

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Epp Muring 154837 IAPB

**RISTMIKU KINNISÕITMISE
ENNUSTAMINE FOORITSÜKLI
OPTIMEERIMISEKS**

Bakalaureusetöö

Juhendaja: Martin Rebane
MSc

Tallinn 2018

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: Epp Muring

21.05.2018

Annotatsioon

Käesoleva töö eesmärgiks on luua tehniline lahendus, mis suudab videokaadrit analüüsides ette ennustada ristmike kinnisõitmisi. Edasiarendusel on võimalik ennustuste põhjal optimeerida rohelise fooritsükli pikkust. Fooritsükli optimeerimise eesmärgiks on vältida olukordi, kui roheline tuli võimaldab ristmikule välja sõita rohkematel autodel kui ristmikult ära sõitmiseks ruumi on. Töö valmib Tallinna Tehnikaülikooli ning Tallinna Linnavalitsuse koostööprojekti raames.

Töö käigus analüüsiti olemasolevaid lahendusi ning otsustati mudeli valikul närvivõrkude kasuks. Närvivõrgu sisendiks vajalik andmestik on koostatud spetsiaalselt käesoleva töö jaoks kasutades selleks Tallinna Transpordiametilt saadud videojäädvustusi A. H. Tammsaare tee ning Sõpruse puiestee ristmikult.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 31 leheküljel, 5 peatükki, 19 joonist, 14 tabelit.

Abstract

Predicting Congestions At An Intersection To Optimize The Traffic Light Cycle

The purpose of this thesis is to create a technical solution that has the ability to predict congestions at an intersection based on frames from video footages. The predictions can be used to optimize the traffic light cycle and to avoid further congestions.

During rush hour there are often cases when there is no more room left in a specific driving direction but since the traffic light is still green, more cars drive out into the intersection and therefore block the other driving directions as well. This situation occurs because the traffic cycle does not take into account how much room there is actually left.

This technical solution uses frames from videos as input to predict the congestions. The technical solution is based on deep neural networks as they have proved to work well on traffic prediction and image classification problems.

All used input datasets are custom made for this specific problem using only video footage from a particular intersection. The datasets include raw images and processed images that highlight object movement between frames.

It was proved that the best solution for this problem would be to use a convolutional neural network using a dataset that consists only of raw image frames. The final model achieved test accuracy of 80%.

The thesis is in Estonian and contains 31 pages of text, 5 chapters, 19 figures, 14 tables.

Lühendite ja mõistete sõnastik

Ahendamine	Protsess, mille eesmärgiks on omaduskaardi dimensioone vähendada nii, et jääks alles kõige olulisem informatsioon (<i>pooling</i>)
<i>Dense optical flow</i>	<i>Optical flow</i> meetod, mis jälgib kaadrite vahel iga piksli liikumist
Aktivatsioonifunktsioon	Mittelineaarne funktsioon, mida kasutatakse võrgusõlme väljundi arvutamisel (<i>activation function</i>)
Esitustäpsus	Mudeli valideerimiseks arvutatav väärtus, mis näitab valepositiivsete ennustuste osakaalu (<i>precision</i>)
F1 skoor	Esitustäpsuse ning tundlikkuse pealt arvutatav väärtus mudeli valideerimiseks (<i>F1 score</i>)
Filter	Maatriks, mille abil leitakse sisendpildist kõige olulisemad omadused
Fooritsükel	Tsükel algab rohelise tulega ning lõppeb punase tule sisse lülitudes
Kinnisõidetud ristmik	Ristmik loetakse kinnisõidetuks, kui vähemalt pool auto kerest jääb ülekäigurajale
Klassifitseerimise täpsus	Protsent, mis leitakse jagades õiged ennustused kõikide tehtud ennustustega
Kulufunktsioon	Näitab erinevust soovitud väljundi ja tegeliku väljundi vahel, proovitakse võimalikult palju minimeerida (<i>cost function</i>)
L2 reguleerimine	Liiga kohandunud mudeli reguleerimise meetod, piirab võrgusõlmede kaalusid
Liiga kohandunud mudel	Mudel kohandub treeningandmete järgi ning ennustustäpsus langeb testandmete puhul järsult (<i>high variance, overfitting</i>)
Liiga üldine mudel	Mudel saavutab madala täpsuse nii treening- kui testandmetel (<i>high bias, underfitting</i>)
Lucas-Kanade meetod	<i>Optical flow</i> meetod, mis kasutab objektide jälgimiseks nurkade tuvastust
Minipartii	Andmestiku väiksem osa (<i>mini-batch</i>)
Närvivõrk	Mudel, mis koosneb sisend-, väljundkihtidest ning nende vahel asuvatest peidetud kihtidest (<i>neural network</i>)
Omaduskaart	Sisaldab sisendpildi kõige olulisemaid omadusi (<i>feature map</i>)
<i>Optical flow</i>	Meetod, millega on võimalik tuvastada objektide kaadritevahelisi liikumisi
Peidetud kihid	Närvivõrgus sisend- ning väljundkihtide vahele jäävad kihid

RGB	Punane – roheline – sinine
Sisendkiht	Närvivõrgu esimene kiht (<i>input layer</i>)
Testandmestik	Andmestik, mille peal valideeritakse mudeli ennustusi (<i>test dataset</i>)
Treeningandmestik	Andmestik, mida kasutatakse mudeli treenimiseks (<i>training dataset</i>)
Tundlikkus	Tõeste positiivsete määr, näitab väärnegatiivsete ennustuste osakaalu (<i>recall, sensitivity, true positive rate</i>)
Väljundkiht	Närvivõrgu viimane kiht (<i>output layer</i>)
Võrgusõlm	Närvivõrgu väikseim üksus (<i>node</i>)
Võrgusõlmede väljalülitamise meetod	Meetod liigse kohandamise kontrollimiseks, millega lülitatakse välja treenimise ajal suvalisi võrgusõlmesid (<i>dropout, dropconnect</i>)
Õppimismäär	Näitab, kui palju kohandatakse iga iteratsioon võrgusõlmede kaalusid (<i>learning rate</i>)

Sisukord

1 Sissejuhatus.....	12
1.1 Nõuded.....	13
1.2 Töökäik	13
2 Analüüs	14
2.1 Võimalikud lahendused	14
2.1.1 Deterministlik lähenemine.....	14
2.1.2 Ennustamisvõimega lahendused	15
2.1.3 Närvivõrgud	16
2.1.4 Sarnased lahendused	17
3 Sisendandmestiku koostamine	19
3.1 Vabakasutuses olevate sisendandmete analüüs	19
3.2 Koostatav sisendandmestik.....	20
3.2.1 Kriteeriumid.....	21
3.2.2 Sisendandmestiku koostamine	23
4 Tehniline lahendus	25
4.1 Sisendi valik.....	25
4.1.1 <i>Optical flow</i>	26
4.2 Mudelite parameetrite valik ning modifitseerimine.....	28
4.2.1 Liiga üldine mudel	29
4.2.2 Liiga kohandunud mudel	29
4.2.3 Kombinatsioonid.....	31
4.3 Täielikult ühendatud närvivõrk.....	32
4.4 Konvolutsiooniline närvivõrk	34
4.5 Närvivõrgu tulemuste valideerimine.....	38
4.6 Edasiarenduse võimalused	41
5 Kokkuvõte.....	42
Kasutatud kirjandus	43
Lisa 1 – Täielikult ühendatud närvivõrgu tulemused 13-kihilise sisendmaatriksi peal..	44

Lisa 2 – Täielikult ühendatud närvivõrgu tulemused 13-kihilise sisendmaatriksi peal L2 piiranguga	45
Lisa 3 – Täielikult ühendatud närvivõrgu tulemused 2-kihilise sisendmaatriksi peal L2 piiranguga	46
Lisa 4 – Täielikult ühendatud närvivõrgu tulemused 3-kihilise RGB töötlemata kaadri peal L2 piiranguga	47
Lisa 5 – Konvolutsioonilise närvivõrgu tulemused 13-kihilise sisendmaatriksi peal.....	48
Lisa 6 – Konvolutsioonilise närvivõrgu tulemused 13-kihilise sisendmaatriksi peal L2 piiranguga	49
Lisa 7 – Konvolutsioonilise närvivõrgu tulemused 2-kihilise sisendmaatriksi peal L2 piiranguga	50
Lisa 8 – Konvolutsioonilise närvivõrgu tulemused 3-kihilise RGB sisendmaatriksi peal L2 piiranguga	51
Lisa 9 – Konvolutsioonilise närvivõrgu tulemused 3-kihilise RGB töötlemata kaadri peal võrgusõlmede väljalülitamisega	52

Jooniste loetelu

Joonis 1. Ristmik koos ära märgitud probleemse sõidusuunaga (aluskaart: Google Maps).	12
Joonis 2. Mudeli valikul peamised rolli mängivad aspektid [2].	15
Joonis 3. Näide 3-kihilisest täielikult ühendatud närvivõrgust [3].	16
Joonis 4. Kaader Berkley DeepDrive andmestikust, kus teekatte markeering on ka pimedas näha [8].	20
Joonis 5. Kaader KITTI andmestikust, kus on madala kaameranurga tõttu ristmikust näha väike osa [9].	20
Joonis 6. Viimased rivas olevad autod takistavad teiste sõidusuundade liiklust.	21
Joonis 7. Teised sõidusuunad saavad normaalselt toimida, kuid autod on ülekäigurajal.	22
Joonis 8. Kõrvalriidades on vaba ruumi, kuid autojuht otsustab jääda pikemasse kolonni.	22
Joonis 9. Kaader udusest ja pilvisest ilmast.	23
Joonis 10. Kaader pimedast ajast.	24
Joonis 11. Kaader päikeselisest ilmast.	24
Joonis 12. Kaks sarnast kaadrit neli sekundit enne tsükli lõppu erineva lõpptulemusega.	26
Joonis 13. Lucas-Kanade meetod. On näha, et parasjagu ristmikule sõitnud autod on jäänud tuvastamata.	27
Joonis 14. <i>Dense optical flow</i> meetod. Kõik ristmikule sõitvad autod on tuvastatud. ...	28
Joonis 15. Erinevad tulemused: (a) liiga üldine, (b) täpselt paras, (c) liiga kohandunud. Despois joonise põhjal [11].	29
Joonis 16. Närvivõrk enne ja peale suvaliste võrgusõlmede eemaldamist. Despois joonise põhjal [11].	30
Joonis 17. Erinevad variatsioonid samast pildist [11].	31
Joonis 18. Tulemus, kus on näha nii liiga üldise kui ka liiga kohandunud mudeli tunnusjooni.	31

Joonis 19. Sisendpildi ja filtri vaheline korrutis. Tulemuse paigutus väljundmaatriksis tuleneb filtri paigutusest sisendpildil [17].....	35
--	----

Tabelite loetelu

Tabel 1. Treening- ja testandmete jaotumine andmestikus.....	24
Tabel 2. Täielikult ühendatud närvivõrgu tulemused 13-kihlise sisendmaatriksi peal...	32
Tabel 3. Täielikult ühendatud närvivõrgu tulemused 13-kihlise sisendmaatriksi peal koos L2 piiranguga.....	33
Tabel 4. Täielikult ühendatud närvivõrgu tulemused 2-kihlise sisendmaatriksi peal koos L2 piiranguga.....	34
Tabel 5. Täielikult ühendatud närvivõrgu tulemused 3-kihlise RGB töötlemata kaadri peal L2 piiranguga.	34
Tabel 6. Konvolutsioonilise närvivõrgu tulemused 13-kihlise sisendmaatriksi peal.	36
Tabel 7. Konvolutsioonilise närvivõrgu tulemused 13-kihlise sisendmaatriksi peal L2 piiranguga.	36
Tabel 8. Konvolutsioonilise närvivõrgu tulemused 2-kihlise maatriksi peal L2 piiranguga.	36
Tabel 9. Konvolutsioonilise närvivõrgu tulemused 3-kihlise RGB töötlemata kaadri peal L2 piiranguga.....	37
Tabel 10. Konvolutsioonilise närvivõrgu tulemused 3-kihlise RGB töötlemata kaadri peal võrgusõlmede väljalülitamisega.....	38
Tabel 11. Närvivõrgu tehtud ennustused tabelina. Brownlee põhjal [20]	39
Tabel 12. Täielikult ühendatud närvivõrgu ennustused grupeeritult.	40
Tabel 13. Konvolutsioonilise närvivõrgu ennustused grupeeritult.	40
Tabel 14. Mudelite esitustäpsused, tõeste positiivsete määrad ning F1 skoorid.	41

1 Sissejuhatus

Suurtel ristmikel näeb tiipitudel tihti olukorda, kui mingis sõidusuunas enam vaba ruumi pole, kuid foorituli on endiselt roheline ning seetõttu jäävad ristmikule välja sõitnud autod teistele sõidusuundadele ette. Selline olukord tekib peamiselt ristmiku läbilaskevõime ning fooritsükli pikkuse erinevusest ehk roheline tuli võimaldab ristmikule välja sõita rohkematel autodel kui ristmikult ära sõitmiseks ruumi on. Selline ristmike kinnisõitmine on kuhjuva kahjuga tegevus, mis võib põhjustada tugevaid liiklushäireid mitmete kilomeetrite raadiuses.

Töö eesmärgiks on luua tehniline lahendus, mis suudab videokaadrit analüüsides ette ennustada ristmike kinnisõitmisi. Edasiarenduse korral on võimalik selle põhjal optimeerida rohelist fooritsükli pikkust.

Antud lahenduse arendamiseks kasutatakse Tallinnas Mustamäel asuvat A. H. Tammsaare tee ning Sõpruse puistee ristmikut. Keskendutakse kõige probleemsemale suunale ehk Õismäe poolt mööda A. H. Tammsaare teed lennujaama poole viivale suunale. Järgneval joonisel on näha vaatluse all olev ristmik ning probleemne sõidusuund.



Joonis 1. Ristmik koos ära märgitud probleemse sõidusuunaga (aluskaart: Google Maps).

Kasutatud kaamerajäädvustused pärinevad Tallinna Transpordiametilt ning keskenduvad eelkõige just tipptundide liiklusele, sest tipptundide perioodid on kõige probleemsemad. Töö valmib Tallinna Tehnikaülikooli ja Tallinna Linnavalitsuse koostööprojekti raames.

1.1 Nõuded

Lahendusel peab olema iseõppimise võime, sest liikluskooormus varieerub tugevasti ning iga fooritsükli ajal võib olla vajadus erineva pikkusega roheliseks tuleks.

Lahendusel peab olema piisavalt pikk ennustamisvõime – kuna tegu on liikuvate autodega, siis ei ole võimalik oodata kuni reaalselt ristmikul ruum otsa saab ning seejärel punane tuli sisse lülitada, vaid tuleb arvestada ka kollase tulega, mille ajal suur osa autosid veel ristmikule sõidab. Seega peaks ristmiku täitumist ette nägema minimaalselt kolm sekundit, väikese varuga neli.

Süsteem peab suutma videopilti analüüsida erinevates ilmastikuoludes (päeval, öösel, hämaras, vihmajärg, udus, lumesajus jms).

Lõplik süsteem peab töötama reaalajas ehk ilma viiteta. Seega peab käesoleva töö käigus leitud prototüüp olema arvutuslikult võimalikult lihtne.

1.2 Töökäik

Esmalt leitakse sarnaseid lahendusi analüüsides potentsiaalne mudel, mis võiks käesoleva probleemi lahendamiseks sobida. Seejärel koostatakse mudeli jaoks võimalikult tasakaalustatud ning mitmekülgsete sisenditega andmestikud, millega mudeli tööd hinnata.

Parima mudeli ning andmestiku kombinatsiooni selgudes analüüsitakse ning valideeritakse saadud tulemusi. Pakutakse välja võimalikke viise, kuidas saadud tulemust veelgi parendada saaks.

2 Analüüs

Selles peatükis analüüsitakse võimalikke lahendusi ning tehnoloogiaid.

2.1 Võimalikud lahendused

Võimalike lahenduste analüüsil võeti aluseks eelkõige teised sarnastel teemadel realiseeritud lahendused.

2.1.1 Deterministlik lähenemine

Kõige esimese lahendusena analüüsitakse ristmiku kinnisõitmise vältimiseks deterministlikku lähenemist – määrame ära, kust maalt võib lugeda vaba ruumi otsa lõppenuks, tuvastame selle momendi ning seejärel muudame fooritule punaseks.

Sellist lahendust saaks realiseerida paigutades ristmikule FLIR kaamera (*forward looking infrared camera*), mis suudab vaatevälja jäävaid objekte tuvastada infrapunakiirguse abil [1]. Paigutades kaamera lubatud ala tagumisele piirile, saame objekti tuvastamisel järeldada, et vaba ruum on otsa lõppenud ning foorituli tuleks keelavaks muuta.

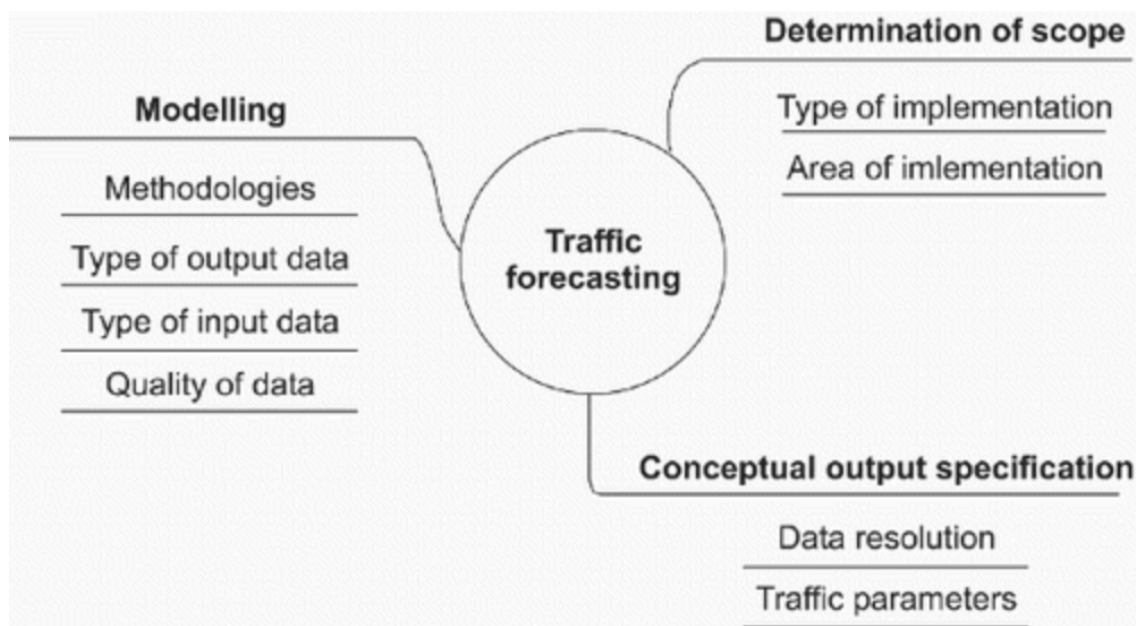
Sellise lahenduse esimene puudus on see, et kaamerast mööduvad ka kõik need autod, mis tegelikult ära mahuvad, kuid oluline oleks tuvastada just selline objekt, mis vaba ruumi lõppu seisma jääb. Et eristada seisma jäänud objekti lihtsalt mööduvast autost, saaks määrata minimaalse aja, kui kaua objekt kaadris peab olema, et objekti liikumatuks lugeda. See aga tähendaks, et tegelikult on ristmikul juba mitu sekundit vaba ruum otsas, kuid foorituli on endiselt roheline.

Isegi kui seisev objekt tuvastada ilma vahepealse viiteta, on ristmiku kinnisõitmise ennetamiseks juba hilja, sest enne punase tule sisse lülitamist tuleb näidata ka kollast tuld, mis kestab vähemalt kolm sekundit ning mille jooksul veel mitmed autod ristmikule sõidavad.

2.1.2 Ennustamisvõimega lahendused

Kuna deterministlikust lahendusest jääb väheks, tuleb vaadata keerulisemaid mudeleid. Vlahogianni et al analüüsivad ning võrdlevad ennustamisvõimega algoritme liiklusennustusprobleemide puhul [2]. Tuuakse välja, et mudeli valik sõltub paljudest erinevatest aspektidest, mida võib jagada kolme pealmisse gruppi: rakendusala, kontseptuaalne väljundi spetsifikatsioon ning modelleerimine.

Rakendusala juurde kuulub kasutuspiirkond (maantee, linnasisene) ning otstarve (liikleja infosüsteem, liikluskorraldussüsteem). Väljundi spetsifikatsiooni puhul sõltub mudeli valik ennustamist vajavatest parameetritest (liiklustihedus, kiirus) ning soovitud ennustamisvõime pikkusest ja sammust. Modelleerimine on seotud meetodi, väljundi tüübi, sisendi tüübi ning andmete kvaliteediga [2]. Joonis 2 illustreerib erinevaid aspekte, mida tuleks mudeli valikul silmas pidada.



Joonis 2. Mudeli valikul peamised rolli mängivad aspektid [2].

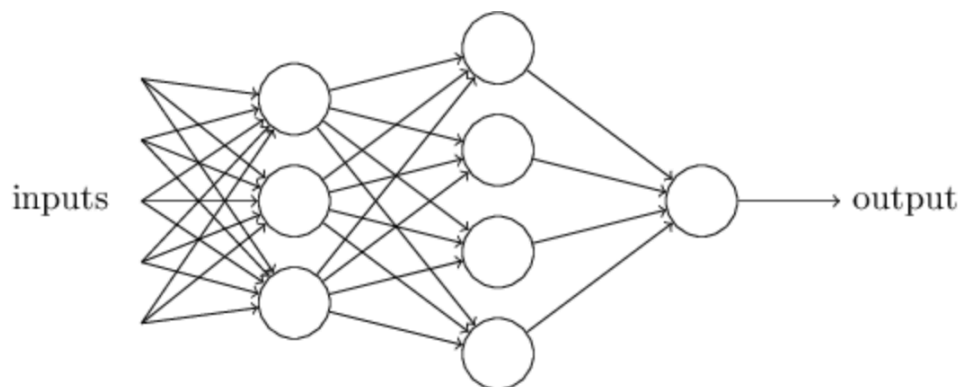
Meetodite valik jagatakse laias laastus kaheks – parameetrilised ning mitteparameetrilised. Parameetrilistest meetoditest on linnasisese liikluse ennustamiseks kasutatud näiteks ARIMA mudelit, kuid selle puhul leiti, et mudel suudab hästi toimida keskpärastes liiklustingimustes, kuid halvasti äärmuslikes olukordades. Reaalsuses on aga linnasisene liiklus just kiiresti muutuv ning kõigub äärmusest äärmusesse. Seepärast leiti, et linnasisese liikluse ennustamiseks sobivad paremini mitteparameetrilised

modelid, mis ei eelda kindlaks määratud näitajaid, vaid mille puhul on kõige olulisem sisendandmestik, mille põhjal hakkab mudel ise korduvaid mustreid leidma [2].

Mitteparameetrilised mudelid on näiteks mitteparameetiline regressiooni mudel ning närvivõrk. Nende kahe võrdluses tuuakse välja, et mitteparameetrilise regressiooni mudeli puhul on vaja eelnevalt kindlaks määrata, kas kasutatakse lineaarset või mittelineaarset modelleerimist, see-eest närvivõrgu puhul ei ole see vajalik. Kuigi mitteparameetrilise regressiooni mudeli struktuur on võrreldes närvivõrguga lihtsam, peetakse närvivõrgu täpsust kõikidest eelnevatest mudelitest paremaks, mis jäi ka antud töö puhul kõige kaalukamaks argumendiks [2].

2.1.3 Närvivõrgud

Närvivõrk on mudel, mis koosneb sisend- ja väljundkihtidest ning nende vahel asuvatest peidetud kihtidest. Iga kiht koosneb N võrgusõlmest ning järjestike kihtide võrgusõlmed on omavahel ühendatud (Joonis 3). Kihtide ja võrgusõlmede arv sõltub kasutusala ja mudelist [3].



Joonis 3. Näide 3-kihilisest täielikult ühendatud närvivõrgust [3].

Et närvivõrk suudaks mingisugust probleemi lahendada, on seda vaja eelnevalt treenida. Selle jaoks on vaja genereerida treeningandmestik, mis koosneb närvivõrkudele antavatest sisenditest ning oodatud väljunditest. Igas kihis ja võrgusõlmes analüüsitakse erinevaid sisendandmete omadusi vastavalt võrgusõlmele määratud kaalule ning tulemus antakse edasi järgmistele kihtidele kuni lõpuks jõutakse väljundini. Kui saadud väljund ei klapi treeningandmestikus oodatud väljundiga, korrigeeritakse võrgusõlmede kaalusid. Mida rohkem treeningandmeid kasutada, seda paremini õpib närvivõrk esinevaid mustreid ära tundma [3].

Närvivõrgu tulemuse hindamiseks kasutatakse testandmestikku ning leitakse, kui suure osa testsisendite puhul jõudis närvivõrk õige väljundini. Närvivõrke on omakorda erinevat tüüpi ning valik sõltub probleemi olemusest.

2.1.4 Sarnased lahendused

Sarnaste lahenduste otsimisel lähtuti kahest vaatenurgast – ühest küljest on tegemist liiklusega seotud probleemiga, teisest küljest pildi klassifitseerimise probleemiga. Seepärast uuriti sobiva lahenduse leidmiseks mõlemat valdkonda.

Liikluse ennustamise probleemide puhul on närvivõrgud väga laialt levinud. Yun et al analüüsivad tagasipöördumisega (*static backpropagation*) täielikult ühendatud närvivõrgu, ajalise viivitusega rekurrentse ehk korduva närvivõrgu (*time-delayed recurrent model*) ning piiratud impulssvastuse mudeli (*finite impulse response, FIR*) tulemusi liiklustiheduse ennustamisel erinevates liiklusoludes, sealhulgas ka linnasisestel ristmikel [4]. Töö käigus leiti, et linnasiseste ristmike puhul saavutab kõige parema tulemuse rekurrentse närvivõrguga, sest see arvestab lisaks sisenditele ka eelnevaid andmeid.

Dia töö eesmärk on ennustada liiklustingimusi kiirtee lõigul, et autojuhid saaksid halbade liiklusolude korral teise tee kasuks otsustada ning nii liiklustihedust ühtlustada [5]. Töö käigus katsetati samuti staatilise tagasipöördumisega täielikult ühendatud närvivõrku, rekurrentset närvivõrku ning hübriidvõrku, mis kasutas rekurrentset närvivõrku ning lühimälu. Selgus, et ka selle probleemi puhul toimus kõige paremini korduv närvivõrk, mis suutis täpselt ennustada kuni 15 minutit tulevikku. Samas said ka hübriidvõrk ning täielikult ühendatud närvivõrk väga hea tulemuse – suudeti 90-94% täpsusega 5 minutit ette ennustada.

Eelnevad lahendused on keskendunud eelkõige liiklustiheduse ennustamisele, kuid kuna käesoleva töö eesmärgiks on kaadri põhjal tuvastada piisavalt vara olukorda, kui ristmik on kinnisõitmise ohus, võib probleemi vaadelda ka kui pildi klassifitseerimise probleemi (on kinnisõidetud või ei ole kinnisõidetud).

Wilkinson ja Kanellopoulos kasutavad pildi klassifitseerimise jaoks tavalist täielikult ühendatud närvivõrku, tuues välja erinevaid viise, kuidas seda kõige efektiivsemalt kasutada [6]. Bengio ja Lecun väidavad samuti, et mõndade pildi klassifitseerimise

probleemide puhul võib juba eelmainitud täielikult ühendatud närvivõrk väga tulemuslik olla, kuid tihti on edukam konvolutsiooniline närvivõrk (*convolutional network*), mis suudab pildilt olulisemad omadused välja tuua [7].

Sarnaste lahenduste uurimisel selgus, et liikluse ning pildi klassifitseerimise probleemide puhul esinevad kõige paremini erinevat tüüpi närvivõrgud – liikluse puhul korduv närvivõrk, klassifitseerimise puhul konvolutsiooniline. Mõlemal juhul saavutati rahuldavaid tulemusi ka lihtsama arhitektuuriga täielikult ühendatud närvivõrgu abil. Kuna käesoleva töö põhjal valmistatava lõpptulemuse üks lõppeesmärkidest on reaalses töötamise võimekus, on vaja leida arvutuslikult kõige lihtsam mudel. Et täielikult ühendatud närvivõrk on arhitektuurilt lihtsam kui konvolutsiooniline ning rekurrentne närvivõrk, arendatakse käesoleva töö käigus kõige põhjalikumalt just täielikult ühendatud närvivõrku.

3 Sisendandmestiku koostamine

Peatükis analüüsitakse erinevaid olemasolevaid sisendandmestikke ning põhjendatakse, miks otsustati selle tehnilise lahenduse jaoks uus andmestik genereerida. Samuti kirjeldatakse sisendandmestiku kriteeriumeid ning koostamise protsessi.

3.1 Vabakasutuses olevate sisendandmete analüüs

Kuna närvivõrgu treenimiseks vajaliku sisendandmestiku koostamine on väga kallis ja ajamahukas töö, on olemas vabakasutuses andmestikke, mida võib uurimistöö eesmärgil kasutada.

Üks sellistest andmestikest on näiteks KITTI¹, mis sisaldab mitusada gigabaiti nii töödeldud kui töötlemata videosid erinevatest piirkondadest, sealhulgas ka linnast ning ristmikest. Lisaks on videod väga täpselt ära annoteeritud: kasutatud on palju erinevaid klasse, seega annab andmestik võimaluse katsetada, mida lahenduse puhul sisendiks valida.

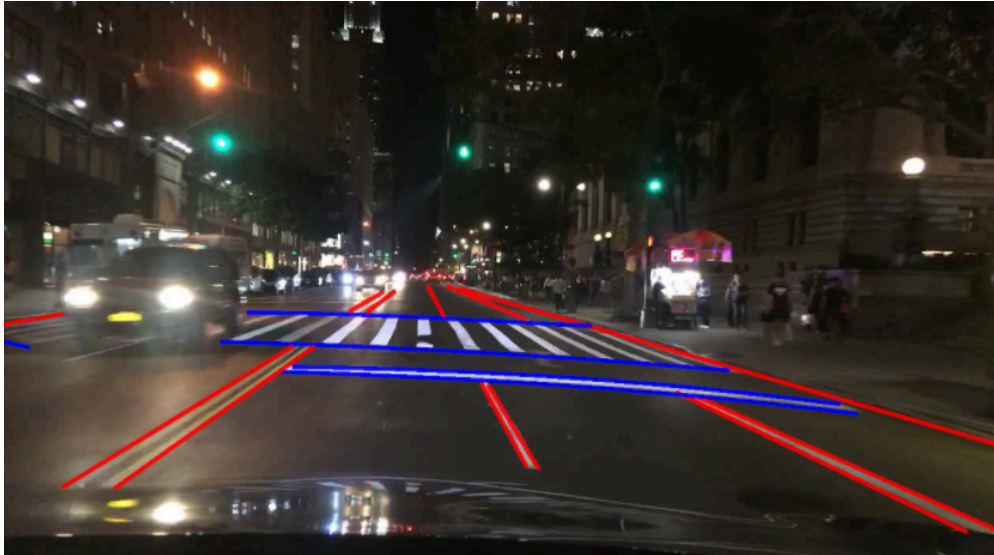
HD1K² andmestik sisaldab üle 1000 kaadri väga hea resolutsiooniga treeningandmeid. Resolutsioon on oluline, sest närvivõrkude puhul mõjutab sisendandmestiku kvaliteet väga tugevalt mudeli tulemusi. Samuti on rõhku pandud erinevatele ilmastiku- ning valgustingimustele, mis aitavad närvivõrgu töökindlust parandada.

Berkeley DeepDrive³ on samuti väga rikkalik andmestik, mis sisaldab lisaks kõigele ülalmainitule veel teekatte markeeringut. See on kasulik, kui on vaja kindlaks määrata, kust näiteks ristmiku peatumisjooned asuvad. Nende põhjal saab otsustada, kas ristmik on kinnisõidetud või ei, kuid pimedal ajal või halbades ilmastikutingimustes ei pruugi jooned alati näha olla. Joonisel 4 on näha kaader Berkley DeepDrive andmestikust, kus on teekatte markeering ka pimedas hästi näha.

¹ www.cvlibs.net/datasets/kitti/raw_data.php

² <http://hci-benchmark.org/>

³ bdd-data.berkeley.edu



Joonis 4. Kaader Berkley DeepDrive andmestikust, kus teekatte markeering on ka pimedas näha [8].

Kuigi eelnevad andmestikud on väga mahukad ja põhjalikud, on nende üks suurimaid puudusi kaameranurk. Kuna andmestikud on mõeldud eelkõige autonoomsete sõidukite arendamiseks, on kogu video filmitud auto esiklaasi kõrguselt ning nii on saadav infohulk väiksem (Joonis 5). Samuti on andmestikud kokku pandud pigem üldisemaks kasutamiseks mitte spetsiifilise rakenduse jaoks ning kuna antud töö on üsna kitsalt määratletud, annab uue andmestiku genereerimine kindlasti täpsema tulemuse.



Joonis 5. Kaader KITTI andmestikust, kus on madala kaameranurga tõttu ristmikust näha väike osa [9].

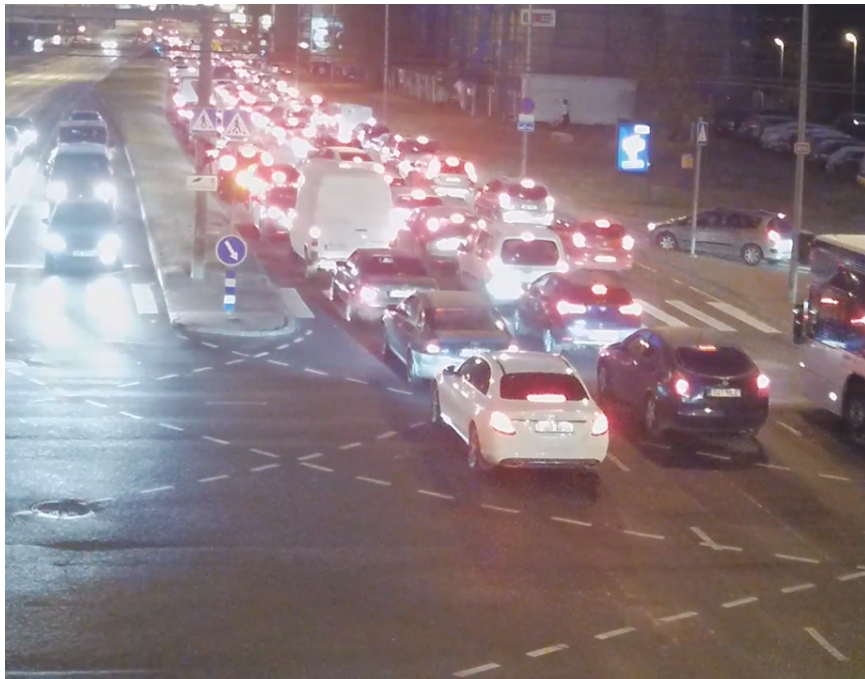
3.2 Koostatav sisendandmestik

Sisendandmestiku koostamiseks on kasutatud Tallinna Transpordiametilt saadud kaamerajäädvustusi Tammsaare ja Sõpruse puistee ristmikult. Andmed on originaalselt saadud 20-40 minutiliste videotena kaadrisagedusega 25 kaadrit sekundis, millest on välja lõigatud vajalikud kaadrid ning millele on rakendatud videotöötlust. Videosalvestised pärinevad peamiselt tipptundidest.

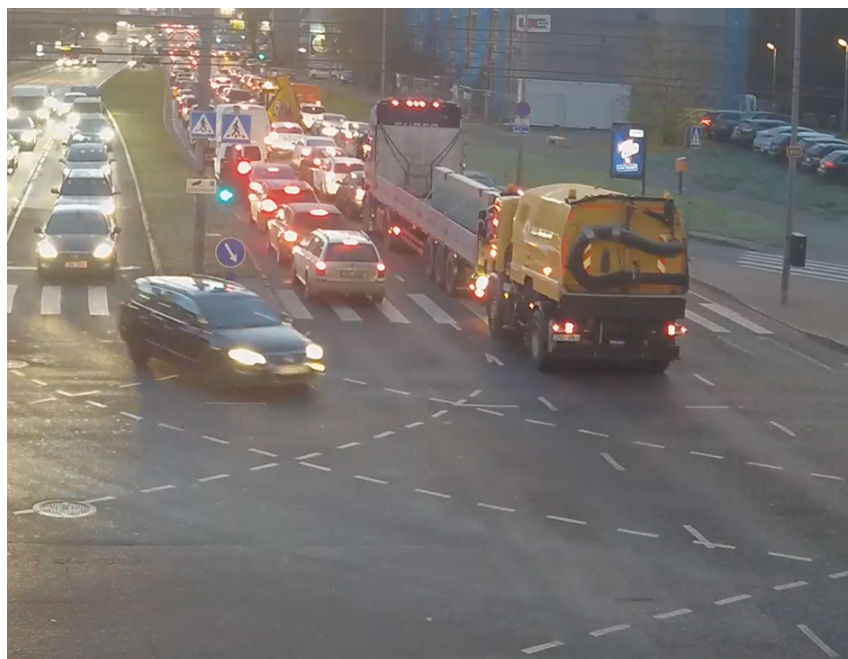
3.2.1 Kriteeriumid

Sisendandmed annoteeriti binaarselt – ristmik on kinni sõidetud (1) või ristmik ei ole kinni sõidetud (0). Kuna annoteerimine on manuaalne töö, pandi kirja kindlad tingimused, kunas võib ristmikku lugeda kinnisõidetuks, et saavutada võimalikult ühtlane ja korrektne sisendandmestik.

Kinnisõitmiseks loetakse juhtusid, kui ristmikule väljasõitnud auto jääb ruumipuuduse tõttu vähemalt poole kerega autodele ettenähtud alast väljapoole. Autodele ettenähtud ala loetakse kuni ülekäigurajani. Seega tähendab see, et kinnisõidetud momendiks ei loeta ainult selliseid juhtusid, kui auto jääb juba teistele sõidusuundadele ette, vaid ka selliseid juhtusid, kui auto peab seisma jääma ülekäigurajale. Joonisel 6 on näha kriitilist olukorda, kui autod häirivad teiste sõidusuundade liiklust ning joonis 7 illustreerib leebemat olukorda, kui sõidukid mahuvad teele ära, kuid on siiski autodele ettenähtud alast väljas. Järgnevad pildid pärinevad autori koostatud andmestikust.

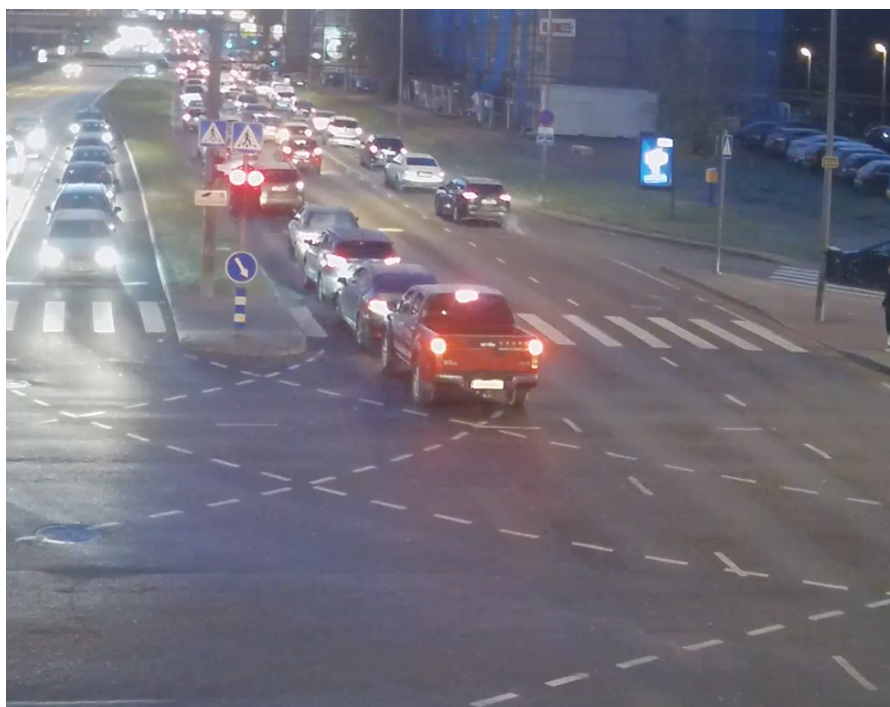


Joonis 6. Viimased ravis olevad autod takistavad teiste sõidusuundade liiklust.



Joonis 7. Teised sõidusuunad saavad normaalselt toimida, kuid autod on ülekäigurajal.

Üks ääreala juhtumitest on ka see, kui kõrvalridades on vaba ruumi, kuid autojuht jääb siiski pikemasse kolonni seisma ja blokeerib ülekäiguraja (Joonis 8). Sellisel juhul ei loeta ristmikku kinnisõidetuks, sest tegelikult on foori läbilaskevõime olnud optimaalne ning blokeerimine ei ole tingitud vaba ruumi puudusest.



Joonis 8. Kõrvalridades on vaba ruumi, kuid autojuht otsustab jääda pikemasse kolonni.

3.2.2 Sisendandmestiku koostamine

Sisendandmestiku koostamiseks kirjutati vastavad programmid, mis igast videost õigel hetkel kaadri salvestavad ning seda vastavalt vajadusele töötlevad. Seejärel vaadati kaadrid või kaadrikogumikud käsitsi üle ning annoteeriti andmed paika pandud reegleid silmas pidades.

Sisendandmestiku koostamisel on oluline jälgida, et ei tekiks suuri klassidevahelisi erinevusi ehk et erinevat tüüpi andmeid oleks võimalikult võrdselt esindatud. Vastasel juhul käitub sisendandmestikku kasutatav programm vähemuses olevatel juhtudel oluliselt halvemini ning programmi töö on ebaühtlane ning vähem usaldusväärne. Käesoleva töö puhul on erinevateks klassideks “kinnisõidetud” ning “mitte kinnisõidetud”.

Et sisendandmestikku veelgi paremaks muuta, tuleb kaasata võimalikult palju erijuhtumeid. Antud töö raames kuuluvad sinna alla näiteks erinevad ilmastikutingimused, mida illustreerivad joonised 9, 10 ja 11. Erijuhtumite kaasamine parandab testandmeid kasutava programmi töökindlust veelgi, sest nii on programm kohandunud ka siis, kui tegemist ei ole ideaaltingimustega.



Joonis 9. Kaader udusest ja pilvisest ilmast.



Joonis 10. Kaader pimedast ajast.



Joonis 11. Kaader päikeselisest ilmast.

Lõplik andmestik on 208 treeningisendiga ning 78 testisendiga. Järgnevas tabelis on näha treening- ning testandmestike jaotumine.

Tabel 1. Treening- ja testandmete jaotumine andmestikus.

	Treeningandmestik	Testandmestik
Kinnisõidetud	104	40
Mitte kinnisõidetud	104	38
Kokku	208	78

4 Tehniline lahendus

Järgnevalt kirjeldatakse tehnilise lahenduse väljatöötamise etappe. Kogu tehniline lahendus on kirjutatud programmeerimiskeeles Python¹, sest see sisaldab videotöötlusteks vajalikku teeki OpenCV² ning matemaatilisteks arvutusteks sobivat teeki Numpy³. Samuti kasutati masinõppele keskendunud teeki Tensorflow⁴. Esimeseks katsetatud lahenduseks oli mitmekihiline täielikult ühendatud närvivõrk (*fully connected network, multi layer perceptron*). Närvivõrkude realiseerimisel võttis autor aluseks Coursera⁵ masinõppe kursuse läbimisel kirjutatud programmikoodi. Programmikoodi on võimalik näha Githubi repositooriumis⁶.

4.1 Sisendi valik

Närvivõrkude puhul on kõige esimesi ning üks kõige olulisemaid etappe sisendi valimine. Võtame näiteks lihtsama mudeli, mille ülesandeks on leida perele sobiv maja. Maja sobivuse puhul mängib kindlasti rolli pere suurus ja eelarve. Kasulikku infot võib anda ka mõni pisem detail, mille peale alguses ei tule, näiteks peres olevate autode arv, sest see mõjutab otseselt seda, kui kaugel töökohast või koolist võib maja asuda. Samas ei tasu korraga sisendiks anda liiga palju infot, sest liigne müra võib närvivõrgu tulemust oluliselt halvendada. Kuna sisendi sobivust saab kontrollida ainult katsetamise teel, on mõistlik proovida erinevaid kombinatsioone.

Käesoleva töö puhul võeti sisendiks videost võetud kaadrid. Kuna töö raames keskenduti ainult ristmiku kõige probleemsemale suunale, lõigati kasutatavatest kaadritest vajalik piirkond välja, et ignoreerida teiste sõidusuundade poolt tekitatud müra. Esimeseks sisendiks valitigi lõigatud töötlemata kaader, mis on tehtud neli sekundit enne fooritsükli lõppu.

¹ www.python.org

² www.opencv.org

³ www.numpy.org

⁴ www.tensorflow.org

⁵ www.coursera.org

⁶ <https://github.com/epmaur/traffic-congestion-prediction>

Ühe töötlemata kaadri pealt võib kinnisõitmist olla raske ennustada, kuna ei ole võimalik tuvastada, kas eespool olevad autod alles liiguvad ja vaba ruumi tekib juurde või on autod juba seisma jäänud. Joonis 12 illustreerib hästi sellist olukorda – on kaks sarnast kaadrit, mis on mõlemad tehtud neli sekundit enne tsükli lõppu, kuid vasakul oleval juhul mahtusid autod tsükli lõpuks ilusti ära, parem oleval juhul jäid autod ettenähtud alast välja.



Joonis 12. Kaks sarnast kaadrit neli sekundit enne tsükli lõppu erineva lõpptulemusega.

Seepärast valiti üheks sisendi variandiks ka kaadrite kogum, millele on rakendatud *optical flow* meetodit.

4.1.1 *Optical flow*

Optical flow meetodiga on võimalik tuvastada kahe järjestikuse kaadri vahelist liikumist vektorite abil. Jälgides ühe piksli liikumist ajas ning eeldades, et piksli intensiivsus ei muutu, on võimalik tuletada *optical flow* võrrand

$$f_x u + f_y v + f_t = 0 \quad (1)$$

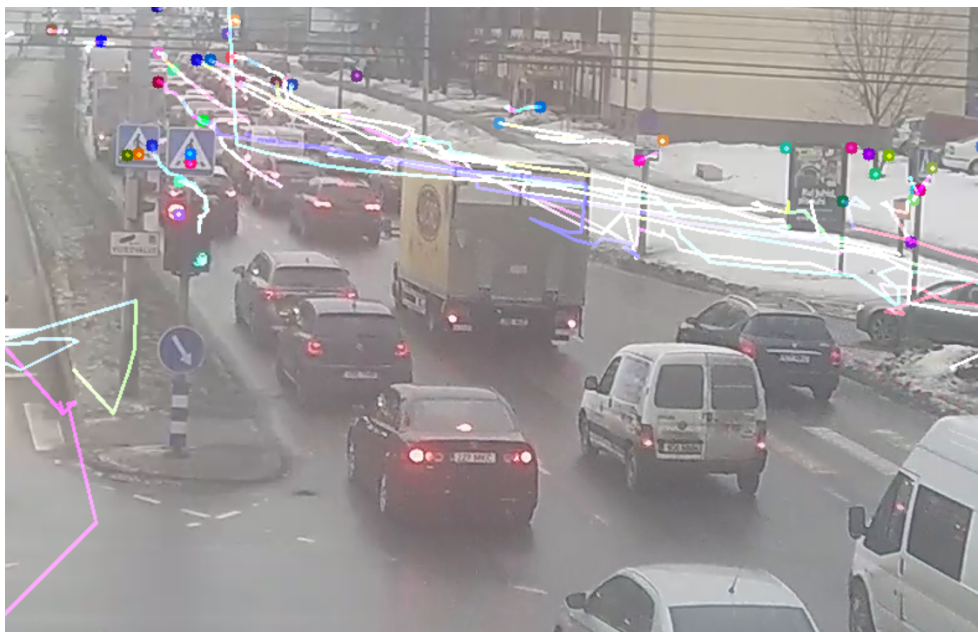
kus parameetrid on vastavalt:

$$f_x = \frac{\partial f}{\partial x}; f_y = \frac{\partial f}{\partial y}; u = \frac{dx}{dt}; v = \frac{dy}{dt} \quad (2)$$

Kuna võrrandisse jääb kaks tundmatut parameetrit u ja v , on vaja rakendada veel lisaks mingit meetodit, et parameetreid välja arvutada [10].

Üks sellistest meetoditest on Lucas-Kanade meetod, mida rakendati tehnilise lahenduse juures kasutades OpenCV'd. Algselt tuvastatakse videos nurgad ning seejärel hakatakse tuvastatud punkte jälgima- iga eelnev punkt antakse edasi järgmisesse kaadrisse ning nii tekivad jooned, mis näitavad objekti liikumise trajektoori. Iga objekti joone jaoks valitakse erinev värv [10].

Lucas-Kanade meetodit videole rakendades oli näha, et väga tihti ei suudetud iga auto puhul nurkasid tuvastada ning seetõttu jäi mitmel autol liikumistrajektor tuvastamata (Joonis 13). Samuti jäävad alles eelmiste fooritsüklite käigus tekkinud jooned, mis tekitavad segadust [10].



Joonis 13. Lucas-Kanade meetod. On näha, et parasjagu ristmikule sõitnud autod on jäänud tuvastamata.

Teise variandina katsetati *dense optical flow* meetodit. *Dense optical flow* erineb Lucas-Kanade meetodist selle poolest, et objektide liikumise jälgimiseks ei tuvastata eelnevalt objektidel nurki, vaid jälgitakse terve kaadri iga punkti liikumist. Objektid värvitakse sõltuvalt liikumissuunast ehk kõik samas suunas liikuvad objektid on sama värvi (Joonis 14). Liikumatu taust värvitakse mustaks. Nii on liikuv objekt paremini eristatav. Samuti suudab meetod tuvastada iga liikuva auto ning on seega antud juhul tunduvalt usaldusväärsem [10].

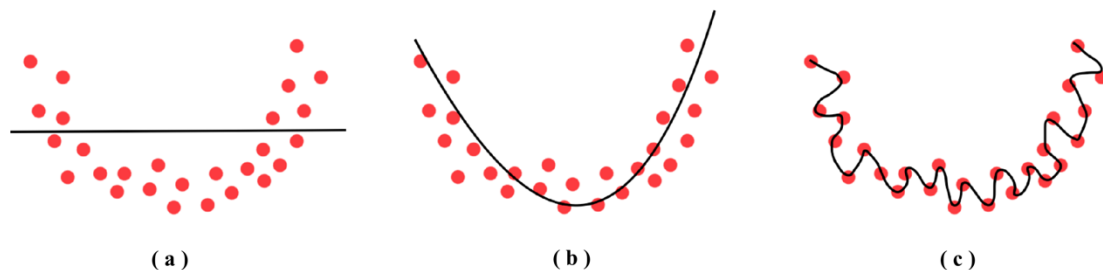


Joonis 14. *Dense optical flow* meetod. Kõik ristmikule sõitvad autod on tuvastatud.

4.2 Mudelite parameetrite valik ning modifitseerimine

Närvivõrgu ehitamisel on vaja paika panna erinevad parameetrid nagu näiteks kihtide arv, võrgusõlmede arv, iteratsioonide arv, õppimismäär. Parameetrite valikuks ei ole olemas kindlat valemit ning parim kombinatsioon leitakse katsetamise teel, kuid on teatud tunnused, mille järgi saab tuvastada, kuidas mudeli parameetreid modifitseerida tuleks ning üldist tulemust parandada saaks.

Kui närvivõrk on etteantud andmestiku peal töö lõpetanud, on võimalik välja arvutada kaks näitajat – treeningandmestiku klassifitseerimise täpsus ning testandmestiku klassifitseerimise täpsus. Esimene näitab, kui hästi õnnestub närvivõrku treenida, ning teine, kui hästi töötab treenitud närvivõrk tundmatute andmete peal. Täpsus arvutatakse jagades õigesti ennustatud korrad kogu ennustuste arvuga. Protsentide põhjal on võimalik tuvastada olukordi, kui närvivõrk on liiga üldine (*high bias, underfitting*), liiga kohandunud (*high variance, overfitting*) või täpselt paras. Kui võtta näiteks närvivõrk, mille ülesanne on leida funktsioon, mis tuvastaks punktikogumiku üldise kuju, siis näeks tulemus erinevatel juhtudel välja järgmine:



Joonis 15. Erinevad tulemused: (a) liiga üldine, (b) täpselt paras, (c) liiga kohandunud. Despois joonise põhjal [11].

4.2.1 Liiga üldine mudel

Liiga üldist tulemust on võimalik tuvastada siis, kui klassifitseerimise täpsus treeningandmetel on väga madal. Madal täpsus annab märku sellest, et närvivõrk määrab treeningandmete sobitamiseks liiga lihtsa funktsiooni ning nii klassifitseeritakse võrdlemisi suur osa andmetest valesti [11].

Sellisel juhul aitab närvivõrgule kihtide lisamine, peidetud kihtidele lisasõlmede lisamine ning rohkemate iteratsioonidega treenimine. Samuti võib katsetada mõnda teist närvivõrgu tüüpi [11].

4.2.2 Liiga kohandunud mudel

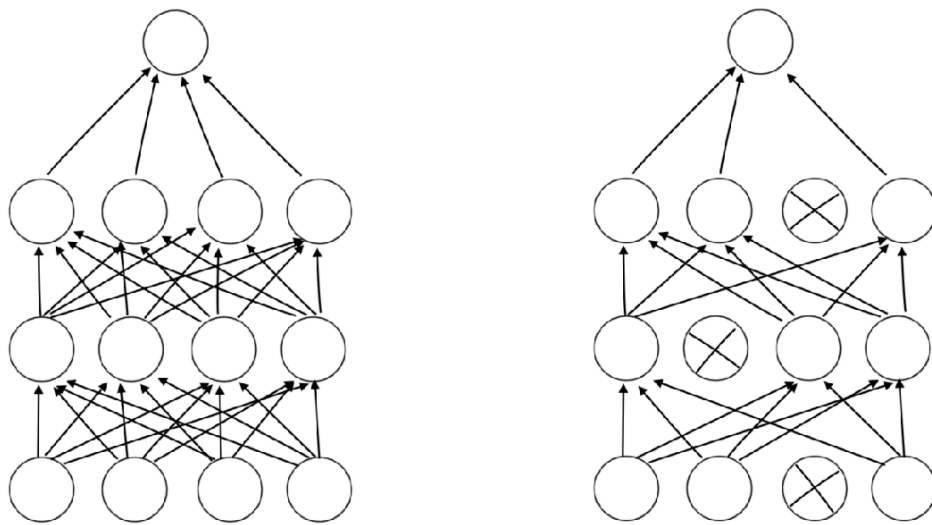
Liiga kohandunud närvivõrguga võib tegemist olla siis, kui mudel saab treeningandmete peal väga hea tulemuse, kuid testandmete peal täpsus kukub järsult. Sellisel juhul kohandub mudel täpselt etteantud treeningandmete järgi ja jätab meelde, mis väärtus iga treeningandme korral õige on. Kui nüüd anda ette testandmestik, ei tööta mudel enam hästi, sest treenides ei töötatud välja üldist mudelit, mis ka võõra andmestikuga ka hakkama saaks [11].

Sellisel juhul aitab näiteks närvivõrgu reguleerimine. Reguleerimise idee seisneb selles, et kulufunktsiooni (*cost function*) arvutamise valemile lisatakse piirang, mis ei lase närvivõrgu kaaludel liiga suureks minna ning sunnib närvivõrku kompromisse tegema. Nii saavutatakse üldisem mudel, mis suudab ka uue andmestiku peal hästi hakkama saada. Üks populaarsemaid meetodeid selleks on L2 piirang [11]. Sellisel juhul lisatakse kulufunktsioonile järgmine avaldis:

$$\lambda \sum_{j=0}^M W_j^2 \quad (3)$$

Valemis tähistab lambda reguleerimise parameetrit, mille parim väärtus võib iga närvivõrgu puhul erinev olla ning mis leitakse katsetamise teel [11].

Samuti kasutatakse võrgusõlmede väljalülitamise meetodit (*dropout*, *dropconnect*), kui iga treeningiteratsiooni käigus muudetakse mõned närvivõrgu sõlmed või nende ühendused mitteaktiivseks. Võrgusõlmede välja lülitamine toimub juhuslikult – selleks kasutatakse parameetrit, mis väljendab iga võrgusõlme alles jäämise tõenäosust. Võrgusõlmede juhuslikul välja lülitamisel ei saa närvivõrk enam kindlatele võrgusõlmedele kindla omadusvektori tuvastamise ülesannet määrata, sest antud võrgusõlme ei pruugi igas iteratsioonis olemas olla [11], [12]. Joonisel 16 on vasakul pool illustratsioon täielikult ühendatud närvivõrgu toimimisest kõikide võrgusõlmede olemasolul ning paremal näide sellest, kuidas võrk toimib, kui mõned sõlmed on välja lülitatud.



Joonis 16. Närvivõrk enne ja peale suvaliste võrgusõlmede eemaldamist. Despois joonise põhjal [11].

Kolmas viis liiga kohandunud mudelit paremaks muuta, on rohkem treeningandmeid lisades. Mudel ei suuda lõpmatu hulga andmestiku kohta infot hoida ning seetõttu sunnib suur andmekogus mudelit jällegi kompromisse tegema.

Mõningatel juhtudel ei pruugi lisaandmete kogumine võimalik olla või osutub väga kalliks. Sellisel juhul on võimalik olemasolevast andmestikust uusi variante genereerida. Olgu tegemist näiteks mingisuguse pildituvastuse probleemiga. Ühest

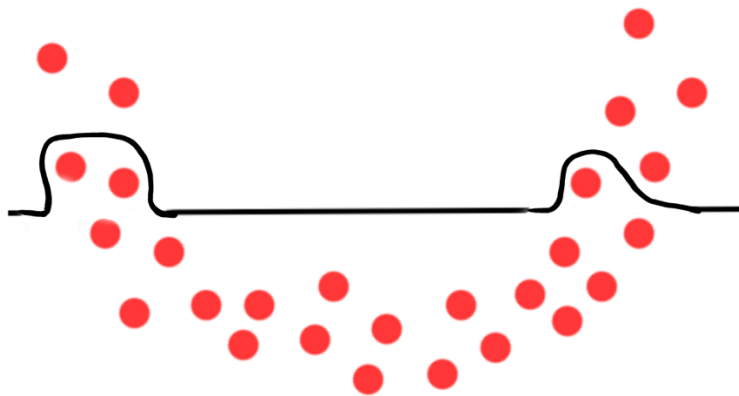
pildist saaks genereerida peegelpildi, suurendatud versiooni, mingi nurga võrra keeratud versiooni, pilditöötlusega manipuleeritud versiooni jne. Kuigi sellisel viisil saadud sisendid ei ole samaväärsed päris uute andmetega, muutub andmestik sellegipoolest veidi suuremaks ja mitmekülgsemaks [11], [13]. Joonis 17 illustreerib, kuidas saaks ühte pilti manipuleerides tekitada juurde uusi sisendeid.



Joonis 17. Erinevad variatsioonid samast pildist [11].

4.2.3 Kombinatsioonid

Liiga üldine ning liiga kohandunud mudel võivad esineda ka koos ning ei ole üksteist välistavad. Ülal mainitud punktikogumiku ülesande puhul võiks mõlema tunnuse koosinemine välja näha midagi sarnast nagu alloleval joonisel kujutatud: suuremal osal pildil on rakendatud lineaarset funktsiooni, kuid teatud punktide puhul teeb funktsioon ebamääraseid kõrvalekaldeid. Tulemuseks on halvasti töötav mudel, mis ei suuda hästi toimida ei treening- ega testandmetel [13].



Joonis 18. Tulemus, kus on näha nii liiga üldise kui ka liiga kohandunud mudeli tunnusjooni.

4.3 Täielikult ühendatud närvivõrk

Esimesena võeti vaatluse alla tavaline täielikult ühendatud närvivõrk. Närvivõrk koosneb sisendikihist, peidetud kihtidest ning väljundkihist. Kihid koosnevad neuronitest ehk võrgusõlmedest (*neuron, node*), mis saavad sisendi eelmiste kihi sõlmedest ning arvutavad väljundi. Igal võrgusõlmel on oma kaal, mis näitab selle tähtsust võrreldes teiste sõlmedega. Funktsiooni, millega sõlmes väljund arvutatakse, nimetatakse aktivatsioonifunktsiooniks (*activation function*) ning see peab olema mittelineaarne. Käesolevas töös valiti närvivõrgu keskmiste kihtide jaoks ReLU funktsioon (*rectified linear unit*) ning väljundsõlme jaoks sigmoidfunktsioon. Väljundsõlme jaoks valiti erinev funktsioon sellepärast, et sigmoidfunktsiooni väljundiks on väärtus 0 ja 1 vahel, mis sobib hästi just binaarse väljundiga probleemi lahendamiseks [14], [15].

Lisaks aktivatsioonifunktsioonile tuleb närvivõrgu puhul määrata peidetud kihtide arv, iga kihi võrgusõlmede arv, treenimise iteratsioonide arv ning õppimismäär (*learning rate*). Kuna parameetrite valiku puhul ei ole kindlat valemit, leitakse sobiv parameetrite komplekt katsetamise teel.

Sisenditest katsetati esimesena 13-kihilist maatriksit, kus üks kihtidest on töötlemata kaader ning töötlemise käigus saadud *optical flow* kaadrid alates neli sekundit enne tsükli lõppu, igast sekundist 3 kaadrit. Kaadreid kasutati must-valgel kujul (*grayscale*). Kaadrid on suuruses 64x64, seega ühe sisendi dimensionaalsus on $64 \cdot 64 \cdot 13 = 53248$. Järgnevas tabelis on välja toodud kolm parimat tulemust, mis erinevate parameetritega saadi. Rohkemate tulemuste jaoks vt Lisa 1.

Tabel 2. Täielikult ühendatud närvivõrgu tulemused 13-kihlise sisendmaatriksi peal.

Kihtide arv	Iga kihi neuronite arv	Iteratsioonide arv	Õppimismäär	Treeningu täpsus	Testi täpsus
5	60, 27, 14, 3, 1	2000	0.01	0.99999	0.77613
4	10, 7, 4, 1	1000	0.01	0.99999	0.76315
2	10, 1	1000	0.08	0.99999	0.72368

Võrreldes treening- ja testandmestike tulemusi, on selgelt näha, et igal juhul on tegemist liiga kohandunud mudeliga, sest kahe tulemuse vahe on ligi 20%. Et liigset

kohandumist vältida, tuleb kasutusele võtta mõni ülalmainitud meetmetest. Praeguse probleemi puhul ei ole võimalik andmeid tehiskult juurde tekitada, sest videos on ainult kindel hulk momente, kui autod uuritavas sõidusuunas liiguvad. Samuti on oluline, et jäädvustus oleks tehtud just neli sekundit enne tsükli lõppu, sest suvalisel hetkel ei ole võimalik kindlalt tuvastada, kas tegemist oleks kinnisõidetud või mitte kinnisõidetud momendiga ning nii võiks palju valeandmeid tekkida. Seepärast võetakse reguleerimiseks kasutusele kõige laialtlevinum L2 piirang. Lisandub veel üks parameeter – L2 piirangu reguleerimisparameeter lambda. Järgnevas tabelis on näha parimad tulemused. Rohkemade tulemuste jaoks vt Lisa 2.

Tabel 3. Täielikult ühendatud närvivõrgu tulemused 13-kihlise sisendmaatriksi peal koos L2 piiranguga.

Kihtide arv	Iga peidetud kihi neuronite arv	Iteratsioonide arv	Õppimis-määr	Reg. parameeter lambda	Treeningu täpsus	Testi täpsus
4	10, 7, 4, 1	1500	0.08	0.03	0.99999	0.73684
4	10, 7, 4, 1	1500	0.05	0.08	0.99999	0.77631
3	17, 10, 1	1500	0.04	0.035	0.99999	0.76315

Tabelist on näha, et mudeli tulemust suudeti väga vähe mõjutada ning üldpildis ei ole suurt muutust näha. See võib olla tingitud sellest, et nii suure dimensionaalsusega sisendite puhul oleks liigse kohandumise vältimiseks vaja kordades suuremat andmestikku.

Et dimensionaalsust vähendada, koostati uus andmestik, kus sisendiks on kahekihiline maatriks, millest üks kiht on jällegi töötlemata kaader ning teine kiht ainult üks *optical flow* kaader, mis on võetud samast hetkest kui tavakaader. Ühe sisendi dimensionaalsus on $64 \times 64 \times 2 = 8192$. Järgnevas tabelis on välja toodud parimad tulemused. Rohkemade tulemuste jaoks vt Lisa 3.

Tabel 4. Täielikult ühendatud närvivõrgu tulemused 2-kihilise sisendmaatriksi peal koos L2 piiranguga.

Kihtide arv	Iga peidetud kihi neuronite arv	Iteratsioonide arv	Õppimis-määr	Reg. parameeter lambda	Treeningu täpsus	Testi täpsus
4	10, 7, 4, 1	3000	0.08	0.1	0.99999	0.72368
3	10, 5, 1	1000	0.01	0.8	0.86160	0.72368
2	10, 1	1000	0.01	0.5	0.87946	0.67105

On näha, et sisendi dimensionaalsuse vähenedes reageerib mudel parameetri muutustele palju selgemini. Kuna on võimalik, et tegelikult ei olegi närvivõrgul *optical flow* kaadrist kasu, katsetatakse sisendina ka ainult töötlemata värvilist kaadrit. See tähendab, et sisendiks on 3-kihiline maatriks, sest värviline pilt koosneb kolmest kihist (RGB). Järgnevas tabelis välja toodud parimad tulemused. Rohkemate tulemuste jaoks vt Lisa 4.

Tabel 5. Täielikult ühendatud närvivõrgu tulemused 3-kihilise RGB töötlemata kaadri peal L2 piiranguga.

Kihtide arv	Iga peidetud kihi neuronite arv	Iteratsioonide arv	Õppimis-määr	Reg. parameeter lambda	Treeningu täpsus	Testi täpsus
4	10, 7, 4, 1	3000	0.01	0.001	0.92307	0.75641
4	40, 25, 7, 1	3000	0.01	0.09	0.87499	0.73076
4	60, 25, 7, 1	1800	0.05	0.08	0.90865	0.75641

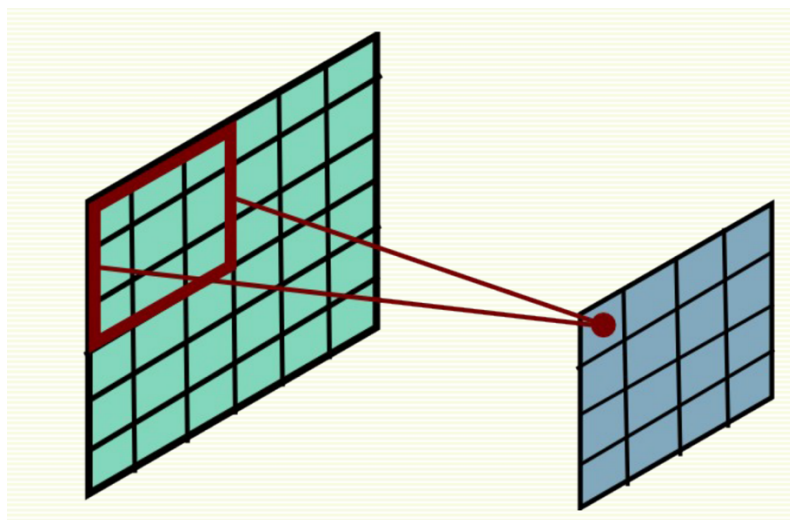
On näha, et ainult töötlemata kaadri tulemus on veidi parem kui koos *optical flow*'ga. Kuna katsetatud andmetike sisendid on võrdlemisi suured ning vajalike omaduste kätte saamine võib olla raskendatud, katsetatakse kõiki eelnevaid andmetikke ka konvolutsioonilise närvivõrgu peal.

4.4 Konvolutsiooniline närvivõrk

Konvolutsiooniline närvivõrk on närvivõrgu tüüp, mida kasutatakse palju piltide tuvastamise ja klassifitseerimise probleemide puhul. Tavalisest täielikult ühendatud

närvivõrgust erineb konvolutsiooniline selle poolest, et sisenditele rakendatakse kõigepealt omaduste eraldamist [7].

Omaduste eraldamiseks kasutatakse filtrit ehk $N \times N$ maatriksit ning sisendpildi piksliväärtusi sisaldavat maatriksit. Filter paigutatakse sisendpildi ülemisse vasakusse nurka ning teostatakse filtri alla jääva sisendpildi ning filtri vahel elemendi kaupa korrutis, mille tulemus läheb vastavalt väljundmaatriksi ülemisse vasakusse nurka (Joonis 19). Filtrit liigutatakse järjest edasi kuni terve sisendpildi maatriks on läbi käidud. Tulemuseks saadud maatriksit nimetatakse omaduskaardiks (*Feature Map*) [16], [17].



Joonis 19. Sisendpildi ja filtri vaheline korrutis. Tulemuse paigutus väljundmaatriksis tuleneb filtri paigutusest sisendpildil [17].

Järgmisena rakendatakse funktsiooni, mis muudab närvivõrgu mittelineaarseks ning seejärel viiakse omaduskaardil läbi ahendamine (*pooling*), mille eesmärgiks on omaduskaardi dimensioone vähendada nii, et jääks alles kõige olulisem informatsioon. Sõltuvalt soovitud lõppmaatriksi suurusest jaotatakse omaduskaart väiksemateks osadeks ning igast piirkonnast läheb väljundisse näiteks suurim arv või piirkonna keskmine (sõltub valitud ahendamise tüübist). Ahendamisest saadud maatriksist saab nüüd tavalise täielikult ühendatud närvivõrgu sisend [16]. Omaduste eraldamise eeliseks on see, et täielikult ühendatud närvivõrgu sisend on võrreldes esialgsuga väiksem ning toob välja kõige dominantsemad omadused [16], [18].

Konvolutsioonilise mudeli optimeerimiseks jaotati andmed väiksemateks osadeks ehk minipartiideks (*mini-batch*). Minipartiide eeliseks on see, et iga kord kulufunktsiooni arvutades ei ole vaja enam eelnevalt tervet andmestikku läbi töödelda, vaid saab juba

väiksemate osade pealt ennustama hakata. Sobiva minipartii suuruse osas on vastakaid arvamusi. Käesoleva töö aluseks võeti Luschi ja Masters'i artikkel, mis on üks värskemaid sellel teemal ning mille kohaselt on kõige stabiilsemalt parim tulemus minipartiidega, mille suurusvahemik jääb 2 ja 32 vahele [19].

Konvolutsioonilise närvivõrgu sisenditena kasutati samu ülalkirjeldatud andmestikke. Järgnevas tabelis välja toodud parimad tulemused 13-kihilise maatriksi peal. Rohkemate tulemuste jaoks vt Lisa 5.

Tabel 6. Konvolutsioonilise närvivõrgu tulemused 13-kihilise sisendmaatriksi peal.

Iteratsioonide arv	Õppimismäär	Minipartii suurus	Omaduskaartide suurused	Treeningu täpsus	Testi täpsus
60	0.01	32	10, 5	1.0	0.78947
40	0.01	20	8, 4	1.0	0.75
60	0.03	20	10, 5	1.0	0.72368

Nagu tabelist näha, esineb 13-kihilise maatriksiga ka konvolutsioonilise närvivõrgu puhul liigne kohandumine. Liigse kohandumise vältimiseks rakendatakse 13-kihilise maatriksi peal samuti L2 piirangut. Järgnevas tabelis välja toodud parimad tulemused. Rohkemate tulemuste jaoks vt Lisa 6.

Tabel 7. Konvolutsioonilise närvivõrgu tulemused 13-kihilise sisendmaatriksi peal L2 piiranguga.

Iteratsioonide arv	Õppimis-määr	Minipartii suurus	Omadus-kaartide suurused	Reg. parameeter lambda	Treeningu täpsus	Testi täpsus
40	0.03	30	10, 5	0.08	0.98660	0.73684
60	0.03	30	10, 5	0.1	0.91517	0.72368
60	0.03	30	10, 5	0.08	0.87946	0.72368

Võrreldes eelmise mudeliga on näha, et konvolutsioonilise mudeli puhul on L2 piirangu mõju paremini näha. Sellegipoolest on mudel liiga kohandunud ning võetakse kasutusele 2-kihiline maatriks. Järgnevas tabelis välja toodud parimad tulemused. Rohkemate tulemuste jaoks vt Lisa 7.

Tabel 8. Konvolutsioonilise närvivõrgu tulemused 2-kihilise maatriksi peal L2 piiranguga.

Iteratsioonide arv	Õppimis-määr	Minipartii suurus	Omadus-kaartide suurused	Reg. parameeter lambda	Treeningu täpsus	Testi täpsus
60	0.05	20	10, 5	0.05	0.81696	0.71052
50	0.05	20	8, 4	0.09	0.78125	0.71052
40	0.05	20	8, 4	0.09	0.83928	0.68421

Võrreldes eelmise tabeliga, on näha, et 2-kihilise maatriksi puhul on sarnaselt esimesele mudelile L2 piirangu mõju tunduvalt suurem. Sisendina katsetatakse ka töötlemata RGB kaadrit ehk 3-kihilist maatriksit. Järgnevas tabelis välja toodud parimad tulemused. Rohkemate tulemuste jaoks vt Lisa 8.

Tabel 9. Konvolutsioonilise närvivõrgu tulemused 3-kihilise RGB töötlemata kaadri peal L2 piiranguga.

Iteratsioonide arv	Õppimis-määr	Minipartii suurus	Omadus-kaartide suurused	Reg. parameeter lambda	Treeningu täpsus	Testi täpsus
75	0.001	30	6, 3	0.0001	0.826923	0.7948718
75	0.001	30	6, 3	0.0003	0.8125	0.7948718
50	0.01	30	8, 4	0.001	0.80288	0.70512

Eelnevast tabelist on näha, et töötlemata kaadri peal andis L2 piirang väga häid tulemusi, sest treening- ning testandmete erinevus on vaid paar protsenti võrreldes esialgse 20%-ga. Kuigi see on juba üsna rahuldav tulemus, katsetatakse töötlemata kaadri peal L2 piirangu asemel ka võrgusõlmede väljajätmise meetodit. Reguleerimisparameetri lambda asemel on nüüd kasutuses parameeter, mis näitab tõenäosust, et iteratsiooni käigus võrgusõlm alles jäetakse. Järgnevas tabelis välja toodud parimad tulemused. Rohkemate tulemuste jaoks vt Lisa 9.

Tabel 10. Konvolutsioonilise närvivõrgu tulemused 3-kihlise RGB töötlemata kaadri peal võrgusõlmede väljalülitamisega.

Iteratsioonide arv	Õppimismäär	Minipartii suurus	Omaduskaartide suurus	Sõlme allesjäämise TN	Treeningu täpsus	Testi täpsus
125	0.0008	30	8, 4	0.9	0.80769	0.70512
105	0.0008	30	8, 4	0.9	0.79326	0.679487
75	0.001	30	6, 3	0.5	0.677884	0.641025

On näha, et võrgusõlmede väljajätmise meetodiga on võimalik samuti liigset kohandumist veidi parendada, kuid L2 piiranguga saadi paremad tulemused.

4.5 Närvivõrgu tulemuste valideerimine

Kõige lihtsam viis närvivõrgu tööd hinnata on klassifitseerimise täpsuse järgi ehk arvutada protsent, kui suure osa testandmetest suutis närvivõrk õigesti ennustada. See protsent võib aga vale informatsiooni anda.

Olgu meil näiteks mingisugune abstraktne 300 sisendiga andmestik, millest 220 juhul on õige vastus 0 ning 80 juhul õige vastus 1. Kui kasutaksime mudelit, mis ennustab iga sisendi puhul tulemuseks 0, saaksime mudeli klassifitseerimistäpsuseks $220 / 300 * 100\% = 73\%$, mis on üsnagi hea tulemus, kuid ilmselgelt ei ole tegemist hea mudeliga. Samamoodi võib hea täpsus tulla ka siis, kui tegemist on väga ebaühtlase testandmestikuga. Näiteks 99% andmetest on tüüpi 0 ja ainult 1% tüüpi 1- kui mudel ennustab alati 0, suudab ta 99% juhtudest õigesti ennustada. Seetõttu kasutatakse mudeli täpsemaks hindamiseks lisameetodeid.

Hea ülevaate saab, kui välja kirjutada närvivõrgu tehtud ennustused tabelina jaotades kõik ennustused positiivseteks ja negatiivseteks ning need omakorda kaheks vastavalt sellele, kas ennustus oli õige.

Tabel 11. Närvivõrgu tehtud ennustused tabelina. Brownlee põhjal [20]

	Positiivne	Negatiivne
Positiivne	Tõene positiivne	Väär positiivne
Negatiivne	Väär negatiivne	Tõene negatiivne

Tabeli põhjal saab veel omakorda välja arvutada erinevaid näitajad, mis aitavad närvivõrku hinnata.

Esitustäpsus (*precision*) on tõeste positiivsete ennustuste arv jagatud tõeste ja väärade positiivsete ennustuste arvuga. Teisisõnu väljendab esitustäpsus tõeste positiivsete ja kõikide tehtud positiivsete ennustuste suhet. Madal esitustäpsus võib viidata suurele arvule valepositiivsetele ennustustele [20].

Tundlikkus ehk tõeste positiivsete määr (*recall, sensitivity, true positive rate*) arvutatakse jagades tõeste positiivsete ennustuste arvu tõeste positiivsete ja väärnegatiivsete ennustuste summaga. Teisisõnu jagatakse õigesti tehtud positiivsete ennustuste arv kõikide positiivsete testandmete hulgaga. Madal tundlikkus võib viidata suurele arvule väärnegatiivsetele ennustustele [20].

F1 skoor (*F1 Score*) võtab arvesse kahte eelmist näitajat ja arvutatakse valemiga:

$$F1 = 2 * \frac{(esitustäpsus * tundlikkus)}{esitustäpsus + tundlikkus} \quad (4)$$

Võrreldes erinevate mudelite grupeeritud ennustustabeleid ning F1 skoori, on võimalik paremini hinnata, milline mudel töötab kõige paremini.

Täielikult ühendatud närvivõrgu puhul kasutatakse analüüsi jaoks töötlemata RGB kaadri ning L2 piiranguga saadud tulemust, kus treeningtäpsus on 87% ning testtäpsus 73%. Järgnevas tabelis on välja toodud mudeli ennustused grupeeritult.

Tabel 12. Täielikult ühendatud närvivõrgu ennustused grupeeritult.

	Tõesed positiivsed ennustused	Väärad positiivsed ennustused	Tõesed negatiivsed ennustused	Väärad negatiivsed ennustused
Treening	104	26	78	0
	Kokku: 130		Kokku: 78	
Test	35	18	22	3
	Kokku: 53		Kokku: 25	

Tabeli põhjal võib järeldada, et treening- ning testandmete klassifitseerimise täpsused andsid õiget informatsiooni, sest on näha, et mudel ei ennusta järjepidevalt ainult sama tulemust. Arvud toovad hästi esile, kuidas testandmete korral mudeli ennustamisvõime tunduvalt kahanes – kui treeningandmestiku peal moodustas näiteks väärade negatiivsete ennustuste arv kogu negatiivsete ennustuste arvust 0%, siis testandmestike peal oli see koguni 12%.

Konvolutsioonilise närvivõrgu analüüsi jaoks kasutatakse samuti töötlemata RGB kaadri ning L2 piiranguga saadud tulemust, mille treeningtäpsus on 81% ning testtäpsus 79%. Järgnevas tabelis on näha konvolutsioonilise mudeli ennustused grupeeritult.

Tabel 13. Konvolutsioonilise närvivõrgu ennustused grupeeritult.

	Tõesed positiivsed ennustused	Väärad positiivsed ennustused	Tõesed negatiivsed ennustused	Väärad negatiivsed ennustused
Treening	97	32	72	7
	Kokku: 129		Kokku: 79	
Test	35	13	27	3
	Kokku: 48		Kokku: 30	

Ka konvolutsioonilise närvivõrgu grupeeritud tulemustest on näha, et treening- ning testandmete klassifitseerimise tulemused olid korrektsed. Võrreldes täielikult ühendatud närvivõrguga on mudel treeningandmetel küll halvema tulemusega, kuid tunduvalt stabiilsem. Treeningandmestiku peal moodustavad väärad negatiivsed ennustused kogu negatiivsete ennustuste hulgast ~9% ning testandmestiku peal 10%.

Järgnevas tabelis on välja arvatud mõlema mudeli esitustäpsus, tõeste positiivsete määr ning F1 skoor nii treening- kui testandmestikel.

Tabel 14. Mudelite esitustäpsused, tõeste positiivsete määrad ning F1 skoorid.

	Esitus- täpsus treening	Tõeste pos. määr treening	F1 treening	Esitus- täpsus test	Tõeste pos. määr test	F1 test
Täielikult ühendatud närvivõrk	0,8	1	0,89	0,66	0,92	0,77
Konvolutsiooniline närvivõrk	0,75	0,93	0,83	0,73	0,92	0,81

Välja arvatud F1 skoorid kinnitavad veelkord grupeeritud ennustuste tabeli põhjal tehtud järeldusi. On selgelt näha, et täielikult ühendatud närvivõrk töötab treeningandmetel paremini kui konvolutsiooniline, kuid see-eest on väga ebastabiilne. Sellepärast sobib antud probleemi lahendamiseks paremini konvolutsiooniline närvivõrk, sest sellega suudeti saavutada ~80% ennustustäpsus ning stabiilne käitumine nii treening- kui testandmetel. Saavutatud tulemus on piisav prototüübi rakendamiseks.

4.6 Edasiarenduse võimalused

Saavutatud ennustustäpsust annaks veelgi parandada, kui järjepidevalt ristmikult infot juurde koguda, andmestikku suurendada ning mudelit treenida.

Kuigi antud töö käigus keskenduti ainult ühele kindlale ristmikule ja sõidusuunale, on liikluse paremaks toimimiseks vaja lahendust edasi arendada nii, et algoritm arvestaks ka teiste ristmikel olevate fooridega.

Seejärel saaks terviklikku lahendust rakendada järjest rohkematele ristmikele, mis on oluline selleks, et tekitada võimalikult pikk “roheline laine”. Võttes arvesse, et kõikidel ristmikel ei ole kaamerat, võiks arendada ka sellise lahenduse, mis ei kasutaks sisendina mitte videopilti, vaid eelneva ristmiku optimeeritud fooritsüklit ning suudaks selle põhjal ka käesoleva ristmiku tsüklit kohandada.

5 Kokkuvõte

Käesolev töö oli koostööprojekt Tallinna Linnavalitsusega, mille eesmärgiks oli arendada tehniline lahendus, mis suudaks ristmiku videokaadrit analüüsides ette ennustada ristmiku kinnisõitmisi. Lahendus suudab ennustada ette vähemalt neli sekundit, et oleks võimalik fooritsükli vastavalt optimeerida ning kinnisõitmisi vältida.

Analüüsi käigus leiti, et sellise probleemi lahendamiseks võiks kõige paremini sobida närvivõrk. Sarnaste tööde põhjal otsustati, et peamiseks mudeliks valitakse tavaline täielikult ühendatud närvivõrk.

Närvivõrgu sisendiks kasutatav andmestik on töö autori poolt koostatud. Andmestiku koostamiseks kasutati Tallinna Transpordiametilt saadud ristmiku videosalvestusi, millest lõigati välja vajalikud kaadrid. Pandi kokku kolm erinevat andmestikku, millest esimene koosneb 13-kaadrist, kus üks on tavaline töötlemata kaader, mis on võetud neli sekundit enne fooritsükli lõppu, ning 12 kaadrit alates neli sekundit enne fooritsükli lõppu kuni tsükli lõpuni, millele rakendati *optical flow* meetodit. Teine andmestik koosneb töötlemata kaadrist ning ainult ühest *optical flow* kaadrist ning kolmas koosneb ainult töötlemata kaadritest. Koostatud treeningandmestiku suuruseks oli 208 sisendit ning testandmestiku suuruseks 78 sisendit.

Täielikult ühendatud närvivõrgu puhul esines iga andmestikuga mudeli liigne kohanemine, mille parandamiseks kasutati L2 piirangut. Kuna piirangu rakendamisega ei suudetud tulemust piisavalt parandada, võeti kasutusele ka konvolutsiooniline närvivõrk. Konvolutsioonilise närvivõrgu puhul esines samuti liigset kohandumist, kuid reguleerimismeetmed mõjusid selle mudeli puhul paremini. Katsetati nii L2 piirangut kui võrgusõlmede väljajätmise meetodit. Leiti, et kõige täpsem tulemus saadi konvolutsioonilise närvivõrguga ainult töötlemata kaadritest koosneva andmestiku peal koos L2 piiranguga. Suudeti saavutada ~80% ennustustäpsus, mida valideeriti kasutades F1 skoori. Saavutatud tulemus on piisav prototüübi rakendamiseks.

Koostööprojekti järgmiseks etapiks on süsteemi rakendamine reaalaja videopildi põhjal, et pikema aja jooksul mudeli toimimist testida, analüüsida ning edasi arendada.

Kasutatud kirjandus

- [1] “Forward looking infrared,” *Wikipedia*. (2017).
- [2] E. I. Vlahogianni, J. C. Golias, and M. G. Karlaftis, “Short-term traffic forecasting: Overview of objectives and methods,” *Transp. Rev.*, vol. 24, no. 5, pp. 533–557 (2004).
- [3] M. A. Nielsen, “Neural Networks and Deep Learning,” (2015).
- [4] S.-Y. Yun, S. Namkoong, J.-H. Rho, S.-W. Shin, and J.-U. Choi, “A Performance evaluation of neural network models in traffic volume forecasting,” *Math. Comput. Model.*, vol. 27, no. 9, pp. 293–310 (1998).
- [5] H. Dia, “An object-oriented neural network approach to short-term traffic forecasting,” *Eur. J. Oper. Res.*, vol. 131, no. 2, pp. 253–261 (2001).
- [6] I. Kanellopoulos and G. G. Wilkinson, “Strategies and best practice for neural network image classification,” *Int. J. Remote Sens.*, vol. 18, no. 4, pp. 711–725 (1997).
- [7] Y. Bengio and Y. Lecun, “Convolutional Networks for Images, Speech, and Time-Series,” (1997).
- [8] “Berkeley Deep Drive.” [Online]. Saadaval: <http://bdd-data.berkeley.edu/>. (18.05.2018).
- [9] “The KITTI Vision Benchmark Suite.” [Online]. Saadaval: http://www.cvlibs.net/datasets/kitti/raw_data.php. (18.05.2018).
- [10] “OpenCV: Optical Flow.” [Online]. Saadaval: https://docs.opencv.org/3.3.1/d7/d8b/tutorial_py_lucas_kanade.html. (16.04.2018).
- [11] J. Despois, “Memorizing is not learning! — 6 tricks to prevent overfitting in machine learning,” *Hacker Noon*, 22.05.2018. [Online]. Saadaval: <https://hackernoon.com/memorizing-is-not-learning-6-tricks-to-prevent-overfitting-in-machine-learning-820b091dc42>. (23.04.2018).
- [12] “Dropout Regularization - deeplearning.ai,” *Coursera*. [Online]. Saadaval: <https://www.coursera.org/learn/deep-neural-network/lecture/eM33A/dropout-regularization>. (18.05.2018).
- [13] “Bias / Variance - deeplearning.ai,” *Coursera*. [Online]. Saadaval: <https://www.coursera.org/learn/deep-neural-network/lecture/ZhcII/bias-variance>. (18.05.2018).
- [14] ujjwalkarn, “A Quick Introduction to Neural Networks,” *the data science blog*, (2016).
- [15] “Activation functions - deeplearning.ai,” *Coursera*. [Online]. Saadaval: <https://www.coursera.org/learn/neural-networks-deep-learning/lecture/4dDC1/activation-functions>. (20.05.2018).
- [16] ujjwalkarn, “An Intuitive Explanation of Convolutional Neural Networks,” *the data science blog*, (2016).
- [17] M. D. Zeiler and R. Fergus, “Visualizing and Understanding Convolutional Networks,” in *Computer Vision – ECCV 2014*, 2014, pp. 818–833.
- [18] “Why Convolutions? - deeplearning.ai,” *Coursera*. [Online]. Saadaval: <https://www.coursera.org/learn/convolutional-neural-networks/lecture/Xv7B5/why-convolutions>. (18.05.2018).
- [19] D. Masters and C. Lusch, “Revisiting Small Batch Training for Deep Neural Networks,” *ArXiv180407612 Cs Stat*, (2018).
- [20] J. Brownlee, “Classification Accuracy is Not Enough: More Performance Measures You Can Use,” *Machine Learning Mastery*, (2014).

Lisa 1 – Täielikult ühendatud närvivõrgu tulemused 13-kihilise sisendmaatriksi peal

Kihtide arv	Iga kihi neuronite arv	Iteratsioonide arv	Õppimismäär	Treeningu täpsus	Testi täpsus
5	60, 27, 14, 3, 1	1000	0.03	0.99999	0.77631
		2000	0.03	0.99999	0.77631
		2000	0.01	0.99999	0.77613
4	10, 7, 4, 1	1000	0.01	0.99999	0.76315
		1000	0.1	0.56249	0.48684
		2500	0.1	0.56249	0.48684
		2500	0.035	0.99999	0.76315
		1000	0.035	0.99999	0.76315
3	7, 4, 1	1000	0.035	0.56249	0.48684
		2500	0.035	0.56249	0.48684
		1000	0.1	0.56249	0.48684
		3000	0.04	0.56249	0.48684
3	17, 10, 1	1500	0.04	0.99999	0.76315
		1000	0.01	0.99999	0.76315
		1000	0.05	0.99999	0.76315
		1000	0.1	0.56249	0.48684

Lisa 2 – Täielikult ühendatud närvivõrgu tulemused 13-kihilise sisendmaatriksi peal L2 piiranguga

Kihtide arv	Iga peidetud kihi neuronite arv	Iteratsioonide arv	Õppimismäär	Reg. parameeter lambda	Treeningu täpsus	Testi täpsus
4	10, 7, 4, 1	1500	0.08	0.03	0.99999	0.73684
		1500	0.08	0.08	0.56249	0.48684
		1500	0.05	0.08	0.99999	0.77631
		1500	0.08	0.05	0.56249	0.48684
3	17, 10, 1	1500	0.04	0.035	0.99999	0.76315
		1500	0.04	0.3	0.99999	0.76315
		1500	0.04	0.9	0.99999	0.76315
2	10, 1	1500	0.04	0.01	0.99999	0.77631
		1000	0.04	0.01	0.99999	0.76315
		1000	0.04	0.1	0.99999	0.77631
		1000	0.08	0.1	0.99999	0.75000
2	5, 1	1000	0.04	0.1	0.99999	0.76315
		1000	0.4	0.1	0.56249	0.48684
		1000	0.4	0.01	0.56249	0.48684
		1000	0.9	0.01	0.56249	0.48684
		2000	0.9	0.01	0.56249	0.48684

Lisa 3 – Täielikult ühendatud närvivõrgu tulemused 2-kihilise sisendmaatriksi peal L2 piiranguga

Kihtide arv	Iga peidetud kihi neuronite arv	Iteratsioonide arv	Õppimismäär	Reg. parameeter lambda	Treeningu täpsus	Testi täpsus
4	10, 7, 4, 1	1000	0.08	0.1	0.99999	0.69736
		1000	0.1	0.1	0.99999	0.67105
		2000	0.08	0.1	0.99999	0.71052
		3000	0.08	0.1	0.99999	0.72368
3	17, 10, 1	1000	0.08	0.1	0.99999	0.68421
		1000	0.08	0.05	0.99999	0.68421
		2000	0.08	0.05	0.99999	0.69736
		2000	0.5	0.05	0.99999	0.68421
3	10, 5, 1	2000	0.5	0.05	0.99999	0.68421
		2000	0.1	0.05	0.99999	0.69736
		2000	0.1	0.001	0.99999	0.69736
		1000	0.01	0.5	0.86160	0.71052
		1000	0.01	0.8	0.86160	0.72368
2	10, 1	2000	0.01	0.01	0.97767	0.68421
		2000	0.01	0.08	0.97767	0.68421
		2000	0.01	0.5	0.96874	0.68421
		1000	0.01	0.5	0.87946	0.67105

Lisa 4 – Täielikult ühendatud närvivõrgu tulemused 3-kihilise RGB töötlemata kaadri peal L2 piiranguga

Kihtide arv	Iga peidetud kihi neuronite arv	Iteratsioonide arv	Õppimismäär	Reg. parameeter lambda	Treeningu täpsus	Testi täpsus
4	10, 7, 4, 1	1500	0.01	0.01	0.50000	0.48717
		2500	0.01	0.001	0.74038	0.67948
		3000	0.01	0.001	0.92307	0.75641
4	40, 25, 7, 1	2000	0.01	0.07	0.88461	0.67948
		2500	0.01	0.07	0.95673	0.73076
		2500	0.01	0.09	0.68269	0.53846
		3000	0.01	0.09	0.87499	0.73076
4	60, 25, 7, 1	3000	0.01	0.08	0.98076	0.75641
		2500	0.05	0.08	0.86057	0.71794
		2500	0.05	0.08	0.86057	0.71794
		1500	0.05	0.08	0.53846	0.50000
		1800	0.05	0.1	0.84615	0.66666
		1800	0.05	0.08	0.90865	0.75641
3	25, 7, 1	1500	0.07	0.01	0.84615	0.65384
		1500	0.01	0.01	0.84134	0.70512
		1500	0.03	0.01	0.86057	0.66666
		2000	0.03	0.3	0.66826	0.62820
		2000	0.03	0.1	0.89423	0.67948
		2000	0.03	0.07	0.97115	0.74358
3	10, 7, 1	1800	0.1	0.1	0.87499	0.66666
		2200	0.01	0.5	0.89903	0.75641
		2200	0.01	0.3	0.971153	0.74358

Lisa 5 – Konvolutsioonilise närvivõrgu tulemused 13-kihilise sisendmaatriksi peal

iteratsioonide arv	Õppimismäär	Minipartii suurus	Omaduskaartide suurused	Treeningu täpsus	Testi täpsus
40	0.01	20	8, 4	1.0	0.75
40	0.05	20	8, 4	0.5625	0.48684
40	0.01	30	8, 4	1.0	0.69736
40	0.01	32	8, 4	1.0	0.69736
60	0.01	32	8, 4	1.0	0.72368
60	0.01	32	10, 5	1.0	0.78947
60	0.01	20	10, 5	1.0	0.75
60	0.03	20	10, 5	1.0	0.72368
60	0.05	20	10, 5	0.57142	0.48684
50	0.05	20	10, 5	0.57142	0.48684
50	0.01	20	10, 5	1.0	0.75

Lisa 6 – Konvolutsioonilise närvivõrgu tulemused 13-kihilise sisendmaatriksi peal L2 piiranguga

iteratsioonide arv	Õppimismäär	Minipartii suurus	Omaduskaartide suurused	Reg. parameeter lambda	Treeningu täpsus	Testi täpsus
60	0.01	20	10, 5	0.003	0.1	0.72368
60	0.01	20	10, 5	0.03	0.1	0.68421
60	0.03	20	10, 5	0.03	0.5625	0.48684
60	0.03	30	10, 5	0.03	0.84821	0.63157
60	0.03	30	10, 5	0.08	0.87946	0.72368
60	0.03	30	10, 5	0.1	0.91517	0.72368
40	0.03	30	10, 5	0.08	0.98660	0.73684
40	0.03	20	10, 5	0.08	0.94642	0.69736
40	0.03	20	8, 4	0.08	0.97767	0.69736
50	0.03	20	8, 4	0.08	1.0	0.73684
50	0.05	20	8, 4	0.08	0.41964	0.36842
60	0.05	20	8, 4	0.08	0.65178	0.5
60	0.05	30	8, 4	0.08	0.5625	0.48684

Lisa 7 – Konvolutsioonilise närvivõrgu tulemused 2-kihilise sisendmaatriksi peal L2 piiranguga

iteratsioonide arv	Õppimismäär	Minipartii suurus	Omaduskaartide suurused	Reg. parameeter lambda	Treeningu täpsus	Testi täpsus
60	0.05	30	8, 4	0.08	0.95535	0.61842
60	0.05	20	8, 4	0.08	0.92857	0.72368
60	0.05	20	8, 4	0.09	0.79910	0.65789
50	0.05	20	8, 4	0.09	0.78125	0.71052
40	0.05	20	8, 4	0.09	0.83928	0.68421
50	0.01	20	8, 4	0.09	1.0	0.65789
50	0.07	20	8, 4	0.09	0.5625	0.48684
50	0.05	10	8, 4	0.09	0.72767	0.61842
50	0.05	30	8, 4	0.09	0.97767	0.68421
50	0.05	30	10, 5	0.09	0.93303	0.67105
50	0.05	20	10, 5	0.05	0.83035	0.64473
60	0.05	20	10, 5	0.05	0.81696	0.71052
70	0.05	20	10, 5	0.05	0.82589	0.65789
80	0.05	20	10, 5	0.05	0.875	0.69736
80	0.05	25	10, 5	0.05	0.91964	0.60526

Lisa 8 – Konvolutsioonilise närvivõrgu tulemused 3-kihilise RGB sisendmaatriksi peal L2 piiranguga

iteratsioonide arv	Õppimismäär	Minipartii suurus	Omaduskaartide suurused	Reg. parameeter lambda	Treeningu täpsus	Testi täpsus
80	0.05	25	8, 4	0.05	0.5	0.51282
60	0.01	25	8, 4	0.05	0.5	0.48717
60	0.01	20	8, 4	0.01	0.5	0.48717
40	0.01	20	8, 4	0.01	0.5	0.51282
40	0.01	30	8, 4	0.001	0.76923	0.67948
50	0.01	30	8, 4	0.001	0.80288	0.70512
60	0.01	30	8, 4	0.001	0.8125	0.73076
70	0.01	30	8, 4	0.001	0.87019	0.73076
70	0.01	29	8, 4	0.001	0.89903	0.67948
70	0.01	30	10, 5	0.001	0.5	0.51282
40	0.01	30	8, 4	0.01	0.67788	0.57692
40	0.01	30	10, 5	0.01	0.69711	0.64102
60	0.01	30	10, 5	0.01	0.80288	0.60256
70	0.01	30	10, 5	0.01	0.82211	0.64102
70	0.01	30	10, 5	0.0003	0.89423	0.65384
70	0.001	30	10, 5	0.0003	0.83173	0.70512
80	0.001	28	10, 5	0.0003	0.87019	0.71794
50	0.001	30	6, 3	0.0003	0.8125	0.75641
60	0.001	30	6, 3	0.0003	0.80288	0.75641
70	0.001	30	6, 3	0.0003	0.82211	0.79487
75	0.001	30	6, 3	0.0003	0.8125	0.79487
75	0.001	30	6, 3	0.0001	0.82692	0.79487

Lisa 9 – Konvolutsioonilise närvivõrgu tulemused 3-kihilise RGB töötlemata kaadri peal võrgusõlmede väljalülitamisega

iteratsioonide arv	Õppimismäär	Minipartii suurus	Omaduskaartide suurused	Sõlme alles-jäämise TN	Treeningu täpsus	Testi täpsus
74	0.001	30	6, 3	0.4	0.69711	0.51282
75	0.001	30	6, 3	0.5	0.67788	0.64102
75	0.001	30	6, 3	0.6	0.74519	0.65384
75	0.001	30	6, 3	0.7	0.73557	0.66666
75	0.001	30	6, 3	0.8	0.73076	0.69230
75	0.001	30	6, 3	0.9	0.79807	0.67948
85	0.001	30	6, 3	0.9	0.84615	0.74358
85	0.001	30	6, 3	0.8	0.82692	0.69230
85	0.001	30	8, 4	0.8	0.76923	0.58974
85	0.001	30	8, 4	0.9	0.76442	0.60256
85	0.0009	30	8, 4	0.9	0.75	0.61538
105	0.0008	30	8, 4	0.9	0.79326	0.67948
125	0.0008	30	8, 4	0.9	0.80769	0.70512
145	0.0008	30	8, 4	0.9	0.84134	0.64102
145	0.0008	30	10, 5	0.9	0.90865	0.75641
145	0.0008	30	10, 5	0.8	0.84134	0.70512
145	0.0008	30	10, 5	0.7	0.86538	0.70512
145	0.0008	30	10, 5	0.6	0.84615	0.73076
145	0.0008	30	10, 5	0.5	0.82692	0.69230