

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Matthias Rajamets

Masinõppeprotsessi automatiseerimine ristmiku kinnisõitmise vähendamise süsteemile

Bakalaureusetöö

Juhendaja: Martin Rebane
MSc

Tallinn 2021

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Matthias Rajamets

29.04.2021

Annotatsioon

Käesoleva töö põhieesmärgiks oli automatiseerida konvolutsioonilise närvivõrgu mudeli jaoks andmete kogumine. Varasemalt on kahe lõputöö raames arendatud mudel, mis suudab 83% täpsusega ennustada 10 sekundit ette, kas ristmik sõidetakse kinni või mitte. Vastavalt vajadusele annab mudel valgusfoorile signaali punase tule varasemaks sisselülitamiseks. See süsteem on loodud ristmike kinnisõitmiste likvideerimiseks, et rohelise tule ajal ei sõidaks rohkem autosid ristmikule kui neid sealt ära saab sõita. Lõputöö on valminud Tallinna Linnavalitsuse abiga, kes on andnud ligipääsu kahele kaamerale Mustamäel Sõpruse-Tammsaare ristmikul.

Antud kaameratest saadud videopiltidest on võimalik koguda ennustusmudeli jaoks treeningandmeid, mille protsessi antud töö käigus automatiseeriti. Töö käigus treeniti välja ka lihtsam mudel, mis automaatselt klassifitseeris ette antud pildid ristmikest kinnisõidetuteks või mitte kinnisõidetuteks. Lisaks katsetati mudeli parandamist suurema treeningandmestiku põhjal ning ka mudeli enda optimeerimist, mille tulemuseks tõsteti ennustustäpsus 83%-lt 85%-le.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 21 leheküljel, 5 peatükki, 13 joonist, 9 tabelit.

Abstract

Automation of Machine Learning Process of Intersection Congestion Reduction System

The main goal of this thesis is to automatize the collection of training data for an existing convolutional neural network model, which can predict intersection blockages real-time from camera stream. Such a model can currently predict an intersection congestion ten seconds into the future with an 83% accuracy. Upon detecting a possible blockage, the model sends a signal to the traffic light, ordering it to switch on the red light. By reducing the amount of cars allowed through the intersection, the goal of this model is to mitigate traffic blockages, and by extension, traffic jams.

Tallinn City Administration has given access to two cameras at Sõpruse-Tammsaare intersection in Mustamäe. These cameras are used to collect training data for the model and the aim is to automatize the process of slicing the video into images and classifying them appropriately. As image classification needs to be done automatically as well, a separate model needs to be trained to detect if an intersection is congested or not.

After collecting more data than what previous models were trained with, the model was trained with 883 images, resulting in no improvement in accuracy. As increasing the size of training dataset alone was not enough, the model itself was optimized as much as possible. Upon testing various loss functions and different hyperparameter combinations, the final accuracy of the model was increased to 85%.

Unfortunately the system could not be tested on a real intersection since at the period of writing this thesis, the amount of traffic blockages was minimal due to the season and Covid-19.

The thesis is in Estonian and contains 21 pages of text, 5 chapters, 13 figures, 9 tables.

Lühendite ja mõistete sõnastik

Fooritsükkel	Tsükkel algab rohelise tulega ning lõppeb punase tule sisse lülitudes
HSV	<i>Hue – saturation - value</i>
Kaofunktsioon	Näitab erinevust soovitud väljundi ja tegeliku väljundi vahel (<i>loss function</i>)
Kinnisõidetud ristmik	Ristmik loetakse kinnisõidetuks, kui vähemalt pool auto kerest jääb ülekäigurajale
Pooljuhendatud õppimine	Masinõppe meetod, kus vähese hulga andmete põhjal õpetatakse mudelit klassifitseerima suuremat andmekogumit
RGB	Punane – roheline – sinine (<i>Red-green-blue</i>)
RTSP	<i>Real Time Streaming Protocol</i>
TalTech	Tallinna Tehnikaülikool
Testandmestik	Andmestik, mille peal valideeritakse mudeli ennustusi
Treeningandmestik	Andmestik, mida kasutatakse mudeli treenimiseks

Sisukord

1 Sissejuhatus	9
2 Olemasoleva mudeli analüüs	11
2.1 Andmete kogumine.....	11
2.2 Olemasolev mudel	12
3 Sisendandmestiku kogumise automatiseerimine	14
3.1 Abimudel	14
3.2 Abimudeli treeningandmete kogumine	15
3.3 Abimudeli treenimine	15
3.4 Video sõelumise automatiseerimine	18
3.4.1 Fooritule tuvastamine	18
3.4.2 Kaadri löikehetke leidmine.....	21
4 Validatsioon.....	24
4.1 Tulemused	24
4.2 Ennustusmudeli optimeerimine	26
4.3 Ennustusmudeli rakendamine ristmikule.....	28
4.4 Edasiarenduse võimalused.....	28
5 Kokkuvõte	30
Kasutatud kirjandus	31
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	33

Jooniste loetelu

Joonis 1. Ristmik ja kaamerad. Hallid täpid on kaamerad ning sinine ja must väli näitavad kaamerate vaatenurka. Punane joon näitab vaatluse all olevat sõidusuunda, mida proovitakse optimeerida (Sander Aasaväli joonistus [2], aluskaart: Google Maps)	12
Joonis 2. Kaamerapilt vaadeldavast ristmikust suunaga Järvevana tee poole, illustreerimaks liikluspiltide erinemist. Vasakpoolne pilt on võetud Epp Muringi tööst [1].	13
Joonis 3. Abimudeli ülesehitus	14
Joonis 4. Kaader koos punasega märgitud huvialaga, mis lõigatakse välja, et eemaldada liigne müra mudeli treenimisel	16
Joonis 5. Tavavalguses võib roheline tuli välja paista rohkem helesinisena illustreerimaks tavavalguses erinevaid toone	18
Joonis 6. Ainult üks mudel suutis pildil olevat valgusfoori ära tunda [14]	19
Joonis 7. Näide: HSV vahemike leidmise abiprogrammi kasutamine	20
Joonis 8. Mudel tuvastab tahtmata ka vasakpoolse valgusfoori tuled	20
Joonis 9. Vasakul pildil on kaader kaamerast, mis on suunatud TalTechi poole. Paremal pildil on mudel leidnud otsitava kollase värvi foorilt	21
Joonis 10. Võetud pildipaar: (a): Punase fooritule algul võetud kaader, mis läheb abimudelisse klassifitseeringuks, (b) 10 sekundit varem võetud pilt, mis, koos abimudelilt tulnud annotatsiooniga, läheb ennustusmudeli treeningandmestikku	22
Joonis 11. Ülevaade sisendandmete kogumise töökäigust	23
Joonis 12. Ennustusmudeli visualisatsioon (Autor: S. Aasaväli) [2].	26
Joonis 13. Lõpliku mudeli visualisatsioon	28

Tabelite loetelu

Tabel 1. Treening- ja testandmete jaotus andmestikus.....	16
Tabel 2. Abimudeli kolme värvikanaliste kaadritega treenimisel kasutatud parimad hüperparameetrid	17
Tabel 3. Abimudeli treeningu tulemused kolme värvikanaliga kaadritest koosneva treeningandmestikuga.....	17
Tabel 4. Abimudeli treeningu tulemused ühe värvikanaliga kaadritest koosneva treeningandmestikuga.....	17
Tabel 5. Ennustusmudeli treenimise hüperparameetrid	24
Tabel 6. Ennustusmudeli lõpliku sisendandmestiku jaotus	25
Tabel 7. Ennustusmudeli ennustused grupeeritult	25
Tabel 8. Ennustusmudeli tulemused väiksema treeningandmestikuga grupeeritult.....	25
Tabel 9. Lõpliku mudeli treenimise hüperparameetrid	27

1 Sissejuhatus

Ristmikute kinnisõitmine on mõnedes kohtades, nagu Mustamäel asuv A. H. Tammsaare tee ning Sõpruse puiestee ristmik, üsnagi levinud nähtus, mis tuleneb olukorrast, kus roheline tuli lubab rohkematel autodel ristmikule tulla kui neid sealt ära jõuab sõita. Fooritsükli lõppedes ristmikule jäänud autod blokeerivad omakorda teisi sõidusuundi ning ka jalakäijate ülekäiguradasid, mille tulemuseks ei saa liikuda ka teistes sõidusuundades ja võivad tekkida suured ummikud.

Üks lahendus taoliste olukordade vältimiseks või vähendamiseks on teha masinõppe abil konvolutsioonilise närvivõrgu mudel, mis suudab ennustada ette, kas ristmik sõidetakse kinni ja selle tulemusena teha otsus, kas lülitada punane tuli varasemalt sisse, mis ei lubaks üleliigsetel autodel sõita ristmikule. Selle tulemusena väheneks ristmike kinnisõitmisest arv, mis säästaks liiklejate aega ning loodust heitgaaside vähendamisega.

Taoline mudel on juba varasemalt tehtud Epp Muringi poolt oma lõputöö raames, mis valmis kasutades Pythoni masinõppele pühendatud teeki Tensorflow. Epp Muringi lõpptulemuseks tuli mudel, mis suutis ennustada 80% täpsusega 4 sekundit ette, kas ristmik sõidetakse kinni või mitte [1]. Sander Aasaväli täiustas oma lõputöö raames eelmainitud mudelit, viies mudeli TensorFlow baasilt üle teisele masinõppe teegile: PyTorchile. Samas tõstes ennustustäpsuse 83%-le ja ennustusaja 10-le sekundile [2].

Selle lõputöö eesmärgiks on, sarnaselt Sander Aasaväli lõputööle, olemasolevat mudelit parandada, kuid teisest küljest, keskendudes rohkem sisendandmestiku ja selle töötlemise automatiseerimisele. Üheks probleemiks praeguse mudeliga on see, et mudel ei saa kusagilt uusi treeningandmeid ning, kuna liikluspilt muutub pidevalt, siis mudel ei käi nende muutustega kaasas ja ajapikku hakkab ta täpsus vähenema. Selle ennetamiseks treenitakse taolised mudelid üle värskemate andmetega [3]. Selle töö raames ongi üheks eesmärgiks võimaldada rohkemate treeningandmete kogumine automaatselt, mis säästaks tundideviisi käsitsi tehtud tööd andmete annoteerimisel. Ei ole välistatud ka mudeli ehituse muutmist, kui paistab, et mudeli efektiivsus ei ole sisendandmete taga kinni.

Planeeritud on kasutada ja treenida üht lihtsamat abimudelit, mis suudaks automaatselt klassifitseerida etteantud kaadrites ristmikud kinnisõidetuks või mitte kinnisõidetuks. Selle abil on võimalik suurel hulgal andmeid koguda, mis käsitsi tehes muutuks liiga ajamahukaks.

Sander Aasaväli lõputööga seoses ei jõutud paigaldada uuendatud mudelit päris ristmikule, nagu oli plaanis [2]. Seega on üheks eesmärgiks lõpus realiseerida see plaan ning võrrelda kinnisõitmiste arve enne ja pärast mudeli paigaldamist. Võimalus on ka riistvara uuendamine Raspberry Pi-lt Jetson Nanole, mille tööjõudlus on suurem ning on rohkem suunatud närvivõrkudele ja sügavõppele [4].

2 Olemasoleva mudeli analüüs

Peatükis analüüsitakse praegust mudelit, andmetekogumist ja lahendusi nende parandamiseks.

2.1 Andmete kogumine

Võttes arvesse vajamineva treeningandmete hulga, siis selle käsitsi tegemine võib kujuneda ajarikkaks ja vaevamahuliseks ettevõtmiseks. Kuigi Sander Aasaväli tegi oma lõputöö raames abiprogrammi `video_slicer.py`, mis lõikas videoklippidest vajalikud kaadrid välja vastavalt klipi nimes olevale ajale, siis kõigi nende kaadrite ja videoklippide annoteerimine on ikkagi ajakulukas töö [2].

Sander Aasaväli ja Epp Muring kasutasid oma töös mudeli treenimiseks umbes 100-300 pilti [1], [2]. Selle töö eesmärgiks oleks andmete kogumise automatiseerimisega see arv mitmekordistada. Selleks tuleb kasutusele võtta pooljuhendatud õppimine, mille mõte on treenida mudel, mis hakkab ise treeningandmeid tegema [5]. Selle töö kontekstis oleks vaja treenida mudel, mis suudab kaadri põhjal aru saada, kas ristmik on kinnisõidetud või mitte ja vastavalt koostada sobiv treeningandme sisend ennustusmodelile.

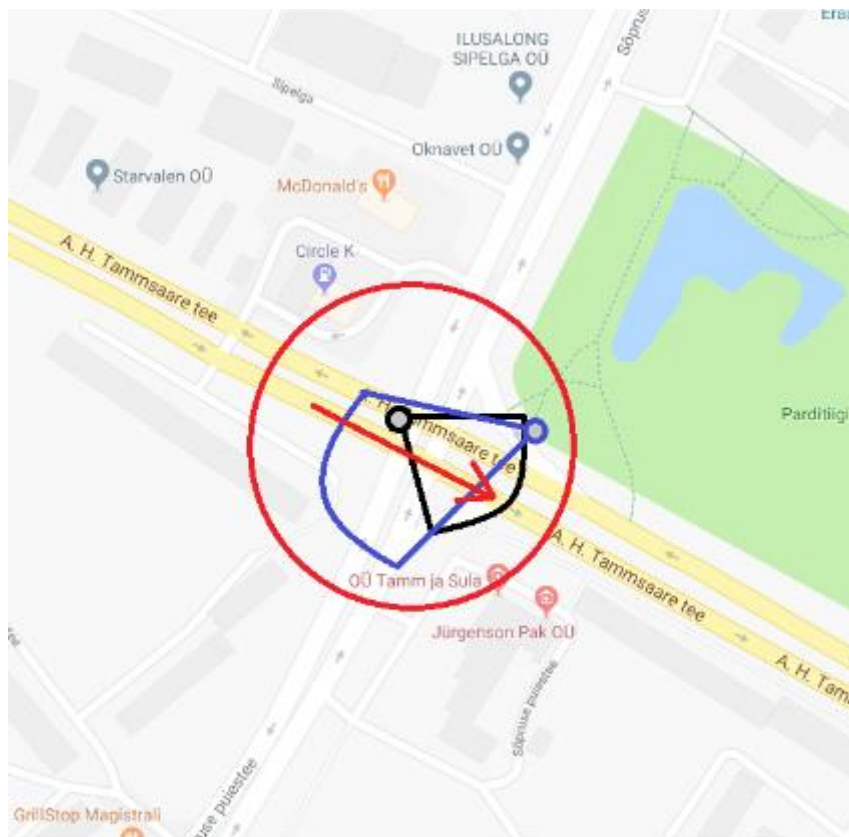
Selle töö raames võetakse sarnaselt eelmistele lõputöödele vaatluse alla Mustamäel asuva A. H. Tammsaare tee ning Sõpruse puiestee ristmik kahel põhjusel:

1. Sama ristmiku peal testimisega saab kõige paremini võrrelda, kui palju mudel on paranenud.
2. Meil on ligipääs kahele ristmikul olevale kaamerale Tallinna Transpordiameti loal, mis võimaldavad kergelt andmeid koguda.

Töö raames keskendutakse joonisel 1 välja toodud A. H. Tammsaare teele suunaga Järvevana tee poole, kus juhtub kõige rohkem kinnisõitmisi [2].

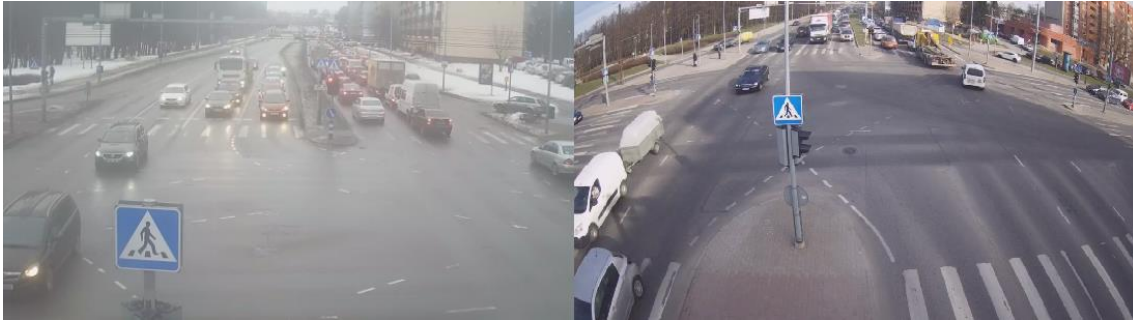
2.2 Olemasolev mudel

Kuna liikluspilt on pidevas muutumises, olgu selleks ilmastik, valgus, erinevad autod, kaamera positsioon või ka ristmik ise, siis mingil hetkel hakkab vananenud andmete tõttu mudeli efektiivsus langema [3]. Selle ärahoidmiseks tuleb koguda uusi andmeid, mille peal saab mudelit üle treenida. Lisaks on tõenäoline, et uute ja suurema mahu



Joonis 1. Ristmik ja kaamerad. Hallid täpid on kaamerad ning sinine ja must väli näitavad kaamerate vaatenurka. Punane joon näitab vaatluse all olevat sõidusuunda, mida proovitakse optimeerida (Sander Aasaväli joonistus [2], aluskaart: Google Maps)

sisendandmetega treenimisel tõuseb mudeli efektiivsus, kuid selleks on alguses vaja koguda suurel hulgal andmeid, mis käsitsi võib võtta väga palju aega.



Joonis 2. Kaamerapilt vaadeldavast ristmikust suunaga Järvevana tee poole, illustreerimaks liikuspiltide erinemist. Vasakpoolne pilt on võetud Epp Muringi tööst [1].

Et hinnata, kui palju on olemasoleva mudeli ennustustäpsus vähenenud, tuleks rakendada mainitud mudelit uute kogutud andmete peal. Andmete kogumisel tuli välja, et arvatavasti COVID pandeemia tõttu oli liikluskoormus vähenenud [6] ning ristmiku kinnisõitmist juhtus vaid mõni üksik kord (täpsem selgitus on välja toodud peatükis 4). Kuna praeguse seisuga pole piisavalt kaadreid kinnisõidetutest ristmikutest ja mitte kinnisõidetutel ristmikel on suur ülekaal, siis mudeli täpsust ei saa veel korrektselt hinnata. Sellegipoolest võime proovida, kui hästi mudel praeguseid mitte kinnisõidetud ristmikke ennustab.

Selleks on Sander Aasaväli valmis teinud `video_slicer.py` ja `inference_tester.py` abiprogrammid, millest esimene võtab käsitsi annoteeritud videod kuhu on kirja pandud, millisel sekundil süttib esimene punane tuli ja seejärel lõikab välja videoklipi 90 sekundiliste intervallidega, kui on annoteeritud, et tegu on tipptunniga, vastasel juhul lõigatakse videoklipp iga 72 sekundi tagant [2]. Klipid salvestatakse isemääratud kohta ning klipidel endil tuleb omakorda ära näidata nimes, millal on punase tule algus ning kas tegu on kinnisõidetud ristmikuga või mitte [2].

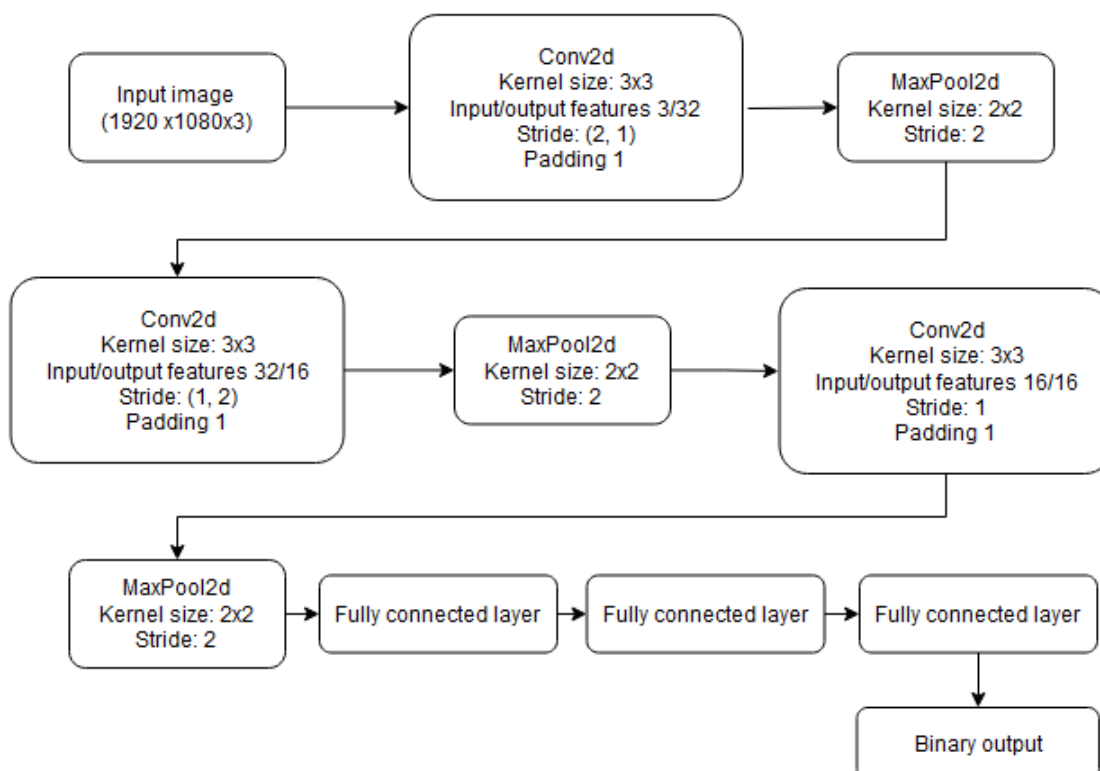
Pärast klippide annoteerimist on võimalik `inference_tester.py` programmiga mudelit testida. Nagu eelnevalt mainitud, koosneb autori poolt 2021. aastal loodud andmestik peaaegu tervenisti mitte kinnisõidetud ristmike näidetest. Täpsemalt jaotub testandmestik 30-ks mitte kinnisõidetud ristmike kaadriteks ja üheks kinnisõidetud kaadriks. Nende andmetega testimisel saadi tulemuseks 87% ennustustäpsus, kus väärraid positiivseid tulemusi oli kolm ning väärraid negatiivseid tulemusi oli üks.

3 Sisendandmestiku kogumise automatiseerimine

Käesolevas peaktükis analüüsitakse ja kirjeldatakse sisendandmete kogumise automatiseerimist. Kogu lahendus on kirjutatud programmeerimiskeeles Python [7], võttes abiks videotötluse teegi OpenCV [8].

3.1 Abimudel

Eemaldamiseks piltide käsitsi annotseerimise vajadust, oli kõige mõistlikum lahendus teha abimudel, mis teeks seda automaatselt. Kuna sisendandmete õigesti klassifitseerimine on väga tähtis, siis abimudeli täpsuse alampiiriks võeti ~95%. Vastasel juhul võib ennustusmudeli treeningandmetesse sattuda liiga palju valesid andmeid ning andmete hulga suurendamisel hakkab ennustuste efektiivsus langema, vastupidiselt meie eesmärgile. Joonisel 3 on välja toodud abimudeli ülesehitus, mis sarnaneb ennustusmudelile, kuigi vähemate kihtidega.



Joonis 3. Abimudeli ülesehitus

3.2 Abimudeli treeningandmete kogumine

Andmestiku koostamiseks oli vaja saada video ristmikul olevast liiklusest. Selleks oli võimalus kasutada varasematel töödel kasutatud ristmikukaameraid Tallinna Linnavalitsuse loal. Sander Aasaväli tegi oma lõputööga seoses skripti, mis võimaldab läbi *Real Time Streaming Protocol* (RTSP) alla laadida tunnipikkuse video TalTechi serverisse [2]. Lisaks oli automatiseeritud *crontabi* abiga iga päev tiptunni ajal Järvevana tee poole suunatud kaamera pildi alla laadimine.

Abimudelile treeningandmestik tuli koostada käsitsi. Tähtis on, et andmeid kogutakse nii erinevatest olukordadest kui võimalik (valgus, ilm), et vältida mudeli ülesobitumist [9]. Sellepärast võeti kaader igast fooritsüklist ja klassifitseeriti vastavalt Epp Muringi reeglitele ehk ristmik loetakse kinnisõidetuks, kui autokere on vähemalt poolenisti üle ülekäiguraja [1].

3.3 Abimudeli treenimine

Kõige esimene treenimiskatse oli 40 kaadrisuuruse treeningandmestikuga ja 10 kaadrisuuruse testandmestikuga. Taoline abimudel ei suutnud ühtegi testpilti tuvastada. Kõige tõenäolisemalt oli probleem liigses müras, mis täissuuruses pildis kaasas oli. Kuna optimeerida oli vaja vaid ühte suunda, siis oli võimalus lõigata välja ainult vajalik osa pildist, mille peal treenida, nagu joonisel 4 on välja toodud. Väljalõikamiseks kasutati PyTorch'i `torchvision.transform` teegi `Lambda` meetodit, kuhu sai parameetriks anda funktsiooni, mis lõikab etteantud pildist otsitud osa välja [10].



Joonis 4. Kaader koos punasega märgitud huvialaga, mis lõigatakse välja, et eemaldada liigne müra mudeli treenimisel

Lõplikus treeningandmestikus on 749 kaadrit ja testandmestikus 118 kaadrit, mille jaotus on ka tabelis 1 välja toodud.

Tabel 1. Treening- ja testandmete jaotus andmestikus

	Kinnisõidetud	Mitte kinnisõidetud
Treeningandmed	364	385
Testandmed	44	74
Kokku	408	459

Abimudeli treenimise käigus testiti erinevaid hüperparameetrite kombinatsioone, mis muutsid märgatavalt klassifitseerimise täpsust. Parim kolme värvikanaliga kaadrite treenimisel saadud mudeli hüperparameetrid ja tulemused on toodud välja vastavalt tabelis 2 ja tabelis 3.

Tabel 2. Abimudeli kolme värvikanaliste kaadritega treenimisel kasutatud parimad hüperparameetrid

Hüperparameeter	Väärtus
Õppimismäär	0.0001
Iteratsioonide arv	20
Minipartii suurus	10

Tabel 3. Abimudeli treeningu tulemused kolme värvikanaliga kaadritest koosneva treeningandmestikuga

	Tõesed positiivsed ennustused	Väärad positiivsed ennustused	Tõesed negatiivsed ennustused	Väärad negatiivsed ennustused
Testandmed	41	3	62	8
Täpsus (%)	91			

Kuna treenimisel saadud 91% täpsusega abimudel jääb liiga palju alla soovitud 95% täpsusest, prooviti ka treeningandmed teisendada üle halltoonidele. Oletatavasti sisaldab kolme värvikanaliga kaader üleliigset infot ning müra, mida klassifitseerimisel vaja ei lähe. Teisendamiseks kasutati PyTorch'i torchvision.transforms teegi Grayscale meetodit [10].

Halltoonideks teisendatud andmestiku peal treenides samade hüperparameetritega nagu tabelis 2, vahetusid väärade positiivsete ja väärade negatiivsete ennustuste osakaalud, mis on välja toodud tabelis 4. Tunduvasti mängib värvikanalite arv suurt rolli klassifitseerimisel, kuid paremat tulemust see lõpuks ei andnud.

Tabel 4. Abimudeli treeningu tulemused ühe värvikanaliga kaadritest koosneva treeningandmestikuga

	Tõesed positiivsed ennustused	Väärad positiivsed ennustused	Tõesed negatiivsed ennustused	Väärad negatiivsed ennustused
Testandmed	30	14	72	2
Täpsus (%)	86.5			

Kahjuks algselt piiriks seatud 95%-st täpsust abimudel ei saavutanud, kuid 91% oli siiski piisavalt lähedal soovitud tulemusele. Kuna enamus valesid ennustusi olid väärad negatiivsed ehk ei tabatud ära, et ristmik oli kinnisõidetud, siis rohkem kinnisõitmisi ei

tohiks see tekitada ning õigete andmete ülekaal suurte andmemahtude korral peaks piisavalt korrigeerima lõplikku ennustusmodeli täpsust.

3.4 Video sõelumise automatiseerimine

Et videost automaatselt õiget kaadrit saada, on vaja teada, millal õige hetk löikamiseks on. Varasemalt kasutasid Sander Aasaväli ja Epp Muring abiprogramme, mis lugesid failinimest õige hetke, millal võtta pilt videost ja sealt edasi võtta iga kindla vahemiku tagant uuesti pilt [1], [2]. Samas tuli õige hetk üles otsida ja manuaalselt failinimesse panna. Selle probleem on samalaadne käsitsi annoteerimisega, sest suurte andmemahtude puhul võtab taoline tegevus tohutult aega ning, kui fooritsükkel nihkub tõrke pärast edasi, siis välja lõigatud pildid võivad olla teisest fooritsükli osast kui vaja.

Videost õige löikamishetke leidmine koos kontrolliga realiseeriti OpenCV abil [8], kus pildi pealt tuvastatakse kindel foorituli, mille pealt leitakse pildi löikehetk ja samaaegselt kontrollitakse, kas löikehetk on õiges fooritsükli staadiumis.

3.4.1 Fooritule tuvastamine

Klassikaline meetod pildilt värvi tuvastamiseks hõlmab pildi teisendamist RGB-st HSV skaalale. HSV skaala on üldlevinud masinnägemise projektides, sest see lubab kergemni määrata värvide intensiivsuste vahemikku, mis on kõige tähtsam taolistes muutuva valgustusega pildides [11].



Joonis 5. Tavavalguses võib roheline tuli välja paista rohkem helesinisena illustreerimaks tavavalguses erinevaid toone

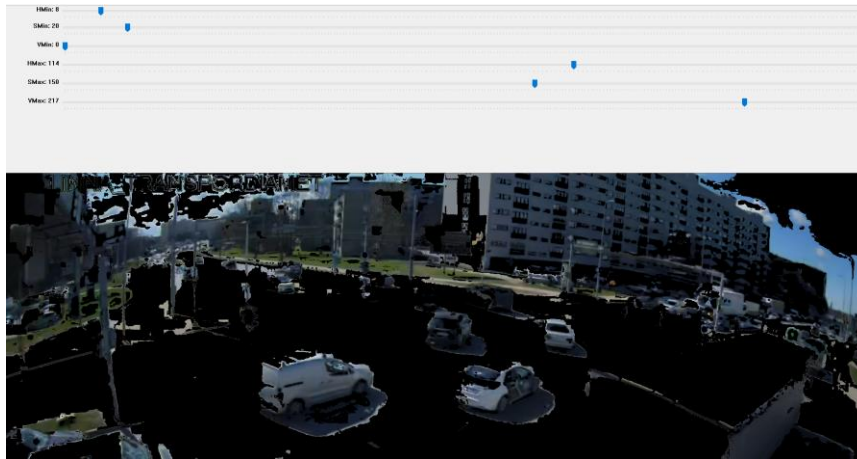
Üks lahendus oleks kaamerast saadud pildi pealt leida fooritulede koordinaadid ning tuvastada, kas tuli põleb või mitte. Sellise lähenemise põhiliseks probleemiks on, kui kaamera nihkub hooldustööde käigus või tormi ajal või sisendpildi suurus muutub, mistõttu tuleks endiseid koordinaate pidevalt korrigeerida. Selle asemel oleks võimalik teha üldisem lahendus, mis leiaks valgusfoori üles pildi pealt.

Töö kergemaks tegemiseks prooviti leida juba valmis treenitud mudel, mis suudaks tuvastada videotel olevaid valgusfoore [12], [13], [14]. Kuna allalaaditav video on suuruses 1280x720 ja kvaliteet ei ole väga hea, siis otsitavad tuled olid vaid mõne piksli suurused, mis tegid ülesande raskemaks. Ainult üks mudel suutis antud kaamerapildilt fooritulesid leida [14].

Sellele mudelile on võimalus anda ette HSV väärtuste vahemik, mida on tarvis otsida ja sellega on võimalik igat värvi fooril ära tunda. Aitamaks õigeid HSV väärtuste vahemikke otsida, kasutati abiprogrammi [15], mis võimaldas leida, kuhu vahemikku otsitava objekti HSV väärtused jäävad.



Joonis 6. Ainult üks mudel suutis pildil olevat valgusfoori ära tunda [14]



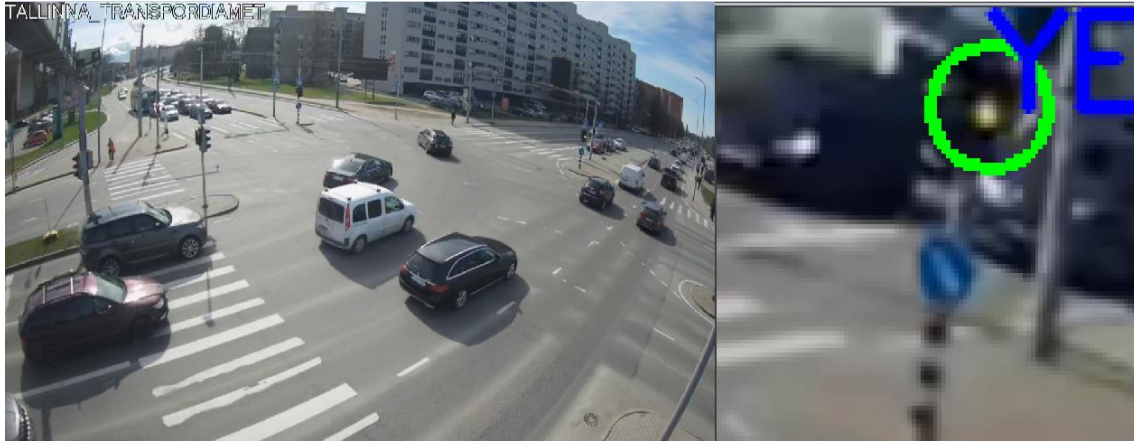
Joonis 7. Näide: HSV vahemike leidmise abiprogrammi kasutamine

Lisaks otsitud foorile, tuvastas mudel tahtmatult ka kõrvaloleva foori tulesid, nagu on näha joonisel 8. Ainult paremapoolse foori pilti jätmise ei toonud ka oodatavat tulemust, sest mudel ei osanud pildi äärel olevat foori ära tunda.



Joonis 8. Mudel tuvastab tahtmata ka vasakpoolse valgusfoori tuled

Kuna Järvevana tee poole vaatavast kaamerast ei suudetud ühtegi teist foori tuvastada, siis järelejäänud lahendus oli kasutada teist, TalTechi poole vaatavat kaamerat. Sellest kaamerast tulnud videost, suutis mudel ühe üksiku foori ära tunda, mis oli piisav.



Joonis 9. Vasakul pildil on kaader kaamerast, mis on suunatud TalTechi poole. Paremalt pildil on mudel leidnud otsitava kollase värvi foorilt.

3.4.2 Kaadri lõikehetke leidmine

Leitud fooritulega oli võimalik kaudselt leida algselt otsitud pildi lõikehetk, mõõtes välja nendel kahel kaameral oleva fooritsüklite suhte ja selgus, et Järvevana tee poole näitava kaamera foor läheb punaseks 15 sekundit enne tuvastatavat foori TalTechi suunas.

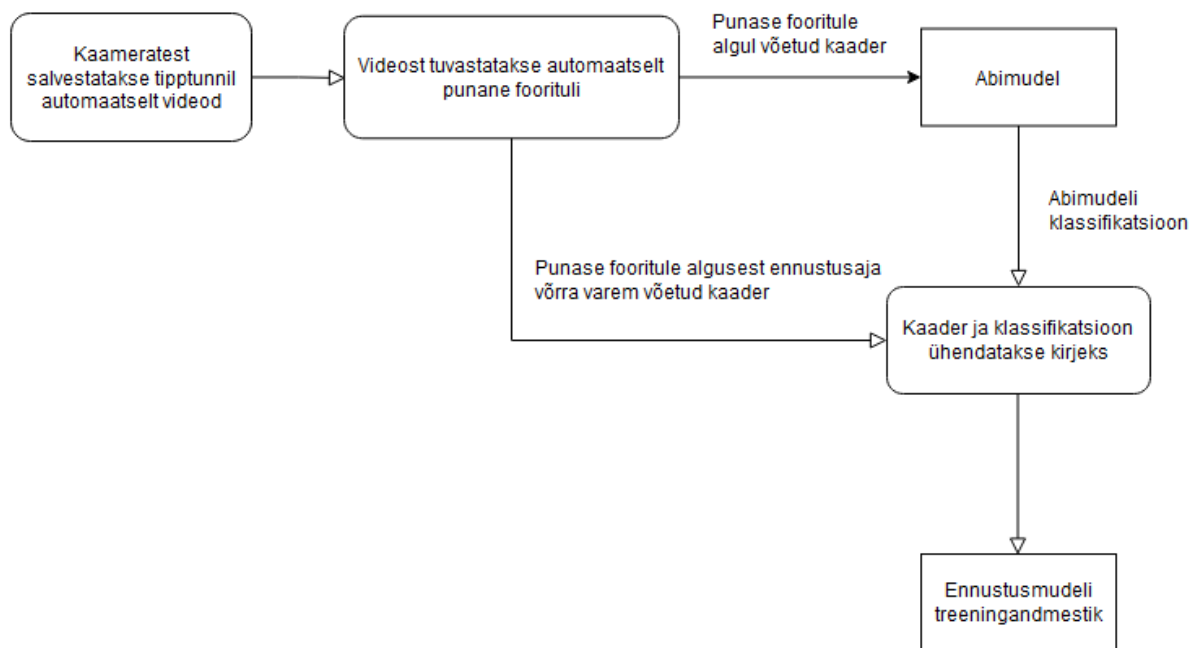
Sealt edasi oli vaja tuvastada TalTechi suunas vaatavast kaamerast punane tuli, mille järel oleks võimalik teisest videost välja lõigata samast hetkest 15 ja 25 sekundit varem olevad pildid, millest esimene läheks edasi abimudelile klassifitseerimiseks ning viimane jõuaks, koos abimudeli klassifitseeringuga, ennustusmudeli treeningandmestikku. Pärast esimest lõikehetke saaks käia Järvevana teele vaatava video läbi eeldefineeritud fooritsükli pikkuste intervallide kaupa, sarnaselt varasematele lahendustele [1], [2].

Kontrollimaks, et punane tuli on tuvastatud kohe tsükli alguses, tuli lisaks teha kontroll, et paar sekundit tagasi oli võimalik tuvastada ka kollast tuld. Selleks oli vaja muuta Sander Aasaväli tehtud programmi `video_parser.py`, mis luges failinimest, millal esimene punane tuli on [2]. Uus lahendus alustas video läbikäimist algusest, kontrollides iga kaader, kas kollane foorivärv oli leitav. Kuna foorivärvid võivad tavavalguses iga sekundi tagant muutuda, oli tähtis, et ei hüppata kogemata üle kollase tule, sest valgustus muutis fooritule mudelile nähtamatuks. Selleks tuli video käia kaaderhaaval läbi, et vältida kollasest tulest üle hüppamist. Kui kollane foorituli oli leitud, otsiti järgneval 3-4 sekundil punast foorituld.

Punase fooritule leidmisega oli pildi lõikehetk käes ning seejärel võeti kaks pilti: üks lõikehetkel ja teine 10 sekundit varem. Hilisem pilt annoteeriti abimudeli poolt ära ning annotatsioon lisati 10 sekundit varem tehtud pildi nimesse, mis andis sobiva treeningisendi ennustumudeli jaoks.



Joonis 10. Võetud pildipaar: (a): Punase fooritule algul võetud kaader, mis läheb abimudelisse klassifitseeringuks, (b) 10 sekundit varem võetud pilt, mis, koos abimudelilt tulnud annotatsiooniga, läheb ennustumudeli treeningandmestikku



Joonis 11. Ülevaade sisendandmete kogumise töökäigust

4 Validatsioon

Sander Aasaväli näitas oma töö käigus, et üksikute piltide peal treenimine on sama tõhus kui mõlemast kaamerast vertikaalselt kokku pandud pilt samast ristmikust [2]. Seepärast otsustati lihtsuse mõttes edasi treenida üksikute piltide peal.

Andmete kogumise käigus märtsis ja aprillis leidis vaid üks ristmiku kinnisõitmine, mistõttu ei suudetud koguda sobilikke andmeid mudeli ületreenimiseks. Arvatavasti põhjustas liikluse vähesust Covid-19 ning kevadine aastaag [6], mistõttu ei olnud võimalik koguda näiteid kinnisõitmistest.

Seepärast otsustati üle minna vanale, 2019. aastal salvestatud videotele, mõttega treenida mudel suurema koguse treeningandmestikuga ja võrrelda, kas puhtalt andmestiku mahu kasvuga paraneb mudeli ennustamise efektiivsus. Kui varasemad mudelid olid treenitud umbes 200-300 arvu ringis treeningandmetega, siis eesmärgiks oli see arv mitmekordistada [1], [2].

4.1 Tulemused

Üheks peamiseks eesmärgiks oli ennustusmudeli täpsuse parandamine treeningandmete mahu suurendamisega. Tabelis 5 on välja toodud hüperparameetreid, millega Sander Aasaväli oma mudelit treenis, ning mida kasutatakse algselt määramaks, kas lihtsalt andmetemahu suurendamine tõstab ennustustäpsust [2].

Tabel 5. Ennustusmudeli treenimise hüperparameetrid

Hüperparameeter	Väärtus
Õppimismäär	0.001
Iteratsioonide arv	100
Minipartii suurus	20
Lambda	0.0001

Valideerimiseks koguti 2019. aasta augustikuu tiptundide ajal saadud videotest kaadreid mille mahuks oli lõpuks 883 sisendit, mille jaotumine treening- ja testandmeteks on välja toodud tabelis 6. Eraldamiseks testandmeid treeningandmetest, võeti testandmed kolmest

viimasest augustikuu päevadest ning treeningandmetesse jäid ülejäänud päevadest võetud kaadrid.

Tabel 6. Ennustusmodeli lõpliku sisendandmestiku jaotus

	Treeningandmed	Testandmed
Kinnisõidetud	361	64
Mitte kinnisõidetud	388	70

Treeningu tulemusena, kasutades tabelis 5 toodud hüperparameetreid, saavutati tulemuseks 81% ennustustäpsus testandmete peal ja 99% ennustustäpsus treeningandmete peal. Tabelis 7 on näha, et enamik valesid ennustusi olid väärad positiivsed, mis võib viidata mudeli ülesobitumisele ning probleemile treeningandmetes [9].

Tabel 7. Ennustusmodeli ennustused grupeeritult

	Tõesed positiivsed ennustused	Väärad positiivsed ennustused	Tõesed negatiivsed ennustused	Väärad negatiivsed ennustused
Treeningandmed	357	1	387	4
Testandmed	59	21	49	5

Üheks võimaluseks treeningandmeid muuta on võtta pooled kaadrid igast päevast ära, vältimaks võimalust, et andmetes on liiga palju duplikaate samadest liikluspiltidest. Sellise meetodiga treeningandmeid muutes paranes ennustusmodeli täpsus 81%-lt 83%-le ehk üheks parandamisviisiks võib olla vähemate kaadrite kogumine päeva jooksul, kuid koguda neid üle rohkemate päevade.

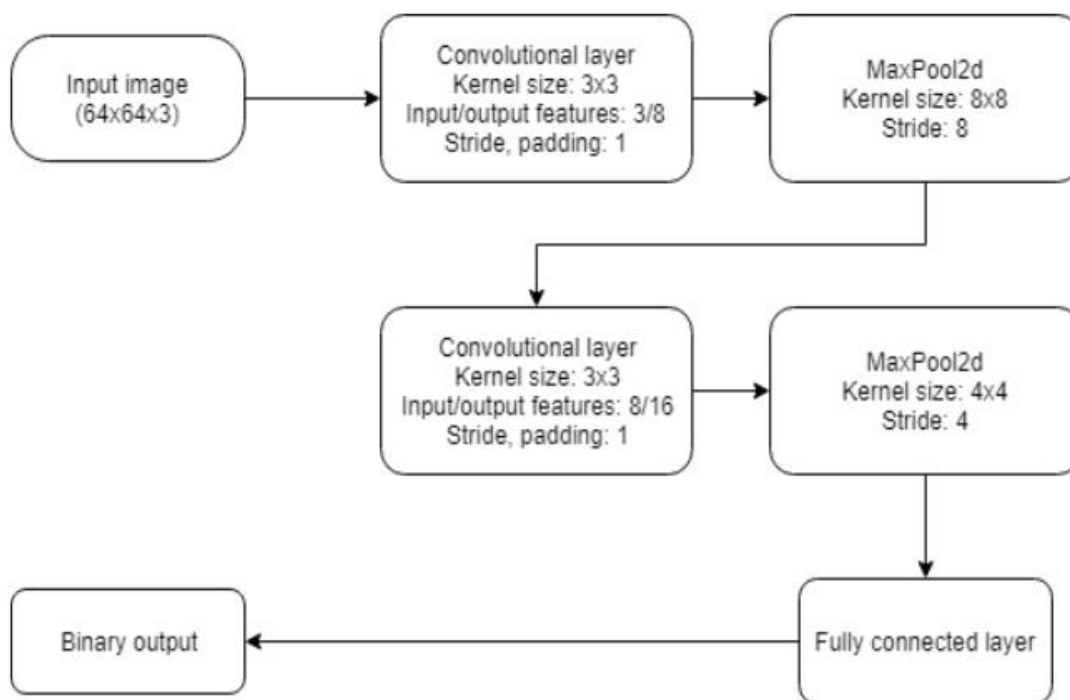
Tabel 8. Ennustusmodeli tulemused väiksema treeningandmestikuga grupeeritult

	Tõesed positiivsed ennustused	Väärad positiivsed ennustused	Tõesed negatiivsed ennustused	Väärad negatiivsed ennustused
Treeningandmed	360	2	386	1
Testandmed	61	19	51	3

Siiski on 83% ennustustäpsus sama, mis Sander Aasavälja poolt saadud tulemus ehk puhtalt andmetehulga neljakordse suurendamisega mudeli efektiivsus ei läinud paremaks [2]. Järgmine variant oleks mudeli ehitust ning parameetreid ise optimeerida, et soovitud tulemus saavutada.

4.2 Ennustusmudeli optimeerimine

Üks lihtsamaid viise närvivõrgu mudelit parandada on kaofunktsiooni muutmise. Joonisel 12 on välja toodud ennustusmudeli skeem, mis on võetud Sander Aasavälja lõputööst [2].



Joonis 12. Ennustusmudeli visualisatsioon (Autor: S. Aasaväli) [2].

Sellele mudeli lõpus on ühendatud kiht, mille lõppu on võimalik rakendada kaofunktsioon, mis ideaalis tõstaks ennustustäpsust.

Sigmoidi ja Softmaxi kaofunktsioonide kordamööda lisamine mudelisse ei andnud soovitud tulemusi. Testandmete täpsus jäi ikkagi vahemikku 80%-83% ning parema efektiivsuse jaoks tuleks ennustusmudelit ikkagi tunduvalt rohkem muuta.

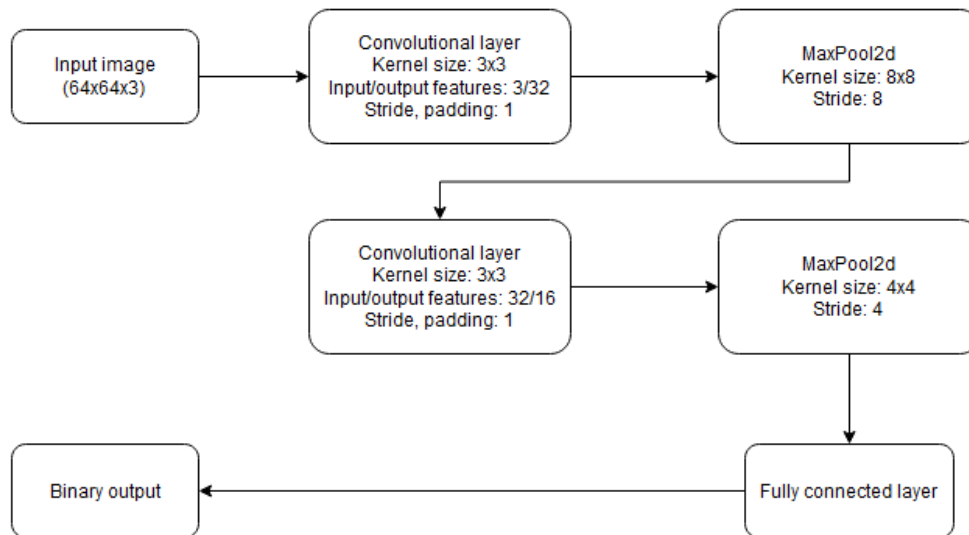
Veel üks asi, mida mudeli optimeerimisel saaks proovida enne suuremat ülesehituse muutmist, oleks konvolutsiooniliste kihtide sisendite ja väljundite arvude muutmine. Muutes esimese kihi väljundite arvu 8-lt 32-le ning sarnaselt teise kihi sisendite arvu 8-lt 32-le, tõusis mudeli ennustustäpsus 85%-le.

Sisendite ja väljundite arvu suurendamisel mudeli efektiivsuse tõusmine viitaks asjaolule, et suurema andmemahu puhul paraneks ennustustäpsus. Seepärast võib ka sügavama mudeli tegemine või teise kaamerapildi lisamine anda soovitud tulemust, mida tulevikus võiks mudeli edasiarendusel proovida.

Kuigi 85% jääb alla algselt eesmärgiks seatud 90%-le, ei tundunud olevat võimalik selle arhitektuuriga mudeli ennustustäpsust 90%-ni tõsta. Selleks oleks vaja ehitada mudel ümber või leida mõni teine võimalus. Hüperparameetrid, millega lõplikku mudelit treeniti ning mudel ise on näha vastavalt tabelis 9 ja joonisel 13.

Tabel 9. Lõpliku mudeli treenimise hüperparameetrid

Hüperparameeter	Väärtus
Õppimismäär	0.0001
Iteratsioonide arv	150
Minipartii suurus	5
Lambda	0.0001



Joonis 13. Lõpliku mudeli visualisatsioon

4.3 Ennustusmudeli rakendamine ristmikule

Algne plaan oli lõputöö raames paigaldada uuendatud mudel ka päriselt ristmikule ning kontrollida selle efektiivsust, kuid, nagu enne mainitud, leiti lõputöö käigus ainult üks kinnisõitmine vaadeldaval ristmikul.

Võttes arvesse, et mudeli paigaldamine on suurem ettevõtmine koos Tallinna Transpordiametiga, siis hinnanguliselt ei olnud lõputöö tegemise ajal mõtet seda teha. Ainus kasu oleks olnud mõõta vääraste positiivsete ennustuste arvu, kuid otstarbekam oleks paigaldusega oodata kuni liiklus tiheneb ja kinnisõitmised hakkavad ka realselt tekkima.

4.4 Edasiarenduse võimalused

Kuigi andmete kogumise automatiseerimine valmis edukalt, ei olnud võimalik kohe kaasaegsete andmetega ennustusmudelit üle treenida, kuna liikluse hõreduse tõttu ei leitud piisavalt näiteid kinnisõidetutest ristmikest. Seepärast oleks vajalik hiljem koguda uuesti andmeid. Kuna Sander Aasavälja lõputöös treenitud mudel on ikka salvestatud, on võimalik ka kontrollida, kui palju mudeli ennustustäpsus kahe aastaga või rohkem on kahanenud.

Lisaks on punase tule tuvastamise süsteemi võimalik parandada, optimeerides tuvastatavaid HSV väärtuste vahmikke või trennides üle sealset kasutatavat mudelit. On võimalik, et kaamerast tuleva pildi liigsel muutumisel, olgu selleks kvaliteet või vaatenurk, võib tekitada tõrkeid mudelis.

Samuti on üheks peamiseks mudeli edasiarenduse viisiks mudelit ennast parandada. Arvestades, et andmete mahu suurendamisel mudeli ennustustäpsus muutus minimaalselt, on mudeli täpsus kinni selle ülesehituse taga või see on juba saavutanud maksimaalse võimaliku ennustustäpsuse.

5 Kokkuvõte

See töö valmis koostöös Tallinna Linnavalitsusega, et vähendada ristmike kinnisõitmisi. Eesmärkideks oli mudeli treeningandmete kogumise automatiseerimine, et lihtsustada olemasoleva konvolutsioonilise närvivõrkude mudeli treenimist, kasutades pooljuhendatud õppimist. Sellega seoses oli sihiks ka mudeli täpsuse parandamine, võttes arvesse varasemate tööde treeningandmete vähest mahtu.

Käesolevas töös valmis põhimudelit abistav mudel, mis treeniti klassifitseerima, kas ristmik on kinnisõidetud või mitte. Abimudel treeniti 280 sisendi suuruse treeningandmestikuga 91% protsendi täpsuseni. Lisaks automatiseeriti videost andmete kogumine: Kui varasemalt oli vaja käsitsi nimetada failid õigele kujule, et nendest treeningandmed välja lõigata, siis foorituvastusmudeli abiga sai automaatselt teada millal õige moment kaadri lõikamiseks on ja samas kontrollides, et kaader lõigatakse välja õigel fooritsükli ajal.

Kuna arvatavasti aastaaja ning Covid-19 tõttu oli liiklus hõredam tavalisest [5], ei leidunud piisavalt näiteid kinnisõidetutest ristmikest. See sundis üle minema varasemalt kogutud 2019. aasta videotele, kus eesmärgiks võeti otseselt võrrelda treeningandmestiku mahu mõju ennustusmudeli täpsusele, treenides mudelit mitmekordselt suurema andmestikuga. Mudelit treeniti 883 sisendi suuruse andmestikuga, mille tulemuseks jäi mudeli ennustustäpsus ka varasemalt olnud 83%-le. Et tõsta mudeli efektiivsust, oli vaja muuta mingil määral ka mudelit ennast, mille tulemusena lõpuks tõsteti ennustustäpsus 83%-lt 85%-le.

Mudeli päris ristmikule rakendamine jäi liikluse hõreduse tõttu ära, kuna otstarbekam oleks võtta see ette tulevikus, kui ristmiku kinnisõitmine hakkab uuesti esinema, mil saab ka treeningandmeid käesolevas töös loodud lahendustega lihtsasti suurel hulgal koguda.

Kasutatud kirjandus

- [1] E. Muring, "Ristmiku kinnisõitmise ennustamine fooritsükli optimeerimiseks", [Bakalaureusetöö], Infotehnoloogia teaduskond, TalTech, Tallinn, Eesti, 2018. [Online]. Loetud aadressil <https://digikogu.taltech.ee/et/Item/1f314994-d64d-414d-a78b-3a1b9f5faac6>
- [2] S. Aasaväli, "Near Real-time Intersection Congestion Prediction Using Convolutional Neural Network", [Magistritöö], Infotehnoloogia teaduskond, TalTech, Tallinn, Eesti, 2019. [Online]. Loetud aadressil <https://digikogu.taltech.ee/et/Item/ae5da556-acd0-47b9-bcf5-111b0c63a16c>
- [3] Y. Wu, E. Dobiban ja S. Davidson, "DeltaGrad: Rapid retraining of machine learning models," in *Proceedings of the 37th International Conference on Machine Learning*, 2020, Virtual, pp. 1-2. Loetud aadressil: <http://proceedings.mlr.press/v119/wu20b.html> Kasutatud 22.04.2021.
- [4] Tom's Hardware, *Nvidia Jetson Nano: the Raspberry Pi of AI?*, 2019, [Online]. Loetud aadressil: <https://www.tomshardware.com/news/jetson-nano-features-price,38856.html> Kasutatud 27.04.21.
- [5] Inside Machine Learning, *Introduction to semi-supervised learning and adversarial training*, 2019, [Online]. Loetud aadressil: <https://medium.com/inside-machine-learning/placeholder-3557ebb3d470> Kasutatud 16.04.21.
- [6] Liiklusloenduse tulemused 2020. aastal, 2020, [Online]. Loetud aadressil: https://www.mnt.ee/sites/default/files/2020._aasta_loendusaruanne.pdf Kasutatud 28.04.21.
- [7] Python 3.8, Python Software Foundation, 2021, [Online]. Loetud aadressil: <https://www.python.org/> Kasutatud 10.04.21.
- [8] OpenCV 4.5.1, Itseez, 2021, [Online]. Loetud aadressil: <https://www.opencv.org/> Kasutatud 10.04.21.
- [9] Elite Data Science, *Overfitting in Machine Learning: What It Is and How to Prevent It*, 2019, [Online]. Loetud aadressil: <https://elitedatascience.com/overfitting-in-machine-learning> Kasutatud 19.04.21.
- [10] Torchvision, *torchvision.transforms*, 2021, [Online]. Loetud aadressil: <https://pytorch.org/vision/stable/transforms.html> Kasutatud 22.04.21.
- [11] Imaginea Labs, *RGB vs HSV for Computer Vision*, 2019, [Online]. Loetud aadressil: <https://labs.imaginea.com/rbg-vs-hsv-for-computer-vision> Kasutatud 22.04.21.
- [12] GitHub, *Traffic Light Detection For Beginners Using OpenCV And YOLOv3*, 2021, [Online]. Loetud aadressil: <https://github.com/initdebugs/Beginner-Traffic-Light-Detection-OpenCV-YOLOv3> Kasutatud 26.04.21.
- [13] GitHub, *Real Time Traffic Light Detection using Deep Learning (YOLOv3)*, 2021, [Online]. Loetud aadressil: <https://github.com/sovit-123/Traffic-Light-Detection-Using-YOLOv3> Kasutatud 26.04.21.

- [14] Github, *GitHub - HevLfreis/TrafficLight-Detector: Simple traffic light detector by opencv python*, 2021, [Online]. Loetud aadressil:
<https://github.com/HevLfreis/TrafficLight-Detector> Kasutatud 26.04.21.
- [15] Stack Overflow, *Choosing the correct upper and lower HSV boundaries for color detecion with 'cv::inRange' (OpenCV)*, 2012, [Online]. Loetud aadressil:
<https://stackoverflow.com/a/59906154> Kasutatud 26.04.21.

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Matthias Rajamets

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Masinõppeprotsessi automatiseerimine ristmiku kinnisõitmise vähendamise süsteemile“, mille juhendaja on Martin Rebane
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

29.04.2021

¹ Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtjaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.