesnis dėl mažesnio kiekio atminties, reikalingo atliekant skaičiavimus, nes „Adam“ optimizatoriuje, matricos daugyba su transponuota savimi, yra pakeičiama dviejų vektorių atitinkamų narių sandauga. „Adam“ optimizatorius saugo praeitos iteracijos gradientu vertes. Pirmo ir antro laipsnio gradientams saugoti yra naudojami šie kintamieji [41]:

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_{t-1}; \quad (9)$$

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_{t-1} \odot g_{t-1}; \quad (10)$$

čia m , v – tendencingi (angl. *bias*) judesio kiekio parametrai, β – smukimo (angl. *decay*) parametras, g – gradientas.

„Adam“ algoritme siūloma naudoti 0,9 ir 0,999 atitinkamų β parametrų vertes [43]. Prieš įkeliant judesio kiekio parametrus į „Adam“ algoritmą, pašalinamas šių parametrų tendencingumas šiomis formulėmis [41,43]:

$$\hat{m}^{(t)} = \frac{m^{(t)}}{1 - \beta_1^t}; \quad (11)$$

$$\hat{v}^{(t)} = \frac{v^{(t)}}{1 - \beta_2^t}; \quad (12)$$

čia t – pakėlimo laipsnis.

„Adam“ algoritmas yra apskaičiuojamas panaudojus šią formulę [41,43]:

$$\theta_j = \theta_{j-1} - \alpha \hat{v}_j^{-\frac{1}{2}} \odot \hat{m}_j; \quad (13)$$

1.6. Panašūs darbai

Lietuvoje vairuotojų, kalbančių telefonu, nustatymui yra naudojamos kameros ir jų įrašus peržiūri paskirti žmonės [44]. Galimai daugumoje šalių yra naudojamas panašus sprendimas. Tai yra monotoniškas darbas, kuris galėtų būti automatizuotas. Šios problemos sprendimas yra aktualus visame pasaulyje. Kiekvienas tyrėjas turi savo unikalų sprendimo variantą.

Kanados „Vaterlo“ Universiteto doktorantūros studentas Chaojie Ou, konsultuojant Dr. Fakhri Karrėjui, sukūrė disertaciją „Deep Learning-based Driver Behavior Modeling and Analysis“ [45]. Disertacija sudaryta iš 3 darbų: 1) išsiblaškiusio vairuotojo atpažinimo pagerinimas, naudojant generatyvinį priešininkų tinklą (angl. *generative adversarial network*); 2) išsiblaškiusio vairuotojo atpažinimas, esant įvairiom apšvietimo sąlygom; 3) vairuotojo manevravimo spėjimo sistema. Pirmame darbe buvo naudoti 2 duomenų rinkiniai: 1) virš 1200 nuotraukų iš interneto; 2) 22800 nuotraukų, nufotografuotų pačio studento ir stacionarios pozicijos. Nuotraukos iš interneto buvo iš transporto priemonės vidaus. Surinktos nuotraukos buvo nufotografuotos iš įvairių transporto priemonės vietų.



1.5 pav. Chaojie Ou surinktų nuotraukų iš interneto pavyzdžiai [45]

Vairuotojai buvo klasifikuojami į neišsiblaškiusius, kalbančius telefonu ir rašančius tekstines žinutes. Šiame darbe buvo tikrinamas tikslumas su 4 modeliais. Siekiant tausoti skaičiavimo išteklius, nes buvo siekiama naudoti modelius mažos skaičiavimo galios prietaisuose, buvo naudojami 3 skaičiavimo ištekliais efektyvūs modeliai ir vienas intensyvus. Ištekliais intensyviausias modelis buvo „ResNet50“. Tyrime taip pat buvo naudoti „ShuffleNet“ [46], „MobileNet“ [47],

„MobileNetv2“ [48]. Buvo naudojamas „Adam“ optimizatorius. Prieš įvedant nuotraukas apmokymui, jos būdavo apdorojamos duomenų papildymo, naudojant generatyvinį priešininkų tinklą (angl. *generative adversarial network*), tačiau šis sprendimas nepasiteisino dėl prastos kokybės (neryškių, be kontūrų) sugeneruotų nuotraukų.

Šios disertacijos rezultatai: tyrime, kai buvo naudojamas vienas duomenų rinkinys apmokymui, o kitas testavimui, buvo pastebėta, kad naudojant įvairios pozicijos nuotraukas apmokymui, buvo gautas didesnis tikslumas (62,61 %), nei naudojant stacionarios pozicijos (45,32 %). Antrame šio darbo tyrime pastebėta, kad „ResNet50“ yra tiksliausias iš 4 modelių [45].

Jungtinių Amerikos Valstijų startuolis „Cellpathy“ yra sukūręs programinę įrangą mobiliam telefonui, kuri atskiria ar telefono naudotojas yra vairuojantis. Programa naudoja telefono giroskopo, akselerometro, GPS, mikrofono duomenis, nuspręsti ar transporto priemonė yra vairuojama [49]. Programoje, galima nustatyti, kad vartotojas yra keleivis, tada yra atliekamos įvairios užduotys, kurios yra sunkiai atliekamos vairuotojui (nustatyti telefoną tam tikru kampu, paliesti ekraną specifinėje vietoje ir suvesti kodą, nufotografuoti vairuotoją) [50]. Esant užduočiai, kad vartotojas turi nufotografuoti vairuotoją, gauta nuotrauka yra apdorojama kompiuterinės regos ir konvoliucinių neuroninių tinklų. Šioje nuotraukoje ieškoma ar matomas vairas ir abi rankos ant vairo [50]. Naudojamo modelio architektūra nėra viešai skelbiama startuolio puslapyje ir patentuose.

Japonijoje „Digital Media Professionals“ yra sukūrę vairuotojo stebėjimo programinę įrangą, nustatyti ar vairuotojas yra išsiblaškęs. Jie šią programinę įrangą nuomoja automobilių nuomos kompanijai „Sumitomo Mitsui Auto Service“. Ši programinė įranga yra skirta vidinėms kameroms. Straipsnyje [51] rašoma, kad vartotojo elgsena yra nustatoma pagal matomos odos kiekio pokyčius keliavimo metu. Jei vairuotojas kalba telefonu ar yra kitaip išsiblaškęs, tuomet yra siunčiami duomenys automobilių nuomos kompanijai, kuri nusprendžia tolimesnę veiklą [51]. Vidinė programos architektūra puslapyje neskelbiama, tačiau kompanijos puslapyje rodoma, kad jie seka vartotojo veidą, laikyseną [52].

Kuriant sistemas, kuriose siekiama aptikti vairuotoją, kalbantį telefonu, susiduriama su sugrupuoto duomenų rinkinio problema [45], nes šie duomenys internete sunkiai atrandami dėl privatumo įstatymų. Atrasti uždarnos televizijos sistemos (angl. *CCTV*) nuotraukas, kuriose vairuotojas kalbėtų telefonu yra beveik neįmanoma, nebent paieška būtų atlikta kinų kalba. Dėl šios priežasties, šia tema atliktiems tyrimams, yra naudojamos nuotraukos iš transporto priemonės vidaus, nes jos yra lengviau pasiekiamos asortimentinių (angl. *stock*) nuotraukų dėka. Susikūrus duomenų rinkinį iš šių nuotraukų, lengva sukurti pačiam savo nuotraukų rinkinį iš transporto priemonės vidaus ir naudoti tai apmokymui arba testavimui. Pasirinkus šį metodą susiduriama su taikymo realiose situacijose problema, nes reikia naudoti programinę įrangą transporto priemonės viduje, o vartotojas šios priemonės savo noru nenaudos (išskyrus siaurą kompanijų sritį (pvz. logistika)). Siekiant savo darbu paveikti kuo didesnę žmonių kiekį, reikia pritaikyti šį sprendimą, kai vairuotojo elgsenys yra nustatomas iš transporto priemonės ribų.

2. Vairuotojo, kalbančio telefonu, nustatymo projektavimas

2.1. Projekto planas

Šiame projekte yra svarbiausios 2 dalys: objektų atpažinimas ir objektų nustatymas. Šiame projekte laikome, kad įvestis bus nuotrauka, o ne vaizdo įrašas. Šis projektas yra atliekamas „Windows 10“ aplinkoje. Testavimo apimties sumetimais, „Linux“ aplinkoje funkcionalumas nėra užtikrinamas. Projekto tikslui pasiekti, galime rinktis iš daugumos programavimo kalbų, tačiau šiam projektui yra pasirinkta „Python“ programavimo kalba, nes ji yra populiariausia programavimo kalba mašininiam mokymui [53] ir turi platų pasirinkimą mašininio mokymo karkasų (angl. *framework*). Renkantis karkasą, kurio pagrindu bus sukurta programa, norima, kad jis būtų vienas iš populiariausių, nes esant plačiai bendruomenei, bus lengviau atrasti atsiradusių problemų sprendimus. Dėl šios priežasties buvo pasirinkta rinktis iš 2 karkasų palaikomų „Google“. „Google“ palaiko „TensorFlow“ ir „Keras“ karkasus. „TensorFlow“ karkaso pagrindinis privalumas: modelių kūrimo specifiškumas, o „Keras“ lengvumas šių modelių kūrimo. Kadangi šiame projekte bus naudojamos jau sukurtos modelių architektūros, patogiau būtų naudoti „Keras“, todėl ši biblioteka ir yra pasirinkama.

Objektų atpažinimui galime rinktis vieną iš anksčiau aptartų modelių, svarbiausia, kad būtų atrandama transporto priemonė virš 95 % tikslumu, esant jai nuotraukoje. Esant pakankamai panašioms treniravimo ir testavimo duomenų rinkiniams, gali užtekti tik transporto priemonės atpažinimo. Turėtumėme gauti aukštesnį tikslumą, jeigu į įvestį duotumėme vairuotojo nuotrauką, nes sumažintumėme tikimybę, kad modelis išmokytų atskirti nuotraukas pagal transporto priemonės bruožus. Iš objektų aptikimo modelio tikimasi, kad jis gali atrasti ir transporto priemonės vairuotojus. Nuotraukoje atradus vairuotoją ir transporto priemonę, prioritetas turėtų būti skiriamas vairuotojo nuotraukos perkėlimui į įvestį objektų atpažinimui.

Objektų nustatyme bus lyginami keturi skirtingi konvoliucinių neuroninių tinklų modeliai. Modelis sprendimui bus parenkamas pagal tikslumą, teisingai spėtų teigiamų ir F vertės rezultatus. Tikimasi, kad F_1 vertė bus aukštesnė nei 50 %. Objektų nustatyme tikimasi, kad bus atrandami bruožai, pagal kuriuos bus nustatoma ar vairuotojas kalba telefonu, esančius vairuotojo pozicijoje (t.y. jeigu įvestis yra transporto priemonė, tikimasi, kad nustatymas nepriklausys nuo transporto priemonės). Objektų nustatymas bus skirstomas į dvi sritis: transporto priemonėje esančio vairuotojo nustatymas ar jis kalba telefonu ir vairuotojo nustatymas ar jis kalba telefonu. Pirmuoju atveju įvestyje yra transporto priemonė ir vairuotojas, antruoju tik vairuotojas. Modelio greitaveikai reikalavimų nėra.

2.2. Nefunkciniai reikalavimai

Kuriamo sprendimo nefunkciniai reikalavimai:

1. Programavimo kalba – Python;
2. naudojama „Keras“ biblioteka;
3. veikia „Windows 10“ operacinėje sistemoje;
4. vairuotojo, kalbančio telefonu, atpažinimui naudojamas „AlexNet“, „VGG“, „GoogLeNet“ arba „ResNet“ modelis;
5. F_1 vertė aukštesnė nei 50 %.

2.3. Funkciniai reikalavimai

Kuriamo sprendimo funkciniai reikalavimai:

1. Aptikti transporto priemonę nuotraukoje;
2. atpažinti ar įvestyje yra vairuotojas, kalbantis telefonu;
3. išsaugoti vairuotojo, kalbančio telefonu, vaizdinę medžiagą.

2.4. Tyrimo aplinkos projektavimas

Vairuotojo, kalbančio telefonu, nustatymo tyrimui atlikti reikia, kad būtų užtikrinama, kad galima apdoroti duomenų rinkinio nuotraukas, naudojant objektų aptikimą, kad duomenų rinkinio nuotraukas būtų galima paruošti modelio mokymui. Modelių mokymams reikia pateikti aplinką, kurioje modeliai galėtų naudoti apdorotas duomenų rinkinio nuotraukas ir mokyti modelius. Modelių svorių rezultatams patikrinti reikia sukurti aplinką, kurioje galima užkrauti apmokytus modelių svorius ir patikrinti jų įvertinimus. Norint, kad vyktų sklandus tyrimas, šias tris dalis reikia atskirti. Skaidymo pagrindiniai privalumai: mažesnė tikimybė, kad bus įvelta klaida, kuri trukdys atlikti kitos srities tyrimą, greitesnė sparta, nes programos krovimas truks trumpiau. Skaidymo trūkumai: kodo pasikartojimas, atliekami pakeitimai didesniame kiekyje failų.

2.4.1. Objektų aptikimo programos projektavimas

Objektų aptikimo programa – pirmasis duomenų rinkinio panaudojimo atvejis. Šios programinės dalies paskirtis yra gauti vaizdinę medžiagą ir joje aptiktus objektus išsaugoti. Iš suprojektuotos programinės dalies tikimasi, kad būtų lengva atrinkti norimos klasės vaizdinę medžiagą, nes ji bus grupuojama.

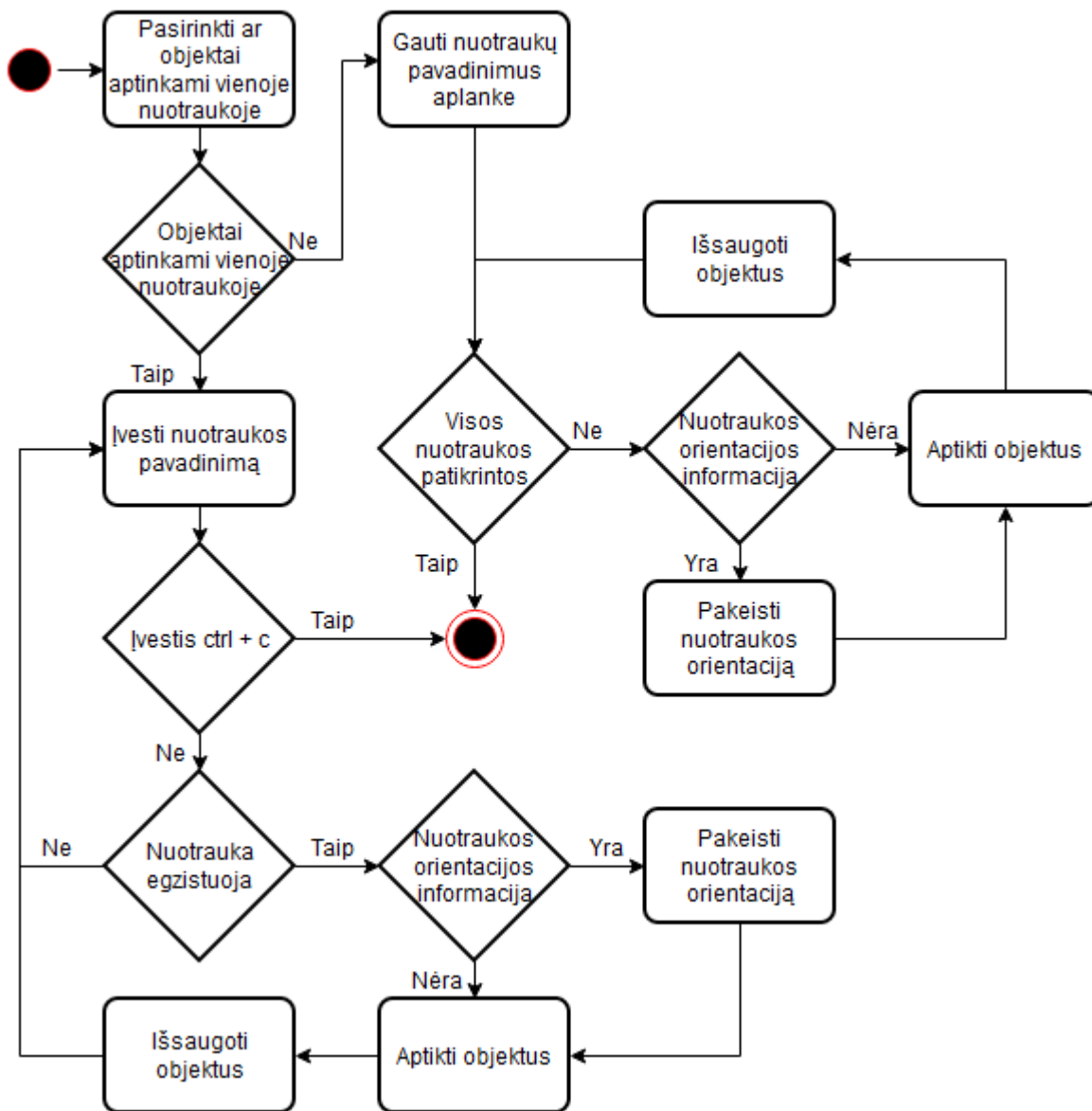
Pagrindinis objektų aptikimo programos reikalavimas – galimybė pasirinkti ar bus apdorojama specifinė nuotrauka ar bus apdorojamas visas aplankas. Ši galimybė pasirinkti yra reikalinga, nes yra atveju, kai norime ištirti kaip veikia objektų aptikimas vienoje nuotraukoje arba norime atlikti tolimesnį tyrimą su viena nuotrauka, todėl mes ją norime paruošti tolimesniam tyrimui. Galimybė pasirinkti, kad galėtumėme naudoti objektų aptikimą visam aplankui yra reikalinga, kai norime atlikti tyrimą visam aplankui, o ne daliai jo. Šiame objektų aptikimo reikalavime taip pat galime pasirinkti galimybę apdoroti aplanke esančio aplanko nuotraukas. Ši galimybė leistų vienu metu apdoroti visą duomenų rinkinį, tačiau šis sprendimo būdas reikalautų papildomų aplankų kūrimo atrastiems objektams rūšiuoti, nes, nesant šiam papildomam aplankų kūrimui, skirtingai suklasifikuotos nuotraukos būtų sukuriamos bendrame aplanke. Jeigu skirtingai suklasifikuotos nuotraukos būtų sukuriamos bendrame aplanke, reikėtų jas išskirti failo pavadinimu, kad būtų galima greitai sugrupuoti gautą vaizdinę medžiagą. Paskutinis aptartas sprendimas padidina programos sudėtingumą ir gali pasunkinti mūsų aptiktų objektų grupavimą, todėl jo yra atsisakoma, siekiant išvengti klaidų. Šiame reikalavime būtų privalumas, jeigu, pasirenkant nuotrauką, po nuotraukos apdoravimo būtų gaunama kita užklausa apie pasirenkamą nuotrauką. Šis privalumas mums leistų pasirinkti kelias nuotraukas, kurias mes norime apdoroti, neatliekant apdoravimo visom aplanke esančiom nuotraukom. Pasirenkant nuotrauką, nuotraukų failų pavadinimai būtų gaunami iš įvesties konsolėje, o ne tekstinio failo ar programos viduje esančių kintamųjų. Jeigu pasirenkama, kad apdorojamas aplankas, mes naudosisime programoje esantį kintamąjį, nes norima išvengti problemų, kada programa gali interpretuoti raidę po pasivirojo brūkšnio kaip teksto formatavimą.

Dalis duomenų rinkinio bus fotografuojama. Fotografuotos nuotraukos saugo nuotraukos savybęse informaciją apie nuotraukos orientaciją. Objektų aptikimo programos veikimo metu yra būtina aptikti šią informaciją ir ją panaudoti, kad būtų paverčiama nuotrauka orientacija, kurioje buvo gaunama nuotrauka, prieš aptinkant objektus nuotraukoje. Neįvykdžius šio reikalavimo, objektų aptikimas

nevyks kaip tikimasi, nes geriausiai modeliai aptinka objektus jų standartinėje orientacijoje. Nuotraukose iš interneto ši informacija gali būti pašalinta, todėl šis reikalavimas yra skirtas užtikrinti fotografuoto duomenų rinkinio panaudojimą.

Aptikti objektai turi būti saugojami. Saugojimo katalogo vieta yra nurodoma kintamajame programme, nes jeigu būtų vedama saugojimo katalogo vieta objektų aptikimo metu, gali įvykti žmogiška klaida ir nuotraukos gali būti išsaugomos nenorimame aplanke. Aptiktų objektų saugojimo metu reikia užtikrinti, kad kiekviena nuotrauka bus pavadinta unikaliumi. Jeigu nuotrauka nebus pavadinta unikaliumi, nauja nuotrauka pakeis senąją. Taip prarastume dalį atrastų objektų nuotraukų. Failo pavadinimo unikalumą galime užtikrinti, jeigu prieš nuotraukos saugojimą patikrinsime ar jau egzistuoja toks failas. Šios realizacijos atlikti nenorima, nes galime užtikrinti nuotraukų pavadinimo unikalumą ir be patikrinimo. Pagrindinė užtikrinimo idėja – pridėti 10 atsitiktinių raidžių ir skaičių eilutę prie failo pavadinimo. Šiame sprendime egzistuoja daugiau nei $8 \cdot 10^{17}$ skirtingų raidžių ir skaičių derinių. Naudodami tik atsitiktinę eilutę galime užtikrinti failų pavadinimų unikalumą. Pagrindinė failo pavadinimo dalis bus kūrimo laikas. Šis sprendimas padės mus greitai suskirstyti nuotraukas, kada jos buvo sukurtos, nenaudojant aplanko rūšiavimo pagal laiką. Norint atlikti greitą sugrupavimą tarp įvairių klasių nuotraukų, mums reikia įvesti šią informaciją į failo pavadinimą. Šiame projekte tai gali būti išsprendžiama, panaudojant išskirtinį simbolį, kuris nėra naudojamas failo pavadinime ir prie šio simbolio parašyti klasės informaciją.

Suprojektuota objektų aptikimo programa pavaizduota UML veiklos diagramoje:



2.1 pav. Objektų aptikimo veiklos diagrama

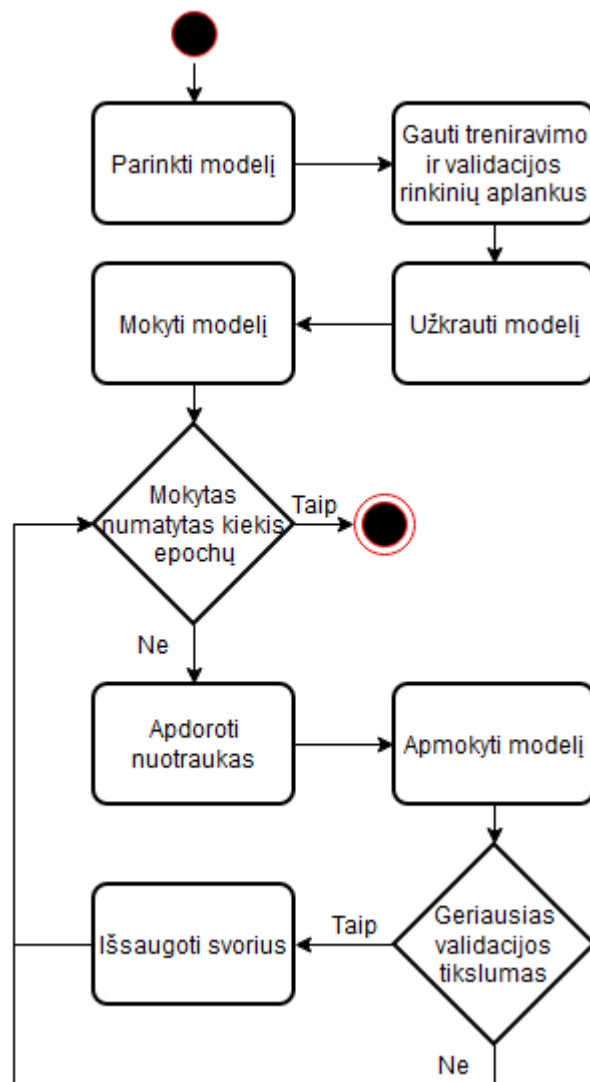
Visus šioje veiklos diagramoje pavaizduotuose veiksmus atliks programa, išskyrus vartotojo pasirinkimą, ar objektai yra aptinkami vienoje nuotraukoje, ir nuotraukų pavadinimų įvestį.

2.4.2. Modelių mokymo programos projektavimas

Modelių mokymo programoje panaudojame duomenų apdorojimo metu gautas nuotraukas ir su jomis apmokome konvoliucinių neuroninių tinklų modelį. Pats svarbiausias šios programos reikalavimas – visi modeliai vienoje programoje. Projekte tiriamų modelių mokymo procesas yra identiškas, todėl visi modeliai turėtų būti vienoje programoje, kad vieno modelio programoje padaryti pakeitimai nebūtų dar kartą rašomi kitiems 3 modeliams. Modelių, aplančių ir kitų kintamųjų vertės turi būti gaunamos programos viduje be vartotojo įvesties. Modelių mokymo metu bus daromi programų pakeitimai, kurie bus susiję su atliekamais tyrimais, todėl programos kodas visada bus atidarytas mokymo metu. Įvestis trukdytų greitai atlikti tyrimus, nes paleidus programą, reikėtų laukti jos užsikrovimo, įvesti vertes ir tikrinti, kad įvestis yra tinkama. Jeigu visos įvestys bus nustatytos programoje, paleidus programą, bus galima grįžti prie projekto tyrimų. Šioje programoje, skirtingai negu objektų aptikimo programoje, mes turime galėti gauti nuotraukas iš aplanke esančio aplanke. Iš

objektų aptikimo programos, nuotraukos bus išskirtos į aplankus, kurių pavadinimai nurodys ar tai yra mokymo ar testavimo duomenų rinkinys, o viduje aplanko nuotraukos bus išskirstytos į aplankus, kuriuose vairuotojai, kalbantys telefonu ir vairuotojai, nekalbantys telefonu. Šiame projekte pasirinkta nuotraukas sugrupuoti pagal klasę aplankuose, ne viename aplanke, kuriame būtų galima atskirti nuotraukų klases pagal failo pavadinimą. Nuotraukų gavimas iš aplanke esančio aplanko leis mums lengviau panaudoti suklasifikuotas nuotraukas. Prieš paduodant nuotraukas modeliui apmokyti, nuotraukas normalizuosime. Modelio mokymo metu turi būti matomas modelio validacijos tikslumas. Tolimesniems modelio tyrimams, reikia, kad modeliai būtų išsaugomi. Be išsaugotų modelių, mes negalėsime patikrinti modelių tikslumo su kitais duomenų rinkiniais, negu naudotais validacijos metu. Modelių saugojimas yra svarbus, nes mes norėsime patikrinti šių modelių klaidų matricas (angl. *confusion matrix*). Modelio mokymo metu turi būti išsaugoti modelio svoriai, kada jų validacijos tikslumas yra aukščiausias, nes, galimai, aukščiausio validacijos tikslumo svoriai leidžia modeliui geriausiai generalizuoti.

Suprojektuota modelių mokymo programa pavaizduota UML veiklos diagramoje:



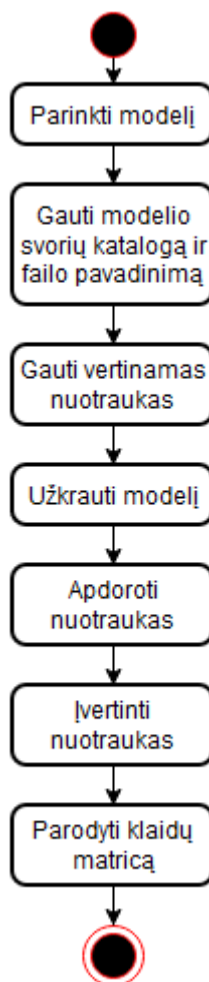
2.2 pav. Modelių mokymo veiklos diagrama

Kaip ir minėta anksčiau, programos veikimo metu, visa veikla yra atliekama programos. Modelio užkrovimo metu yra užkraunamas tik vienas modelis. Jeigu modelis turi „ImageNet“ svorius, jie yra

užkraunami šio proceso metu. Veiklos diagramoje laikoma, kad, mokymo metu, validacijos tyrimai yra mokymo proceso dalis. Nuotraukų apdorojime yra normalizacijos procesas ir kiti galimi nuotraukos modifikavimai, mokymo metu.

2.4.3. Modelio svorių vertinimo programos projektavimas

Modelio svorių vertinimo programos pagrindinė paskirtis yra patikrinti išsaugotų modelių svorių įvertinimus ir gauti jų statistiką. Idetiškomis sąlygomis, kaip ir modelių mokymo programoje, turėtų būti gaunamos aplanke esančio aplanko nuotraukos. Kaip ir praeitoje programoje, mes turėsime visus modelius vienoje programoje. Modelio ir jo svorio pavadinimo kategorijos kintamieji bus apibrėžti programoje, nes norima, kad vyktų sklandesni ir greitesni modelių vertinimų tyrimai. Šioje programoje nebus diferencijuojama į aplanko ir nuotraukos įvertinimą, nes, norint įvertinti vieną nuotrauką, bus galima sukurti aplanką, kurioje yra viena nuotrauka. Pats svarbiausias reikalavimas šioje programoje – po nuotraukų įvertinimo proceso parodyti modelio vertinimo klaidų matricą. Šis reikalavimas yra būtinas, jeigu norime gauti F_1 vertę, sužinoti ar modelis įvertina visas nuotraukas kaip viena klasę, sužinoti, kurias nuotraukas modelis teisingiau atskiria: vairuotojų, kalbančių telefonu ar vairuotojų, nekalbančių telefonu.



2.3 pav. Modelio svorių vertinimo veiklos diagrama

2.5. Tyrimo planas

Tyrimas turėtų susidaryti iš 2 pagrindinių dalių: modelių apmokymui reikalingo duomenų rinkinio gavimas ir modelių apmokymo tyrimas.

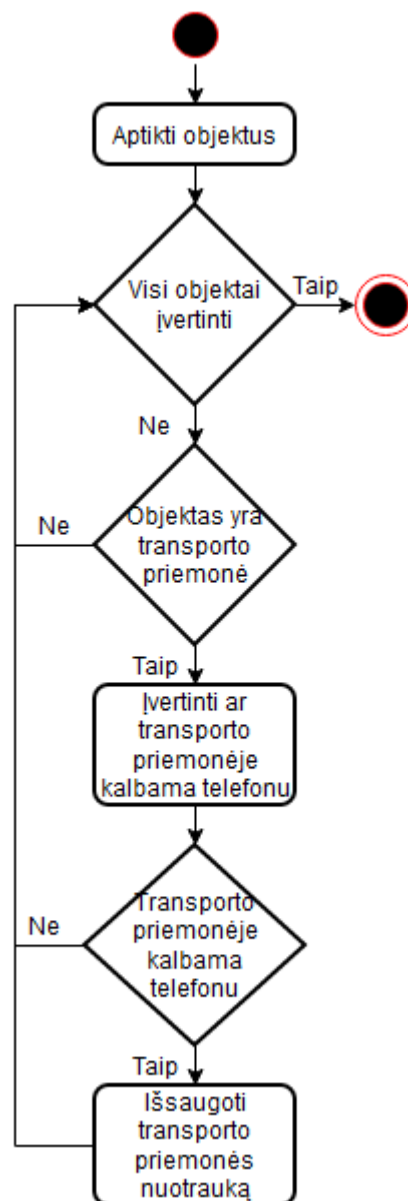
Svarbiausia šio tyrimo dalis yra atrasti duomenų rinkinį, kurį būtų galima iširti. Didžiausią, šiam tyrimui, duomenų rinkinį Lietuvoje turėtų turėti Lietuvos Respublikos policija. Jeigu būtų galima gauti anonimizuotas vairuotojų, kalbančių telefonu, nuotraukas be galimybės jomis dalintis, reikėtų pasinaudoti šia galimybe. Lietuvos Respublikos policija dalyvauja įvairiuose renginiuose, norėdami pagerinti savo įvaizdį visuomenėje ir populiarinti profesijas, esančias policijoje. Pirmasis tikslas bus susisiekti su policijos atstovais įvairiuose renginiuose ir išgauti informaciją apie galimą šių duomenų naudojimą mokymo tikslams. Šis duomenų rinkinio gavimo būdas yra mažai tikėtinas, nes, galimai, duomenys policijos duomenų bazėje nėra anonimizuojami, pašaliniamis šie duomenys negali būti suteikti pagal galiojančius nuostatus, policija rizikuotų savo reputacija, jeigu duomenys būtų gauti tretiesiems asmenims. Kitas duomenų rinkinio rinkimo variantas gali būti šio duomenų rinkinio fotografavimas. Būtų geriausia, kad šios nuotraukos būtų gaunamos fotografuojant pravažiuojančias transporto priemones iš aukšto, imituojant policijos kamerų poziciją vairuotojų atžvilgiu. Šis sprendimo variantas būtų sunkus įgyvendinti, nes reikėtų susirasti tinkamas fotografavimui vietas, nuotraukų kokybė priklausytų nuo oro sąlygų, nuotraukos būtų gaunamos retokai, fotografuojami asmenys gali sukelti grėsmę sveikatai, tačiau šis duomenų rinkinys būtų gautas panašiausiomis sąlygomis į nuotraukas, kurias turi Lietuvos Respublikos policija. Dar vienas galimas sprendimo variantas – nufotografuoti duomenų rinkinį, kai transporto priemonė nejudą. Šiuo atveju reikėtų antro asmens, kuris būtų vairuotojas arba fotografuotojas ir fotografavimo metu būtų gautos nuotraukos su įvairiomis vairuotojo pozicijomis. Pagrindinis šio sprendimo privalumas – nuotraukos gaunamos greitai, nuotraukos atvaizduoja panašias sąlygas į realias. Šio sprendimo pagrindinis trūkumas – nuotraukos bus panašios. Paskutinis sprendimo variantas, kuris bus apžvelgiamas – nuotraukų surinkimas iš interneto. Internete galima rasti įvairių nuotraukų, tarp jų turėtų būti galima atrasti ir vairuotojų, kalbančių telefonu. Šis sprendimo būdas būtų lėtesnis už duomenų rinkinio fotografavimą stovinčioje transporto priemonėje, tačiau šis duomenų rinkinys, galimai, turėtų įvairesnes nuotraukas. Šis duomenų rinkinio rinkimo variantas būtų sudėtingesnis už Chaojie Ou duomenų rinkinio gavimą [45], nes jo atliktame darbe buvo renkamos visos nuotraukos su vairuotoju, kalbančiu telefonu, tačiau šio projekto atveju, šios nuotraukos bus tik iš transporto priemonės išorės. Yra tikimybė, kad nebus atrasta pakankamai daug nuotraukų, kuriose vairuotojas kalba telefonu, nes Europoje, Šiaurės Amerikoje tai gali drausti šalyse galiojantys įstatymai.

Modelių apmokymo tyrimas bus skirstomas į dvi dalis: modelių apmokymą, kada naudojame surinktą duomenų rinkinį ir modelių tyrimą, kada naudojame pakeistą duomenų rinkinį. Pirmoje dalyje bus tikrinamas modelių tikslumas, kada mokymo metu naudojame transporto priemonių duomenų rinkinį, žmonių duomenų rinkinį ir abu duomenų rinkinius bendrai. Iš gautų rezultatų nuspręsimė, kurį duomenų rinkinį naudoti kituose tyrimuose. Šiuo metu nežinome būsimų duomenų rinkinių raiškos, tačiau yra tikimybė, kad rinkinių raiškos skirsis. Todėl antroje dalyje, jeigu bus duomenų rinkiniai su skirtingomis raiškomis, reikėtų atlikti tyrimą, kada vienas iš duomenų rinkinių sumažinamas iki kito duomenų rinkinio dydžio. Galimai šis sprendimas padidins modelių tikslumus, nes duomenų rinkiniai supanašės. Vienas iš tyrimų, kuris turi būti atliktas – mokymo rinkiniui panaudojamas duomenų papildymas. Duomenų papildymas yra naudojamas, kad būtų gaunami geresni rezultatai su duomenų rinkiniu. Jeigu duomenų rinkinys yra sąlyginai mažas, duomenų papildymas padeda apmokyti modelį be papildomo duomenų rinkinio rinkimo. Paskutinis tyrimas, kurį norima atlikti – modelių mokymas,

kai duomenų rinkinys yra pakeičiamas jo kraštinių nuotraukomis. Šiame tyrime bus norima ištirti ar galime vairuotojų, kalbančių telefonu, nustatyme naudoti kraštinių nuotraukas. Šiam tyrimui pasitvirtinus, turėtų būti spartesnis modelių apmokymas. Visi kiti būsiami tyrimai bus situaciniai ir priklausys nuo gautų rezultatų minėtuose tyrimuose.

2.6. Sprendimo programos projektavimas

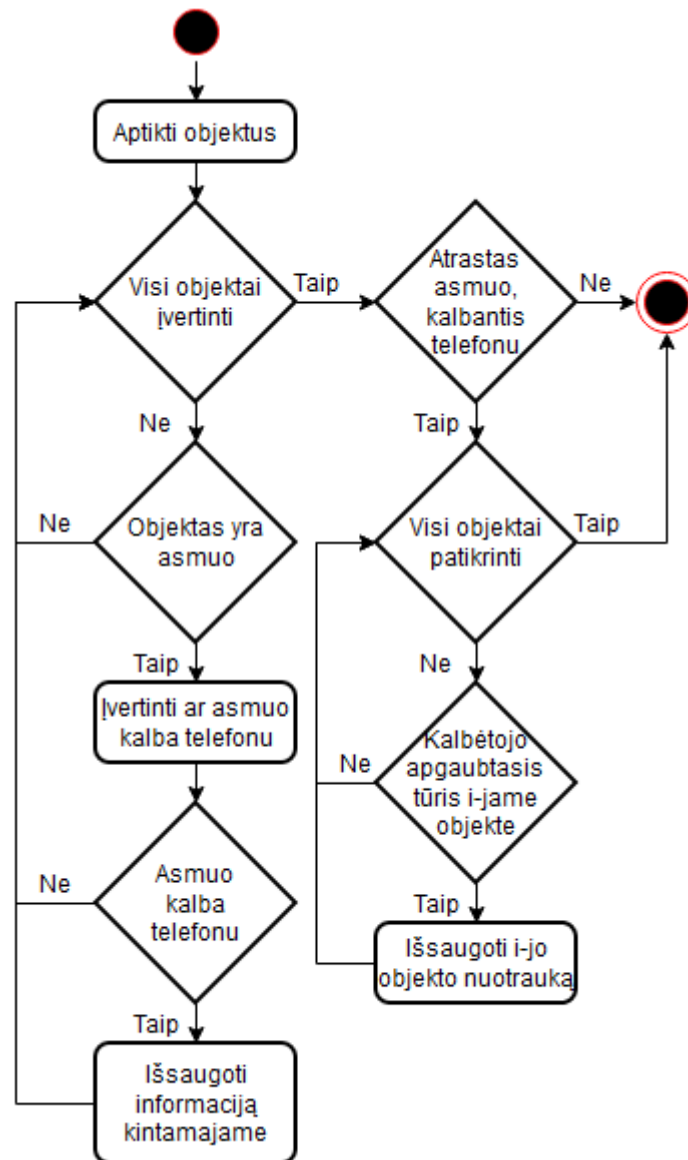
Sprendimo programa – objektų aptikimo ir modelio svorių vertinimo programų sąjunga, nes sprendime mums nėra reikalingas modelių svorių apmokymas. Galimi trys sprendimo programos variantai: 1) vairuotojas, kalbantis telefonu, nustatomas iš asmens nuotraukos; 2) vairuotojas, kalbantis telefonu, nustatomas iš transporto priemonės nuotraukos; 3) naudojami abu variantai vienu metu. Kai vairuotojas, kalbantis telefonu, yra nustatomas iš transporto priemonės nuotraukos, ši sprendimo realizacija būtų lengviausia. Galima sprendimo realizacija parodyta UML veiklos diagramoje:



2.4 pav. Sprendimo algoritmas, nustatant vairuotoją, kalbantį telefonu, iš transporto priemonės nuotraukos

Šioje UML diagramoje laikome, kad pradžios taškas yra, kai mes gauname nuotrauką, kurią turime iširti. Šiame algoritme tikriname visas transporto priemonių rūšis (automobilius, sunkvežimius, autobusus ir pan.). Šio viso algoritmo esmė – jeigu transporto priemonėje pamatome asmenį, kalbantį telefonu, mes išsaugome transporto priemonės nuotrauką. Priklausomai nuo modelio mokymo, asmuo kalbantis telefonu gali būti vairuotojas arba keleivis. Modelį galime išmokyti atpažinti ar asmuo kalba telefonu pagal vieną pusę, tada gausime, kad modelis bus išmokytas atpažinti vairuotojus, jeigu nuotraukoje vairas bus įprastoje pusėje.

Kitas sprendimo būdas – mes vairuotoją, kalbantį telefonu, nustatome pagal asmens nuotrauką. Šis sprendimo variantas pavaizduotas UML diagramoje:



2.5 pav. Sprendimo algoritmas, nustatant vairuotoją, kalbantį telefonu, iš asmens nuotraukos

Šis sprendimas yra sudėtingesnis, negu vairuotojo, kalbančio telefonu, nustatyme iš transporto priemonės nuotraukos. Pagrindinė sudėtingumo priežastis – kada nustatome, kad asmuo kalba telefonu, mes negalime išsaugoti šios nuotraukos, nes be transporto priemonės ji yra bevertė, todėl mes turime surasti objektą, kurio apgaubtajame tūryje yra visas asmens apgaubtasis tūris. Šio algoritmo metu įvertiname visus apgaubtuosius tūrius, kurie yra asmens klasės. Jeigu pastebime, kad

asmuo kalba telefonu, mes šią informaciją išsaugome kintamajame. Kitame etape, mes ieškome objektų, kurių apgaubtajame tūryje yra asmuo, kalbantis telefonu. Jeigu asmens apgaubtasis tūris nėra visiškai kito viduje, galimai, šis asmuo yra praeivis, todėl mes kito objekto apgaubtojo tūrio nesaugome.

Pasirinkimas tarp šių 2 programų algoritmų bus atliktas po atliktų projekto tyrimų. Pati svarbiausia informacija – su kuriuo duomenų rinkiniu yra pasiekiami geresni rezultatai. Sprendimo algoritmas, kada yra naudojami abu variantai vienu metu nėra apžvelgiamas, nes yra maža tikimybė, kad abu duomenų rinkiniai turės panašius tikslumus ir būtų verta juos naudoti greta.

3. Realizacija

3.1. Aparatūrinė ir programinė įranga

Eksperimentas yra atliktas naudojant kompiuterį turintį šiuos techninius ir programinės įrangos parametrus:

3.1 lentelė. Naudoto kompiuterio techniniai ir įrangos parametrai

Procesorius	Intel® Core™ i5-6600K CPU
Procesoriaus dažnis	3500 MHz
Operacinė sistema	Windows 10 Education version 1909
Atmintis	16 GB
Vaizdo plokštė	NVIDIA GeForce GTX 1060
Vaizdo plokštės dažnis	1500 MHz
Vaizdo plokštės atmintis	6 GB
Vaizdo plokštės CUDA gijų skaičius	1280

Eksperimentai buvo atliekami „Anaconda Navigator“ aplinkoje. Šioje aplinkoje buvo įrašytos šios programinės įrangos versijos:

3.2 lentelė. Naudotoje „Anaconda Navigator“ aplinkoje parsųstos programinių įrangų versijos

„Cudatoolkit“	10.0.130
„Cudnn“	7.3.1
„H5py“	2.8.0
„Keras“	2.2.4
„Matplotlib“	3.1.0
„Numpy“	1.16.3
„OpenCV“	3.4.2
„Pandas“	0.24.2
„Pillow“	6.0.0
„Python“	3.7.3
„Tensorflow“	1.13.1

3.2. Duomenų rinkinio kūrimas

Šiame darbe išvesties rezultatai turi 2 galimus variantus: 1) Vairuotojas kalba telefonu; 2) Vairuotojas nekalba telefonu. Atsižvelgiant į galimus 2 išvesties variantus, buvo sukurta vaizdų duomenų bazė, kurioje nuotraukos yra suskirstytos į šias grupes. Vaizdinės medžiagos rinkimas vyko 2 etapais. Pirmajame etape buvo atliekama duomenų paieška internete. Paieškos metu (2018 metų rudenį), atvirai pasiekiami duomenų rinkiniai buvo sukurti iš transporto priemonės vidaus [54]. Šie duomenys eksperimentui yra netinkami, nes siekiame nustatyti vairuotoją, kalbantį telefonu, iš transporto priemonės išorės. Siekiant atrasti duomenų rinkinį, kuriame būtų vairuotojai, buvo ieškomi ir transporto priemonių atpažinimo duomenų rinkiniai [55,56], tačiau tik retu atveju transporto priemonėje būdavo vairuotojas. Atlikus šią duomenų rinkinių paiešką buvo nuspręsta susirinkti vaizdinės medžiagos rinkinį, naudojant vaizdinę paiešką. Paieškai buvo naudotos 4 paieškos sistemos

– „Google“, „Bing“, „DuckDuckGo“, „Baidu“. Paieška buvo atliekama naudojant užklausą „driver talking on the phone“ bei šios užklausos vertimus į ispanų, portugalų, vokiečių, kinų kalbas. Anglų, ispanų, portugalų ir vokiečių pagrindiniai paieškos rezultatai buvo „stock photos“ ir nuotraukos vairuotojų, kalbančių telefonu, iš transporto priemonės vidaus. Tikriausiai gauti tokie rezultatai dėl privatumo įstatymų Europos Sąjungoje ir kitose šalyse. „Google“ vertimas į kinų kalbą buvo netinkamas duomenų paieškai. Mokėdamas kinų kalbos pradmenis, sukūriau užklausą „开车打电话“ (liet. *kalbėjimas telefonu vairuojant*). Šioje užklausoje buvo daug nuotraukų iš transporto priemonės išorės, kai vairuotojas kalba telefonu. Išmokus daugiau kinų kalbos žodžių, buvo atliktos paieškos ir su šiomis užklausomis: „开车打电话闭路电视“ (liet. *kalbėjimas telefonu vairuojant, uždara televizijos sistema* (angl. *CCTV*)), „开车打电话天眼“ (liet. *kalbėjimas telefonu vairuojant, dangaus akis*), „开车打电话照相机“ (liet. *kalbėjimas telefonu vairuojant, kamera*). Nuotraukos buvo parsųstos automatiškai, naudojant internetinį naršytoją (angl. *web crawler*). Nuotraukos buvo atrinktos ir sugrupuotos rankiniu būdu. Atrinkimo metu buvo išsaugomos nuotraukos, kurios buvo fotografuotos transporto priemonės priekyje [-30;30] laipsnių kampu iš įvairaus aukščio ir įvairiu paros metu. Atrinkus visas netinkamas nuotraukas, buvo sukurtas 415 nuotraukų rinkinys, kuriame 313 nuotraukų vairuotojas kalba telefonu, o 102 vairuotojas nekalba telefonu. Dėl priežasties, kad šiame duomenų rinkinyje yra 3 kartus daugiau nuotraukų, kuriose vairuotojas kalba telefonu, buvo nuspręsta atrasti daugiau nuotraukų, kuriose vairuotojas nekalba telefonu. Buvo atliekama paieška kiniškuose portaluose ir buvo atrastas puslapis [57], kuriame yra rodomi įvairūs kelių eismo taisyklių pažeidimai. Buvo atlikta papildoma paieška, naudojant „驾驶员未按规定使用安全带“ (liet. *vairuotojas nesinaudoja saugos diržais, kaip reikalaujama*) užklausa. Atrastos 225 tinkamos nuotraukos. Dėl sąlyginai mažo kiekio nuotraukų, nuspręsta šias nuotraukas naudoti modelio testavime, nustatant kaip gerai modelis išmoksta apibendrinti.



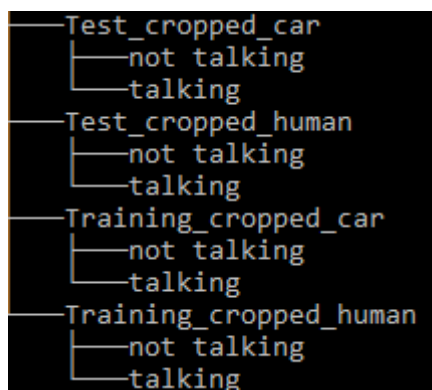
3.1 pav. Atsitiktinės duomenų rinkinio nuotraukos

Antrojo etapo metu buvo kuriamas duomenų rinkinys, esant nejudančiai transporto priemonei su vairuotoju. Fotografavimas buvo atliekamas įvairiais [-15;15] laipsnių kampais iš priekio, siekiant

imituoti policijos kamerų kampus transporto priemonės atžvilgiu. Fotografavimo metu gautos nuotraukos 1 megapikselyje raiškios. Siekiant išvengti susiliejusiu nuotraukų, fotografavimas vyko stacionariai. Fotografavimo metu transporto priemonės pozicija ir antrasis planas nesikeitė. Nufotografuota 1701 nuotrauka, kurioje vairuotojas kalba telefonu ir 2189 nuotraukos, kuriose vairuotojas nekalba telefonu.

3.3. Duomenų rinkinio paruošimas

Automatiniam objektų atradimui buvo pasirinktas „YOLOv3“ modelis. Pagrindinės priežastys: „YOLOv3“ aukštesnis tikslumas, nei „SSD“ [58], atrastas viešos prieigos modelis [59], turintis nuotraukos ir video objektų atradimo funkcijas. Atlikus kodo pritaikymą šiam projektui, buvo automatiškai patikrinamas visas aplankas nuotraukų, jos išsaugomos su atrasta objektų klase failo pavadinime. Apmokymo nuotraukos buvo saugotos JPG formatu, nes fotografuotos nuotraukos buvo aukštos raiškios, o testavimo nuotraukos saugotos PNG formatu, nes jos yra mažos raiškios ir norime išsaugoti kuo daugiau informacijos. Nuotraukos, kuriose pagrindinis atrastas objektas buvo pašalinis (pėstieji, tolumo transporto priemonės ir pan.), buvo ištrintos. Likusios nuotraukos buvo suskirstytos į 2 skirtingus aplankus: vairuotojo ir transporto priemonės. Transporto priemonės aplanke yra visos nuotraukos, kuriose objektų atradimo modelis nustatė klasę „car“, „bus“ arba „truck“, o vairuotojų, kuriose nustatyta klasė „person“.



3.2 pav. Paruoštų duomenų aplankai

2019 metų sausio mėnesį buvo atliktas „YOLOv3“ apmokyto modelio tikslumo tyrimas. Šiame tyrime naudota jau minėtas „YOLOv3“ modelis [59] bei naudoti jau pamokyti „YOLOv3“ svoriai [33]. Testavimo metu buvo naudota išvestis, kuri nurodė esamų elementų būvimą. Tyrimui atlikti buvo naudotos 313 nuotraukų, kuriuose vairuotojas kalba telefonu ir 102 nuotraukos, kuriose nėra vairuotojo, kalbančio telefonu. Kiekvieno dydžio „YOLO“ apgaubtasis tūris gali pasirodyti tik vieną kartą. Transporto priemone yra laikoma mašinos, sunkvežimio išvestis.

Tyrimo metu gauti šie rezultatai:

3.3 lentelė. „YOLO“ modelio objektų aptikimo tyrimo rezultatai

Duomenų rinkinys	Duomenų kiekis	Transporto priemonė	Asmuo ir transporto priemonė	Be asmens ar transporto priemonės
Vairuotojas, kalbantis telefonu	313	307	231	82
Vairuotojas, nekalbantis telefonu	102	101	82	20

Rezultatuose matome, kad transporto priemonė buvo aptikta 98 % nuotraukų, o asmenų ir transporto priemonių 75 %. Transporto priemonės galėjo būti neatrastos dėl vakarų ir rytų šalyse paplitusių transporto priemonių skirtumų. Galimai „YOLOv3“ modelis nėra apmokytas aptikti tik rytų šalyse esančias transporto priemones. Asmenys transporto priemonėse yra sunkiai aptinkami. Viena iš pagrindinių priežasčių, kodėl neaptinkamas asmuo – cenzūravimas, po kurio dingsta asmens galvos bruožai. Kita priežastis – asmens rūbų ir interjero spalvos susilieėjimas. Nuotraukose, kuriose asmuo vilki rūbus interjero apšvietimo spalvos, matomi asmens bruožai yra tik galva ir rankos, todėl modelis gali neaptikti asmens. Šis reiškinys yra dažnesnis nuotraukose gautose tamsiu paros metu. Nuotraukos gali būti per mažos rezoliucijos, kad juose neatrandami asmens bruožai. Kaip ir minėta anksčiau, naudojame unikalios dydžio apgaubtuosius tūrius, todėl kiekvienų matmenų apgaubtieji tūriai yra naudojami tik vieną kartą. Dėl šios priežasties, jeigu vairuotojas ir keleivis abu yra tinkami vienam apgaubtajam tūriui, yra 50 % tikimybė, kad apgaubtasis tūris bus skirtas keleiviui, o ne vairuotojui. Ši problema gali būti išsprendžiama pakėlus limitą iki 2 apgaubtųjų tūrių su ta pačia rezoliucija. Šis sprendimas dvigubai pailgintų „YOLO“ modelio aptikimo trukmę.

Vaizdinės duomenų bazės paruošimo metu buvo pastebėta, kad „YOLOv3“ modelis gali gražinti transporto priemonės kėdę su asmens klase. Taip pat buvo pastebėta, kad aukštesnės raiškos nuotraukose modelis gali aptikti telefoną rankose. Aptiktos telefonų nuotraukos šiame projekte nėra naudojamos, nes testavimo atveju šios informacijos panaudoti negalėtume, nes duomenų rinkinyje nuotraukos yra mažos raiškos ir telefonai jose nėra aptinkami.

Paruošus duomenų rinkinius mokymams ir testavimui, turima 3.4 lentelėje esančių nuotraukų kiekį.

3.4 lentelė. Paruoštų duomenų rinkinių nuotraukų kiekiai

Duomenų rinkinys	Transporto priemonė su vairuotoju		Vairuotojas	
	Kalbančiu telefonu	Nekalbančiu telefonu	Kalbantis telefonu	Nekalbantis telefonu
Mokymo	1704	2189	1577	2162
Testavimo	432	787	241	340

3.4. Modelių tyrimas su nepakeistais duomenų rinkiniais

Šiame projekte yra naudojami toliau aprašyti numatytieji parametrai iki kol nėra pasakoma, kad parametrai yra pakeičiami. „VGG“, „GoogLeNet“ ir „ResNet“ modeliai naudojo apmokytus „Imagenet“ svorius. „AlexNet“ nenaudojo jokių pradinių svorių. „VGG“, „GoogLeNet“ ir „ResNet“ sluoksniai buvo užšaldomi. Naudoti modeliai nebuvo modifikuoti (t.y. nebuvo pridėta sluoksnių, tokių kaip „Dropout“), išskyrus „VGG“, siekiant išskirti jį iš visų modelių savo raiška. „VGG“ įvesties raiška buvo pakeista į 64x64. Modelių mokymo metu buvo mokoma atpažinti 2 klases. Klasių pavadinimai yra aplankų pavadinimai, kuriuose yra sugrupuotos nuotraukos. Modeliai buvo mokomi 50 epochų. Modeliai mokymo metu yra leidžiami vieną kartą.

Pagal suprojektuotą sprendimą, mums svarbu sužinoti kokius rezultatus pasiekia modeliai su duomenų rinkiniais, kurie yra išskirstyti į žmones ir transporto priemones. Šiuose tyrimuose gauti rezultatai padės nuspręsti kokias nuotraukas naudoti iš duomenų rinkinių mokymams ir testavimui. Tyrimuose bus tikrinamas tikslumas, teisingi teigiami, teisingi neigiami ir F vertė. Teisingi teigiami ir teisingi neigiami yra apskaičiuojami pagal šias formules:

$$Teisingi\ teigiami = \frac{TT}{TT + NT}; \quad (14)$$

$$Teisingi\ neigiami = \frac{TN}{TN + NN}; \quad (15)$$

čia TT – teisingai įvertinti kaip teigiami, NT – neteisingai įvertinti kaip teigiami, TN – teisingai įvertinti kaip neigiami, NN – neteisingai įvertinti kaip neigiami.

Pirmame tyrime buvo naudojamos visos turimos mokymo ir testavimo nuotraukos. Šiame tyrime buvo atskirai tiriamas tikslumas transporto priemonių ir vairuotojų duomenų rinkiniams.

3.5 lentelė. Modelių mokymo rezultatai, naudojant transporto priemonių duomenų rinkinį

Modelis	Tikslumas, %	Teisingi teigiami, %	Teisingi neigiami, %	F vertė, %
„AlexNet“	51,6	55,1	45,5	59,2
„VGG“	47,2	54,8	45,1	30,4
„GoogLeNet“	55,1	55,0	100	80,0
„ResNet“	53,3	55,2	46,4	65,1

3.6 lentelė. Modelių mokymo rezultatai, naudojant vairuotojų duomenų rinkinį

Modelis	Tikslumas, %	Teisingi teigiami, %	Teisingi neigiami, %	F vertė, %
„AlexNet“	54,0	46,6	68,5	57,3
„VGG“	59,7	52,3	62,4	40,9
„GoogLeNet“	58,5	0 dalyba iš 0	58,5	0 dalyba iš 0
„ResNet“	59,2	52,9	60,1	23,8

Iš pradinių rezultatų matome, kad tikslumas, naudojant žmonių duomenų rinkinį, lyginant su transporto priemonių duomenų, vidutiniškai tikslumas yra aukštesnis 12,1 %. Iš rezultatų galime pastebėti, kad modelis persimoko ir visiškai neatranda požymių įvestyje esant transporto priemonei, o įvestyje kur žmogus tik truputį geresnė situacija. Šiame ir likusiuose modelių tyrimuose su nepakeistais duomenimis, mokymosi gylis yra iki 5 epochų. Abiejuose atvejuose „GoogLeNet“ spėjo tik vieną klasę. Šiame tyrime galime pastebėti F vertės trūkumą, kai visos vertės yra spėjamos kaip tiesa, mes gauname aukštą F vertę, nes F vertė neatsižvelgia į neteisingai įvertintus kaip neigiamus rezultatus.

Tokiomis pačiomis sąlygomis yra atliekamas ir greitaveikos tyrimas, kurio rezultatai vėliau bus naudojami kaip atskaitos taškas. Šiame greitaveikos tyrime tikrinsime 10 epochų mokymosi greičius, jų medianas ir vidurkius.

3.7 lentelė. Greitaveikos tyrimo rezultatai, naudojant transporto priemonių duomenų rinkinį

Modelis	Lėčiausiai, s	Greičiausiai, s	Mediana, s	Vidurkis, s
„AlexNet“	60	54	54,5	55,4
„VGG“	57	51	52,5	52,8
„GoogLeNet“	61	54	56	56,6
„ResNet“	107	54	56	60,6

3.8 lentelė. Greitaveikos tyrimo rezultatai, naudojant vairuotojų duomenų rinkinį

Modelis	Lėčiausiai, s	Greičiausiai, s	Mediana, s	Vidurkis, s
---------	---------------	-----------------	------------	-------------

„AlexNet“	15	9	9	9,9
„VGG“	8	5	5	5,6
„GoogLeNet“	21	14	16	16,1
„ResNet“	27	23	23	23,7

Visų greitaveikos tyrimų pirmoji epocha buvo lėčiausia. Vidutiniškai mokymosi trukmė yra 42,5 s ilgesnė, naudojant transporto priemonių duomenų rinkinius. Lėtesnė mokymo trukmė yra paaiškinama tuo, kad transporto priemonių rinkinys užima 19 kartų daugiau vietos nei vairuotojų.

Surinktuose iš interneto duomenyse, dauguma nuotraukų žmonių duomenų rinkinyje yra aukštesnės raiškos nei 32x32, o transporto priemonėse nei 64x64. Aukštinant nuotraukų raišką, dingsta jų bruožai, todėl norima patikrinti modelių tikslumus, kai mes naudojame nuotraukas, kurios atitinka raiškos reikalavimus. Naudojant aukštesnės raiškos nuotraukas, mes nebenaudojame žemesnės raiškos nuotraukų, kurios turi didesnę tikimybę būti neteisingai įvertintos. Šiuo tyrimu siekiame tai patvirtinti. Žmonių duomenų rinkiniui buvo pasirinkti 48 ir 64 pikselių raiškos atskaitos taškai, nes norime išsaugoti kuo daugiau duomenų rinkinio, tačiau taip pat ir patikrinti su nuotraukomis, kurios yra aukštesnės raiškos nei dauguma. Dėl tos pačios priežasties buvo tiriami ir transporto priemonių duomenų rinkiniai, kada jų raiškos prasideda nuo 96 ir 128 pikselių. Siekiant taupyti mokymo laiką, buvo sukurti papildomi aplankai, kuriuose yra nuotraukos atitinkančios raiškos kriterijus.

Transporto priemonių duomenų rinkinio tyrime su aukštesnės raiškos validacijos nuotraukomis visos sąlygos yra tos pačios kaip ir praeitame tyrime.

3.9 lentelė. Modelių mokymo rezultatai, naudojant virš 96x96 raiškos transporto priemonių duomenų rinkinį

Modelis	Tikslumas, %	Teisingi teigiami, %	Teisingi neigiami, %	F vertė, %
„AlexNet“	51,4	17,1	51,1	19,0
„VGG“	51,6	51,0	50,2	52,2
„GoogLeNet“	56,2	61,4	54,0	44,9
„ResNet“	54,2	53,2	56,5	62,5

3.10 lentelė. Modelių mokymo rezultatai, naudojant virš 128x128 raiškos transporto priemonių duomenų rinkinį

Modelis	Tikslumas, %	Teisingi teigiami, %	Teisingi neigiami, %	F vertė, %
„AlexNet“	52,4	51,0	54,3	55,3
„VGG“	51,0	49,8	52,1	48,9
„GoogLeNet“	54,8	57,9	53,9	36,8
„ResNet“	56,7	54,0	62,6	63,2

Kaip matome, naudojant aukštesnės raiškos validacijos rinkinį, gaunami panašūs rezultatai. Vienas iš mokymo, naudojant transporto priemonių duomenų rinkinį, sunkumų – siekiama, kad modelis išmoktų įvertinti nuotrauką pagal regioną (vairuotojo vietą), tačiau pats modelis yra mokomas pagal visos nuotraukos parametrus. Treniravimo metu tarpusavyje yra pasiekimas 100 % tikslumas, galima priežastis – modeliai išmoksta atpažinti transporto priemonės pokyčius, o ne vairuotojo. Šiame atvejuje priežastis galėjo atsirasti, nes visos mokymo nuotraukos gautos lauke, su ta pačia transporto priemone, esant saulėtai dienai. Fotografavimo metu galėjo kisti nuotraukų ryškumas ir pagal šią sąlygą apmokyti parametrus. Tai galėjo būti išvengta, jeigu būtų atliktas fotografavimas sąlygose, kuriose dominuotų nenatūralus apšvietimas patalpose. Kita galima priežastis – parinkti per mažos

raiškos atskaitos taškai, todėl vis dar galėjo būti dauguma nuotraukų, kurios yra per mažos raiškos, kad tiksliai nustatytų ar vairuotojas kalba telefonu.

Tokiomis pačiomis sąlygomis modeliai buvo ištirti ir su vairuotojo duomenų rinkiniu.

3.11 lentelė. Modelių mokymo rezultatai, naudojant virš 48x48 raiškos vairuotojų duomenų rinkinį

Modelis	Tikslumas, %	Teisingi teigiami, %	Teisingi neigiami, %	F vertė, %
„AlexNet“	55,0	36,6	64,0	34,9
„VGG“	59,1	38,6	64,4	28,0
„GoogLeNet“	63,8	0 dalyba iš 0	0	0 dalyba iš 0
„ResNet“	59,1	40,0	65,0	31,5

3.12 lentelė. Modelių mokymo rezultatai, naudojant virš 64x64 raiškos vairuotojų duomenų rinkinį

Modelis	Tikslumas, %	Teisingi teigiami, %	Teisingi neigiami, %	F vertė, %
„AlexNet“	62,0	44,8	67,8	42,3
„VGG“	61,1	38,5	66,1	26,3
„GoogLeNet“	66,7	54,8	68,6	32,1
„ResNet“	67,1	70,0	67,0	16,5

Atlikus tyrimą su aukštesnės raiškos duomenų rinkiniais, gauname aukštesnes tikslumo vertes. Modeliuose, kurių įvesties raiška yra 224x224, lyginant be filtruoto duomenų rinkinio, vidutiniškai tikslumas pakilo 12,3 %, o „VGG“ tik 2,3 %. Šio staigaus tikslumo pokyčio galimos priežastys: 1) tyrimo hipotezė yra patvirtina, buvo išimtos nuotraukos, kurios neteisingai būtų buvusios įvertintos priešinga klase; 2) buvo sėkmingas 224x224 raiškos modelių inicializavimas.

Atlikus šiuos tyrimus buvo norima sužinoti koks yra pasiekiamas tikslumas, jeigu yra naudojami abu duomenų rinkiniai vienu metu. Šiuo tyrimu buvo bandoma patikrinti ar naudojant abu duomenų rinkinius, gausime aukštesnį tikslumą. Pagrindinė šio tyrimo idėja – esant dviem skirtingiems duomenų rinkiniams, mokymo proceso metu, transporto priemonių duomenų rinkinio klasifikatorius kartu išmoks atpažinti ir vairuotojo bruožus, o tai padėtų atrasti transporto priemonėje vairuotoją ir atpažinti ar jis kalba telefonu.

Šiame tyrime bus naudojamos tos pačios sąlygos kaip ir praeituose tyrimuose. Šiame tyrime bus naudojami bendrai sugrupuoti duomenų rinkiniai. Vairuotojų ir transporto priemonių duomenų rinkiniai yra išsaugoti bendrame sugrupuotame aplanke. Šiame tyrime buvo parinktos vairuotojų nuotraukos virš 48x48 raiškos ir transporto priemonių, kuriuos atitinka virš 96x96 raiškos reikalavimą.

3.13 lentelė. Modelių mokymo tikslumai, naudojant virš 64x64 raiškos žmonių ir virš 96x96 raiškos transporto priemonių duomenų rinkinius kartu

Modelis	Tikslumas, %	Teisingi teigiami, %	Teisingi neigiami, %	F vertė, %
„AlexNet“	53,3	46,2	54,6	22,5
„VGG“	52,8	45,5	54,4	25,5
„GoogLeNet“	56,3	52,4	58,6	47,5
„ResNet“	52,4	48,4	59,8	57,0

Kaip matome iš rezultatų, hipotezė nepasitvirtino. Šio tyrimo rezultatuose tikslumas žymiai nepakito, lyginant su tik transporto priemonių duomenų rinkinio tyrimu. Iš šio tyrimo suprantame, kad mokymui reikės pasirinkti vieną iš duomenų rinkinių, o ne abu bendrai.

3.5. Modelių tyrimas su keičiamais duomenų rinkiniais

Iš tyrimų su nepakeistais duomenų rinkiniais, matome, kad modeliai persimoko, jų gylis labai negilus, pasiekiamas labai žemas tikslumas su duomenų rinkiniu iš interneto. Šie rezultatai reiškia, kad yra problemų su duomenų rinkinio kiekiu, jo įvairumu, duomenų rinkinių nepanašumu. Šioms problemoms spręsti buvo sumąstyta, kad reikia keisti apmokymo duomenų rinkinį. Šiuose tyrimuose pasirinkta atlikti tyrimus su žmonių duomenų rinkiniu, nes su juo buvo pasiektas aukštesnis tikslumas ir jis bus pagrindinis duomenų rinkinys, kuris bus naudojamas sistemoje.

3.5.1. Mokymo duomenų rinkinio raiškos mažinimo tyrimas

Kadangi visos duomenų rinkinio nuotraukos yra aukštesnės raiškos nei 224x224, iš aukštos kokybės nuotraukos, kai ji yra sumažinama, mes sukoncentruojame bruožus, tačiau internetinių duomenų atveju, nuotraukų raiška bus didinama ir šių nuotraukų bruožai bus išsklaidomi. Dėl nuotraukų raiškos didinimo, „AlexNet“, „GoogLeNet“ ir „ResNet“ atveju, 56x56 nuotrauka bus paaukštinama iki 224x224 raiškos, t.y. 1 pikselis bus atvaizduojamas 4 pikselių aukščiau ir pločiu. Dėl šios priežasties internetinio duomenų rinkinio nuotraukos išgauname skirtingus nuotraukų bruožus, modeliai yra apmokomi atskirti bruožus, kai nuotraukos yra su sumažinta raiška. Sumažinant aukštos raiškos mokymo duomenų rinkinį į testavimo duomenų rinkinio raišką, aukštinant nuotraukų raišką turėtų būti išgaunami panašūs bruožai. Modelių apmokymo metu, jeigu yra išmokstami didesnės raiškos bruožai, modelis turėtų lengviau išmokti sugrupuoti nuotraukas. Šiame tyrime raiška bus mažinama iki 48x48, nes norime patikrinti veiksmingumą ir „VGG“ modelyje, nes jeigu parinktume 64x64 raišką, „VGG“ rezultatai nekistų. Šio tyrimo tikslas - patikrinti kaip keičiasi tikslumas, kai mokymo duomenų rinkinio raiška yra pritaikoma prie testavimo rinkinio raiškos.

Tyrimo metu buvo naudotas mokymo duomenų rinkinys, kurio raiška buvo sumažinta iki 48x48. Šis rinkinys buvo išsaugotas diske, kad naudotume mažiau resursų mokymo metu. Rezultatams tikrinti buvo naudotas virš 48 raiškos dimensijų vairuotojų duomenų rinkinys. Šiame tyrime epochų skaičius – 30. Atlikus tyrimą su visais modeliais, gauti 3.14 lentelėje pavaizduoti rezultatai.

3.14 lentelė. Modelių mokymo tikslumai, naudojant sumažintą iki 48x48 vairuotojų duomenų rinkinį

Modelis	Tikslumas, %	Teisingi teigiami, %	Teisingi neigiami, %	F vertė, %
„AlexNet“	61,2	41,8	64,9	25,8
„VGG“	62,1	47,5	70,0	46,9
„GoogLeNet“	63,8	0 dalyba iš 0	63,8	0 dalyba iš 0
„ResNet“	63,8	42,5	67,0	40,7

Lyginant su duomenimis, kai buvo naudojamas nesumažintas duomenų rinkinys, pastebime, kad „GoogLeNet“ modelis vėl įvertino visas nuotraukas viena klase, o kitų 3 modelių tikslumo vertė, lyginant su virš 48x48 duomenų rinkinio tyrimu, vidutiniškai pakilo 7,43 %. Dėl aukštesnio tikslumo buvo nuspręsta tolimesniuose tyrimuose naudoti duomenų rinkinį su sumažinta raiška. Šis duomenų rinkinio pokytis neišsprendžia problemos, kad modelių mokymo gylis yra negilus.

Pasirinkus, kad tolimesni tyrimai bus atliekami, naudojant iki 48x48 raiškos sumažintą duomenų rinkinį, reikia atlikti greitaveikos tyrimus, siekiant turėti atskaitos taškus, kaip kinta mokymo greitis, pridėdant papildomų sąlygų. Šis greitaveikos tyrimas yra atliekamas vienodomis sąlygomis kaip ir žmonių mokymo greitaveikos tyrime. Vienintelis skirtumas – mokymo metu naudojamas iki 48x48 raiškos sumažintas duomenų rinkinys. Greitaveikos tyrimo rezultatai pavaizduoti 3.15 lentelėje:

3.15 lentelė. Greitaveikos tyrimo rezultatai, naudojant sumažintą iki 48x48 raiškos vairuotojų duomenų rinkinį

Modelis	Lėčiausiai, s	Greičiausiai, s	Mediana, s	Vidurkis, s
„AlexNet“	14	9	9	9,8
„VGG“	8	5	5	5,4
„GoogLeNet“	18	13	13	13,8
„ResNet“	25	21	21	21,2

Iš rezultatų matome, kad naudojant mažesnės raiškos duomenų rinkinį, gauname greitesnį mokymo procesą. „AlexNet“ ir „VGG“ modelių mokymo laikas sumažėjo nežymiai, o „GoogLeNet“ ir „ResNet“ vienos epochos metu sutaupo virš 1s.

3.5.2. Mokymo duomenų rinkinio papildymo tyrimas

Jeigu nėra galimybės padidinti duomenų rinkinio kiekio ir įvairovės, yra naudojamas duomenų papildymas (angl. *data augmentation*), kuris padeda sulėtinti modelio permokymą [60]. Šis sprendimas gali padidinti mokymosi gylį. Atlikdami šį tyrimą, siekiame gauti aukštesnį modelio tikslumą, nei nenaudojant jokio duomenų papildymo.

Duomenų papildymas buvo atliekamas naudojant „Keras“ „ImageDataGenerator“ funkciją. Kiekviename duomenų papildymo tyrime buvo naudojama „rescale = 1./255“ ir „horizontal_flip = True“. Rescale yra naudojamas, kad normalizuotume duomenis, o horizontal_flip, kad gautume daugiau skirtingų veiksmingų failų, nes horizontalus apvertimas nepanaikina vairuotojo, kalbančio telefonu, bruožų. Šiame tyrime buvo naudojamas tik „GoogLeNet“ modelis, nes jo tikslumas, naudojant virš 64x64 raiškos nuotraukas buvo vienas iš aukščiausių, F vertė dvigubai aukštesnė nei „ResNet“, buvo norima patikrinti ar šis duomenų papildymas padės išspręsti „GoogLeNet“ problemą, kad modelis išmoksta įvertinti validacijos nuotraukas tik teigiamai arba neigiamai. Tyrimas atliekamas tik su „GoogLeNet“, nes „ImageDataGenerator“ pailgina modelio apmokymo trukmę, o atliekant tyrimą ir su kitais modeliais, ypatingai skirtingų rezultatų neturėtume gauti. Šiame tyrime geriausi modelių svoriai nebuvo saugomi. Naudotas sumažintas iki 48x48 raiškos duomenų rinkinys, nes buvo norima apmokyti modelį su kuo panašesnėmis sąlygomis. Visi tyrimai buvo atlikti vieną kartą. Atradus, kad parametras padidina tikslumą, jis yra naudojamas ir tolimesniuose parametru tyrimuose. Duomenų papildymo tyrimo rezultatai pateikti 3.16 lentelėje.

3.16 lentelė. Duomenų papildymo tyrimo rezultatai

Parametrai	Vertės	Tikslumas, %
rotation_range	45	64,4
rotation_range	30	65,3
rotation_range	15	66,2
rotation_range, width_shift_range	15, 0,25	65,0

rotation_range, width_shift_range	15, 0,15	65,0
rotation_range, height_range_shift	15, 0,25	68,5
rotation_range, height_range_shift	15, 0,125	70,9
rotation_range, height_range_shift, brihtness_change	15, 0,125, [0,5; 1]	68,8
rotation_range, height_range_shift, brihtness_change	15, 0,125, [1; 1,5]	64,2
rotation_range, height_range_shift, shear_change	15, 0,125, 0,2	63,8
rotation_range, height_range_shift, shear_change	15, 0,125, 0,1	63,8
rotation_range, height_range_shift, zoom_change	15, 0,125, 0,2	63,8
rotation_range, height_range_shift, zoom_change	15, 0,125, 0,1	63,8

Visais atvejais modelis persimokydavo. Mokymo gylis nepagerėjo. Geriausi rezultatai buvo gauti, naudojant „rotation_range“ ir „height_shift_range“ parametrus, tačiau aukštesnis tikslumas nebuvo pasiektas atliekant tyrimus su daugiau papildomų parametru. Aukščiausi tikslumai dažniausiai pasiekti modelio mokymo pradžioje.

3.5.3. Veiksmingumo tyrimas, naudojant duomenų rinkinio kraštines

Konvoliuciniai neuroniniai tinklai įvertina nuotraukas pagal jų bruožus. Dažniausiai nuotraukos turi papildomos informacijos, kuri yra nebūtina modeliui teisingai nustatyti nuotraukos klasę. Ši papildoma informacija gali būti nuotraukos spalvos. Modeliui teisingai nustatyti ar objektas yra apskritimas, nebūtina turėti spalvotos nuotraukos, nes modelis išmoks atpažinti kraštines. Pagal šią idėją yra atliekamas šis tyrimas. Šio tyrimo tikslas – sužinoti, ar galime naudoti nuotraukų kraštines vietoje įprastų nuotraukų, taip pašalindami nebūtiną nuotraukos informaciją.

Šio tyrimo metu naudosime duomenų rinkinius, kurie buvo pakeisti, naudojant kraštinių aptikimo algoritmus. Šiame tyrime buvo išsaugotos nuotraukos, naudojant „Laplacian“ ir „Sobel“ operatorius. Šie operatoriai buvo panaudoti ant iki 64x64 raiškos sumažintas vairuotojų apmokymo duomenų rinkinys. 64x64 raiška buvo pasirinkta, nes buvo norima išlaikyti daugiau informacijos negu su 48x48 raiškos nuotraukomis. Tyrimo validacijai naudotas virš 64x64 raiškos vairuotojų duomenų rinkinys. Mokymo metu nebuvo naudotas duomenų papildymas.

Tyrimo rezultatai: naudotas „GoogLeNet“ algoritmas, gautas tikslumas žemesnis nei 63,8 % su abiem operatoriais. Peržiūrėjus duomenų rinkinius, pastebėta, kad mokymo duomenų rinkinyje kraštinės matosi puikiai, tačiau validavimo rinkinyje jos buvo nepastebimos (pagrindinė kraštinė prieš normalizaciją dažniausiai būdavo visiškai nesusijusi su vairuotoju, todėl šios kraštinės būdavo labai mažos vertės) arba jos retai būdavo geros būklės kaip ir mokymosi duomenų rinkinio. Duomenų rinkiniai yra per daug didelio kokybės skirtumo, kad būtų galima naudoti kraštinių nuotraukas modelių apmokymui.



3.3 pav. Mokymo rinkinio kraštinių nuotraukos (kairėje pusėje) ir validacijos rinkinio kraštinių nuotraukos

3.5.4. Naujo modelio kūrimo tyrimas

Tyrimo pradžioje minėta, kad galimai problema su permokymu yra iš duomenų. Reikia įvertinti galimybę, kad modeliai yra per daug geri mano duomenų rinkiniui ir todėl yra pasiekiami aukšti mokymosi rezultatai, tačiau ne testavimo metu. Viena iš galimų priežasčių kodėl modeliai persimoko – per didelis modelio parametrų skaičius. Kuo daugiau modelis turi parametrų, tuo didesnė jo „talpa“, o tai reiškia, kad modelis gali išmokti atpažinti nuotraukas taip gerai, kad jas įsimins ir nemokės generalizuoti su kitomis nuotraukomis [61]. Šiame tyrime yra tikrinama ar tikrai modeliai yra per daug geri duomenų rinkiniui ir todėl modelis persimoko.

Tyrimai buvo atlikti naudojant „AlexNet“ architektūrą kaip pagrindą. Šiame tyrime buvo tiriama kaip modelis veikia, jeigu yra naudojama mažiau sluoksnių, filtrų, „dense“ sluoksnio parametrų, naudojant skirtingus filtrų dydžius, pridėdant „Dropout“ (su įvairomis reikšmėmis) modelio pabaigoje ar tarp sluoksnių, pridėjus „gaussian_blur“ modelio pradžioje.

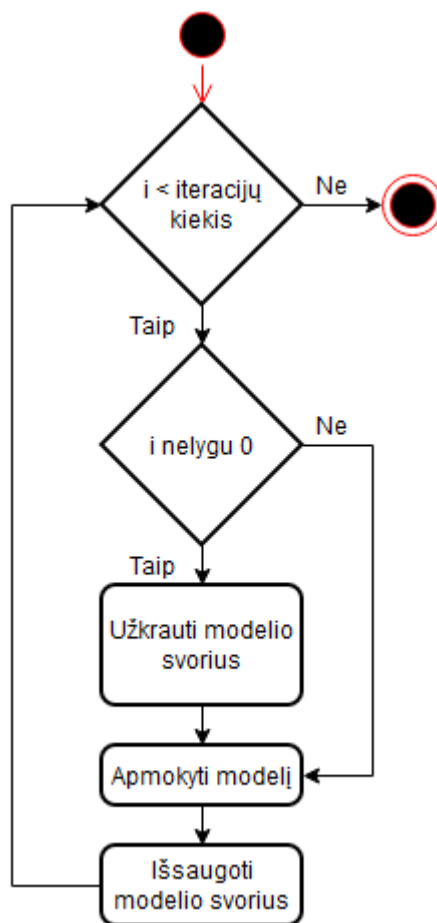
Šiame tyrime buvo pastebėta, kad „dropout“ sluoksnis padeda sumažinti permokymą, tačiau modelio architektūros keitime kitų permokymo mažinimo rezultatų nebuvo gauta. Šio tyrimo rezultatai patvirtino hipotezę, kad problema yra duomenų rinkinyje, o ne modelyje.

3.5.5. Duomenų rinkinio papildymas naudojant duomenų papildymo grupę

Naujo modelio kūrimo tyrimo rezultatai parodo, kad problema yra duomenų rinkinyje. Papildomas duomenų rinkimas turėtų pagerinti šių modelių rezultatus, tačiau nufotografuotos nuotraukos gali būti nepakankamai įvairios ir vėl susidursime su ta pačia problema. Duomenų papildymo tyrime susiduriame su problema, kad modeliai persimoko net ir su duomenų papildymu. Duomenų papildymo pagrindinės problemos: 1) naudojant daugumą parametrų vienu metu, gali susidaryti parametrų kombinacijos, kada bruožai, nurodantys, kad vairuotojas kalba telefonu, gali dingti iš įvesties; 2) dauguma jo naudotojų mano, kad jo pridėjimas papildė jų duomenų rinkinį [62] ir tai padeda sumažinti permokymo tikimybę. Pirmosios problemos vienas iš lengviausiai suprantamų pavyzdžių: tarkime naudojame „rotation“ ir „height“ parametrus vienu metu, gali atsirasti tokia situacija, kada nuotrauka yra pasukama taip, kad ranka su telefonu gali atsirasti aukščiau/žemiau ir nuotrauka pakeliama/nuleidžiama pakankamai aukštai/žemai, kad būtų panaikinama iš vaizdo įvesties. Tai sumažina mūsų galimybę laisvai naudoti visus parametrus, nes mes turime apgalvoti visas galimas parametrų sąveikos pasekmes. Esant šiai suvaržytai laisvei naudoti parametrus, mums tenka rinktis tarp vieno iš kelių parametrų, kuri galėtumėme panaudoti (pvz. „height“ ir „zoom“ atskirai veiktų gražiai, bet kartu lengvai gali panaikinti esminius nuotraukų bruožus). Ši priežastis labai sumažina duomenų papildymo efektyvumą. Antroji problema: mūsų duomenų rinkinys yra pakeičiamas nauju. Tai reiškia, kad jeigu turime problemą su permokymu, mes tai nukeliame tik trupam į ateitį. Jeigu nauduosime kokį parametą, esant mažam kiekiui duomenų rinkinio, modelis tiesiog išmoks atpažinti tą parametą ir modelis toliau bus permokomas (žinoma, to galime išvengti, tačiau turime išspręsti pirmą problemą arba pagerinti duomenų rinkinį). Žinoma, galima duomenų papildytas nuotraukas išsaugoti diske ir taip papildyti savo duomenų rinkinį, tačiau tada turėtumėme pertikrinti visas sukurtas nuotraukas, nes kitu atveju gali būti dar didesnė tikimybė dėl pirmos problemos.

Sugalvotas sprendimo variantas: naudoti ne vieną duomenų papildymo funkciją, o grupę jų. Buvo ieškota šio sprendimo būdų „StackOverflow“ ir „Google“, tačiau šio sprendimo būdo nebuvo rasta. Šiam sprendimui įgyvendinti reikia imageDataGenerator funkcijos veikimo apėjimo (angl.

workaround) ir jis nėra gražus. Šios funkcijos apėjimas yra keisti vidinę šios funkcijos struktūrą (netinka, nes su kiekvienu versijos atnaujinu šią vietą gali reikėti atnaujinti iš naujo, asmenims, kurie bandytų atkartoti rezultatus, gali nepavykti pakeisti funkcijos į reikiama, esant daugumai projektų, jeigu kitas projektas naudotų šią funkciją, tas projektas gali sugriūti ir pan.) arba naudoti esamas funkcijas neefektyviai. Kai `imageDataGenerator` yra inicializuojamas, jo vertės negali būti pakeičiamos. Kintamasis, kuris gauna `imageDataGenerator` vertes yra treniravimo generatorius, kuris naudoja funkciją `flow_from_directory`. Kintamasis, turintis `flow_from_directory` funkciją, ją kviečia tik vieną kartą, todėl jeigu naudotume atsitiktinį generatoriaus parinkimą, būtų parinktas tik vienas `imageDataGenerator` iš kelių ir jis nebūtų keičiamas kol modelis yra apmokomas. Treniravimo generatoriaus vertė yra duodama `model.fit_generator` funkcijai. Čia irgi neveikia atsitiktinis generatoriaus parinkimas (neveikia ir funkcija, kuri gražintų reikšmę su „yield“). Buvo atlikti bandymai su atsitiktiniu parinkimu iš masyvo ir funkcijos kvietiniu. Funkcija buvo iškviesta tik 1 kartą per visą mokymo laiką. Kadangi `model.fit_generator` yra iškviečiamas tik 1 kartą mokymo metu, buvo patikrinta kaip modelis elgiasi, jeigu yra iškviečiamas antrą kartą. Iškvietus antrą kartą `model.fit_generator` funkciją, yra panaudojami nauji parametrai, todėl atsitiktinis treniravimo generatoriaus parinkimas yra galimas, tačiau susiduriame su problema, kad modelio svoriai yra mokomi iš naujo. Išspręstas sprendimo variantas pavaizduotas 3.4 pav:



3.4 pav. Modelio mokymo veiklos diagrama

Šis sprendimas išsprendžia problemą, kad kiekvienos iteracijos metu yra apmokomi modeliai su inicializuotais svoriais. Modelio svoriai yra išsaugomi kietajame diske, nes modelio užkrovimo funkcija `load_weights` reikalauja failo vietos (angl. *filepath*) argumento. Kaip ir minėta anksčiau, šiame projekte, vidinės funkcijos nėra keičiamos optimizavimo tikslu.

Tyrimas buvo atliktas su keturiais modeliais. Kiekvienam modeliui, išskyrus „VGG“ naudotas „Dropout“ su verte 0,5. Siekiant sumažinti modelių tikslumą, buvo naudojama papildoma transporto priemonių klasė. Duomenų papildytojai buvo parenkami atsitiktinai. „GoogLeNet“, „ResNet“, „VGG“ sluoksniai buvo užšaldomi. Atlikus tyrimą, buvo tikrinama ar didėja tikslumas, atšaldant sluoksnius. Mokymai truko 30 epochų. Naudotas iki 48x48 raiškos sumažintas vairuotojų duomenų rinkinys ir virš 48x48 raiškos duomenų rinkinys iš interneto. Tyrimai atlikti po vieną kartą. Gauti rezultatai pavaizduoti 3.17 lentelėje:

3.17 lentelė. Modelių mokymo rezultatai, naudojant iki 48x48 raiškos sumažintą vairuotojų duomenų rinkinį ir grupę treniravimo generatorių

Modelis	Tikslumas, %	Teisingi teigiami, %	Teisingi neigiami, %	F vertė, %
„AlexNet“	64,4	48,4	66,5	32,4
„VGG“	75,9	72,2	76,8	61,0
„GoogLeNet“	63,8	-	-	-
„ResNet“	63,8	-	-	-

„GoogLeNet“ ir „ResNet“ modelių svoriai buvo ištrinti per klaidą, tačiau iš tikslumo tendencijos galime nuspėti, kad ir vėl buvo spėjama tik viena klasė. „AlexNet“ tikslumas padidėjo 9,4 procento, o „VGG“ 16,8 procento. Šis rezultatas buvo gautas, atliekant tyrimą, kai buvo atšaldyti paskutiniai 4 „VGG“ modelio sluoksniai. „GoogLeNet“ ir „ResNet“ sluoksnių atšaldymas tikslumo nepadidino. „AlexNet“ geriausias pasiektas tikslumas buvo 64,4 %, tačiau mokymosi metu vertės pagrindu svyravo [45;55] % režyje, taigi, tikėtina, kad šis rezultatas yra gautas atsitiktinumo dėka. Įvertinimo metu „AlexNet“ ir „VGG“ nuotraukų vertinimui nepasirinko trečios klasės. Galime pastebėti, kad 224x224 raiškos įvesties modeliams šie duomenų papildymai nepadėjo. Pagrindinė priežastis – įvestys yra padidinamos 4,66 karto, kad iš 48x48 raiškos taptų 224x224. Tiek daug kartų paaukštinant įvesties nuotrauką, nuotrauka praranda savo kontūrus ir nuotraukos yra atskiriamos ne pagal tikruosius jų bruožus. Galime pastebėti, kad modeliui, kuris yra panašios raiškos, šis sprendimas padėjo.

Po šių tyrimų buvo atliktas greitaveikos tyrimas, siekiant sužinoti kiek ilgiau užtrunka naudoti duomenų generatorių grupę ir ar laiko sąnaudos nepakinta labai žymiai dėl svorių saugojimo ir užkrovimo. Duomenų rinkinio papildymo tyrime nebuvo saugomi svoriai, bei nebuvo atliekamas greitaveikos tyrimas. Duomenų rinkinio papildymo grupei įvertinti, duomenų rinkinio papildymo tyrimo greitaveikos rezultatai nėra reikalingi, nes vienas iš grupėje naudojamų duomenų papildytojų yra su identiškais duomenų papildytojo tyrimo parametrais (rotation_range, height_range_shift, brihtness_change). Žinodami šio duomenų papildytojo greičius, galėsime sužinoti ar vidutinis duomenų papildytojų grupės apmokymo laikas yra greitesnis, nei naudojant vieną duomenų papildytoją su dauguma parametru. Šių dviejų tyrimų greitaveikos skirtumas yra modelių svorių saugojimo ir užkrovimo trukmė. Duomenų papildytojų grupę bus verta naudoti, jeigu jų mokymo epochų vidutinis apmokymo laikas bus greitesnis už apmokymo trukmę, kai yra naudojamas vienas papildytojas su dauguma parametru be saugojimo ir užkrovimo trukmės.

Greitaveikos tyrimas buvo atliekamas su visais naudojamais duomenų papildytojais vieną kartą (išskyrus pradinį, kuris kartojosi su vienu iš vėliau naudojamų papildytojų). Pirmojo duomenų papildytojo laiko į trukmės skaičiavimus neįtraukiame (pirmojo ir antrojo greitaveikos tyrimų rezultatai rodo, kad pirmosios epochos mokymo laikas visada (išskyrus nenumatytus atvejus, kurie šiame projekte neatsirado) yra ilgesnis už kitus keliomis sekundėmis). Šiame greitaveikos tyrime

modelių svorių saugojimas ir užkrovimas nėra atskiriami nuo epochos laiko. Kiekvienas tyrimas buvo atliktas vieną kartą. Tam, kad galėtumėme sužinoti kiek ilgiau užtrunka naudoti duomenų papildytoją, mokymui naudojame iki 48x48 raiškos sumažintą duomenų rinkinį. Mokymo procese yra apmokomos 2 klasės. Greitaveikos tyrimo rezultatai, naudojant sumažintą iki 48x48 raiškos vairuotojų duomenų rinkinį

3.18 lentelė. Greitaveikos tyrimo rezultatai, naudojant iki 48x48 raiškos sumažintą vairuotojų duomenų rinkinį ir grupę treniravimo generatorių

Modelis	Lėčiausias, s	Greičiausias, s	Mediana, s	Vidurkis, s
„AlexNet“	43	9	42	31,3
„VGG“	8	5	7	6,7
„GoogLeNet“	46	12	45	34,0
„ResNet“	47	19	45,5	36,9

Rezultatų lentelėje nerodoma, tačiau identiškas duomenų papildytojas su trejais parametrais visada buvo tarp lėčiausių duomenų papildytojų. Paprastumo sumetimais, laikysime, kad jo epochos mokymo trukmė buvo lygi medianai. Lyginant su laiku, kada buvo nenaudojamas duomenų papildytojas, vidutinis mokymo trukmės ilgumas išaugo įvairiai: „VGG“ +24 %; „ResNet“ +74 %; „GoogLeNet“ +146 %; „AlexNet“ +219 %. Lengviausiai pastebimas rezultatas – duomenų grupės vidutinis apmokymo laikas yra greitesnis už vieno duomenų papildytojo su dauguma parametru. Norint patvirtinti hipotezę, kad jeigu turime vietos diske ir galime daug kartų įrašinėti svorius, laiko sąnaudų atžvilgiu, duomenų papildytojų grupę yra geriau naudoti negu vieną su dauguma parametru, mums dar reikia sužinoti saugojimo ir užkrovimo trukmę.

Saugojimo ir užkrovimo trukmės nustatyme naudojame tuos pačius duomenų rinkinius, kaip ir praeitame greitaveikos tyrime. Šiame tyrime yra nagrinėjamas laikas, per kurį epochos pabaigoje svoriai yra išsaugomi ir kitos epochos pradžioje jie yra užkraunami. Šio proceso trukmė kiekviename modelyje yra tikrinama 3 iteracijose. Šio proceso trukmių vidurkis yra pavaizduotas rezultatų lentelėje:

3.19 lentelė. Modelių svorių saugojimo ir užkrovimo trukmės tyrimo rezultatai

Modelis	Saugojimo ir užkrovimo trukmė, ms	Modelis	Saugojimo ir užkrovimo trukmė, ms
„AlexNet“	76	„GoogLeNet“	92
„VGG“	15	„ResNet“	73

Buvo tikimasi, kad saugojimo ir užkrovimo procesas truks [1;2] sekundžių intervale. Modelio mokymo metu yra didesni laiko nuokrypiai, negu šio proceso trukmė. Pridėjus šį procesą į mokymo procesą, pailgėjusi mokymo trukmė nebus pajaučiama. „VGG“ atveju, jeigu modelis būtų mokomas 100 iteracijų, mokymo trukmė, esant svorių saugojimui ir užkrovimui, pailgėtų tik 1,5s. Gavus šiuos rezultatus, galime drąsiai teigti, kad duomenų papildytojų grupės vidutinė mokymo trukmė yra greitesnė už vieno duomenų papildytojo su dauguma parametru.

Gavus gerus rezultatus su „VGG“ treniravimo modeliu, buvo norima patikrinti kokį tikslumą gaus modelis, jeigu mokymo procese naudosisime iki 64x64 raiškos sumažintą ir validacijai panaudosisime virš 64x64 raiškos nuotraukas. Tyrimas atliktas identiškomis duomenų papildytojų grupės tyrimo sąlygomis.

3.20 lentelė. Modelių mokymo rezultatai, naudojant iki 64x64 raiškos sumažintą vairuotojų duomenų rinkinį ir grupę treniravimo generatorių

Modelis	Tikslumas, %	Teisingi teigiami, %	Teisingi neigiami, %	F vertė, %
„AlexNet“	63,0	-	-	-
„VGG“	78,2	76,7	72,0	43,8
„GoogLeNet“	69,4	0 dalyba iš 0	65,3	0 dalyba iš 0
„ResNet“	69,4	0 dalyba iš 0	65,3	0 dalyba iš 0

Su aukštesnės raiškos nuotraukomis rezultatai yra geresni, tačiau ir vėl „AlexNet“ neišmoko bruožų atpažinimo ir svyravo [45;55] % ribose. „AlexNet“ išsaugoti modelio svoriai buvo per klaidą ištrinti. „GoogLeNet“ ir „ResNet“ vėl visus rezultatus įvertino kaip vieną klasę. „VGG“ modelio tikslumas pakilo 2,3 % ir taikliau įvertina nuotraukas, kuriose vairuotojas kalba telefonu.

Šio sprendimo privalumai: 1) galima turėti daugiau nei vieną treniravimo generatorių; 2) lengviau testuoti, kurios treniravimo generatoriaus parametrų sąjungos gali panaikinti bruožus būtinus apmokyti modelį; 3) mažesnė permokymo tikimybė, nei naudojant vieną treniravimo generatorių; 4) galima turėti tikslesnius generatorius, su kuriais būtų labiau pabrėžiami bruožai; 5) veikia greičiau negu vienas treniravimo generatorius.

Šio sprendimo trūkumai: 1) modelio paleidimo metu reikia užkrauti daugiau treniravimo generatorių; 2) svoriai turi būti įrašomi į diską ir vėl užkraunami modeliui; 3) modelio mokymo proceso metu gautų tikslumų atvaizdavimui reikia keisti verčių saugojimo būdą; 4) nerekomenduojama naudoti šį sprendimą, jeigu naudojamas SSD.

3.5.6. Tyrimas, kai mokymo metu naudojamas transporto priemonės nuotraukos ketvirtis

Objektų aptikimo modelio efektyvumo tyrime buvo gauta, kad tik 75 % asmenų buvo atrandama nuotraukose, kuriose yra vairuotojas. Pagal dabartinį sprendimą, mes nuotraukų įvertinimui naudojame tik vairuotojo nuotraukas, vadinasi šis modelis nevertina 25 % nuotraukų. Ši nuotraukų nevertinimo problema gali būti sprendžiama: 1) pakeičiant objektų aptikimo modelį; 2) iš naujo apmokant duomenų aptikimo modelį; 3) atrandant regioną, kuriame didžiausia tikimybė, kad bus vairuotojas. Pirmąjį sprendimą būtų reikėję priimti projekto pradžioje, nes atradus, kad modelis bus geresnis, reikės perdaryti pusę nustatymo programos. Antrasis sprendimas būtų tinkamas šiame projekte, nes būtų galima išmokyti modelį atpažinti žmones tik pagal viršutinę žmogaus pusę. Mokymui reikėtų surasti žmonių duomenų rinkinį ir naudoti tik viršutinę nuotraukos pusę arba naudoti veido atpažinimą ir iškirpti 3 kartus didesnę apgaubtąjį tūrį (šiuo atveju turėtų būti pakankamai daug ploto atpažinti ar vairuotojas iš tiesų kalba telefonu). Tačiau pagrindinis šio sprendimo trūkumas – laiko sąnaudos dėl nebūtinai geresnio rezultato, todėl šio sprendimo irgi atsisakoma. Trečio sprendimo variantas būtų atrasti transporto priemonės langą ir naudoti tik tą pusę, kurioje yra vairuotojas (priklausomai nuo vairavimo pusės), jau anksčiau minėtas veido aptikimas ir trigubai didesnio apgaubtojo tūrio naudojimas arba tiesiog iškirpti nuotraukos ketvirtį, kuriame yra didžiausia tikimybė, kad yra vairuotojas (priklausomai nuo vairavimo pusės). Veido aptikimas šiuose duomenų rinkiniuose gali netikti, nes nuotraukų rinkinyje iš interneto yra daug pašalinio garso ir yra nuotraukų, kuriose akys yra uždengtos juosta. Transporto priemonės lango aptikimas būtų geriausias sprendimas iš pasiūlytų, tačiau reikėtų sukurti papildoma objekto aptikimo modelį, todėl yra pasirenkamas tikimiausias ketvirčio sprendimas.

Šiame tyrime buvo iškerpamas aukštesnysis dešinės pusės ketvirtis iš duomenų rinkinio, kuriame yra transporto priemonės virš 96x96 pikselių raiškos. Nuotraukos nebuvo atrenkamos, nes buvo imituojama reali veikimo situacija, kai „YOLO“ modelis neatranda asmens transporto priemonėje. Tikslumo tikrinimui buvo naudojamas „VGG“ modelis. Naudoti 78,2 % tikslumo svoriai iš praeito tyrimo, nes šis modelis turėjo geriausias rezultatus. Priminimas, kad tame tyrime buvo naudojamos 3 klasės. Tyrimo rezultatai pateikti lentelėje:

3.21 lentelė. Vairuotojo, kalbančio telefonu, nustatymas, naudojant transporto priemonės nuotraukos ketvirtį

Modelis	Tikslumas, %	Teisingi teigiami, %	Teisingi neigiami, %	F vertė, %
„VGG“	46,2	49,3	46,4	59,0

Iš gautų rezultatų matome, kad naudoti šio transporto priemonės vairuotojo atpažinimo šiame modelyje yra neverta dėl jo tikslumo. Šie rezultatai buvo pasiekti, nes iškerpant ketvirtį nuotraukos, atsiranda nuotraukų, kuriose nėra vairuotojo arba jis matosi tik mažoje dalyje nuotraukos. Autobusų, sunkvežimių atveju yra matomas ir ilgas transporto priemonių stogas, todėl ketvirtis gali būti tiesiog transporto priemonės stogas. Rezultatai būtų geresni, naudojant transporto priemonės lango aptikimo algoritmą. Teigiama, kad atliktame tyrime, su paprastu filtru buvo atrastas transporto priemonės stiklas 95 % tikslumu [63]. Naudojant šį algoritmą kartu su transporto priemonės aptikimo modeliu, galėtumėme sumažinti nepatikrintų nuotraukų skaičių 20 %.

3.5.7. Sumažintos įvesties modelių tyrimas

Šis tyrimas yra atliekamas, nes pastebėta, kad geriausi rezultatai yra pasiekiami su „VGG“ modeliu, kurio įvestis yra sumažinta iki 64x64 raiškos. Kaip ir minėta anksčiau, didinant nuotraukų raišką, nuotrauka praranda savo bruožus. Galimai kiti 3 modeliai šiame projekte pasirodė prasčiau, nes jų įvesties raiška yra 3,5 karto didesnė, negu „VGG“, todėl norima ištirti ar sumažinus šių modelių įvesties raišką gausis panašūs rezultatai į „VGG“. Taip pat norima sužinoti koks yra pasiekiamas modelių mokymo gylis, naudojant grupę duomenų rinkinio generatorių.

Šio tyrimo metu modeliai yra apmokomi 50 epochų. Naudojami 48x48 raiškos duomenų rinkiniai. Visų modelių vidinė architektūra yra identiška duomenų papildytojų grupės tyrimo metu. Pakeitus „AlexNet“ modelio įvestį į 64x64, gaunama klaida, kad modelio įvestis yra per mažos raiškos modeliui sujungti. Buvo apsvarstyti 3 sprendimo variantai: 1) naikinti 2,3,4 ir 5-tą sluoksnius; 2) naikinti pirmą sluoksnį; 3) sumažinti modelio pirmo sluoksnio filtro žingsnį. Pirmasis sprendimo variantas turėjo tik vieną sluoksnį, mokymo metu jis nepagaudavo bruožų ir su mokymo duomenų rinkiniu. Antrasis ir trečiasis sprendimai yra labai panašūs, todėl buvo pasirinktas trečiasis variantas, kad būtų išlaikoma kuo panašesnė „AlexNet“ modelio struktūra. „VGG“ viskas buvo identiška kaip ir duomenų papildymo grupės tyrimo. „GoogLeNet“ sumažinus iki 64x64 būdavo rašoma, kad minimali galima įvesties raiška yra 75x75. Buvo galima rinktis arba naikinti „GoogLeNet“ paskutinius modulius arba padidinti įvestį iki 75x75. Siekiant išlaikyti kuo panašesnį modelį į pradinį, buvo pasirinkta įvesties raišką padidinti iki 75x75. Sumažinus „ResNet“ modelio įvestį, problemų neatsirado. Tyrimo rezultatai pateikti lentelėje:

3.22 lentelė. Sumažintos įvesties modelių tyrimo rezultatai

Modelis	Tikslumas, %	Teisingi teigiami, %	Teisingi neigiami, %	F vertė, %
„AlexNet“	62,1	38	64	9
„VGG“	74,4	65	74	55

„GoogLeNet“	67,9	50	65	13
„ResNet“	66,7	0	64	0

Tyrimo metu buvo tiriamas ir modelių mokymo gylis. Modelių mokymo gylio tyrime pastebėta, kad „dropout“ gali padidinti modelio tikslumą, tačiau šis tikslumas bus pasiekiamas sėkmingos inicializacijos dėka. Šiame tyrime, naudojant „dropout“ sluoksnį, sumažėjo modelio mokymo patikimumas, validacijos tikslumai pradėjo svyruoti labiau nei įprastai. Taip pat buvo pastebėta, kad naudojant „dropout“ sluoksnį, modeliai yra labiau linkę įvertinti visas nuotraukas kaip vieną klasę ir neišmokti bruožų. Todėl visi lentelėje pavaizduoti rezultatai yra be „dropout“ sluoksnio (išskyrus „AlexNet“, kurio architektūroje yra „dropout“ sluoksniai). Šiuo tyrimu nėra sakoma, kad „dropout“ neturėtų būti naudojamas modeliuose, tačiau šiuo atveju, kai mums trūksta stabilumo validacijos tikslume, yra rekomenduojama atsisakyti „dropout“ sluoksnio“.

Šiame tyrime „AlexNet“ modelis vėl nesugebėjo atskirti bruožus, didžioji modelio mokymo dalis buvo [40;55] % validacijos tikslume. „VGG“ modelio tyrimuose buvo pastebėta, kad geriausi rezultatai, dažniausiai, yra gaunami iki pirmųjų 10 epochų pabaigos, po 10 epochų modelis pradėdavo svyruoti smarkiau. „GoogLeNet“ atvejis yra labai įdomus, nes su sumažinta įvestimi modelis ir toliau spėdavo vieną klasę, tačiau skirtingai negu praeitais kartais, šiame tyrime „GoogLeNet“ modelis stengdavosi atrasti kokius nors bruožus (modelio tikslumas nebuvo konstanta, kaip praeitais atvejais), tačiau bruožų ieškojimas nepasiteisino, nes „GoogLeNet“ keisdavo savo pagrindinę spėjamą klasę kas 1-5 epochas, o tikslumas kisdavo neženkliai nuo įvertinimo, kad visos klasės yra vienodos (kitimas vyko 5 % režiuose). „ResNet“ mokymas neišsiskyrė nuo kitų mokymų rezultatų, išskyrus tuo, kad šis tikslumas buvo pasiektas atsitiktiniu ženkliau tikslumo padidėjimu. „ResNet“ ir „GoogLeNet“ buvo tiriami, kai visi arba visi iki paskutinių 6/31 („ResNet“/„GoogLeNet“) arba 0 sluoksnių yra užšaldyti. Šie papildomi tyrimai nedavė išskirtinių rezultatų, visais atvejais tikslumas buvo panašus ir buvo stengiamasi atspėti vieną klasę.

Iš rezultatų matome, kad ne visiems modeliams jų įvesties sumažinimas padidina tikslumą, jeigu testavimo metu naudojamas mažesnės raiškos duomenų rinkinys. Iš šio tyrimo galime spręsti, kad gautumėme geresnius validacijos rezultatus, reikėtų pagerinti duomenų rinkinį. Šio duomenų rinkinio trūkumai, kurie galėjo padaryti įtaką rezultatams: nuotraukose yra pašalinio garso, atrandami objektai yra per mažos raiškos, kad būtų gerai atliekama analizė, kai kurių nuotraukų galvos pozicijoje yra uždėtas „blur“ filtras. Pagrindinis duomenų rinkimo trūkumas – jis buvo surinkas iš „Baidu“ paieškos rezultatų, todėl buvo praleisti puslapiai, kuriuose yra daugiau geresnės kokybės nuotraukų, „Baidu“ vienu metu buvo pakeitęs nuotraukų atvaizdavimo sistemą, todėl paieškos rezultatuose rodė mažesnės raiškos nuotraukas. Magistrinio projekto metu buvo bandyta naršyti pačiuose paieškos rezultatų puslapiuose, tačiau buvo susidurta su problema, kad mano kinų kalbos įgūdžiai nėra tokie pažengę, kad galėčiau ieškoti duomenų pačiuose puslapiuose, o „Chrome“ naršyklės automatinis puslapio vertimas naudos neduoda, ypač dėl priežasties, kad kinai mėgsta navigacijos objektus vaizduoti paveikslėliais. Vienos iš atrastų pažeidėjų nuotraukų puslapyje buvo pranešama, kad vieno Kinijos miesto policija pasidalino dideliu kiekiu pažeidėjų nuotraukų (tarp jų buvo paviešinta ir atrasta nuotrauka), tačiau naršant to miesto policijos svetainę, joks duomenų rinkinys nebuvo atrastas. Jeigu būtų renkamas visas duomenų rinkinys rankiniu būdu, būtų duomenų rinkinio rinkimas žymiai lėtesnis, nebent būtų atrasta viešai prieinama pažeidėjų duomenų bazė, tačiau pats duomenų rinkinys būtų aukštesnės kokybės. Apibendrinus, šio projekto geresniam duomenų rinkinio įgyvendinimui reikėtų geresnių kinų kalbos žinių arba laiko.

Išvados

1. Buvo sukurtas nuotraukų duomenų rinkinys, susidedantis iš 415 nuotraukų gautų iš interneto ir 3890 nufotografuotų nuotraukų;
2. su asmens nuotrauka yra gaunamas 12,1 % aukštesnis tikslumas, negu naudojant transporto priemonę, nustatant ar vairuotojas kalba telefonu;
3. sumažinus aukštesnės raiškos mokymo duomenų rinkinio raišką iki internetinio duomenų rinkinio raiškos, vidutiniškai gautas 7,4 % tikslumo pagerėjimas;
4. duomenų papildymo metu, apmokymas, naudojant grupę duomenų papildytojų, yra spartesnis, negu vieno duomenų papildytojo, galima išgauti įvairesnį mokymo duomenų rinkinį;
5. modelio įvesties sumažinimas iki testavimo duomenų rinkinio raiškos negarantuoja padidėjusio modelio tikslumo.

Literatūros sąrašas

1. HINKELMANN, K. Neural Networks. In [interaktyvus]. [žiūrėta 2020-05-18]. . Prieiga per internetą: <http://didattica.cs.unicam.it/lib/exe/fetch.php?media=didattica:magistrale:kebi:ay_1718:ke-11_neural_networks.pdf>.
2. TING, K.M. Confusion Matrix. In SAMMUT, C. - WEBB, G.I.Sud. *Encyclopedia of Machine Learning* [interaktyvus]. Boston, MA: Springer US, 2010. p. 209. ISBN 978-0-387-30164-8Prieiga per internetą: <https://doi.org/10.1007/978-0-387-30164-8_157>.
3. The History of Artificial Intelligence - Science in the News. In [interaktyvus]. 2017. [žiūrėta 2020-03-05]. Prieiga per internetą: <<http://sitn.hms.harvard.edu/flash/2017/history-artificial-intelligence/>>.
4. Kasparov vs. Deep Blue | The Match That Changed History - Chess.com. In [interaktyvus]. 2018. [žiūrėta 2020-03-05]. Prieiga per internetą: <<https://www.chess.com/article/view/deep-blue-kasparov-chess>>.
5. The DARPA Grand Challenge: Ten Years Later. In [interaktyvus]. 2014. [žiūrėta 2020-03-05]. Prieiga per internetą: <<https://www.darpa.mil/news-events/2014-03-13>>.
6. NOVET, J. Microsoft researchers say their newest deep learning system beats humans -- and Google | VentureBeat. In [interaktyvus]. 2015. [žiūrėta 2020-03-05]. Prieiga per internetą: <<https://venturebeat.com/2015/02/09/microsoft-researchers-say-their-newest-deep-learning-system-beats-humans-and-google/>>.
7. Distracted Driving | NHTSA. In [interaktyvus]. [žiūrėta 2019-01-24]. Prieiga per internetą: <<https://www.nhtsa.gov/risky-driving/distracted-driving>>.
8. HIGHWAY TRAFFIC SAFETY ADMINISTRATION, N. - DEPARTMENT OF TRANSPORTATION, U. [interaktyvus]. 2016. [žiūrėta 2019-01-24]. Prieiga per internetą: <<https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812426>>.
9. Kas antras vairuotojas telefonu naudojasi pažeisdamas taisykles | GAZAS.LT. In [interaktyvus]. [žiūrėta 2019-01-24]. Prieiga per internetą: <<https://www.15min.lt/gazas/naujiena/saugukelyje-lt/kas-antras-vairuotojas-telefonu-naudojasi-pazeisdamas-taisykles-787-712685>>.
10. Atviri duomenys | Valstybės įmonė REGITRA. In [interaktyvus]. [žiūrėta 2019-01-24]. Prieiga per internetą: <<https://www.regitra.lt/lt/atviri-duomenys/keliu-transporto-priemoniu-vairuotoju-registre-iregistruotu-vairuotoju-turinciu-galiojancius-vairuotojo-pazymejimus-skaicius>>.
11. NG, A. Unsupervised Learning - Introduction | Coursera. In [interaktyvus]. [žiūrėta 2020-03-05]. Prieiga per internetą: <<https://www.coursera.org/lecture/machine-learning/unsupervised-learning-oLRZo>>.
12. Section 7 (Week 7). In [interaktyvus]. [žiūrėta 2020-05-21]. Prieiga per internetą: <<http://cs230.stanford.edu/section/7/>>.
13. Introduction to Unsupervised Learning | Algorithmia Blog. In [interaktyvus]. 2018. [žiūrėta 2020-03-05]. Prieiga per internetą: <<https://algorithmia.com/blog/introduction-to-unsupervised-learning>>.
14. How do neurons work? - Queensland Brain Institute - University of Queensland. In [interaktyvus]. [žiūrėta 2019-01-25]. Prieiga per internetą: <<https://qbi.uq.edu.au/brain-basics/brain/brain-physiology/how-do-neurons-work>>.
15. MA, T. et al. .
16. GOODFELLOW, I. et al. *Deep Learning* [interaktyvus]. . [s.l.]: MIT Press, 2016. 327–328 p.
17. The AlexNet model - Deep Learning for Computer Vision [Book]. In [interaktyvus]. [žiūrėta 2019-01-24]. Prieiga per internetą: <<https://www.oreilly.com/library/view/deep-learning-for/9781788295628/1ca58a07-6a2a-44ea-9b14-55b07ffebe8f.xhtml>>.
18. ALOM, Z. et al. The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches. In [interaktyvus]. [žiūrėta 2019-01-24]. . Prieiga per internetą: <<https://arxiv.org/ftp/arxiv/papers/1803/1803.01164.pdf>>.
19. ImageNet Large Scale Visual Recognition Competition 2012 (ILSVRC2012). In [interaktyvus]. [žiūrėta 2019-01-24]. Prieiga per internetą: <<http://image-net.org/challenges/LSVRC/2012/results.html>>.
20. JAIN, P. Complete Guide of Activation Functions - Towards Data Science. In [interaktyvus]. 2019. [žiūrėta 2020-03-05]. Prieiga per internetą: <<https://towardsdatascience.com/complete-guide-of-activation-functions-34076e95d044>>.
21. KRIZHEVSKY, A. et al. ImageNet Classification with Deep Convolutional Neural Networks. In [interaktyvus]. [žiūrėta 2019-01-24]. . Prieiga per internetą: <<http://code.google.com/p/cuda-convnet/>>.
22. 1. Introduction to Deep Learning - TensorFlow for Deep Learning [Book]. In [interaktyvus]. [žiūrėta 2019-01-24]. Prieiga per internetą: <<https://www.oreilly.com/library/view/tensorflow-for-deep/9781491980446/ch01.html>>.
23. SIMONYAN, K. - ZISSERMAN, A. VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION. In [interaktyvus]. 2015. [žiūrėta 2019-01-24]. . Prieiga per internetą: <<http://www.robots.ox.ac.uk/>>.
24. The History of Neural Networks - Dataconomy. In [interaktyvus]. [žiūrėta 2019-01-26]. Prieiga per internetą: <<https://dataconomy.com/2017/04/history-neural-networks/>>.
25. SERENGIL, S.I. Deep Face Recognition with VGG-Face in Keras | sefiks.com. In [interaktyvus]. 2018. [žiūrėta

- 2020-03-07]. Prieiga per internetą: <<https://sefiks.com/2018/08/06/deep-face-recognition-with-keras/>>.
26. SZEGEDY, C. et al. Going deeper with convolutions. In [interaktyvus]. [žiūrėta 2019-01-26]. . Prieiga per internetą: <<https://arxiv.org/pdf/1409.4842.pdf>>.
 27. NIELSEN, M.A. Neural Networks and Deep Learning. In [interaktyvus]. 2015. [žiūrėta 2019-01-26]. . Prieiga per internetą: <<http://neuralnetworksanddeeplearning.com/chap5.html>>.
 28. FUNG, V. An Overview of ResNet and its Variants - Towards Data Science. In [interaktyvus]. 2017. [žiūrėta 2020-03-07]. Prieiga per internetą: <<https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035>>.
 29. NG, A. ResNets - deeplearning.ai | Coursera. In [interaktyvus]. [žiūrėta 2020-03-07]. Prieiga per internetą: <<https://www.coursera.org/learn/convolutional-neural-networks/lecture/HAhz9/resnets>>.
 30. HE, K. et al. Deep Residual Learning for Image Recognition. In [interaktyvus]. [žiūrėta 2019-01-26]. . Prieiga per internetą: <<http://image-net.org/challenges/LSVRC/2015/>>.
 31. GRM, K. et al. Strengths and Weaknesses of Deep Learning Models for Face Recognition Against Image Degradations. In [interaktyvus]. [žiūrėta 2020-03-08]. . Prieiga per internetą: <<https://github.com/kgm/face-recog-eval>>.
 32. SACHAN, A. Zero to Hero: Guide to Object Detection using Deep Learning: Faster R-CNN, YOLO, SSD – CV-Tricks.com. In [interaktyvus]. [žiūrėta 2020-03-08]. Prieiga per internetą: <<https://cv-tricks.com/object-detection/faster-r-cnn-yolo-ssd/>>.
 33. REDMON, J. et al. You Only Look Once: Unified, Real-Time Object Detection. In [interaktyvus]. [žiūrėta 2019-01-26]. . Prieiga per internetą: <<http://pjreddie.com/yolo/>>.
 34. LIU, W. et al. SSD: Single Shot MultiBox Detector. In [interaktyvus]. [žiūrėta 2019-01-26]. . Prieiga per internetą: <<https://github.com/weiliu89/caffe/tree/ssd>>.
 35. CHEN, V. - CHOU, E. Practical Object Detection and Segmentation. In [interaktyvus]. [žiūrėta 2019-01-26]. . Prieiga per internetą: <http://cs231n.stanford.edu/slides/2018/cs231n_2018_ds06.pdf>.
 36. NG, A. CS229 Lecture notes. In [interaktyvus]. [žiūrėta 2020-03-09]. . Prieiga per internetą: <<http://cs229.stanford.edu/notes/cs229-notes1.pdf>>.
 37. NG, A. Normal Equation - Stanford University | Coursera. In [interaktyvus]. [žiūrėta 2019-06-18]. Prieiga per internetą: <<https://www.coursera.org/learn/machine-learning/lecture/2DKxQ/normal-equation>>.
 38. GROSSE, R. Lecture 8: Optimization. In [interaktyvus]. [žiūrėta 2020-03-09]. Prieiga per internetą: <[https://www.cs.toronto.edu/~rgrosse/courses/csc321_2018/readings/L08 Optimization.pdf](https://www.cs.toronto.edu/~rgrosse/courses/csc321_2018/readings/L08%20Optimization.pdf)>.
 39. PERLA, J. Notes on AdaGrad. In [interaktyvus]. [žiūrėta 2019-06-17]. . Prieiga per internetą: <<https://web.archive.org/web/20150330033637/http://seed.ucsd.edu/mediawiki/images/6/6a/Adagrad.pdf>>.
 40. RUDER, S. An overview of gradient descent optimization algorithms. In [interaktyvus]. 2016. [žiūrėta 2020-03-15]. Prieiga per internetą: <<https://ruder.io/optimizing-gradient-descent/index.html>>.
 41. ZHANG, J. Gradient Descent based Optimization Algorithms for Deep Learning Models Training. In [interaktyvus]. [žiūrėta 2019-06-17]. . Prieiga per internetą: <<https://arxiv.org/pdf/1903.03614.pdf>>.
 42. KARPATY, A. A Peek at Trends in Machine Learning. In [interaktyvus]. 2017. [žiūrėta 2020-03-15]. Prieiga per internetą: <<https://medium.com/@karpathy/a-peek-at-trends-in-machine-learning-ab8a1085a106>>.
 43. KINGMA, D.P. - LEI BA, J. ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION. In [interaktyvus]. [žiūrėta 2019-06-18]. . Prieiga per internetą: <<https://arxiv.org/pdf/1412.6980v8.pdf>>.
 44. Plepiems vairuotojams ateina sunkūs laikai: juos stebės vaizdo kameros - DELFI Auto. In [interaktyvus]. 2018. Prieiga per internetą: <<https://www.delfi.lt/auto/audos-kvitas/plepiems-vairuotojams-ateina-sunkus-laikai-juos-stebes-vaizdo-kameros.d?id=78006325>>.
 45. OU, C. *Deep Learning-based Driver Behavior Modeling and Analysis* [interaktyvus]. [s.l.]: Waterloo, 2019. Prieiga per internetą: <https://uwspace.uwaterloo.ca/bitstream/handle/10012/15334/Ou_Chaojie.pdf?sequence=3&isAllowed=y>.
 46. ZHANG, X. et al. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In [interaktyvus]. [žiūrėta 2020-03-15]. . Prieiga per internetą: <<https://arxiv.org/pdf/1707.01083.pdf>>.
 47. HOWARD, A.G. et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. In [interaktyvus]. [žiūrėta 2020-03-15]. . Prieiga per internetą: <<https://arxiv.org/pdf/1704.04861.pdf>>.
 48. SANDLER, M. et al. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In [interaktyvus]. [žiūrėta 2020-03-15]. . Prieiga per internetą: <<https://arxiv.org/pdf/1801.04381.pdf>>.
 49. CELLEPATHY INC. Restricting mobile device usage. In [interaktyvus]. 2012. Prieiga per internetą: <<https://patents.justia.com/patent/9800716>>.
 50. VERIFY™ — Distracted Driving Prevention Technology - YouTube. In [interaktyvus]. 2016. [žiūrėta 2020-03-15]. Prieiga per internetą: <<https://www.youtube.com/watch?v=NndeKlyDPko>>.
 51. AI on the lookout for not-so-good drivers - Nikkei Asian Review. In [interaktyvus]. [žiūrėta 2019-01-24]. Prieiga per

- interneta: <<https://asia.nikkei.com/Business/Technology/AI-on-the-lookout-for-not-so-good-drivers>>.
52. Digital Media Professionals Inc. » DMP, going into full scale in professional services for safe driving support system development. In [interaktyvus]. [žiūrėta 2020-03-15]. Prieiga per internetą: <<https://www.dmprof.com/en/news/2019/15333/>>.
 53. State of the developer nation q1 2017. In [interaktyvus]. 2017. Prieiga per internetą: <<https://mwc.gr/presentations/2017/konstantinou.pdf>>.
 54. State Farm Distracted Driver Detection | Kaggle. In [interaktyvus]. [žiūrėta 2020-03-30]. Prieiga per internetą: <<https://www.kaggle.com/c/state-farm-distracted-driver-detection>>.
 55. KRAUSE, J. Cars Dataset. In [interaktyvus]. [žiūrėta 2020-03-30]. Prieiga per internetą: <https://ai.stanford.edu/~jkrause/cars/car_dataset.html>.
 56. Vehicle Re-Identification for Automatic Video Traffic Surveillance [ATS-CVPR 2016] | Traffic Research. In [interaktyvus]. [žiūrėta 2020-03-30]. Prieiga per internetą: <<https://medusa.fit.vutbr.cz/traffic/research-topics/detection-of-vehicles-and-datasets/vehicle-re-identification-for-automatic-video-traffic-surveillance-ats-cvpr-2016/>>.
 57. 伊州区机动车不文明交通行为曝光. In [interaktyvus]. [žiūrėta 2020-04-11]. Prieiga per internetą: <<https://new.qq.com/omn/20190817/20190817A0OMYJ00.html>>.
 58. You only look once (YOLO) is a state-of-the-art, real-time object detection system. In [interaktyvus]. [žiūrėta 2020-03-05]. Prieiga per internetą: <<https://pjreddie.com/darknet/yolo/>>.
 59. A Keras implementation of YOLOv3 (Tensorflow backend) - qqwweee/keras-yolo3. In [interaktyvus]. [žiūrėta 2020-03-05]. Prieiga per internetą: <<https://github.com/qqwweee/keras-yolo3>>.
 60. HUANG, L. et al. Data Augmentation for Deep Learning-based Radio Modulation Classification. In [interaktyvus]. [žiūrėta 2020-05-10]. . Prieiga per internetą: <<https://arxiv.org/pdf/1912.03026.pdf>>.
 61. Overfit and underfit | TensorFlow Core. In [interaktyvus]. [žiūrėta 2020-05-10]. Prieiga per internetą: <https://www.tensorflow.org/tutorials/keras/overfit_and_underfit>.
 62. ROSEBROCK, A. Keras ImageDataGenerator and Data Augmentation. In [interaktyvus]. 2019. [žiūrėta 2020-04-26]. Prieiga per internetą: <<https://www.pyimagesearch.com/2019/07/08/keras-imagedatagenerator-and-data-augmentation/>>.
 63. HAO, X. et al. Vehicle window detection based on line features. In *Proceedings - 2012 International Symposium on Instrumentation and Measurement, Sensor Network and Automation, IMSNA 2012* [interaktyvus]. [s.l.]: IEEE, 2012. p. 261–263. [žiūrėta 2020-05-11]. Prieiga per internetą: <<http://ieeexplore.ieee.org/document/6324563/>>.

Priedai

Priedas 1. Apmokytų modelių klaidų matricos, mokymo metu naudojant transporto priemonių duomenų rinkinį

„AlexNet“	<pre> Classification Report precision recall f1-score not talking 0.45 0.37 0.41 talking 0.55 0.64 0.59 accuracy 0.52 macro avg 0.50 0.50 0.50 weighted avg 0.51 0.52 0.51 </pre>
„VGG“	<pre> Classification Report precision recall f1-score not talking 0.45 0.79 0.57 talking 0.55 0.21 0.30 accuracy 0.47 macro avg 0.50 0.50 0.44 weighted avg 0.50 0.47 0.43 </pre>
„GoogLeNet“	<pre> Classification Report precision recall f1-score not talking 1.00 0.01 0.01 talking 0.55 1.00 0.71 accuracy 0.55 macro avg 0.78 0.50 0.36 weighted avg 0.75 0.55 0.39 </pre>
„ResNet“	<pre> Classification Report precision recall f1-score not talking 1.00 0.01 0.01 talking 0.55 1.00 0.71 accuracy 0.55 macro avg 0.78 0.50 0.36 weighted avg 0.75 0.55 0.39 </pre>

Priedas 2. Apmokytų modelių klaidų matricos, mokymo metu naudojant vairuotojų duomenų rinkinį

„AlexNet“	<pre> Classification Report precision recall f1-score not talking 0.69 0.40 0.50 talking 0.47 0.74 0.57 accuracy 0.54 macro avg 0.58 0.57 0.54 weighted avg 0.59 0.54 0.53 </pre>
-----------	---

„VGG“	<pre> Classification Report precision recall f1-score not talking 0.62 0.78 0.69 talking 0.52 0.34 0.41 accuracy 0.60 macro avg 0.57 weighted avg 0.58 </pre>
„GoogLeNet“	<pre> precision recall f1-score not talking 0.59 1.00 0.74 talking 0.00 0.00 0.00 accuracy 0.59 macro avg 0.29 weighted avg 0.34 </pre>
„ResNet“	<pre> Classification Report precision recall f1-score not talking 0.60 0.90 0.72 talking 0.53 0.15 0.24 accuracy 0.59 macro avg 0.56 weighted avg 0.57 </pre>

Priedas 3. Apmokytų modelių klaidų matricos, mokymo metu naudojant virš 96x96 raiškos transporto priemonių duomenų rinkinį

„AlexNet“	<pre> Classification Report precision recall f1-score not talking 0.51 0.44 0.47 talking 0.52 0.59 0.55 accuracy 0.51 macro avg 0.51 weighted avg 0.51 </pre>
„VGG“	<pre> Classification Report precision recall f1-score not talking 0.50 0.48 0.49 talking 0.51 0.53 0.52 accuracy 0.51 macro avg 0.51 weighted avg 0.51 </pre>
„GoogLeNet“	<pre> Classification Report precision recall f1-score not talking 0.54 0.77 0.64 talking 0.61 0.35 0.45 accuracy 0.56 macro avg 0.58 weighted avg 0.58 </pre>

„ResNet“	Classification Report			
		precision	recall	f1-score
	not talking	0.57	0.32	0.41
	talking	0.53	0.76	0.62
	accuracy			0.54
	macro avg	0.55	0.54	0.52
	weighted avg	0.55	0.54	0.52

Priedas 4. Apmokytų modelių klaidų matricos, mokymo metu naudojant virš 128x128 raiškos transporto priemonių duomenų rinkinį

„AlexNet“	Classification Report			
		precision	recall	f1-score
	not talking	0.54	0.45	0.49
	talking	0.51	0.60	0.55
	accuracy			0.52
	macro avg	0.53	0.53	0.52
	weighted avg	0.53	0.52	0.52
„VGG“	Classification Report			
		precision	recall	f1-score
	not talking	0.52	0.54	0.53
	talking	0.50	0.48	0.49
	accuracy			0.51
	macro avg	0.51	0.51	0.51
	weighted avg	0.51	0.51	0.51
„GoogLeNet“	Classification Report			
		precision	recall	f1-score
	not talking	0.54	0.81	0.65
	talking	0.58	0.27	0.37
	accuracy			0.55
	macro avg	0.56	0.54	0.51
	weighted avg	0.56	0.55	0.51
„ResNet“	Classification Report			
		precision	recall	f1-score
	not talking	0.63	0.38	0.47
	talking	0.54	0.76	0.63
	accuracy			0.57
	macro avg	0.58	0.57	0.55
	weighted avg	0.58	0.57	0.55

Priedas 5. Apmokytų modelių klaidų matricos, mokymo metu naudojant virš 48x48 raiškos vairuotojų duomenų rinkinį

„AlexNet“	<pre> Classification Report precision recall f1-score not talking 0.64 0.67 0.66 talking 0.37 0.33 0.35 accuracy 0.55 macro avg 0.50 weighted avg 0.54</pre>
„VGG“	<pre> Classification Report precision recall f1-score not talking 0.64 0.80 0.71 talking 0.39 0.22 0.28 accuracy 0.59 macro avg 0.52 weighted avg 0.55</pre>
„GoogLeNet“	<pre> precision recall f1-score not talking 0.64 1.00 0.78 talking 0.00 0.00 0.00 accuracy 0.64 macro avg 0.32 weighted avg 0.41</pre>
„ResNet“	<pre> Classification Report precision recall f1-score not talking 0.65 0.78 0.71 talking 0.40 0.26 0.32 accuracy 0.59 macro avg 0.53 weighted avg 0.56</pre>

Priedas 6. Apmokytų modelių klaidų matricos, mokymo metu naudojant virš 64x64 raiškos vairuotojų duomenų rinkinį

„AlexNet“	<pre> Classification Report precision recall f1-score not talking 0.70 0.74 0.72 talking 0.45 0.40 0.42 accuracy 0.62 macro avg 0.57 weighted avg 0.61</pre>
„VGG“	<pre> Classification Report precision recall f1-score not talking 0.66 0.83 0.74 talking 0.38 0.20 0.26 accuracy 0.61 macro avg 0.52 weighted avg 0.57</pre>

„GoogLeNet“	<pre> Classification Report precision recall f1-score not talking 0.69 0.90 0.78 talking 0.55 0.23 0.32 accuracy 0.67 macro avg 0.62 0.56 0.55 weighted avg 0.64 0.67 0.62 </pre>
„ResNet“	<pre> Classification Report precision recall f1-score not talking 0.67 0.98 0.80 talking 0.70 0.09 0.16 accuracy 0.67 macro avg 0.68 0.54 0.48 weighted avg 0.68 0.67 0.58 </pre>

Priedas 7. Apmokytų modelių klaidų matricos, mokymo metu naudojant virš 64x64 raiškos vairuotojų duomenų rinkinį ir virš 96x96 raiškos transporto priemonių duomenų rinkinius kartu

„AlexNet“	<pre> Classification Report precision recall f1-score not talking 0.55 0.86 0.67 talking 0.46 0.15 0.22 accuracy 0.53 macro avg 0.50 0.50 0.45 weighted avg 0.51 0.53 0.47 </pre>
„VGG“	<pre> Classification Report precision recall f1-score not talking 0.54 0.82 0.65 talking 0.45 0.18 0.26 accuracy 0.53 macro avg 0.50 0.50 0.46 weighted avg 0.50 0.53 0.47 </pre>
„GoogLeNet“	<pre> Classification Report precision recall f1-score not talking 0.59 0.67 0.63 talking 0.52 0.43 0.48 accuracy 0.56 macro avg 0.56 0.55 0.55 weighted avg 0.56 0.56 0.56 </pre>
„ResNet“	<pre> Classification Report precision recall f1-score not talking 0.60 0.38 0.47 talking 0.48 0.69 0.57 accuracy 0.52 macro avg 0.54 0.54 0.52 weighted avg 0.55 0.52 0.51 </pre>

Priedas 8. Apmokytų modelių klaidų matricos, mokymo metu sumažinant mokymo duomenų rinkinį iki 48x48 raiškos

„AlexNet“	<pre> Classification Report precision recall f1-score not talking 0.65 0.85 0.74 talking 0.42 0.19 0.26 accuracy 0.61 macro avg 0.53 0.52 0.50 weighted avg 0.57 0.61 0.56 </pre>
„VGG“	<pre> Classification Report precision recall f1-score not talking 0.70 0.71 0.70 talking 0.47 0.46 0.47 accuracy 0.62 macro avg 0.59 0.59 0.59 weighted avg 0.62 0.62 0.62 </pre>
„GoogLeNet“	<pre> precision recall f1-score not talking 0.64 1.00 0.78 talking 0.00 0.00 0.00 accuracy 0.64 macro avg 0.32 0.50 0.39 weighted avg 0.41 0.64 0.50 </pre>
„ResNet“	<pre> Classification Report precision recall f1-score not talking 0.67 0.70 0.68 talking 0.42 0.39 0.41 accuracy 0.59 macro avg 0.55 0.55 0.55 weighted avg 0.58 0.59 0.58 </pre>

Priedas 9. Apmokytų modelių klaidų matricos, mokymo metu sumažinant mokymo duomenų rinkinį iki 48x48 raiškos ir naudojant duomenų generatorių grupę

„AlexNet“	<pre> Classification Report precision recall f1-score not talking 0.67 0.85 0.75 talking 0.48 0.24 0.32 accuracy 0.63 macro avg 0.57 0.55 0.54 weighted avg 0.60 0.63 0.59 </pre>
-----------	--

„VGG“	Classification Report			
		precision	recall	f1-score
	not talking	0.77	0.88	0.82
	talking	0.72	0.53	0.61
	accuracy			0.76
	macro avg	0.75	0.71	0.72
	weighted avg	0.75	0.76	0.75

Priedas 10. Apmokytų modelių klaidų matricos, mokymo metu sumažinant mokymo duomenų rinkinį iki 64x64 raiškos ir naudojant duomenų generatorių grupę

„VGG“	Classification Report			
		precision	recall	f1-score
	not talking	0.72	0.95	0.82
	talking	0.77	0.31	0.44
	accuracy			0.73
	macro avg	0.74	0.63	0.63
	weighted avg	0.74	0.73	0.69
„GoogLeNet“	Classification Report			
		precision	recall	f1-score
	not talking	0.65	1.00	0.79
	talking	0.00	0.00	0.00
	accuracy			0.65
	macro avg	0.33	0.50	0.39
	weighted avg	0.43	0.65	0.52
„ResNet“	Classification Report			
		precision	recall	f1-score
	not talking	0.65	1.00	0.79
	talking	0.00	0.00	0.00
	accuracy			0.65
	macro avg	0.33	0.50	0.39
	weighted avg	0.43	0.65	0.52

Priedas 11. Apmokytų modelių klaidų matricos, mažesnės raiškos įvesties tyrime

„AlexNet“	Classification Report			
		precision	recall	f1-score
	not talking	0.64	0.95	0.77
	talking	0.38	0.05	0.09
	accuracy			0.63
	macro avg	0.51	0.50	0.43
	weighted avg	0.54	0.63	0.52
„VGG“	Classification Report			
		precision	recall	f1-score
	not talking	0.74	0.86	0.79
	talking	0.65	0.47	0.55
	accuracy			0.72
	macro avg	0.70	0.66	0.67
	weighted avg	0.71	0.72	0.71

„GoogLeNet“	<pre> Classification Report precision recall f1-score not talking 0.65 0.96 0.77 talking 0.50 0.07 0.13 accuracy 0.64 macro avg 0.57 0.52 0.45 weighted avg 0.59 0.64 0.54 </pre>			
„ResNet“	<pre> precision recall f1-score not talking 0.64 1.00 0.78 talking 0.00 0.00 0.00 accuracy 0.64 macro avg 0.32 0.50 0.39 weighted avg 0.41 0.64 0.50 </pre>			