VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
KOMPIUTERIJOS KATEDRA

Magistro baigiamasis darbas

Kelių atpažinimas iš oro nufotografuotuose vaizduose

Atliko:
Artūras Grigorjevas parašas

Vadovas:
dr. Tadas Meškauskas

Vilnius
2018

Turinys

Santrauka

Šiame darbe yra išsikeliamas uždavinys kaip įmanoma tiksliau atpažinti kelius iš oro nufotografuotuose vaizduose. Tam tikslui pasiekti yra realizuojamas konvoliucinis neuronų tinklas ir darbo metu sukurti po-apdorojimo (angliškai *post-processing*) algoritmai.

Darbe yra atliekama 20 eksperimentų su konvoliuciniais neuronų tinklais. Kiekvieno eksperimento metu yra derinamas vis kitas konvoliucinio neuronų tinklo parametras. Šiais eksperimentais yra bandoma surasti tokią konvoliucinio tinklo architektūrą, kuri labiausiai tinka kelių atpažinimui. Atradus šią architektūrą, tinklo atpažinimo rezultatai yra papildomai apdorojami su siekiu, kad jie taps tikslesni. Galutiniams rezultatams yra atliekamas specialiai kelių atpažinimui išrastas kokybės įvertinimas.

Darbo algoritmai pasiekia aukštos kokybės rezultatus kaimo kelių analizėje. Miesto kelių atpažinimo rezultatai yra mažiau sėkmingi. Pagrindinė to priežastis yra negebėjimas korektiškai atpažinti uždaras kilpas keliuose.

Summary

Road Recognition in Aerial Images

Road recognition in aerial images is an important area of research, because having access to up-to-date road network data is essential in GIS applications and GPS navigation. Since manual road mapping is an expensive and tedious process, automated solutions are actively being studied and developed. Since road recognition is a specialized image classification task, in this paper we use a state of the art image classification tool - convolutional neural networks. In addition, we propose custom post-processing algorithms to futher improve the achieved results.

We conduct 20 wide-scale experiments related to convolutional neural networks. Each experiment attempts to adjust a single parameter in the network. The goal of these experiments is to design a convolutional neural network, whose architecture is best-suited for the task of recognizing roads in aerial images. After such architecture is achieved, the output of the network undergoes additional post-processing, such as cleaning of false positives and center pixel detection. Most of the post-processing algorithms used are custom-designed in this paper. We also make use of some well-known graph theory algorithms for additional post-processing. The final results of the recognition are then put through an evaluation algorithm, which is specifically tailored to measure the success of road reacognition tasks.

We test our method with two datasets - one containing images of urban roads, the other - of rural roads. We achieve a high accuracy result when testing with the rural data set. We achieve a less stellar result when testing with the urban data set - the most prevalent problem being the inability to properly detect closed loops in the road. The reason for this, is that urban images contain a lot more noise and occlusions, such as buildings, bgridges and cars. All of these factors impact the performance of our algorithms in urban environments.

Nevertheless, based on our results, we conclude that convolutional neural networks are indeed a suitable tool for road recognition tasks, especially in rural areas. We also conclude, that the problems in urban areas can be alleviated if another algorithm could detect loops in the road network and close them. To that end, we recommend looking into fully-convolutional networks.

Įvadas

Padėtimi pagrįstų sistemų plėtra pastaraisiais metais sparčiai auga. Tą galima pastebėti atkreipus dėmesį į geografinių informacinių sistemų paplitimą bei tyrimus, atliekamus autonominių automobilių navigacijos srityje. Tiksli, korektiška, ir naujausia informacija apie kelių tinklus yra privaloma siekiant užtikrinti sklandų minėtų technologijų funkcionavimą.

Vienas pagrindinių šaltinių, renkant duomenis apie kelius, yra iš oro nufotografuoti vaizdai - nuotraukos, atliktos palydovų arba nepilotuojamų orlaivių (angliškai *unmanned aerial vechicle*). Prasmingų duomenų išrinkimas iš tokių nuotraukų rankiniu būdu atima daug laiko ir yra gana brangus. Dėl to yra dedamos pastangos į efektyvių automatinių kelių atpažinimo algoritmų plėtrą.

Šiuo metu didelį susidomėjimą ir pasisekimą vaizdų atpažinimo uždaviniuose rodo konvoliucinių neuronų tinklų modeliai. Šie tinklai yra specializuoti sudėtingų objektų (pavyzdžiui, kačių, lėktuvų, mašinų) aptikimui nuotraukose. Tokie modeliai demonstruoja įspūdingus rezultatus įvairiuose konkursuose ir praktiniuose pritaikymuose. Šiame darbe įvairių eksperimentų pagalba yra bandoma sukurti tokį konvoliucinį neuronų tinklą, kurį naudojant būtų gaunami geriausi įmanomi kelių atpažinimo rezultatai. Šio tinklo rezultatas yra papildomai apdorojamas tam, kad atpažintas kelias būtų dar tikslesnis.

Pirmajame šio darbo skyriuje dalykinė sritis yra suskirstoma į kategorijas, bei yra pristatomos problemos, su kuriomis susiduriama bandant atlikti kelių atpažinimą. Taip pat yra atliekama susijusių darbų apžvalga. Antrajame skyriuje yra pristatomi besimokančių sistemų veikimo principai bei detaliai analizuojamas konvoliucinių neuronų tinklų teorinis pagrindas. Pirmieji du darbo skyriai sutelkia dėmesį į teorinę medžiagą ir yra paimti iš autoriaus mokslo tiriamojo darbo projekto [12]. Trečiajame skyriuje yra detalizuojama visa praktinė darbo dalis. Į ją įeina duomenų paruošimas, plataus masto eksperimentai kurių pagalba yra kuriamas optimalus konvoliucinis neuronų tinklas. Taip pat yra pristatomi autoriaus sukurti po-apdorojimo (angliškai *post-processing*) algoritmai. Galiausiai, darbo rezultatai yra įvertinami naudojant specializuotą kelių atpažinimo rezultatų vertinimo būdą.

Darbe sukurti algoritmai yra taikomi trijų tipų keliams - kaimo, miesto ir nežinomiems. Pagal savo kokybę, kelių atpažinimo rezultatai gali būti išrikiuoti tokia mažėjimo tvarka - kaimo, nežinomi, miesto.

1. Keliai iš oro nufotografuotuose vaizduose

Šiame skyriuje yra pristatoma darbo temos problematika. Iš oro nufotografuotų vaizdų dalykinė sritis yra suskirstoma į keletą kategorijų. To motyvacija yra ta, kad norint atlikti sėkmingą nuotraukų analizę reikia išmanyti vaizdus su kuriais bus dirbama, bei gerai suprasti ieškomų objektų charakteristikas. Skyriuje taip pat yra atliekama darbų, susijusių su kelių atpažinimu iš oro nufotografuotuose vaizduose, apžvalga.

1.1. Dalykinės srities analizė

Norint korektiškai atpažinti kelius vaizduose pirmiausia reikėtų suprasti kaip kelio fizinės savybės įtakoja vizualines charakteristikas, kadangi vaizdų analizė bus paremta būtent vizualinėmis charakteristikomis. Fizinių kelio savybių, kurios įtakoja vizualines savybes, aprašymas pateiktas [7] ir [11]:

1. Kelio paviršius yra tvirtas ir glotnus. Kadangi kelias gaminamas iš asfalto ar betono, jo spektrinės savybės nuotraukose gali sutapti su šių medžiagų savybėmis. Deja, praktikoje kelio pikselių intensyvumas nebūna pastovus, o svyruoja dėl kelio žymių, kelio ženklų, medžių ir jų šešėlių bei mašinų.

2. Kelio statumas būna apribotas, dėl to kalnuotose vietovėse keliai būna vingiuoti. Ši informacija negali būti lengvai gauta iš oro nufotografuotuose vaizduose ir, jei jos reikia, turi būti patiekta kitokiais 3D šaltiniais.

3. Kelio plotis būna pastovus ir priklauso nuo kelio svarbos. Pavyzdžiui, autostrada bus platesnė negu miesto gatvė ar kaimo kelias. Pločio informacija lengviau išgaunama iš aukštos rezoliucijos vaizdų.

4. Kelio kreivumas taip pat priklauso nuo svarbos. Kreivumas vienodai gerai gali būti išgautas tiek iš aukštos, tiek iš žemos rezoliucijos vaizdų.

5. Kelių ir jų jungiamų objektų tankumas gali skirtis pagal teritorijos tipą. Miesto keliai ir jų tinklai būna tankesni už kaimo kelių tinklus.

Pagrindinės problemos su kuriomis susiduriama automatiniame kelių atpažinime yra okliuzija (kelias uždengtas medžiu, namo šešėliu arba automobiliu), nevienodos kelių spalvos arba pločiai. Taip pat problemą pasunkina labiau komplikuotų kelio ypatybių (sankryžų, žiedų) buvimas. Atpažinimo rezultatus įtakoja ir nuotraukų ypatybės - prastos oro sąlygos fotografavimo metu, prastas spalvų kontrastas bei faktas, kad kai kurių objektų (pvz. namų stogų) spalva gali sutapti su kelio spalva.

Kelių atpažinimo vaizduose metodikos gali drastiškai skirtis priklausomai nuo to, su kokiais vaizdais yra dirbama. Dėl to yra pravartu suprasti, kaip keliai atrodo skirtingo tipo vaizduose. Literatūroje, o taip pat ir susijusių darbų rinkinyje [11], kelių vaizdai skirstomi į dvi kategorijas:

- Žemos rezoliucijos vaizdai. Tokiuose vaizduose viename pikselyje telpa daugiau nei 2 metrai vaizdo. Šiuose vaizduose keliai atrodo kaip plonos linijos.

- Aukštos rezoliucijos vaizdai. Vienas pikselis talpina nuo 0.5 iki 1 metro. Tokiuose vaizduose keliai atrodo kaip ištįsę plotai, turintys lygiagrečias briaunas.

Taip pat naudinga yra išskirti kelio tipą - ar tai miesto, ar kaimo kelias. Tai yra svarbu, kadangi kaimo keliuose būna mažiau okliuzijos - kelią gali uždengti tik medžiai. Miesto kelią gali uždengti daugiau objektų - medžių, mašinų, pastatų šešėliai, tiltai. Kaip atrodo skirtingų kategorijų ir tipų keliai iliustruoja 1 pav.

1 pav. Kelių tipai. Kairėje - žemos rezoliucijos, kaimo keliai. Dešinėje - aukštos rezoliucijos, miesto keliai.

1.2. Susijusių darbų apžvalga

Šiame darbe yra išsikeltas netrivialus uždavinys - kaip įmanoma geriau atpažinti kelių tinklą iš oro nufotografuotame vaizde. Prieš kuriant naują algoritmą, skirtą spręsti bet kokiai problemai, reikia pasidomėti apie jau egzistuojančius, tam pačiam uždaviniui skirtus, metodus. Dėl to, sekančiose pastraipose yra apžvelgiami moksliniai darbai, susiję su kelių atpažinimu iš oro nufotografuotuose vaizduose.

Ankstesniuose darbuose, parašytuose tuomet, kai vaizdo apdorojimo technologijos buvo mažiau pažengusios, kelių atpažinimas prasideda nuo kraštų aptikimo (angliškai *edge detection*). Tai matosi ir pirmajame darbe apie kelių atpažinimą [7]. Kraštų aptikimas yra pirmas žingsnis, kuris pats iš savęs nepagamina patenkinamų kelių atpažinimo rezultatų, kadangi keliai nuotraukose būna užgožti medžių, šešėlių arba mašinų. Vienas iš modelių okliuzijos problemai spręsti yra aktyvaus kontūro modelis, kitaip vadinamas gyvate. Pirmą kartą toks modelis pasiūlytas 1988m. [18]. Šis modelis buvo praplėstas kaspino gyvatėmis (angliškai *ribbon snakes*) 2000m. [8]. Kaspino gyvatės sugeba efektyviai išnaudoti mažą kiekį naudingos informacijos, kuri egzistuoja uždengtuose keliuose. Tačiau, šis būdas labiau tinka kaimo keliams, kuriuose pagrindinį uždengimą sukuria medžiai, o ne mašinos.

Naujesnis darbo su okliuzijomis sprendimas buvo pasiūlytas 2009m. [19]. Šiame darbe autoriai pristato centrinio pikselio ir remiamojo apskritimo (angliškai *reference circle*) sąvokas. Pirmas šio algoritmo žingsnis taip pat yra kraštų aptikimas. Nagrinėjant pikselį p, jo remiamasis apskritimas yra didžiausio spindulio apskritimas, kurio centras yra pikselis p ir kuriame nėra nė vieno kraštui priklausančio pikselio. Pikselis p yra laikomas centriniu, jei jo remiamasis apskritimas yra didesnis už visų kaimyninių pikselių remiamuosius apskritimus tam tikrame paieškos plote

(pvz. 3×3). Naudojantis šiomis sąvokomis algoritmas atlieka keletą etapų, kurių išvedimas yra atpažintas vaizdo kelių tinklas.

Algoritmas, pristatytas [19] viename iš savo etapų naudoja spalvų spektrinę informaciją tam, kad patikrintų ar atskiri segmentai yra to pačio kelio dalis. Klasikiniai kelių atpažinimo algoritmai dirbdavo su nespalvotais vaizdais. Spalvinė informacija gali suteikti daug papildomos naudos bandant sugrupuoti atskirus kelių segmentus į vieną tinklą. Kai spalva yra naudojama kažkokiuose skaičiavimuose, standartiškai ji yra modeliuojama RGB erdvėje. Tačiau, RGB erdvės skirtumai tarp gretimų spalvų gali būti per dideli. Dėl to kyla poreikis transformuoti spalvą į kitokią erdvę. Viena tokių alternatyvių erdvių yra CIELAB erdvė, kuri kelių atpažinimo tikslams buvo panaudota darbe [21]. Ši erdvė yra daug didesnė nei RGB erdvė. [21] darbas parodo spalvinės informacijos svarbą kelių atpažinimo uždaviniui. Minėto darbo algoritmas nėra visiškai automatizuotas, kadangi pradiniame jo žingsnyje yra rankiniu būdu parenkamas kelio regionas, pagal kurį bus ieškoma likusių segmentų nuotraukoje.

Pastaraisiais metais atsiranda vis daugiau kelių atpažinimo algoritmų, kurie neapsiriboja vien tik rastriniais duomenimis iš vaizdų. Šie metodai stengiasi išnaudoti įvairius papildomus duomenis apie kelius, tokius kaip erdvinės duomenų bazės, automobilių GPS trajektorijos ir panašiai. Vienas tokių darbų naudoja grubų kelių tinklo vektorinį pavidalą [24]. Šio darbo rezultatai demonstruoja atpažintus kelių segmentus turinčius tikslų plotą. Metodas taip pat nėra pilnai automatinis.

Pagal atliktą apžvalgą ryškėja tendencija - kelių atpažinimo problema nėra sprendžiama tradiciniais duomenų analizės būdais (pavyzdžiui, *SVM* klasifikatoriumi (angliškai *Support Vector Machine*)) arba šablonų derinimu (angliškai *template matching*). Apžvelgtų darbų autoriai siūlo hibridinius, arba visiškai savitus sprendimo metodus. Kadangi šiame darbe yra operuojama tik rastriniais vaizdais ir neturima jokios papildomos informacijos - taškų debesų (angliškai *point cloud*) ar vektorinių duomenų - uždaviniui spręsti reikia taikyti netrivialų metodą. Toks metodas - konvoliucinis neuronų tinklas - yra pristatomas sekančiame skyriuje.

2. Vaizdų klasifikavimas ir konvoliuciniai neuronų tinklai

Šiame skyriuje yra aptariamas vaizdų klasifikavimo uždavinys ir detaliai pristatomi konvoliuciniai neuronų tinklai kaipo įrankis šiam uždaviniui spręsti. Taip pat yra pagrindžiama šių tinklų taikymo kelių atpažinimui motyvacija.

2.1. Vaizdų klasifikavimas

Kelių atpažinimas iš oro nufotografuotuose vaizduose priklauso vaizdų klasifikavimo uždavinių grupei, kur analizuojamam vaizdui yra priskiriama tam tikros, iš anksto apibrėžtos, klasės reikšmė - sveikas skaičius. Viename vaizde ieškant keleto skirtingų klasių objektų ši reikšmė yra priskiriama ne visam vaizdui, o atskiriems jo segmentams (smulkiausias vaizdo segmentas gali būti ir pavienis pikselis). Konkrečios klasių reikšmės priklauso nuo sprendžiamos problemos sudėtingumo ir realizacijos detalių.

Nors žmogaus smegenims atlikti objektų atpažinimą nuotraukose yra trivialus uždavinys, kompiuterinės regos vaizdų klasifikavimo metodai susiduria su dideliu kiekiu iššūkių. Toliau yra išvardijama keletas tokių iššūkių:

- Stebėjimo padėties kitimas - tos pačios klasės objektas gali atrodyti skirtingai priklausomai nuo to, iš kurio taško ir kokiu kampu buvo atlikta nuotrauka.

- Okliuzija - nagrinėjamas objektas yra uždengtas. Ieškant kelių okliuziją dažniausiai sukelia aplinkinių objektų (pvz. medžių, pastatų) šešėliai.
- Masto skirtumai - net ir tos pačios klasės objektai gali būti skirtingo dydžio. Ieškant kelių šis skirtumas pasireiškia tarp skirtingos svarbos kelių.
- Apšvietimas - priklausomai nuo esamo apšvietimo, skirtingų klasių objektų spalvos gali būti nebeatskiriamos. Pavyzdžiui, prasto oro sąlygomis atliktoje nuotraukoje kelio spalva gali būti neatskiriama nuo pastatų spalvos.

Moderniausi vaizdų klasifikavimo algoritmai naudoja duomenimis grįstas besimokančias sistemas (angliškai *data-driven machine learning*). Šis būdas leidžia kurti vaizdų atpažinimo algoritmus kurie neprivalo išreikštinai žinoti kiekvienos dominančios klasės objektų pavidalo. Vietoje to, šie metodai kaip įvestį priima didelius kiekius sužymėtų duomenų (konkrečiai - daug nuotraukų, iš kurių kiekvienai yra iš anksto priskirta teisingos klasės reikšmė) ir, naudojantis įvairiais algoritmais, patys save apmoko kaip turi atrodyti kiekvienos klasės objektai. Perspektyviausias ir labiausiai šiomis dienomis plėtojamas toks metodas yra konvoliuciniai neuronų tinklai (angliškai *convolutional neural networks*). Kasmetiniame vaizdų atpažinimo ir klasifikavimo konkurse ILSVRC (angliškai *ImageNet Large Scale Visual Recognition Challenge*) šie tinklai demonstruoja įspūdingus rezultatus, ypač lyginant su nekonvoliucinių tinklų metodais. Keletas tokių rezultatų pateikiama toliau:

- **AlexNet** [15]. 2012 metais minėtą konkursą laimėjęs tinklas pasiekė 15.3% klaidų kiekį ir smarkiai pralenkė antros vietos laimėtoją, pasiekusį 26.2% klaidų kiekį.
- **ZFNet** [10]. 2013 pasiekė 14.8% klaidų kiekį. Šis modelis yra kiek modifikuotas anksčiau paminėtas **AlexNet**.
- **GoogLeNet** [6]. 2014 metų ILSVRC pasiekė 6.67% klaidų kiekį. Šis tinklas įdomus tuo, kad pristatė naują elementą, kurį pavadino pradžios moduliu (angliškai *inception module*), kuris paremtas keletu labai mažų konvoliucijų ir smarkiai sumažina tinko parametrų kiekį.
- **ResNet** [13]. 2015 metais pasiekė 3.57% klaidų kiekį.
- **Faster R-CNN**. 2016 metais pasiekė 2.99% klaidų kiekį apjungiant prieš tai laimėjusių modelių savybes.

Dėl aukščiau aptarto konvoliucinių neuronų tinklų pasisekimo, šiame darbe kelių atpažinimo problema taip pat yra sprendžiama pasitelkiant konvoliucinių neuronų tinklų modelį. Šis modelis yra išsamiai nagrinėjamas sekančiuose poskyriuose.

2.2. Konvoliucinių neuronų tinklų modelis

2.2.1. Besimokančios sistemos ir neuronų tinklai

Kaip ir kitokio tipo neuronų tinklai, konvoliuciniai tinklai yra apibrėžiami tam tikru kiekiu sluoksnių, iš kurių kiekvienas priima jam tinkamo pavidalo įvesties duomenis, su jais atlieka sau būdingas operacijas ir pagamina tam tikro pavidalo išvesties duomenis. Kiekvieno sluoksnio išvesties duomenys yra po jo sekančio sluoksnio įvesties duomenys. Kiekvienas sluoksnis susideda iš tam tikro kiekio ir pavidalo neuronų. Savo ruožtu, kiekvieną neuroną sudaro svorių rinkinys w,

poslinkis (angliškai - *bias*) b ir aktyvavimo funkcija f. Darbo metu kiekvienas neuronas skaičiuoja savo svorių w ir įvesties duomenų x skaliarinę sandaugą, prie jos prideda poslinkį b ir gautą rezultatą kaip įvestį pateikia aktyvavimo funkcijai f, kuri praktiniuose neuronų tinklų pritaikymuose yra netiesinė [14].

Prieš bandant klasifikuoti norimus vaizdus, neuronų tinklas privalo būti apmokytas - jis turi išanalizuoti didelį kiekį vaizdų (kuo didesnį - tuo geriau), kurie jau anksčiau buvo sužymėti teisingomis klasių reikšmėmis. Šios teisingos klasių reikšmės kartu su vaizdais tinklui turi būti pateiktos suprantamu formatu (pavyzdžiui, *.csv* failu). Toks vaizdų ir teisingų klasių reikšmių rinkinys, pateikiamas tinklui apmokymo tikslais, yra vadinamas mokymo duomenų rinkiniu.

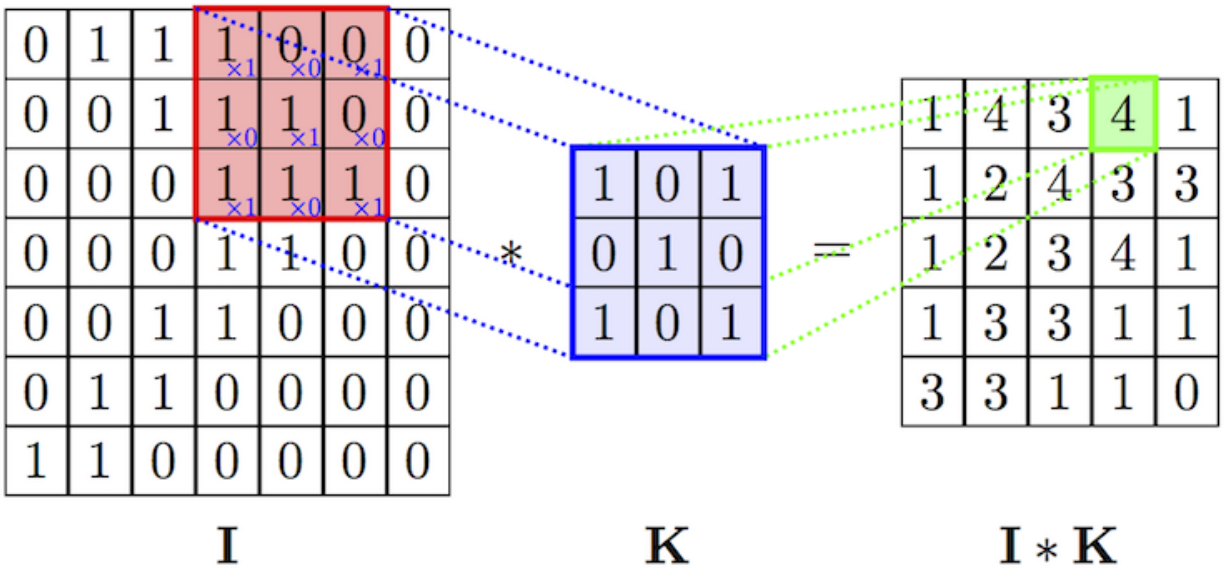
Mokymosi proceso pradžioje kiekvieno tinkle esančio neurono svoriai ir poslinkiai yra inicijuojami atsitiktinėmis reikšmėmis, kurioms taikomi amplitudės ir pasiskirstymo apribojimai. Besimokantis tinklas analizuoja jam pateiktus vaizdus pasluoksniui transformuodamas jų pikselių reikšmes į klasės reikšmę ir pateikia išvadas apie tai, kuriai klasei turėtų būti priskirtas kiekvienas vaizdas. Suprantama, jog proceso pradžioje tinklo išvados bus, jei ne visiškai tai bent dalinai, klaidingos. Klaidų kiekis neuronų tinkle matuojamas pasitelkiant netekties (angliškai *loss*) funkciją. Ši funkcija apskaičiuoja, kiek stipriai tinklo pateiktos išvados kiekvienam vaizdui skiriasi nuo tikrų, kartu su vaizdais pateiktų, klasių reikšmių. Esant skirtumui, tinklo svorių ir poslinkių reikšmės yra pareguliuojamos ir procesas yra iteratyviai kartojamas. Neuronų tinklo mokymosi tikslas yra kaip įmanoma labiau sumažinti netekties funkcijos reikšmę. Toks tikslas mokymosi procesą paverčia savotišku optimizavimo uždaviniu, kadangi tinklo mokymui yra pasitelkiamos tokios optimizavimo technikos kaip gradiento nuolydis (angliškai *gradient descent*). Konkretūs šiame darbe naudojamų konvoliucinių neuronų tinklų mokymo niuansai yra plačiau aptariami 3.2 ir 3.3 poskyriuose.

Pabaigus neuronų tinklo apmokymą jis turi būti įvertintas. Tam yra naudojamas įvertinimo duomenų rinkinys, analogiškas mokymo rinkiniui, susidedantis iš vaizdų ir teisingų jų klasių reikšmių. Įvertinimo proceso metu tinklas transformuoja įvertinimo rinkinio vaizdus į klasės reikšmes ir pateikia tas reikšmes kaip išvadas. Šios išvados, kaip ir mokymosi procese, yra palyginamos su tikromis klasių reikšmėmis naudojant netekties funkciją ir yra tikimasi, kad netektis bus kaip įmanoma arčiau nulio. Reikia atkreipti dėmesį, kad įvertinimo duomenų rinkinys privalo susidėti iš tinklui dar nematytų vaizdų. Priešingų atveju, nebus aišku ar tinklas buvo tinkamai paruoštas naujų vaizdų klasifikavimui. Darant prielaidą, kad mokymo proceso metu netektis buvo sėkmingai sumažinta iki priimtinos reikšmės ir tinklas pateikė teisingas išvadas įvertinimo metu, modelis skaitomas paruoštu ir gali būti diegiamas realiems klasifikavimo uždaviniams spręsti. Savaime suprantama, jog modelis apmokytas atpažinti tam tikrus vaizdus nesugebės klasifikuoti vaizdų, kurie priklauso jam nežinomai klasei (pavyzdžiui, tinklas specializuotas kelių atpažinimui nesugebės atskirti katės nuo šuns).

2.2.2. Konvoliucinis sluoksnis

Lyginant su kitokio tipo daugiasluoksniais neuronų tinklais, konvoliuciniai neuronų tinklai išsiskiria tuo, kad juose yra plačiai naudojamas specialus konvoliucinis sluoksnis. Konvoliucinis sluoksnis kaipo įvestį priima $W \times H \times D$ formos tūrį (pirmajame žingsnyje šis tūris yra W ilgio, H aukščio ir D spalvos kanalų nuotraukos pikseliai). Teoriškai šio sluoksnio įvesties duomenų ilgis ir plotis gali nesutapti, tačiau literatūroje ir praktiniuose pritaikymuose įvesties duomenys yra kvadrato formos. Dėl to šiame darbe taip pat bus aptariami konvoliuciniai sluoksniai, kurie analizuoja tik vienodo ilgio ir pločio $N = W = H$ duomenis.

Vienas konvoliucinis sluoksnis susideda iš vieno, arba keleto, F dydžio kvadrato formos filtrų, dar vadinamų branduoliais (angliškai *kernel*), kurie savo ruožtu susideda iš F^2 individualių svorių. Konvoliucijos operacijos metu šie filtrai „slenka“ per nuotrauką ir atlieka matricų sandaugą tarp savo svorių ir nuotraukoje nagrinėjamos srities pikselių reikšmių. Konvoliucijos operaciją viename kanale iliustruoja 2 pav.



2 pav. Konvoliucijos operacija viename kanale. Filtrui K slenkant per įvesties tūrį I, skaičiuojama filtro svorių ir įvesties tūrio pikselių matricų sandauga. To rezultatas - išvesties tūris $I \times K$. Šaltinis: [22].

Konvoliucijos metu šio sluoksnio filtrai „mokosi“ atpažinti įvairias nuotraukos ypatybes (angliškai *features*). Pirmieji konvoliuciniai sluoksniai atpažįsta žemo lygio ypatybes - kraštus, linijas, spalvų dėmes. Tolesni konvoliuciniai sluoksniai atpažįsta vis sudėtingesnes aukštesnio lygio ypatybes [14]. Aukšto lygio ypatybių pavyzdžiai yra snapai, letenos, arba (kaip siekiama šiame darbe) keliai.

Atliekant konvoliucijos operaciją reikia atkreipti dėmesį į keletą niuansų. Pirma, egzistuoja galimybė reguliuoti filtro slinkimo per įvesties tūrį žingsnį. Standartiniu atveju, žingsnio reikšmė būna 1, kas reiškia jog filtras slinks per kiekvieną nuotraukos pikselį. Didinant šią reikšmę, filtras slinks per nuotrauką šuoliais. Filtro žingsnis yra vienas iš keleto reguliuojamų konvoliucinio sluoksnio parametrų. Kitas reguliuojamas parametras yra papildymas nuliais (angliškai *zero padding*). Naudojant šį parametrą prieš kiekvieną konvoliucijos operaciją įvesties tūrio kraštai yra papildomi nuliais. Papildymo parametras praktikoje leidžia reguliuoti konvoliucijos išvesties dydį ir užtikrinti, kad jis bus toks pats kaip ir įvesties dydis. Apibendrinant, konvoliucinis sluoksnis turi šiuos atributus:

- Įvestis - $N \times N \times D$ dydžio tūris;
- Parametrai:
  - Papildymas nuliais P;
  - Filtro dydis F;
  - Filtro žingsnis S;

- Išvestis - $N_1 \times N_1 \times D$ dydžio tūris. $N_1 = (N - F)/S + 1$.

Reikia pastebėti, jog programuojant konvoliucinius sluoksnius yra svarbu, kad visi minėti parametrai (filtro dydis, žingsnis ir t.t.) sukurtų sveiko dydžio rezultatą - kiekvienas filtras turi „derėti“ su savo įvesties tūriu ir praslinkti juo neišeinant už ribų. Pavyzdžiui, turint 10×10 dydžio įvesties tūrį ir nustačius filtro dydžio, papildymo nuliais bei filtro žingsnio reikšmes atitinkamai 3, 0 ir 2 gautųsi, jog filtras per įvestį slinktų 4.5 karto. Toks rezultatas yra klaidingas.

2.2.3. Išlygintas tiesinis vienetas

Kaip minėta 2.2.1 dalyje, kiekvieno neurono išvestis yra aktyvavimo funkcijos reikšmė, naudojant savo svorių ir įvesties reikšmių skaliarinę sandaugą kaip tos funkcijos argumentą. Ši funkcija turi būti netiesinė [14]. Priešingu atveju, neuronų tinklas pastoviai skaičiuos tiesines reikšmių kombinacijas arba tiesines tiesinių funkcijų kombinacijas. Tokių skaičiavimų rezultatas visada bus tiesinis. Naudojant netiesines funkcijas ir jų kombinacijas neuronų tinklas sugeba apibrėžti sudėtingesnius netiesinius modelius. Tokia savybė yra svarbi vaizdų klasifikavime, kadangi konvoliucinių sluoksnių tikslas yra atrasti sudėtingos formos ypatybes vaizduose, o tokių ypatybių neišeitų apibrėžti tiesiškai. Istoriškai klasifikatoriuose buvo naudojamos tokios netiesinės funkcijos kaip *sigmoid* ir *tanh*, apibrėžtos 2.1 ir 2.2 formulėse:

$$\sigma(x) = 1/(1 + e^{-x}) \tag{2.1}$$

$$tanh(x) = 2\sigma(2x) - 1 \tag{2.2}$$

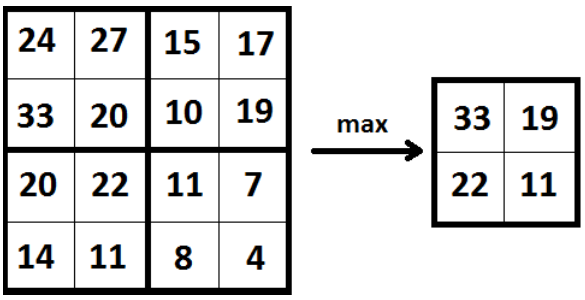
[15] darbo autoriai atrado, kad geriausi rezultatai pasiekiami naudojant išlyginto tiesinio vieneto (angliškai *rectified linear unit*) funkciją $f(x) = \max(0, x)$. Lyginant su sigmoid ir tanh funkcijomis, išlygintas tiesinis vienetas pagreitina mokymosi procesą bei yra paprasčiau apskaičiuojamas, kadangi jo esmė yra tiesiog neigiamas reikšmes paversti nuliais. Dėl to, visuose šio darbo konvoliuciniuose tinkluose yra naudojama išlyginto tiesinio vieneto funkcija.

2.2.4. Sutelkimo sluoksnis

Po konvoliucinio sluoksnio tinkle įprastai seka sutelkimo sluoksnis (angliškai *pooling layer*). Sutelkimo sluoksnio vaidmuo yra sumažinti jam pateikiamo įvesties tūrio dydį. Tai, panašiai kaip ir konvoliuciniame sluoksnyje, yra atliekama pasitelkiant $F_{pooling}$ dydžio kvadrato formos langą (be svorių), kuriuo tam tikru žingsniu $S_{pooling}$ yra slenkama per įvesties tūrį. Kaip ir konvoliuciniame sluoksnyje, $F_{pooling}$ ir $S_{pooling}$ yra reguliuojami tinklo parametrai. Slinkimo metu yra atliekama į filtrą patenkančių elementų kiekio mažinimas pasitelkiant vidurkio arba maksimumo operacijas. Sutelkimo sluoksnio paskirtis yra sumažinti galutinį konvoliucinio tinklo apmokomų parametrų (svorių, poslinkių) ir operacijų su jais kiekį.

Literatūroje, taip pat ir šiame darbe paminėtuose modeliuose, sutelkimo sluoksniuose yra naudojama maksimumo operacija. Tokio sutelkimo sluoksnio, kurio parametrų reikšmės yra $F_{pooling} = 2$, $S_{pooling} = 2$, pavyzdį iliustruoja 3 pav.

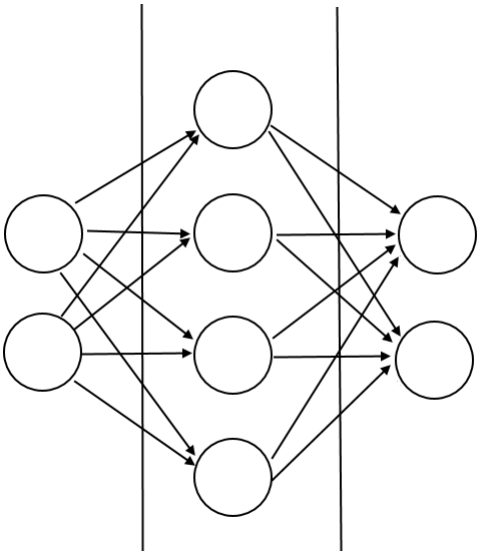
Sutelkimo sluoksnių praktinė nauda yra ta, kad jie mažina galutinį konvoliucinio tinklo parametrų kiekį. Tai pagreitina atliekamus skaičiavimus.



3 pav. Sutelkimo sluoksnis, atliekantis maksimumo operaciją, kuriame $F = 2$, $S = 2$. Iš sluoksniui pateiktų reikšmių lieka jų maksimumai.

2.2.5. Pilnai sujungtas sluoksnis

Didžiąją 2.1 poskyryje išvardintų modelių sluoksnių dalį sudaro tam tikras kiekis konvoliucinių sluoksnių su išlyginto tiesinio vieneto aktyvavimu. Po kai kurių iš jų seka sutelkimo sluoksnis tam, kad sumažintų bendrą tinklo parametrų kiekį. Paskutinieji minėtų tinklų sluoksniai būna pilnai sujungti sluoksniai (angliškai *fully connected layer*), susidedantys iš tam tikro kiekio neuronų. Kiekvienas neuronas pilnai sujungtame sluoksnyje yra sujungtas su su kiekvienu gretimų sluoksnių neuronu, tačiau tame pačiame sluoksnyje neuronai neturi jokių jungčių. Pilnai sujungto sluoksnio pavyzdį iliustruoja 4 pav. Tokiame sluoksnyje priekinio perdavimo (angliškai *forward pass*) metu yra atliekama standartinė įvesties ir svorių sandauga bei poslinkio pridėjimas kartu su aktyvavimo funkcija. Šis sluoksnis transformuoja konvoliucijos metu atpažintas sudėtingų formų ypatybes į ieškomų klasių reikšmių rezultatus(skaičius, reprezentuojančius pradinio vaizdo priklausymą analizuojamoms klasėms). Vėliau šie rezultatai yra interpretuojami ir transformuojami išvesties sluoksnyje.



4 pav. Pilnai sujungtas sluoksnis. Kiekvienas viduriniojo sluoksnio neuronas turi jungtį su kiekvieno gretimų sluoksnių neuronu, tačiau neturi jokių jungčių su to pačio sluoksnio neuronais.

2.2.6. Išvesties sluoksnis

Po pilnai sujungto sluoksnio neuronų tinkle seka paskutinis išvesties sluoksnis. Šio sluoksnio vaidmuo yra transformuoti gautus klasių reikšmių rezultatus į galutinį tinklo išvedimą. Galutinio

išvedimo pavidalas priklauso nuo naudojamos netekties funkcijos, kurią bandoma minimizuoti. Literatūroje ir praktiniuose darbuose yra naudojama kryžminės entropijos netektis (angliškai *cross-entropy loss*) su *softmax* funkcija:

$$L_i = \log(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}), y_i \neq j \tag{2.3}$$

Šioje funkcijoje skaičiai f_k yra interpretuojami kaip nenormalizuotos, logaritminės tikimybės, kad vaizdas priklauso klasei k. Eksponentė $e^{f_{y_i}}$ reiškia nenormalizuotą tikimybę, kad vaizdas i priklauso tikrajai jo klasei y_i. Padalinus tikimybę $e^{f_{y_i}}$ iš likusių klasių tikimybių sumos $\sum_j e^{f_j}$ ji yra normalizuojama. Tokia f_k interpretacija yra intuityvi, kadangi kiekviena tikimybė e^{f_k} yra intervale $[0; 1]$ ir jos visos susisumuoja į 1 [17].

2.3. Parametrų derinimas besimokančiose sistemose

Pasirinkus norimą konvoliucinio neuronų tinklo sluoksnių išdėstymą ir turint apmokymo duomenų rinkinį galima pradėti mokymosi procesą. Šio proceso pasisekimas priklauso nuo didelio kiekio parametrų. Kai kurie parametrai (pavyzdžiui, filtrų kiekis) yra susiję su tinklo struktūra. Keičiant šiuos parametrus keisis atpažįstamų ypatybių formos, taigi ir viso tinklo rezultatai. Parinkus neteisingus parametrus tinklas nesugebės vaizduose atrasti sudėtingų formų ir dėl to klasifikacija bus nesėkminga.

Mokymosi procesas taip pat priklauso nuo parametrų, nesusijusių su tinklo architektūra. Šie parametrai apibrėžia mokymosi proceso eigą. Svarbiausi tokių parametrų - mokymosi greitis ir jo mažėjimas - yra aptariamas 3.3 poskyryje.

3. Algoritmų realizacija

3.1. Bendros nuostatos

Norint kuo tiksliau atpažinti nuotraukose matomus kelius, šio darbo algoritmų realizacijos buvo padalintos į tris kategorijas - duomenų paruošimą, konvoliucinio neuronų tinklo optimizavimą ir po-apdorojimą. Panašų mechanizmą pasiūlė [16] straipsnio autoriai, tačiau jie naudojo nekonvoliucinį neuronų tinklą. Visi šio darbo eksperimentai buvo atliekami su miestų kelių nuotraukomis. Taip yra todėl, kad miesto kelių atpažinimas yra žymiai sudėtingesnis uždavinys - miesto nuotraukose yra daugiau triukšmo ir pašalinių objektų, kurie trukdo atlikti kokybišką kelio analizę. Darbe yra daroma prielaida jog, atradus optimalų algoritmą miesto kelių atpažinimui, jį bus galima taip pat pritaikyti kaimo keliams, kadangi kaimo kelių atpažinimas yra lengvesnis uždavinys.

Tinklo optimizavimo metu buvo atlikta 20 eksperimentų. Kiekvieno eksperimento metu buvo reguliuojamas atskiras konvoliucinio tinklo parametras. Tokiu būdu yra siekiama atrasti, kokią įtaką atskiri tinklo parametrai turi galutiniam rezultatui. Šių eksperimentų atlikimas įgalina sukurti tokį konvoliucinį tinklą, kurio architektūra labiausiai tinka kelių atpažinimui. Visi eksperimentai, susiję su konvoliuciniu neuronų tinklu, buvo vykdomi eilės tvarka. Atradus optimalią vieno parametro reikšmę, ji yra fiksuojama ir naudojama visuose sekančiuose eksperimentuose, kuriuose yra derinamas jau kitas parametras. Šie eksperimentai yra aprašyti 3.2 poskyryje.

Kadangi konvoliucinis neuronų tinklas sprendžia klasifikavimo uždavinį (nurodo, ar nuotraukoje yra kelias ar ne) jo rezultatai yra apytiksliai. Mokslo tiriamojo darbo projekte [12] konvoliucinis tinklas sugebėjo atlikti grubų kelių tinklo atkūrimą iš stambių fragmentų. Dėl to yra reikalinga atlikti tam tikrus po-apdorojimo žingsnius, kurie pasmulkintų tinklo išvesties rezultatus. Po-apdorojimo metodai yra aprašomi 3.5 poskyryje.

Algoritmai, kuriuose dirbama su konvoliuciniais neuronų tinklais, buvo įgyvendinti naudojantis *Python* kalba ir *TensorFlow* atviro kodo biblioteka. Ši biblioteka yra skirta įvairių besimokančių sistemų uždavinių sprendimui. Ji pateikia žemo lygio programavimo sąsają, leidžiančią kontroliuoti visą besimokančių sistemų gyvavimo ciklą - kurti konvoliucinių (ir ne tik) neuronų tinklų modelius, atlikti jų apmokymą ir įvertinimą, stebėti šių procesų eigą, bei pateikti vartotoją dominančias statistikas. Darbo kodas, naudojantis *TensorFlow*, yra parašytas laikantis bibliotekos mokomojoje medžiagoje randamų konvencijų. Iš viso šio darbo programinio kodo, tik ta dalis, kuri yra atsakinga už mokymosi sesijos paleidimą yra randama *TensorFlow* mokomojoje medžiagoje [1]. Žemiau yra pateikiamas sąrašas, įvardijantis unikalų šio darbo indėlį:

- Duomenų paruošimas.
- Duomenų sužymėjimas teisingomis klasių reikšmėmis.
- Konvoliucinio tinklo pritaikymas duomenims.
- Eksperimentiniai tinklų modeliai.
- Konvoliucinio tinklo išvedimo įvertinimas.
- Konvoliucinio tinklo išvedimo sujungimas į bendrą vaizdą.
- Po-apdorojimo algoritmai.
  - Klaidingai teigiamų fragmentų pašalinimas.
  - Centrinių gyvačių algoritmas.
- Rezultatų įvertinimo metodo pritaikymas rastriniams duomenims.

TensorFlow biblioteka leidžia skaičiavimams panaudoti kompiuterio grafinį procesorių (angliškai *graphics processing unit - GPU*). Kadangi konvoliucinio tinklo apmokymas yra daug resursų reikalaujantis procesas, GPU panaudojimas yra itin svarbus, nes atliekami skaičiavimai sutrumpėja keletu kartų. Pavyzdžiui, mokslo tiriamojo darbo projekto metu vieno tinklo apmokymas (naudojant tik centrinį procesorių) truko virš trijų dienų. Pasitelkus GPU, ilgiausias konvoliucinio tinklo apmokymas truko 6 valandas.

Visi šio darbo eksperimentai buvo atliekami dviejuose kompiuteriuose. Konvoliucinio tinklo apmokymai buvo atliekami stacionariame kompiuteryje, naudojant *NVIDIA GeForce GTX 780* GPU. Visi po-apdorojimo darbai buvo atlikti nešiojamame kompiuteryje, naudojant *NVIDIA Quadro M2000M* GPU ir *Intel i7-6820HQ* centrinį procesorių.

Darbe taip pat yra pasitelkiama grafų teorija - minimalaus jungiančiojo medžio algoritmas. Šiam algoritmui įgyvendinti naudojama *Python* kalbos biblioteka *NetworkX*.

3.1.1. Duomenų kiekis besimokančiose sistemose

Kaip minėta 2.2.1 poskyryje, norint apmokyti bet kurio tipo neuronų tinklą yra reikalingas apmokymo duomenų rinkinys. Konvoliucinio neuronų tinklo atveju, toks duomenų rinkinys susideda iš tam tikro kiekio vaizdų, kur kiekvienam vaizdui yra priskirta teisinga jame matomos klasės reikšmė. Ši reikšmė turi būti pateikiama tinklui suprantamu pavidalu. Naudojantis duomenimis grįstu mokymo būdu, mokymo duomenų rinkinio apimties didinimas pagerina modelio rezultatus, kadangi kuo daugiau skirtingų nuotraukų modelis analizuoja, tuo daugiau ypatybių yra išmokstama konvoliuciniuose sluoksniuose. To pasekoje tinkle esantys svoriai gali būti korektiškiau keičiami.

Dėl didelio duomenų kiekio reikalavimo, konvoliucinių neuronų tinklų mokymas yra atliekamas saujomis (angliškai *batches*). Tai reiškia, kad apmokymo metu nuotraukos yra analizuojamos ne po vieną, o panaudojant reguliuojamą parametrą - saujos dydį B_{size}. Be to, nuskaitymo metu duomenų saujos yra sumaišomos ir skaitomos ne eilės tvarka. Žingsnių kiekis, po kurio neuronų tinklas yra išanalizavęs visas galimas duomenų saujas ir matęs visus galimus duomenis, yra vadinamas epocha (angliškai *epoch*). Šios sąvokos yra svarbios, kuomet prireikia reguliuoti neuronų tinklo apmokymo greičio mažėjimą. Detalus eksperimentuose naudotas mokymo greičio mažėjimas yra aprašomas 3.3.2 skirsnyje.

Didelio duomenų kiekio reikalavimas pristato praktinę konvoliucinių tinklų apmokymo problemą - kokiu būdu galima greitai ir efektyviai sužymėti didžiulius kiekius duomenų? Vienas iš būdų yra panaudoti egzistuojančius duomenų šaltinius. Minėtame ILSVRC konkurse yra naudojami *ImageNet* duomenys. *ImageNet* yra didelės apimties, laisvai prieinama vaizdų bazė, kurioje kiekvienas vaizdas yra suklasifikuotas rankiniu būdu. Kitoks galimas duomenų sužymėjimo būdas yra pritaikytas technologijų kompanijos *Google* - jų platinama interneto svetainių apsaugos nuo automatinių atakų sistema *reCAPTCHA* pateikia vartotojams nedidelį vaizdų rinkinį ir prašo jų pažymėti visus vaizdus, kuriuose yra matomi tam tikros klasės objektai. Iš pažymėtų vaizdų yra kuriami duomenų rinkiniai, kuriuos kompanija toliau naudoja besimokančiose sistemose.

Kelių atpažinimo iš oro nufotografuotuose vaizduose užduotis yra specifinė ir jai išspręsti neegzistuoja patogaus, lengvai prieinamo mokymo duomenų rinkinio. Kadangi vaizdų žymėjimas rankiniu būdu yra lėtas procesas, šiame darbe buvo išplėtotas pusiau automatinis mokymo duomenų rinkinio paruošimo metodas.

3.1.2. Duomenų rinkinio paruošimas

Visų eksperimentų metu buvo naudojamos ortografinės nuotraukos, paimtos iš dviejų, laisvai prieinamų, duomenų šaltinių - *Google Maps* [4] ir lietuviško erdvinių duomenų rinkinio *geoportal.lt* [2]. Darbe yra naudojamasi abiem šaltiniais, kadangi lietuviškojo duomenų šaltinio nuotraukos yra aukštesnės kokybės ir kontrasto. Analizuojant miesto kelius, aukštesnis spalvų kontrastas leidžia pasiekti geresnius klasifikavimo rezultatus. Taip yra todėl, kad aukštesnio kontrasto nuotraukose pilka kelio spalva yra lengviau atskiriama nuo pilkšvos pastatų spalvos. Kaimo kelių analizei buvo panaudotos *Google Maps* nuotraukos, kadangi kaimo kelio spalva smarkiai skiriasi nuo aplinkos spalvos. Be to, kaimo kontekste aukštas kontrastas neturi tokios didelės reikšmės, kadangi nuotraukose yra mažai objektų.

Abiejų šaltinių atveju, kiekvienai ortografinei nuotraukai buvo sugeneruota papildoma nuotrauka, kurioje yra matomi visi tikri originaliosios nuotraukos keliai. Ši papildoma nuotrauka buvo generuojama skirtingai, priklausomai nuo duomenų šaltinio.

Dirbant su *Google Maps*, teko pasinaudoti jų pateikiama programavimo sąsaja (angliškai *application programming interface* - API). Ši sąsaja įgalina programuotojus išfiltruoti nepageidaujamus

geografinius objektus iš žemėlapio. Keliai yra vienas iš sąsajoje egzistuojančių objektų tipų. Programuotojai taip pat gali reguliuoti žemėlapio elementų spalvas. Naudojantis šiomis galimybėmis buvo sukurtas specialiai šiam darbui pritaikytas tikrų kelių žemėlapis, matomas 5 pav.

5 pav. Kairėje - originali ortografinė nuotrauka. Dešinėje - specialus tikrų kelių žemėlapis, naudojamas konvoliuciniame neuronų tinkle.

Dirbant su *geoportal.lt* duomenimis buvo sugeneruotas panašus kelių žemėlapis. Šiuo atveju, žemėlapis buvo sugeneruotas naudojant georeferencinių duomenų rinkinį GDR10LT [3]. Kiekvienas objektas šio rinkinio vaizde yra užkoduotas unikalia spalva. Miesto keliai tokiuose vaizduose yra baltos spalvos. GDR10LT vaizdams buvo sukurtas specialus išankstinio apdorojimo (angliškai *preprocessing*) algoritmas. Algoritmo metu yra analizuojamas kiekvienas vaizdo pikselis. Tie pikseliai, kurių RGB reikšmė yra mažesnė už tam tikrą slenkstinę reikšmę P (reguliuojamas algoritmo parametras), yra nuspalvinami juodai. Kadangi būtent miesto kelių spalva GDR10LT rinkinyje yra balta, po tokio apdorojimo yra gaunamas dvispalvis vaizdas, kuriame yra matomi balti kelių tinklai. Visa kita vaizdo dalis yra tiesiog juoda, kaip ir 5 pav.

Reikia atkreipti dėmesį, kad visų sugeneruotų kelių žemėlapių centras sutampa su originalaus ortografinio vaizdo centru. Dėl to, pikseliai, kurie kelių žemėlapiuose yra balti, idealiai atitinka originalios nuotraukos kelių tinklą.

Juodai-balta kelių žemėlapio spalva nėra pasirinkta atsitiktinai. Kelių žemėlapio kūrimas taip pat atlieka savotišką duomenų kodavimo/klasifikavimo vaidmenį. Kiekvienas žemėlapio pikselis turi reikšmę iš klasių aibės $K = \{0, 1\}$. Šioje aibėje reikšmė 1 atitinka baltą spalvą ir reiškia, kad pikselis priklauso keliui. Reikšmė 0 atitinka juodą spalvą ir reiškia, kad pikselis nepriklauso keliui.

Kadangi konvoliucinis neuronų tinklas yra vaizdų klasifikatorius, jo užduotis yra analizuojamam vaizdui priskirti klasės reikšmę. Šiuo atveju, konvoliucinis tinklas vaizdams priskirs reikšmes iš aukščiau apibrėžtos aibės K. Be abejo, analizuojant didelę nuotrauką, nepakanka jai priskirti vienos klasės reikšmės - tai būtų labai grubus atpažinimo rezultatas. Šiame darbe yra siekiama kaip įmanoma tiksliau suklasifikuoti atskirus nuotraukos regionus tam, kad atkurti pilną kelių tinklą. Šiam tikslui pasiekti, kiekviena ortografinė nuotrauka bei jos kelių žemėlapis yra padalinama į

$N_{frag} \times N_{frag}$ dydžio fragmentus pagal tokią formulę:

$$N_{frag} = \sqrt{\left\lceil \frac{W \times H \times P_{frag}}{100} \right\rceil} \tag{3.1}$$

Šioje formulėje W ir H - originalios nuotraukos ilgis ir plotis, P_{frag} - originalios nuotraukos dalis (procentais), kurią užima sukurtas fragmentas.

Šiame darbe naudotų ortografinių nuotraukų mastelis yra $1 : 5000$. Šis mastelis buvo pasirinktas stengiantis išlaikyti pusiausvyrą tarp matomo kelio detalių ir aprėpiamo ploto dydžio. Kiekvienos nuotraukos dalinimo metu papildomai buvo naudojamas slenkančio lango žingsnio parametras S_{frag}. Atlikus keletą padalinimo eksperimentų su tokiomis nuotraukomis buvo nustatyta, jog optimali P_{frag} reikšmė - 0.07%, o optimali S_{frag} reikšmė - 18. Su šiomis reikšmėmis yra sugeneruojami 40×40 dydžio fragmentai, kurių kiekis priklauso nuo originalios nuotraukos matmenų. Pavyzdžiui, panaudojus penkias nuotraukas, kurių dydis yra 1920×1055, pilna apmokymo duomenų rinkinio aibė yra 31565 nuotraukų fragmentų.

Sugeneravus minėtą duomenų rinkinį, lieka vienas žingsnis iki to, kad duomenys būtų paruošti neuronų tinklui - fragmentų sužymėjimas. Kaip jau minėta, kiekvienas nuotraukos fragmentas turi juodai-baltą savo atitikmenį specialiame kelių žemėlapyje. Konvoliucinis neuronų tinklas, kaip įvestį priima du dalykus - originalios nuotraukos fragmentą ir jo klasės reikšmę. Kelių žemėlapio fragmentai patys iš savęs nėra tinkamo pavidalo, kad būtų traktuojami kaip klasės reikšmės. Dėl to, juos reikia transformuoti į skaičius iš aibės K. Tai yra padaroma sumuojant kiekvieno juodai-balto fragmento pikselius. Pažymėjus fragmente esančių baltų pikselių kiekį W_{pixel}, o juodų pikselių kiekį B_{pixel}, fragmento klasės reikšmė $C \in K$ yra priskiriama pagal 3.2 formulę:

$$C = \begin{cases} 1, \dfrac{W_{pixel}}{B_{pixel}} \geq 0.05 \\ 0, \dfrac{W_{pixel}}{B_{pixel}} < 0.05 \end{cases} \tag{3.2}$$

Kiekvieno fragmento klasės reikšmė C yra yra įrašoma į *.csv* formato failą. Reziumuojant, konvoliuciniam neuronų tinklui kaip įvestis yra pateikiami tokie duomenys:

1. Originalios ortografinės nuotraukos fragmentai.
2. Kiekvieno fragmento klasės reikšmė C, gauta pagal aukščiau aprašytą metodiką.

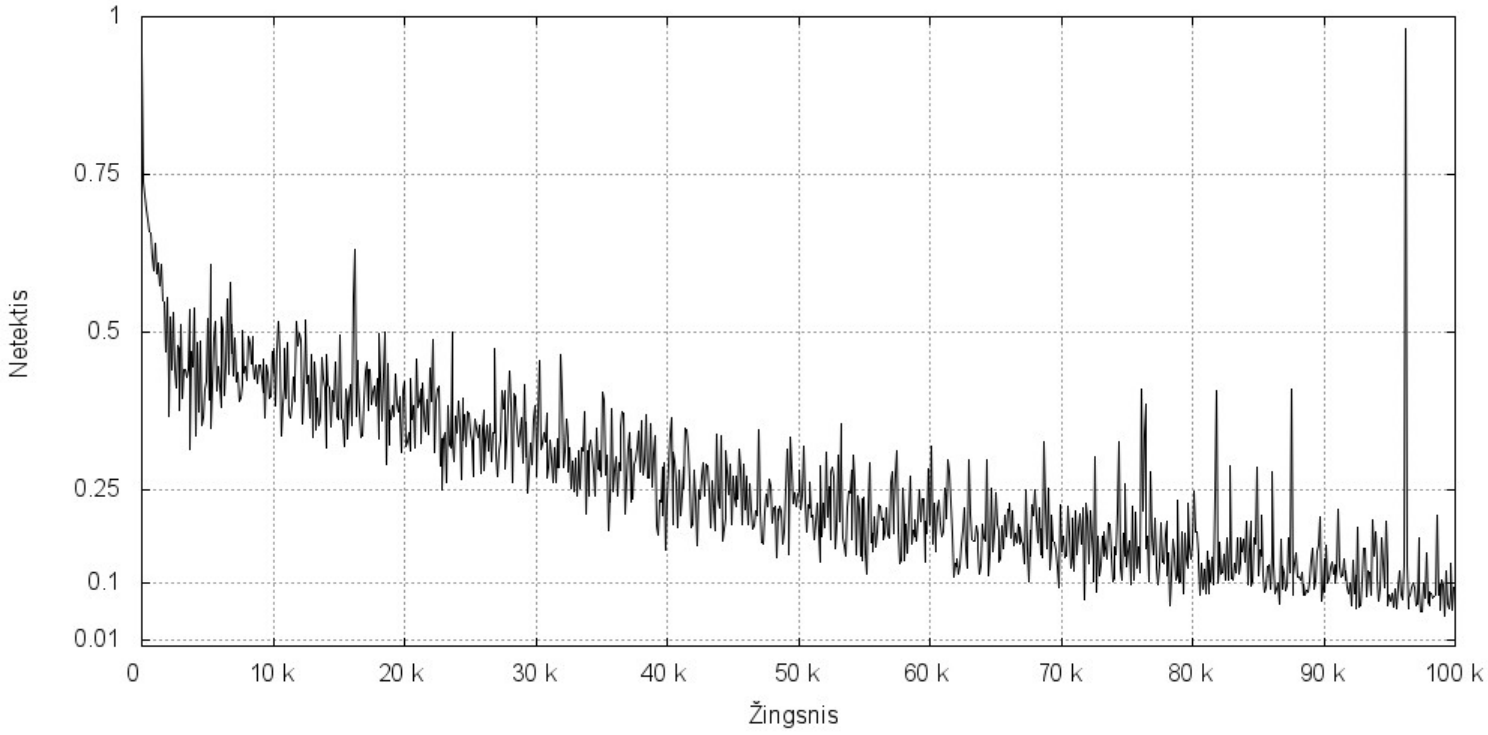
3.2. Tinklo architektūros derinimas

Šiame poskyryje yra aprašoma, kokį poveikį bendram klaidų kiekiui turi tinklo architektūrą keičiantys parametrai. Reikia pastebėti, kad kol kas mokslinėje literatūroje nėra aiškių gairių, kuriomis vadovaujantis, būtų galima sukurti „tobulos“ architektūros konvoliucinį neuronų tinklą. Kiekvienam atskiram uždaviniui yra kuriami vis kitokios formos neuronų tinklai.

Prieš atliekant bet kokį eksperimentą, reikia pristatyti bazinį konvoliucinį neuronų tinklą, su kuriuo bus lyginami visi šiame darbe sukurti tinklai. Bazinis tinklas nepasižymi sudėtinga architektūra - jis turi du konvoliucinius sluoksnius. Po kiekvieno konvoliucnio sluoksnio taip pat seka sutelkimo sluoksnis. Šis bazinis tinklas buvo panaudotas autoriaus mokslo tiriamojo darbo projekte [12].

Pirminė konvoliucinio tinklo apmokymo proceso išvestis - apmokymo kreivė. Horizontali kreivės ašis nusako apmokymo žingsnio numerį. Vertikali ašis - klaidų kiekį bet kuriame einamajame apmokymo žingsnyje. Ši kreivė atsako į svarbiausią apmokymo proceso klausimą - ar tinklas, laikui bėgant, sugeba teisingai mokytis. Geidžiamas rezultatas yra kreivėje įžvelgti mažėjimo tendenciją. Tai reiškia, kad, laikui bėgant, tinklas savo svorius reguliuoja taip, jog klaidų kiekis laipsniškai mažėja. Kreivė yra laikoma tuo geresne, kuo mažesnį klaidų kiekį ji sugeba pasiekti viso apmokymo metu.

Analizuojant kreivės mažėjimo tendenciją (angliškai *trend*) galima daryti išvadas apie tinklo parametrų tinkamumą. Tinklo, kurio parametrai yra parinkti teisingai, kreivėje turėtų vyrauti mažėjimo tendencija, panaši į eksponentinę [17]. Be to, ši kreivė yra osciliuojanti. Kuo mažesnė yra kreivės osciliavimo amplitudė, tuo mažiau tinklas „mėtėsi“ apmokymo metu. Tai reiškia, kad jo svoriai reguliavosi kryptingai po truputį, o ne padrikai, spėliojant naujas svorių reikšmes. Bazinio konvoliucinio tinklo kreivė yra matoma 6 pav. Apmokius kiekvieną naują konvoliucinį tinklą, jo apmokymo proceso kreivė buvo palyginta su šia bazine kreive.



6 pav. Bazinio konvoliucinio neuronų tinklo apmokymo proceso kreivė.

Eksperimentas laikomas sėkmingu, jei jo apmokymo kreivė pasižymi bent viena (arba daugiau) iš šių savybių, lyginant su bazinio tinklo kreive:

- Kreivė pasižymi eksponentine nykimo tendencija.
- Kreivė pasiekė mažesnį globalų minimumą.
- Kreivės osciliavimo amplitudė yra mažesnė.

Reikia atkreipti dėmesį jog, analizuojant eksperimentų rezultatus, yra lyginamos tik tos kreivių savybės, kurios skiriasi (tiek teigiamai, tiek neigiamai). Nepakitusios savybės darbe detaliai nėra aptariamos.

3.2.1. Konvoliucinių sluoksnių kiekis

Kaip buvo minėta 2.2.2 poskyryje, konvoliucinis sluoksnis mokosi atpažinti įvairias nuotraukose matomas ypatybes. Duomenims pereinant iš pradinių tinklo sluoksnių į gilesnius, didėja atpažįstamų ypatybių sudėtingumas. Kai kuriuose modeliuose konvoliucinių sluoksnių kiekis gali perkopti keletą dešimčių, kaip, pavyzdžiui, *Google Inception-v3* architektūroje. Reikia atkreipti dėmesį į tai, kad kiekvieno naujo konvoliucinio sluoksnio pridėjimas padidina skaičiavimų apimtį, kadangi atsiranda papildomos matricų sandaugos. Taip pat didėja ir atminties reikalavimai, kadangi atsiranda poreikis saugoti daugiau filtrų reikšmių. Dėl šių priežasčių atsiranda tikimybė „atsiremti“ į resursų trūkumą.

Šio darbo metu konvoliucinių sluoksnių kiekis tinkle yra didinamas po vieną. Kaskart pridėjus naują sluoksnį, tinklo apmokymo procesas yra pakartojamas. Po kiekvieno apmokymo yra tikrinamas mažiausias pasiektas klaidų kiekis, apmokymo laikas ir netekties reikšmių mediana. Eksperimentai yra kartojami tol, kol naujo konvoliucinio sluoksnio pridėjimas rezultatą pablogins. Eksperimentai taip pat gali būti nutraukti, jeigu naujų konvoliucinių sluoksnių pridėjimas rezultatą pradėtų gerinti labai nežymiai. To priežastis yra ta, kad, kaip jau minėta, šie sluoksniai didina skaičiavimų apimtį. Dėl to, labai nežymus rezultato pagerinimas nėra vertas išaugusių reikalavimų. Kadangi konvoliucinių sluoksnių kiekio didinimo eksperimentų rezultatai yra didelės apimties, jie yra pateikiami 1 lentelėje, o ne atskiromis apmokymų kreivėmis.

1 lentelė. Klaidų kiekio, laiko, ir reikšmių medianos priklausomybė nuo konvoliucinių sluoksnių kiekio.

Sluoksniai	Klaidų kiekis	Laikas, min	Mediana
1	0.061796695	89	0.34
2	0.045487009	117	0.23
3	0.035402618	123	0.19
4	0.023879064	133	0.16
5	0.020447411	134	0.15
6	0.020054489	167	0.09

Kaip matoma, pasiekus šešių konvoliucinių sluoksnių ribą apmokymo rezultatas pagerėja nežymiai. Be to, kiekvieno sluoksnio pridėjimas padidina apmokymo laiką vidutiniškai 15 minučių. Tačiau, yra svarbu atkreipti dėmesį į klaidų kiekio medianą. Nors šeštas konvoliucinis sluoksnis ženkliai padidina skaičiavimų trukmę, jis sumažina klaidų kiekio medianą. Tai reiškia, jog šeštojo konvoliucinio sluoksnio pridėjimas sumažina apmokymo kreivės osciliavimo amplitudę. Ši ypatybė yra laikoma pakankamai svarbia, kad šeštas konvoliucinis sluoksnis dar būtų įtrauktas į galutinę konvoliucinio neuronų tinklo architektūrą.

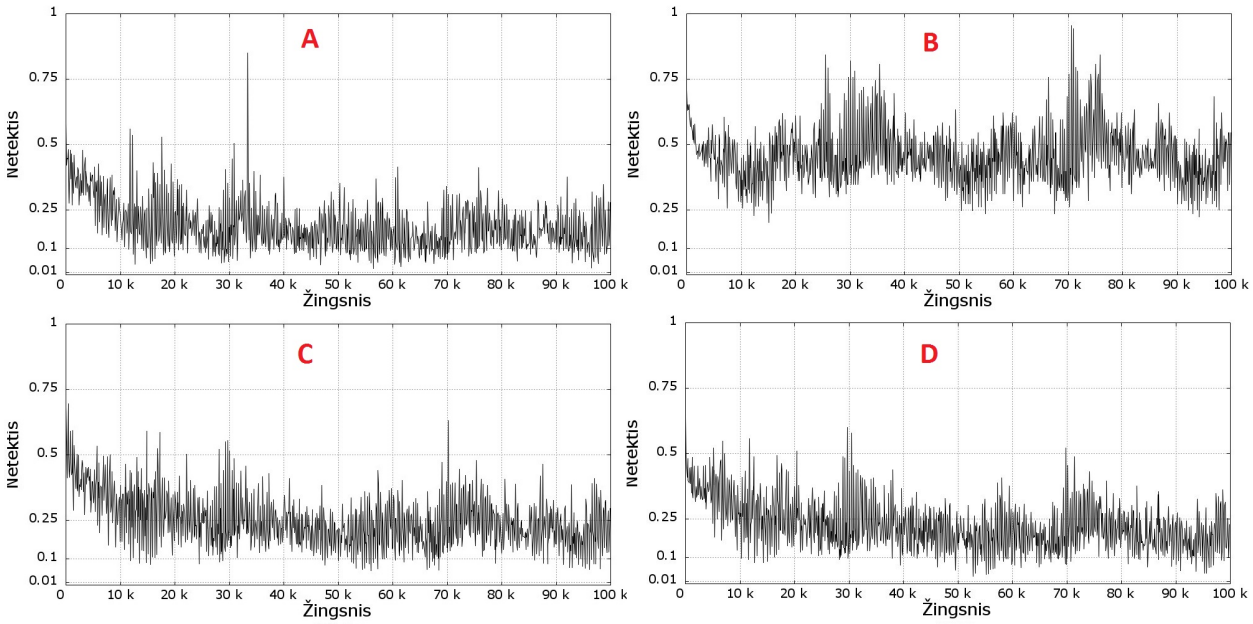
3.2.2. Filtro parametrai

Kaip jau minėta 2.2.2 skirsnyje, kiekvieno filtro parametrai yra filtro dydis, filtro žingsnis ir papildymas nuliais. Šiame skirsnyje yra aptariami tik filtro dydis ir žingsnis. To priežastis yra ta, kad papildymo nuliais poveikis yra atvirkštinis sutelkimo sluoksnio poveikiui. Dėl to, minėti parametrai yra aptariami kartu 3.2.3 skirsnyje.

Tiriant filtro dydžio ir žingsnio poveikį buvo atlikti 4 eksperimentai. Kiekvienas tinklas buvo pavadintas pagal šias taisykles:

- A - tinklas su 3×3 dydžio filtru.
- B - tinklas su 7×7 dydžio filtru.
- C - tinkas A su 2×2 dydžio žingsniu.
- D - tinklas B su 2×2 dydžio žingsniu.

Eksperimentų rezultatai yra keturios apmokymo procesų kreivės. Jos yra apjungtos į vieną iliustraciją ir matomos 7 pav.



7 pav. Konvoliucinio sluoksnio parametrų derinimo eksperimentai.

Pagal 7 pav. kreives matoma, jog, lyginant su baziniu konvoliuciniu tinklu, filtro parametrų keitimas arba nepagerino, arba netgi pablogino mokymosi proceso eigą. Eksperimentų A, C ir D tinklai pernelyg greitai įsisotino aplink savo vidutinę reikšmę - buvo tikėtasi išgauti palaipsniui konverguojančią kreivę, kuri reikštų po truputį gerėjantį mokymo procesą. Be to, šių eksperimentų kreivės labiau osciliuoja, lyginant su bazinio tinklo rezultatu. Tai reiškia, jog tinklo atliekami svorių reguliavimai neprivedė tinklo prie nuotraukose esančių ypatybių atpažinimo. Eksperimento B atveju, visos aukščiau minėtos problemos taip pat galioja. Be to, šio tinklo kreivė įsisotino aplink didesnę klaidų kiekio reikšmę. Tai reiškia, jog tinklas B visomis prasmėmis pasirodė prasčiau už bazinį tinklą.

Iš aptartų eksperimentų rezultatų matoma, jog filtro parametrų derinimas nepagerino apmokymo proceso. Pagal tai yra daroma išvada jog, su turimais duomenimis, optimaliausi filtro parametrai jau buvo pasiekti baziniame konvoliuciniame neuronų tinkle.

3.2.3. Papildymas nuliais ir sutelkimo sluoksniai

Kaip jau minėta 3.2.2 skirsnyje, papildymo nuliais ir sutelkimo sluoksnių parametrai gali būti aptariami kartu. Taip yra todėl, kad jie turi tą pačią paskirtį - modifikuoti konvoliucinio sluoksnio įvesties ir išvesties dydį. Sutelkimo sluoksnis šį dydį sumažina, o papildymas nuliais - padidina.

Egzistuoja du galimi papildymo nuliais metodai - validus (angliškai *valid*) ir tapatus (angliškai *same*). Naudojant validų metodą, sluoksnio įvesties duomenys nėra papildomi nuliais. Tokiu atveju, sluoksnio išvestis būna mažesnė už įvestį. Programuotojas pats turi užtikrinti, kad jo tinklas pagamins teisingus rezultatus ir kad duomenys, keliaudami jo tinklu, nesumažės iki 0×0 dydžio. Naudojant tapatų metodą, įvesties duomenys papildomi tokiu kiekiu nulių, kurį panaudojus, galios lygybė:

$$n_o = \left\lceil \frac{n_i}{S_{padding}} \right\rceil \tag{3.3}$$

Šioje lygybėje n_o reiškia išvesties dydį, n_i - įvesties dydį, o $S_{padding}$ - filtro žingsnį [5].

Sutelkimo sluoksniai reguliuojami vienu parametru - sutelkimo sluoksnių kiekiu. Sutelkimo sluoksnių kiekis nustato, kiek kartų, ir iki kokio dydžio, mažėja analizuojami duomenys (paveiksliukų ypatybės).

Baziniame konvoliuciniame tinkle buvo naudojamas validus papildymo metodas ir du sutelkimo sluoksniai. Eksperimento, kurio metu buvo pakeistas papildymo metodas, apmokymo kreivė yra matoma B priede, 21 pav. Pagal šia kreivę yra matoma, jog apmokymo proceso rezultatai ženkliai pablogėjo visais kriterijais:

- Apmokymo kreivė neturi nykimo tendencijos - ji lieka įsisotinusi aplink vieną reikšmę viso mokymosi metu.
- Apmokymo kreivė pasiekė didesnį globalų minimumą.
- Apmokymo kreivė osciliuoja smarkiau, nei bazinio tinklo kreivė.

Atliekant eksperimentus su sutelkimo sluoksniais, jų kiekis buvo tik mažinamas, o ne didinamas. Pridėjus daugiau sutelkimo sluoksnių, analizuojamos ypatybės taptų pernelyg mažos, kadangi pradinių nuotraukų fragmentų dydis yra 40×40. Eksperimentų metu nustatyta, jog sutelkimo sluoksnių pašalinimas neturi reikšmingos įtakos tinklo apmokymo kreivei. Tačiau, kaip ir tikėtasi, šių sluoksnių pašalinimas padidino skaičiavimo laiką. Rezultatas, kuomet nepakitusi kokybė yra pasiekiama per ilgesnį laiką, nėra patenkinamas. Dėl to, darbe yra nuspręsta palikti sutelkimo sluoksnių kiekį tokį patį, kaip ir baziniame tinkle.

3.2.4. Neuronų kiekis pilnai sujungtame sluoksnyje

Dar vienas tinklo architektūros parametras - neuronų kiekis pilnai sujungtame sluoksnyje. Nagrinėjant įvairią literatūrą ir 2.1 poskyryje pristatytų tinklų modelius, buvo pastebėta, jog pilnai sujungtuose sluoksniuose neuronų kiekis visuomet yra koks nors dvejeto laipsnis. [6] ir [10] naudoja sluoksnius su 1024 neuronais, o [15] - 2048.

Baziniame konvoliuciniame tinkle šis neuronų kiekis buvo nustatytas į 4. Toks skaičius buvo parinktas ne atsitiktinai - mokslo tiriamojo darbo projekte buvo iškelta hipotezė, jog neuronų kiekis pilnai sujungtame sluoksnyje turėtų būti 2^N, kur $N = |K| = 2$ - klasių aibės galia [12]. Šio darbo eksperimentai toliau patvirtino šią hipotezę - bandant didinti šį neuronų skaičių, mokymosi proceso kreivė pablogėjo - jos absoliutaus minimumo reikšmė tapo didesnė, nei naudojant 4 neuronus. Be to, kreivė prarado savo nykimo tendenciją. Pasiekus 32 neuronus, mokymo procesas ėmė diverguoti ir programą teko nutraukti. Kreivės, vaizduojančios mokymo procesą su 2, 8 ir 16 pilnai sujungto sluoksnio neuronų, yra B priede, 22 pav.

3.2.5. Lokalaus atsako normalizavimas

Viena iš nepageidautinų neuronų tinklo būsenų yra neuronų įsisotinimas (angliškai *saturation*). Neuronai tinkle yra laikomi įsisotinę tada, kuomet jų išvesties reikšmės sutampa su jų aktyvavimo funkcijos minimumu arba maksimumu. Kuomet neuronai išveda tokius rezultatus, jų aktyvavimo funkcijos išvestinė turi labai mažą reikšmę. Ši maža išvestinės reikšmė, savo ruožtu, slopina visą neurono signalo srautą į neurono svorius. Dėl mažo signalo srauto, neurono svoriai negali atlikti ženklaus savo pasikeitimo ir pradeda „nebesimokyti“ [17].

Būdas, kurio dėka galima išvengti neuronų įsisotinimo, vadinasi normalizavimu. Literatūroje minimas ne vienas normalizavimo metodas, tačiau **AlexNet** autoriai pasiūlė metodą, vadinamą lokalaus atsako normalizavimu (angliškai *local response normalization*) [15]. Šio metodo formulė atrodo taip:

$$b^i_{x,y} = a^i_{x,y}/(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N_{filter}-1,i+n/2)} (a^j_{x,y})^2)^\beta \tag{3.4}$$

Šios formulės nariai reiškia:

- $a^i_{x,y}$ - i-tojo konvoliucijos filtro išvestis.
- N_{filter} - einamojo konvoliucijos filtro numeris.
- n - gretimojo konvoliucijos filtro numeris.
- k, α, β - reguliuojami parametrai.

Šiame darbe buvo atlikti 6 eksperimentai, susiję su lokalaus atsako normalizavimu - tiek pat kiek tinkle yra konvoliucinių sluoksnių. Individualaus eksperimento metu, po kiekvieno konvoliucinio sluoksnio išvesties buvo atliekamas normalizavimas. Atliekant eksperimentus buvo stebima, kokį poveikį normalizavimo didinimas turi tinklo mokymosi kreivei.

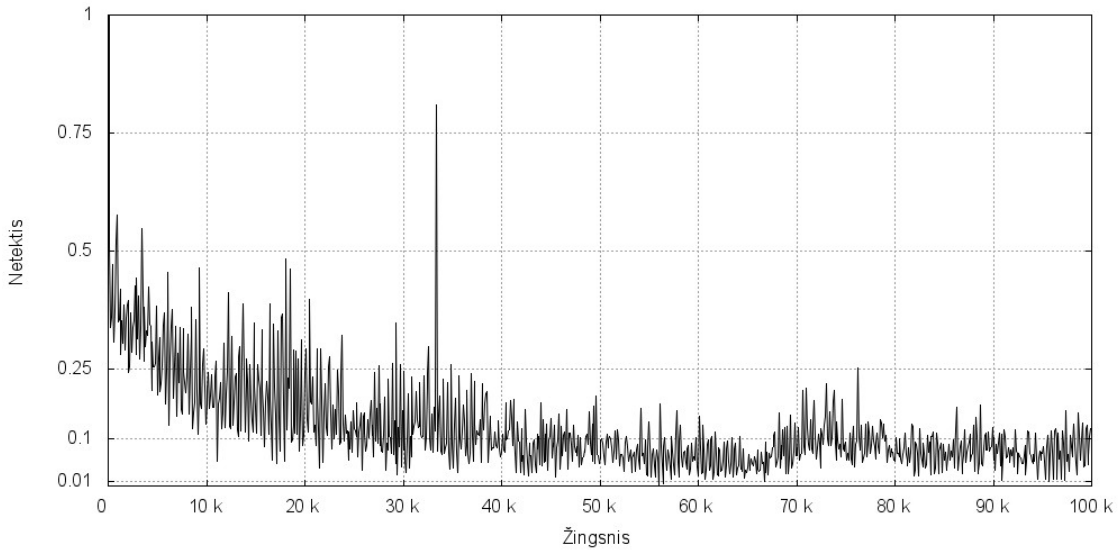
Atlikti eksperimentai parodė, jog didinant normalizavimų kiekį tinklo apmokymas pasiekia vis geresnius rezultatus. Geriausias apmokymo rezultatas buvo pasiektas paskutiniame eksperimente, kuriame po kiekvieno konvoliucinio sluoksnio buvo taikomas normalizavimas. Geriausio pasiekto apmokymo rezultato kreivė iliustruota 8 pav.

Pagal pateiktą kreivę matoma, jog lokalaus atsako normalizavimas pagerino konvoliucinio neuronų tinklo apmokymo rezultatą, kadangi:

- Lyginant su baziniu tinklu, apmokymo kreivės nykimo tendencija panašesnė į eksponentinę.
- Apmokymo kreivė pasiekė mažesnį globalų minimumą.
- Apmokymo kreivė proceso eigoje ima mažiau osciliuoti.

3.3. Mokymosi greičio derinimas

Pagrindinis parametras, kuris lemia konvoliucinio tinklo mokymosi proceso eigą, yra mokymosi greitis (angliškai *learning rate*). Gradiento nuolydžio kontekste, šis parametras yra žinomas kaip žingsnio dydis (angliškai *step size*). Gradiento nuolydžio metu, funkcijos išvestinė nurodo, kuria



8 pav. Apmokymo rezultatas, pasiektas taikant lokalaus atsako normalizavimą po kiekvieno konvoliucinio sluoksnio.

kryptimi funkcijos reikšmė turėtu pakisti, kad būtų pasiektas maksimalus sumažėjimas. Tačiau išvestinė nenurodo, „kiek toli“ reikia žengti minėta kryptimi. Į pastarąjį klausimą atsako būtent mokymosi greitis [17]. Optimalus šio parametro parinkimas yra svarbus, nes:

1. Jei jis per žemas, mokymasis vyks per lėtai ir truks per ilgai.
2. Jei jis per aukštas, svorių atnaujinimai bus pernelyg chaotiški ir nesugebės nusistovėti ties optimalia netekties funkcijos reikšme.

Konvoliucinio tinklo mokymosi greitis pilnai apibrėžiamas trimis parametrais:

1. Pradinis mokymosi greitis.
2. Mokymosi greičio mažėjimo koeficientas.
3. Mokymosi greičio mažėjimo dažnis.

Toliau yra aptariama, su kokiomis šių parametrų reikšmėmis yra pasiekiamas geriausias tinklo apmokymo rezultatas.

3.3.1. Pradinis mokymosi greitis

Pradinis mokymosi greitis yra išreiškiamas vienu skaičiumi. Šis skaičius skiriasi, priklausomai nuo sprendžiamo uždavinio ir tinklo. Dėl to, tinkama jo reikšmė turi būti nustatyta eksperimentų būdu. Šiame darbe, kaip ir kitoje literatūroje, pradinis mokymosi greitis yra išreiškiamas dešimties laipsniais. Eksperimentais buvo nustatyta, jog tinkamiausias pradinis mokymosi greitis šio darbo tinklui yra 10^{-3}.

Pakėlus šį greitį iki 10^{-2} ir aukščiau, mokymosi rezultatas pablogėja, kadangi proceso kreivė ima smarkiau osciliuoti ir nebesimokyti. Ši kreivė randasi B priede, 23 pav. Nustačius pradinį mokymosi greitį į 10^{-1}, klaidų kiekis proceso eigoje ima didėti, kas yra visiškai nepriimtina. Pradiniam greičiui priskyrus reikšmę 10^{-4}, arba mažiau, mokymosi proceso kreivė labai greitai įsisotina apie nepriimtiną globalų minimumą, ir nebejuda iš vietos. Ši kreivė randasi B priede, 24 pav.

3.3.2. Mokymosi greičio mažėjimas

Praktiniuose neuronų tinklų pritaikymuose, norint pasiekti patenkinamą apmokymo rezultatą, vien tik pradinio mokymosi greičio nepakanka. Literatūroje yra rekomenduojama apmokymo metu mažinti mokymosi greitį [9]. Kaip jau minėta 3.3 poskyryje, mokymosi greičio mažėjimas apibrėžiamas dviem skaičiais - mažėjimo koeficientu ir mažėjimo dažniu. Mažėjimo dažnis nusako, kas kiek žingsnių mokymosi greitis bus keičiamas. Mažėjimo koeficientas apibrėžia, kaip stipriai pasikeis mokymosi greitis per vieną mažėjimo žingsnį.

Kaip ir daugelio konvoliucinio tinklo parametrų atveju, optimalios mažėjimo koeficiento ir dažnio reikšmės priklauso nuo sprendžiamos problemos ir pačio modelio [17]. Šiame darbe mokymosi greitis yra mažinamas kas H_{step} žingsnių, apskaičiuojamų pagal šią formulę:

$$H_{step} = \frac{T_{step} \times B_{size}}{3 \times D_{size}} \tag{3.5}$$

Šioje formulėje T_{step} reiškia mokymosi žingsnių kiekį, B_{size} - duomenų partijos dydį, D_{size} - duomenų rinkinio dydį. Ši formulė užtikrina, jog mokymosi greitis tolygiai sumažės tris kartus per visą mokymosi procesą.

Šiame darbe, remiantis kitais nagrinėtais modeliais, mokymosi greičio mažėjimas yra eksponentinis.

3.4. Papildomi parametrai

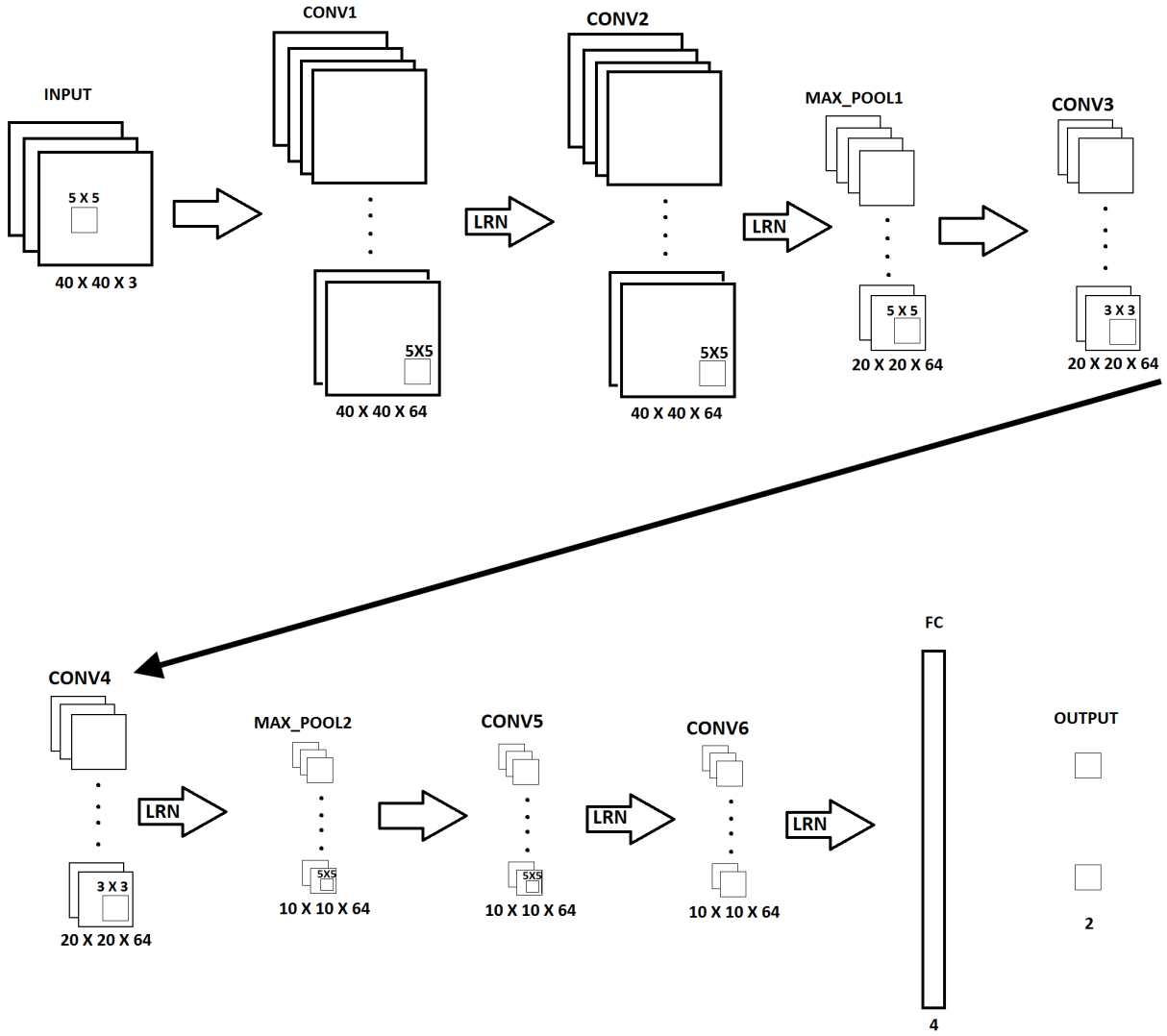
Be visų aukščiau aptartų parametrų egzistuoja dar viena aibė dydžių, kurie yra reguliuojami konvoliuciniame neuronų tinkle. Šie parametrai nekeičia konvoliucinio tinklo architektūros ir nedaro reikšmingos įtakos mokymosi procesui. Dėl to jie yra laikomi antraeiliais. Šių parametrų reikšmės yra fiksuotos visų eksperimentų metu. Jos gautos analizuojant literatūrą arba egzistuojančius kodo pavyzdžius. Tam, kad visi šio darbo eksperimentų rezultatai būtų atkartojami, šių parametrų reikšmės yra išvardijamos žemiau:

- Konvoliucinio sluoksnio svorių standartinis nuokrypis: 0.04.
- Konvoliucinio sluoksnio svorių irimas (angliškai *decay)*): 0.
- Konvoliucinio sluoksnio filtrų kiekis: 64.
- Konvoliucinio sluoksnio poslinkis: 0.1.
- Lokalaus atsako normalizavimo k: 4
- Lokalaus atsako normalizavimo α: 10^{-4}.
- Lokalaus atsako normalizavimo β: 0.75.
- Pilnai sujungto sluoksnio standartinis nuokrypis: 0.04.
- Pilnai sujungto sluoksnio svorių irimas: 0.004.
- Pilnai sujungto sluoksnio poslinkis: 0.1.
- Išvedimo sluoksnio standartinis nuokrypis: 0.25.

- Išvedimo sluoksnio svorių irimas: 0.
- Išvedimo sluoksnio poslinkis: 0.

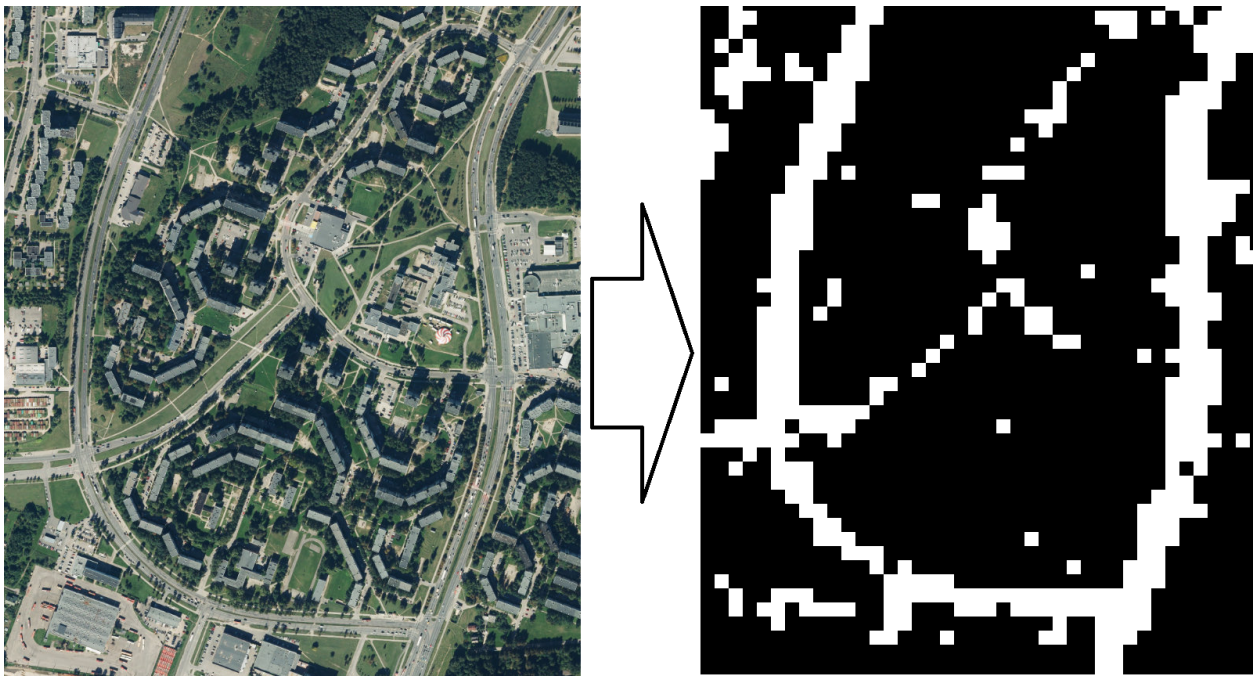
3.5. Po-apdorojimo algoritmai

Galutinis mokymosi proceso rezultatas yra apmokytas konvoliucinis neuronų tinklas turintis sureguliuotus svorius. Šie svoriai turi tokias reikšmes, su kuriomis klasifikuojant vaizdus buvo pasiektas mažiausias klaidų kiekis. Toks tinklas yra paruoštas įvertinimui - naujų, nematytų nuotraukų klasifikavimui bei rezultatų tikrinimui. Galutinio konvoliucinio neuronų tinklo architektūra iliustruota 9 pav.



9 pav. Galutinio konvoliucinio neuronų tinklo architektūra.CONV - konvoliuciniai sluoksniai, MAX_POOL - sutelkimo sluoksniai.

Kadangi šiame darbe naudojamas konvoliucinis tinklas yra klasifikuojamojo pobūdžio, jis geba pasakyti, ar konkrečiame nuotraukos fragmente yra matomas kelias, ar ne. Visi suklasifikuoti vienos nuotraukos fragmentai tuomet gali būti sujungti į naują nuotrauką tam, kad sukurtų pilną atpažinto kelio rekonstrukciją. Tačiau, norint kaip įmanoma tiksliau atkurti nuotraukoje matomą kelių tinklą, vien tik to nepakanka.



10 pav. Kairėje - analizuojamas kelių tinklas. Dešinėje - kelio rekonstrukcija, gauta apjungus suklasifikuotus kairiosios nuotraukos fragmentus.

Kaip matoma 10 pav., atpažintas kelių tinklas yra tikrojo kelių tinklo aproksimacija. Paprastas atpažintų fragmentų sujungimas sukuria pernelyg platų ir grubų kelio atvaizdą. Be to, rezultate galima pastebėti klaidingai teigiamai atpažintų fragmentų. Norint, kad atpažinimo rezultatas būtų tikslesnis, klasifikuotam vaizdui reikia pritaikyti papildomus po-apdorojimo algoritmus.

Po-apdorojimo algoritmų užduotis - „pasmulkinti“ konvoliucinio tinklo atpažinimo rezultatą tam, kad geriau atsiskleistų atpažinto kelio topologija. Be to, minėti algoritmai pasirūpina klaidingai teigiamų fragmentų eliminavimu. Būtent tokie algoritmai yra aptariami sekančiuose darbo poskyriuose. Šie algoritmai yra sugalvoti ir realizuoti paties autoriaus (neskaitant grafų teorijos minimalaus jungiančiojo medžio algoritmo, aptariamo 3.5.3 skirsnyje). Taip pat reikia pastebėti, kad visi tolesniuose skyriuose demonstruojami rezultatai yra pasiekti naudojant duomenis, kurių konvoliucinis tinklas nėra matęs apmokymo metu. Be to, kiekvienas naujas po-apdorojimo algoritmas operuoja su prieš tai buvusio algoritmo išvestimi.

3.5.1. Klaidingai teigiamų fragmentų šalinimas

Atkreipus dėmesį į 10 pav. kelių tinklą, galima pastebėti keletą klaidingai teigiamų fragmentų. Šie fragmentai pasireiškia pavieniais baltais kvadratais tose vietose, kur originalioje nuotraukoje nėra kelio. Šie fragmentai turi vieną pageidautiną ypatybę - jie nėra susijungę su likusiu rekonstruotu keliu. Dėl to, darbe yra sukurtas algoritmas, pašalinantis tokius fragmentus. Algoritmas susideda iš šių žingsnių:

1. Iš nuotraukos iškerpamas $N_{false} \times N_{false}$ dydžio fragmentas F_{frag}.
2. Susumuojami fragmente F_{frag} esantys pikseliai. Juodi pikseliai neprisideda prie sumos, nes jų reikšmė yra 0.
3. Jei suma viršija skaičių $P_{threshold}$ - į naują nuotrauką įklijuojamas fragmentas F_{frag}.

4. Priešingu atveju - į naują nuotrauką įklijuojamas $N_{false} \times N_{false}$ dydžio juodų pikselių fragmentas.

5. Slenkama per N_{false} pikselių į dešinę ir grįžtama į 1 žingsnį.

Parametrai N_{false} ir $P_{threshold}$ - derinami. Reikia pastebėti, kad šie parametrai gali ir nesutapti su konvoliucinio tinklo analizuojamų fragmentų dydžiu. Dėl to, šis algoritmas gali būti priderintas prie skirtingų vartotojo poreikių bei vaizdų.

Neišvengiamai, šis algoritmas gali panaikinti mažą dalį tikrojo kelių tinklo. Taip atsitinka jei fragmento F_{frag} pozicija sutampa su pozicija, kurioje randasi tikras kelių tinklas, tačiau fragmente baltų pikselių mažiau, nei $P_{threshold}$. Tai nesukelia didelės problemos, kadangi praktiniai algoritmo pritaikymai parodė, kad tikrojo kelio dalis, kuri prarandama, yra kraštas. Netgi priešingai - stambaus kelio kraštų praradimas yra teigiamas reiškinys, kadangi kelio smulkinimas yra pageidautinas rezultatas. Šį rezultatą pasiekti taip pat yra bandoma metodais, aprašytais sekančiuose skirsniuose.

3.5.2. Centrinių pikselių paieška

Vienas iš konvoliucinio tinklo klasifikavimo trūkumų - atpažinto kelio stambumas. Norint gauti smulkesnę kelio reprezentaciją, nuotraukoje pirmiausia reikia atrasti centrinius atpažinto kelio pikselius. Tam tikslui pasiekti buvo sukurtas algoritmas, pavadintas „centrine gyvate“ . Algoritmo užduotis - iteruoti per dvejetainį juodai-baltą (balta spalva - kelias, juoda - ne kelias) vaizdą ir atrasti kiekvieno kelio segmento centrinį pikselį. Tam tikslui pasiekti yra naudojama aktyvios arba ne aktyvios gyvatės būsena. Algoritmas susideda iš šių žingsnių:

1. Iteracija pradedama viršutiniame-kairiajame kampe.

2. Jei einamasis pikselis baltas:

   (a) Jei gyvatė neaktyvi - pradedama nauja gyvatė su ilgiu 1.

   (b) Jei gyvatė aktyvi - jos ilgis padidinamas 1.

3. Jei einamasis pikselis juodas:

   (a) Jei gyvatė neaktyvi - vykdomas žingsnis 4.

   (b) Jei gyvatė aktyvi - įrašoma gyvatės vidurio pikselio pozicija. Ji apskaičiuojama pagal einamojo pikselio koordinatę ir gyvatės ilgį. Gyvatė tampa neaktyvi.

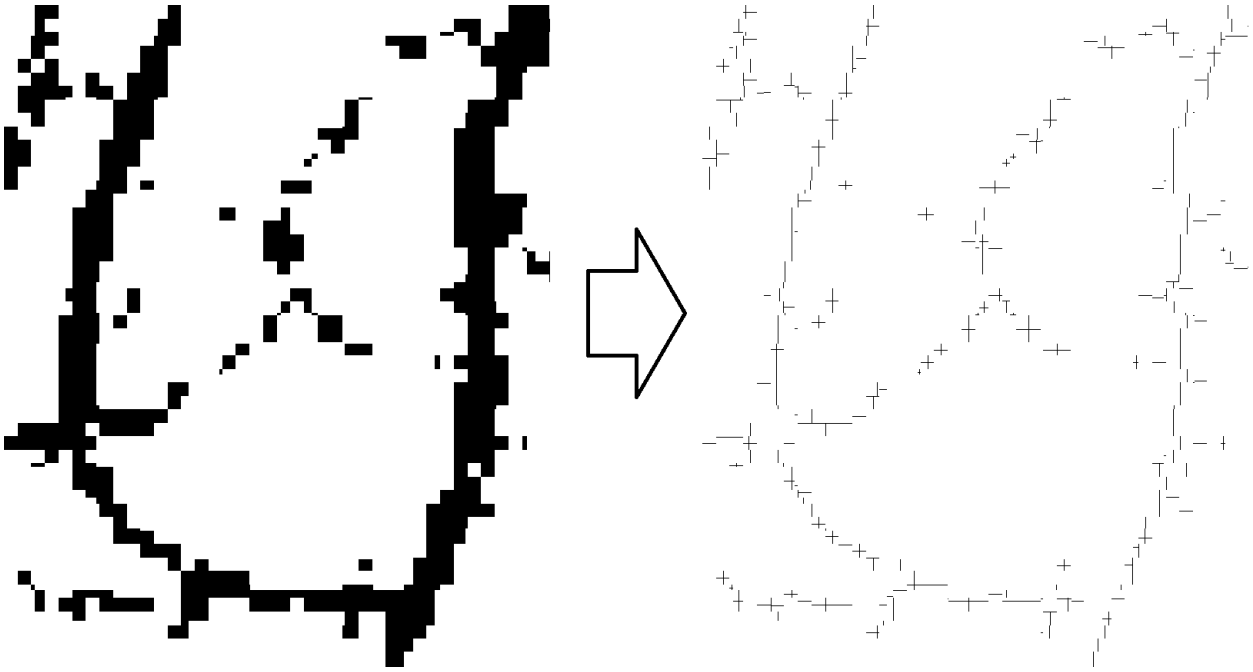
4. Pajudama vienu pikseliu į dešinę.

Algoritmo išvestis - gyvačių centrai. Viena gyvatė atitinka vienoje nuotraukos eilėje esančią baltų pikselių seką. Dėl to, gyvatės centras sutampa su pikselių sekos centru. Pritaikius šį algoritmą visoms nuotraukos eilėms yra pagaminami visų baltų pikselių sekų centrai.

Reikia atkreipti dėmesį, jog vienetinis šio algoritmo panaudojimas pagamina tik dalinai korektišką rezultatą. Dėl savo specifinio judėjimo, gyvatės žymi tik vertikalių kelio komponentų centrinius pikselius.

Tam, kad nuotraukoje matomas kelias būtų atpažintas korektiškai, neužtenka surasti vien tik jo vertikaliųjų dalių centrinius pikselius. Šie pikseliai neatvaizduoja horizontalių kelio dalių. Dėl to yra svarbu centrinės gyvatės algoritmą pakartoti antrą kartą su nuotrauka, pasukta 90 laipsnių kampu. Taip yra randami horizontalių kelio komponenčių centriniai pikseliai.

Horizontalių ir vertikalių centrinių pikselių atradimo pakanka tam, kad susidaryti pirminę smulkesnio kelio versiją. Taip yra todėl, kad tinklo suklasifikuotas vaizdas susideda iš didelio kiekio kvadratinių fragmentų. Kelio dalys, kurios yra pasuktos kokiu nors kampu, bet kokiu atveju yra sudarytos iš kvadratų. Dėl to, atradus horizontalius ir vertikalius centrinius pikselius, minėtųjų dalių topologija nėra prarandama. Apjungtus horizontalius ir vertikalius centrinius pikselius iliustruoja 11 pav.



11 pav. Kairėje - konvoliucinio neuronų tinklo išvestis. Dešinėje - apjungti vertikalūs ir horizontalūs centriniai pikseliai. Matomumo dėlei, vaizdų spalvos yra atvirkščios.

3.5.3. Minimalaus jungiantysis medis

Sujungus vertikalius ir horizontalius centrinius pikselius į vieną nuotrauką, susidaro smulkesnis tikrojo kelių tinklo vaizdas. Tačiau, tokio vaizdo dar negalima laikyti galutiniu rezultatu, kadangi tarp centrinių pikselių vis dar egzistuoja tarpai. Be to, apjungti centriniai pikseliai estetiškai neprimena kelio, o tik padrikas jo užuomazgas. Dėl to, atsiranda poreikis užpildyti tarpus tarp centrinių pikselių tam, kad atpažintas kelias neturėtų plyšių. Tuo tikslu, šiame darbe yra pasitelkiami grafų teorijos algoritmai.

Sujungtų centrinių pikselių nuotrauka yra interpretuojama kaip grafas, kurį sudaro tik viršūnės ir kuris neturi briaunų. Kadangi yra dirbama su vaizdais, grafo viršūnė yra apibrėžiama kaip pora (X, Y), kur X ir Y - pikselio koordinatės horizontalioje ir vertikalioje ašyje, atitinkamai. Žinoma, ne kiekvienas vaizdo pikselis yra grafo viršūnė. Vaizdo pikselis yra laikomas grafo viršūne, jei jis yra baltos spalvos ir atitinka vieną iš šių sąlygų:

1. Jis yra „pliuso“, gauto sujungus vertikalius ir horizontalius centrinius pikselius, viduryje.
2. Jis yra vertikalios centrinių pikselių sekos kraštas.

Remiantis šiomis prielaidomis, toliau yra atliekami šie veiksmai:

1. Pagal aukščiau apibrėžtas taisykles yra sukuriamas grafas G be briaunų.

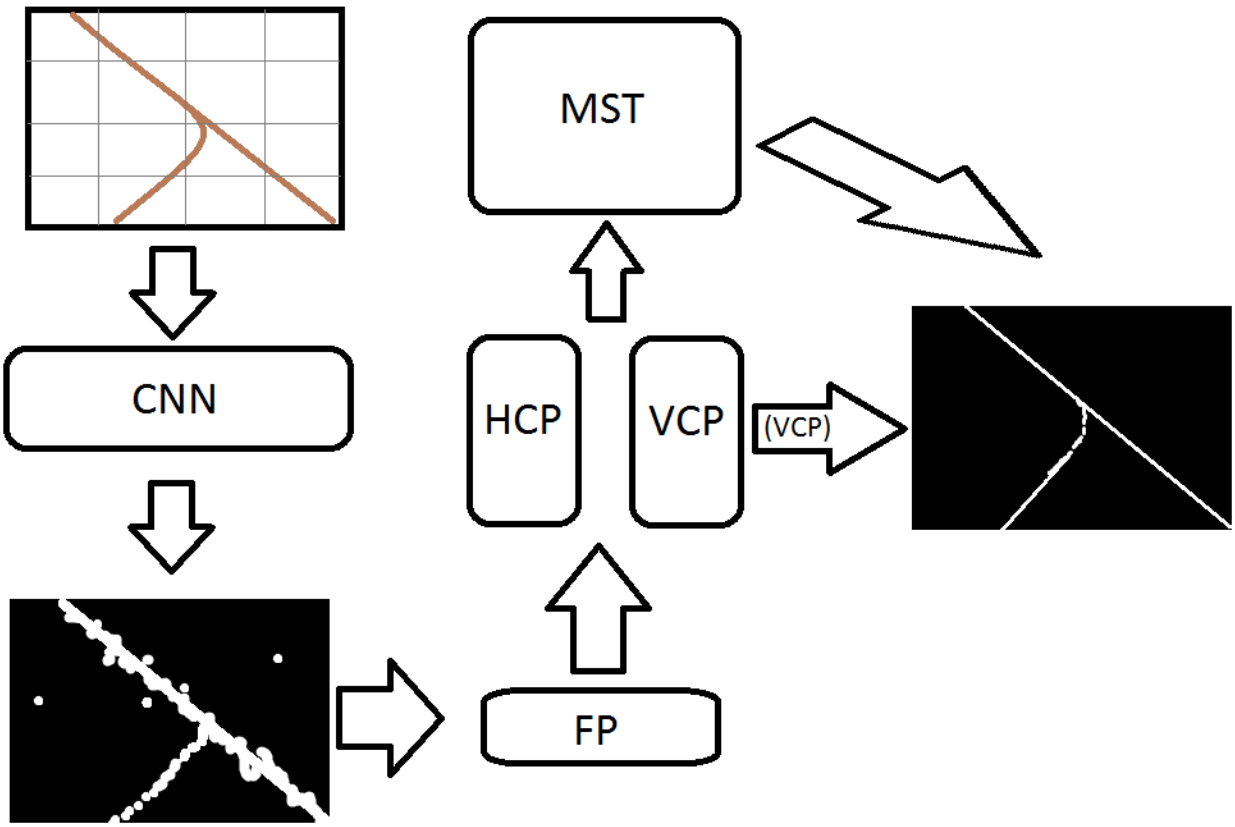
2. Kiekviena grafo viršūnė briaunomis yra sujungiama su visomis likusiomis grafo viršūnėmis, taip sukuriant pilną grafą G_p.
3. Kiekvienos grafo G_p briaunos svoriu tampa atstumas tarp briaunos viršūnių (pagal pikselių koordinates).
4. Grafui G_p yra pritaikomas minimalaus jungiančiojo medžio algoritmas, sukuriantis grafą G_m.
5. Grafo G_m vizualinė reprezentacija yra sujungiama su vertikaliais centriniais pikseliais.

Grafas G_m yra papildomai sujungiamas su vertikaliais centriniais pikseliais todėl, kad pats vienas nesukuria išbaigto kelių tinklo vaizdo. Analizuojant minimalaus jungiojo medžio algoritmo rezultatus buvo pastebėta, jog jame trūksta būtent didelio ilgio vertikalių kelio komponenčių. Dėl to, kaipo paskutinis po-apdorojimo žingsnis, buvo įvykdytas minėtas apjungimas. Šis apjungimas padėjo atkurti labiau išbaigtą kelių tinklą.

Šiame darbe yra naudojamas *Prim* algoritmas minimaliam jungiančiajam medžiui rasti [20]. Kito minimalaus jungiančiojo medžio algoritmo naudojimas nesukėlė reikšmingo rezultato pokyčio.

3.6. Rezultatų įvertinimas

Reziumuojant, visa šiame darbe sukurta ir išplėtota kelių atpažinimo algoritmų eiga yra iliustruota 12 pav.



12 pav. Kelių atpažinimo iš oro nufotografuotuose vaizduose eiga.

Ši eiga susideda iš tokių žingsnių:

1. Nuotrauka, kurioje yra kelias, padalijama į fragmentus.
2. Fragmentai pateikiami konvoliuciniam neuronų tinklui (*CNN*).
3. Konvoliucinis tinklas pateikia dvejetainį nuotraukos klasifikavimo rezultatą.
4. Klasifikavimo rezultate panaikinami klaidingai teigiami fragmentai (*FP*).
5. Randami horizontalūs ir vertikalūs centriniai pikseliai (*HCP* ir *VCP*).
6. Apjungtiems *HCP* ir *VCP* pritaikomas minimalaus jungiančiojo medžio algoritmas (*MST*).
7. *MST* apjungiamas su *VCP* tam, kad gauti galutinį rezultatą.

Tolimesniuose skirsniuose ši eiga yra pritaikoma keletui nuotraukų, kuriose bandoma atpažinti kelius. Algoritmas taikomas skirtingo tipo vaizdams - kaimo, miesto ir nežinomiems. Atpažinti keliai yra įvertinami pagal 3.7 poskyryje aprašytą metodiką. Reikia atkreipti dėmesį, jog pagrindinėje darbo dalyje yra įtraukta tik keleto nuotraukų atpažinimo įvertinimas. Papildomos įvertinimų nuotraukos yra priede A.

3.7. Kelių atpažinimo rezultatų įvertinimas

Nepriklausomai nuo turimų vaizdų ir kelių juose tipo, yra svarbu kokybiškai įvertinti atliktų eksperimentų rezultatus. Tam, kad tai padaryti, reikia turėti tikrus duomenis (angliškai *ground truth*), kuriuose yra tiksliai (pavyzdžiui, rankiniu būdu) surasti visi keliai ir atlikti keletą palyginimų. Kelių atpažinimo eksperimentų įvertinimo būdas yra aprašytas [23]. Darbo autoriai siūlo vertinimą atlikti dviem etapais - suderinimu (angliškai *matching*) ir kokybės analize.

Suderinimo etape abu kelių tinklai - tikrasis ir atpažintas - yra padalinami į trumpus, vienodo ilgio, segmentus. Tuomet aplink tikrojo tinklo segmentus yra konstruojamas tam tikro pločio buferis (buferio plotis - algoritmo parametras). Atpažinto tinklo segmentai, kurių kryptis nesiskiria nuo tikrų segmentų daugiau negu nustatyta riba (irgi algoritmo parametras) yra laikomi suderintais. Suderinti atpažinto tinklo segmentai yra laikomi teisingai teigiami (angliškai *true positive*) ir pilnas jų ilgis yra pažymimas TP. Atpažinto tinklo segmentai, kurie nėra suderinti yra laikomi klaidingai teigiamais (angliškai *false positive*), kadangi tose vietose kur algoritmas rado kelią, tikro kelio nėra. Pilnas klaidingai teigiamų segmentų ilgis žymimas FP.

Suderinimo etapas turi antrą žingsnį, kuriame yra atliekami tokie patys veiksmai, tik šįkart buferis yra konstruojamas aplink atpažinto kelių tinklo segmentus ir prie jų yra derinami tikro tinklo segmentai. Nesuderinti tikro kelio segmentai yra laikomi klaidingai neigiamais (angliškai *false negative*) ir pilnas jų segmentų ilgis pažymimas FN.

Antrajame vertinimo etape yra pateikiami atpažinimo algoritmo kokybę nusakantys parametrai:

- Išbaigtumas $= \frac{TP}{TP + FN}$;
- Korektiškumas $= \frac{TP}{TP + FP}$;
- Kokybė $= \frac{TP}{TP + FP + FN}$;
- RMS - vidutinis atstumas tarp atpažintų ir tikrų kelių ;

- Spragos - spragų kiekis viename atpažintų kelių kilometre.

Šis vertinimo būdas yra sukurtas vektorinių duomenų palyginimui. Dirbant tik su rastriniais vaizdais jis nėra pilnai pritaikomas. Dėl to, šiame darbe yra naudojamas kiek pamodifikuotas vertinimo metodo variantas. Atliktos modifikacijos priklauso nuo konkrečių šio darbo atpažinimo rezultatų pavidalo.

Modifikuotas algoritmas lygiagrečiai slenkančių langu iteruoja dvi nuotraukas - specialų tikrų kelių žemėlapį (žr. 5 pav.) ir galutinį kelių atpažinimo rezultatą. Iteravimo metu yra atliekami šie veiksmai:

1. Apskaičiuojama einamojo slenkančio lango pikselių suma tikruose duomenyse S_R.
2. Apskaičiuojama einamojo slenkančio lango pikselių suma klasifikuotuose duomenyse S_C.
3. Jei $S_R > 0$ ir $S_C > 0$ - padidinamas TP - teisingai teigiamų fragmentų - skaitliukas.
4. Jei $S_R = 0$ ir $S_C = 0$ - padidinamas TN - teisingai neigiamų fragmentų - skaitliukas.
5. Jei $S_R = 0$ ir $S_C > 0$ - padidinamas FP - klaidingai teigiamų fragmentų - skaitliukas.
6. Jei $S_R > 0$ ir $S_C = 0$ - padidinamas FN - klaidingai neigiamų fragmentų - skaitliukas.
7. $N_{eval} \times N_{eval}$ dydžio slenkantis langas per S_{eval} pikselių pajuda į dešinę abiejose nuotraukose.

N_{eval} ir S_{eval} - reguliuojami algoritmo parametrai. Kadangi vaizdai, kuriais iteruojama, yra dvejetainiai, juodi pikseliai neprisideda prie bendros fragmento sumos. Kokybiniai įverčiai yra apskaičiuojami pagal tokias pat formules, kaip aukščiau.

Šis metodas negali pateikti dviejų originalaus algoritmo statistikų - atstumo ir spragų kiekio. Taip yra dėl to, kad rastriniai duomenys neturi erdvinės informacijos, kuri yra reikalinga, norint pateikti šiuos dydžius.

Prieš konkrečių darbe gautų rezultatų analizę reikia aptarti kokybinių įverčių interpretavimą. Bet kuris įvertis, ar tai būtų išbaigtumas, korektiškumas, ar kokybė, yra laikomas aukštu, jei jis peržengė 75% ribą. Dėl to, šią sąlygą tenkinantys įverčiai rezultatų analizėje įvardijami kaip aukšti. Įvertis, kurio intervalas yra nuo 50% iki 75% yra laikomas vidutinišku. Įvertis, žemesnis nei 50% yra laikomas žemu.

3.7.1. Kaimo kelių atpažinimo rezultatai

Kelias, atpažintas 13 pav. pasižymi tokiais kokybės įverčiais:

- Išbaigtumas: 85,48 %.
- Korektiškumas: 70,06 %.
- Kokybė: 63,09 %.

Kaip ir tikėtasi, algoritmų pritaikymas kaimo keliams pasiekia aukštus arba vidutinius kokybinius įvertinimus. Tačiau, net ir šie rezultatai nėra idealūs. Pagrindinė problema kaimo kelių atpažinime yra klaidingai teigiami fragmentai. Tai paliudija aukštas išbaigtumo įvertis ir žemesnis kokybės įvertis. Pagal 3.7 poskyrio formules, išbaigtumą mažina klaidingai neigiami fragmentai,

13 pav. Atpažintas kaimo kelių tinklas, perdengtas virš originalios nuotraukos.

o kokybę - klaidingai teigiami. Dėl to yra aišku, kad būtent klaidingai teigiamų fragmentų kiekis, šiuo atveju, yra problema.

Klaidingai teigiamų fragmentų buvimas kaimo keliuose yra paaiškinamas tuo, jog kaimo kelio spalva nuotraukose sutampa su kitokių objektų - laukų, dirbamų žemės plotų ir pan. - spalva. Be to, skirtingai nei miesto nuotraukose, vienodos spalvos objektai kaimo vietovėse nesukuria šešėlių. Šešėlių buvimas miesto miesto nuotraukose leidžia tos pačios spalvos objektus traktuoti kaip kitas ypatybes. Šešėlių pikseliai užtikrina, jog konvoliucijos rezultatas tokios pačios spalvos ypatybei bus skirtingas . Dėl to, konvoliucinis tinklas sugeba atskirti kelią nuo, pavyzdžiui, namo. Nesant jokiems šešėliams būna sunkiau atskirti gelsvą dirbamo lauko plotą nuo kelio. Būtent tokius artefaktus - kuomet kelio spalvos objektas yra klaidingai pažymimas keliu - galima pastebėti 13 pav.

3.7.2. Miesto kelių atpažinimo rezultatai

Kelias, atpažintas 14 pav. pasižymi tokiais kokybės įverčiais:

- Išbaigtumas: 46,32 %.
- Korektiškumas: 85,41 %.
- Kokybė: 42, 93 %.

Aukštas korektiškumo įvertinimas liudija, kad algoritmas sugeneruoja nedaug klaidingai teigiamų fragmentų. Vadinasi, algoritmas sėkmingai neklasifikuoja tų vietų, kuriose kelio iš tikrųjų nėra. Žemesni išbaigtumo ir kokybės rezultatai parodo, kad algoritmą reikia tobulinti su tikslu, kad jis atpažintų mažiau klaidingai neigiamų fragmentų. T.y. - algoritmas atpažino nepakankamai nuotraukoje esančių kelių.

Pagrindinis algoritmo trūkumas - klaidingai neigiamų fragmentų buvimas - yra sukeliamas dėl specifinio neuronų tinklo išvesties pavidalo. Specialiame žemėlapyje kelių kreivės yra glodžios.

14 pav. Atpažintas miesto kelių tinklas, perdengtas virš originalios nuotraukos.

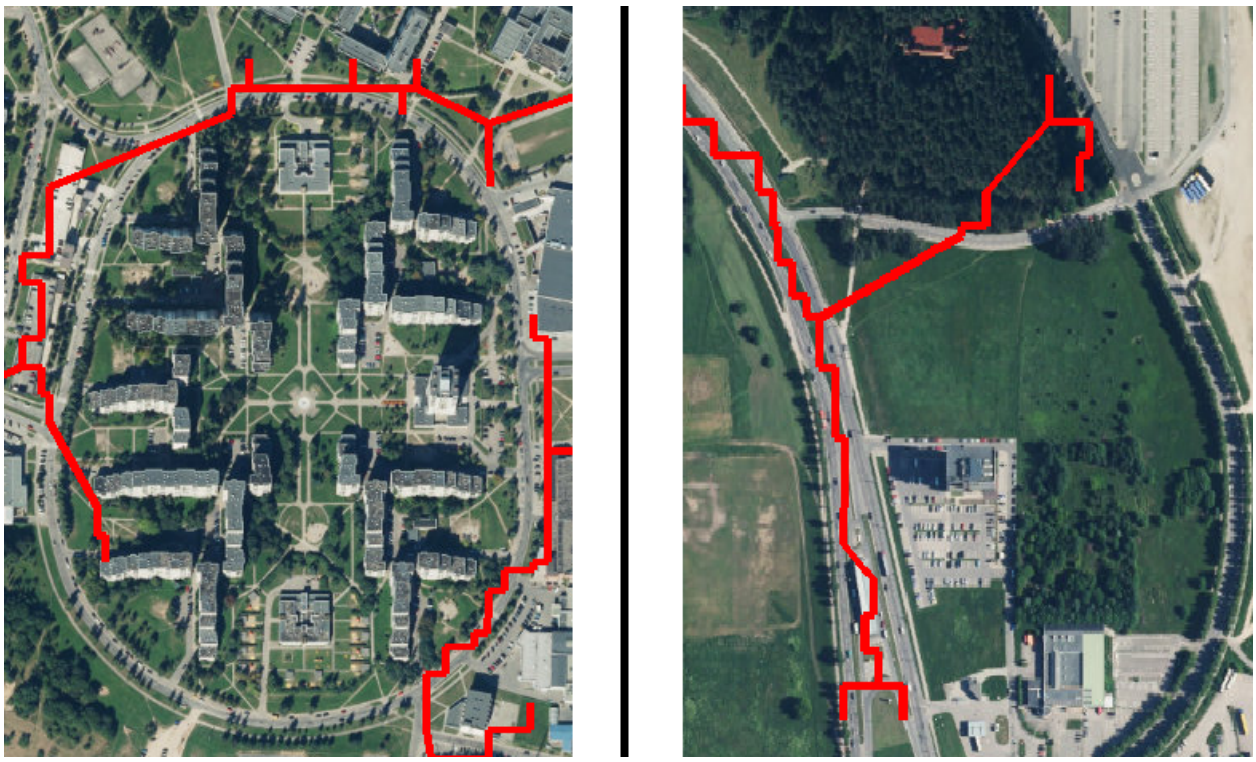
Konvoliucinis tinklas išvedą šių kreivių aproksimaciją, sudėtą iš atskirų fragmentų. Šių fragmentų dydis yra daug didesnis, negu originalių kreivių plotis. Po-apdorojimo algoritmai plonina stambius fragmentus ir iš jų kuria grafą. Šios eigos metu atsiranda tikimybė, jog galutinių kreivių absoliuti pozicija bus pasislinkusi, lyginant su originalių kreivių pozicija. Tokį artefaktą galima pastebėti 14 pav., atkreipus dėmesį dešinį-viršutinį kampą. Atpažinta vertikali kreivė yra kiek pasislinkusi į kairę, lyginant su originaliu keliu. Vizualiai, tai sukelia nedidelį susierzinimą, tačiau griežtas vertinimo algoritmas, analizuodamas tikrą kelio fragmentą, nekreipia dėmesio į kreivę, esančią „beveik“ ant kelio.

Klaidingai neigiami fragmentai taip pat pasireiškia tose vietose, kuriose yra uždara kelio kilpa. To priežastis yra minimalaus jungiančiojo medžio algoritmo taikymas visam kelio grafui. Šio algoritmo užduotis yra sukurti beciklį grafą. Ši mintis kertasi su fizine kelio topologija, kurioje tikrai pasitaiko kilpų. Dėl to, neatpažinta kelio kilpos dalis yra pažymima kaip klaidingai neigiamas fragmentas. Tokį rezultatą iliustruoja 15 pav. kairė dalis.

Kitas, mažesnis algoritmo trūkumas yra klaidingai teigiamų fragmentų buvimas. Jo nebuvo išvengta iki galo, nepaisant to, kad vieno iš po-apdorojimo žingsnių užduotis yra tokių fragmentų pašalinimas. Klaidingai teigiamų fragmentų šalinimo žingsnis ieško pavienių fragmentų, neturinčių jungčių su pagrindiniu keliu. Dėl to, klaidingai teigiami fragmentai, kurie nėra pavieniai, arba kurie randasi arti pagrindinio kelio tinklo, nėra pašalinami. Grafo kūrimo metu jie yra paverčiami grafo viršūnėmis. To pasekoje, šie fragmentai sudaro jungtis su sau artimiausiomis grafo viršūnėmis. Dėl to atsiranda artefaktas, kuomet atpažintas kelias nesutampa su tikruoju keliu. Šį artefaktą iliustruoja 15 pav. dešinioji dalis.

3.7.3. Nežinomų kelių atpažinimo rezultatai

Aukščiau įvertintos nuotraukos yra panašios kategorijos, kaip ir apmokymo duomenys. Jos, žinoma, nėra tos pačios nuotraukos, su kuriomis tinklas buvo mokomas, tačiau jų ypatybės panašios. Tiek apmokymo, tiek įvertinimo duomenyse objektų spalvos yra panašios - kaimo kelias yra gels-



15 pav. Problematiški miesto kelių atpažinimo rezultatai. Kairėje - klaidingai neigiamai atpažinta kelio kilpa. Dešinėje - viršutinėje dalyje klaidingai teigiamai atpažintas kelio fragmentas.

vos spalvos, aplink kelią - žalia žolė ir pan. Be to, apmokymo ir įvertinimo nuotraukų kontrastai yra panašūs.

Išbaigtumo dėlei, apmokytą tinklą vertėtų įvertinti ir su visiškai skirtingos kategorijos duomenimis. Savaime suprantama, kad tinklas, kuris apmokytas atpažinti kelius, neatpažins katės ar šuns. Tačiau, galima įvertinti, kaip tinklas sugeba atpažinti visiškai nežinomo tipo kelius - tokius, kurių spalva ir aplinkos kontrastas skiriasi nuo jam matytų.

Pirmoje nuotraukoje (žymima A) kelio spalva sutapo su mokymo duomenų kelio spalva, tačiau skyrėsi kelią supančios aplinkos spalva - ji buvo nebe žalia, o rusva. Antroje nuotraukoje (žymima B) skyrėsi tiek kelio, tiek aplinkos spalvos ir, papildomai, tarp šių dviejų objektų buvo parinktas mažesnis kontrastas. Kaip ir tikėtasi, visiškai skirtingos nuotraukos atpažinimas nepasiekė tokių rezultatų, kaip nuotraukos, kuri yra panašesnė į apmokymo duomenis. A ir B nuotraukos yra iliustruotos 16 pav., kairėje ir per vidurį, atitinkamai. A ir B nuotraukų kokybiniai įverčiai randasi 2 lentelėje,

Miesto kelių atveju, atpažinimui buvo parinkta tokia nuotrauka, kurioje kelio spalva sutapo su mokymo duomenų kelio spalva, tačiau pastatų ir augmenijos spalvos skyrėsi (žymima C). Be to, įvertinimo nuotraukoje specialiai buvo parinktas labai tankus miesto fragmentas. Tokiu būdu yra siekiama maksimaliai ištestuoti algoritmų ribas. Šios nuotraukos kokybiniai įverčiai taip pat randami 2 lentelėje.

Nors miesto nuotraukos C kokybiniai įvertinimai nėra perdėtai blogi, jie prasilenkia su realybe. Atkreipus dėmesį į 16 pav. dešiniausią dalį, matosi, jog atpažintas kelių tinklas daugelyje vietų neatitinka tikrojo. Vidutiniškas kokybės įverčių reikšmes galima pagrįsti dideliu kelio tankumu. Kai originalus kelių tinklas yra tankus, atpažinimo rezultatas taip pat yra tankus grafas. Kadangi įvertinimo algoritmas operuoja pikselių sumomis tam tikro dydžio buferyje, tankaus tinklo ir tankaus grafo derinys sugeneruoja didelį kiekį teisingai teigiamų reikšmių. Tokiu atveju, tik atlikus

empirinę analizę galima pastebėti, jog „teisingos“ reikšmės neatitinka realybės.

16 pav. Nežinomų kelių atpažinimo rezultatų fragmentai. Kairėje - kaimo kelias, kurio aplinkos spalva nematyta. Viduryje - kaimo kelias, kurio visų objektų spalvos nematytos. Dešinėje - tankaus miesto kelias, kurio aplinkos spalvos nematytos.

2 lentelė. Nežinomų kelių kokybiniai įverčiai.

Nuotrauka	Išbaigtumas	Korektiškumas	Kokybė
A	68,87%	87,39%	62,65%
B	35,52%	40,60%	23,37%
C	57,10%	64,32%	43,37%

Išvados ir rekomendacijos

Atlikus plačios apimties eksperimentus buvo sukurtas konvoliucinis neuronų tinklas, kurio architektūra yra specialiai pritaikyta kelių atpažinimui iš oro nufotografuotuose vaizduose. Šis konvoliucinis tinklas, apjungtas su sukurtais po-apdorojimo algoritmais, demonstruoja aukštos kokybės rezultatus kaimo kelių atpažinime. Tą liudija pasiekti 85% išbaigtumo ir 70% kokybės rezultatai kaimo keliuose (žr. 3.7.1 poskyrį). Sukurtas metodas dalinai sugeba atpažinti netgi visiškai nežinomą kaimo kelią, kas demonstruoja gerą algoritmų prisitaikymą prie visiškai naujų užduočių.

Naudojami metodai taip pat yra taikytini vidutinio tankumo miesto keliams atpažinti. Tą liudija pasiekti 85% korektiškumo ir 46% išbaigtumo įverčiai. Miesto kelių atveju, realizuoti metodai veikė kiek prasčiau tose vietose, kuriose yra kelio kilpos (žr. 3.7.2 poskyrį).

Darbe yra nuodugniai ištirta įvairių konvoliucinio neuronų tinklo parametrų įtaka mokymosi procesui. Yra atrasta, jog kai kurių parametrų derinimas pagerina mokymosi proceso rezultatus(žr. 3.2.1 ir 3.2.5 poskyrius). Darbe taip pat yra atrasta, jog tam tikri tinklo parametrai nepagerina, arba pablogina mokymosi proceso rezultatą (žr. 3.2.2 ir 3.2.3 poskyrius). Tie parametrai, kurie sėkmingai pagerina mokymosi rezultatus, atitinka visus išsikeltus kriterijus:

- Pakeičia mokymosi kreivės mažėjimo tendenciją į eksponentinę.
- Sumažina globalų mokymosi kreivės minimumą.
- Sumažina mokymosi kreivės osciliavimo amplitudę.

Darbe realizuoti po-apdorojimo algoritmai sugebėjo pagerinti galutinį atpažinimo rezultatą. Jų dėka buvo pašalinti akivaizdžiausi klaidingai teigiami fragmentai. Po-apdorojimo algoritmai taip pat sėkmingai paplonino konvoliucinio neuronų tinklo išvedimą, o minimalaus jungiančiojo medžio algoritmas sugebėjo užpildyti spragas tarp nesujungtų kelio segmentų.

Tačiau, po-apdorojimo algoritmai neišsprendė visų kelio atpažinimo problemų. Atpažinti keliai vis dar pasižymi grubiu susijungimu tarp greta esančių segmentų. Tai atpažintam keliui suteikia netolydžią kreivių išvaizdą. Be to, minimalaus jungiančiojo grafo algoritmo nepakanka tam, kad atpažinti kilpas keliuose. Dėl to, ypač miestų keliuose, atsiranda klaidingai neigiamų atpažinimo tarpų.

Apibendrinant - konvoliuciniai neuronų tinklai yra neabejotinai tinkamas instrumentas kelių atpažinimui. Kalbant apie kaimo kelius, darbe išsikeltas tikslas yra laikomas pasiektu. Miesto kelių atveju, darbo tikslas laikomas dalinai pasiektu, kadangi dar yra akivaizdžių algoritmų tobulinimo krypčių.

Ateities tyrimų gairės

Norint papildomai pagerinti konvoliucinio neuronų tinklo išvedimą, galima bandyti taikyti pilnai konvoliucinius tinklus. Šie tinklai išskirtiniai tuo, kad yra pritaikyti spręsti semantinio segmentavimo uždavinį. Semantinio segmentavimo uždavinio tikslas yra sugrupuoti bendras savybes turinčius nuotraukos pikselius į segmentus. Tokio tinklo išvedimas gali būti naudojamas kaip papildomas informacijos apie nuotrauką šaltinis atliekant klasifikavimą.

Tam, kad neieškoti konvoliucinio tinklo architektūros rankiniu būdu, galima realizuoti augančių topologijų (angliškai *augmented topologies*) metodus. Augančių topologijų esmė yra genetinio algoritmo pagrindu evoliucionuoti neuronų tinklus. Evoliucijos metu yra keičiami ne tik tinklų svoriai, bet ir tinklų architektūros. Tai leistų dar labiau optimizuoti konvoliucinio tinklo architektūrą.

Galiausiai, rekomenduojama po galutinio rezultato išvedimo pridėti dar vieną po-apdorojimo žingsnį, kurio užduotis būtų uždarų kilpų kelyje atpažinimas. Šį tikslą taip pat galima pasiekti apmokant papildomą konvoliucinį tinklą, kurio užduotis būtų būtent tokių kilpų atpažinimas.

Literatūros šaltiniai

[1] Convolutional neural networks | tensorflow. https://www.tensorflow.org/tutorials/deep_cnn.

[2] Geoportal.lt žemėlapis. http://www.geoportal.lt/map.

[3] Georeferencinių duomenų rinkinys gdr10lt. http://www.geoportal.lt/map#portalAction=openService&serviceUrl=http://menuo:8399/arcgis/rest/services/NZT/GDR10LT/MapServer.

[4] Google Žemėlapiai. https://maps.google.com.

[5] Neural network | tensorflow. https://www.tensorflow.org/versions/master/api_guides/python/nn.

[6] D. Anguelov, D. Erhan, Y. Jia, W. Liu, A. Rabinovich, S. Reed, P. Sermanet, C. Szegedy, and V. Vanhoucke. Going deeper with convolutions. In *CVPR*, 2015.

[7] R. Bajcsy and M. Tavakoli. Computer recognition of roads from satellite pictures. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6(9):623–637, 1976.

[8] A. Baumgartner, W. Eckstein, I. Laptev, T. Lindeberg, H. Mayer, and C. Steger. Automatic extraction of roads from aerial images based on scale-space and snakes. Technical report, 2000.

[9] Y. Bengio, A. Courville, and Ian Goodfellow. Deep learning. Book in preparation for MIT Press, 2016.

[10] R. Fergus and M. D. Zeiler. Visualizing and understanding convolutional networks. In *Computer Vision - ECCV*, pages 818–833. Springer, 2014.

[11] M.-F. Auclair Fortier, D. Ziou, C. Armenakis, and S. Wang. Survery of work on road extraction in aerial and satellite images. Technical report, Département de mathématiques et informatique, Université de Sherbrooke, 2013.

[12] A. Grigorjevas. Kelių atpažinimas iš oro nufotografuotuose vaizduose. Mokslo tiriamojo darbo projektas, Vilniaus Universitetas, 2017.

[13] K. He, S. Ren, J. Sun, and X. Zhang. Deep residual learning for image recognition. In *CVPR*, 2016.

[14] S. Hijazi, R. Kumar, and C. Rowen. Using convolutional neural networks for image recognition. Technical report, IP Group, Cadence, 2016.

[15] G. E. Hinton, A. Krizhevsky, and I. Sutskever. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems 25*, pages 1106–1114, 2012.

[16] G. E. Hinton and V. Mnih. Learning to detect roads in high-resolution aerial images.

[17] J. Johnson, F. Li, and S. Yeung. Cs231n: Convolutional neural networks for visual recognition. http://cs231n.github.io/.

[18] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[19] R. Briggs Y. Li. Automatic extraction of roads from high resolution aerial and satellite images with heavy noise. *International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:3, No:6*, pages 1571–1577, 2009.

[20] R. C. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36(6):1389–1401, 1957.

[21] B. Sirmaçek and C. Ünsalan. Road detection from remotely sensed images using color features. In *Recent Advances in Space Technologies (RAST), 2011 5th International Conference*, pages 112–115. IEEE.

[22] P. Veličković. Deep learning for complete beginners: convolutional neural networks with keras. https://s3-eu-west-1.amazonaws.com/com.cambridgespark.content/tutorials/convolutional-neural-networks-with-keras/figures/convolve.png.

[23] C. Wiedemann, C. Heipke, H. Mayer, and O Jamet. Empirical evaluation of automatically extracted road axes. *CVPR Workshop of Empirical Evaluation Methods in Computer Vision*, pages 172–187, 1998.

[24] J. Yuan and A. M. Cheriyadat. Road segmentation in aerial images by exploiting road vector data. In *Computing for Geospatial Research and Application (COM.Geo), 2013 Fourth International Conference*, pages 16–23. IEEE.

Priedai

Dokumentą sudaro du priedai: A priede yra iliustruojama daugiau kelių atpažinimo rezultatų, kurie nėra įtraukti į pagrindinę darbo dalį. B priede yra iliustruojamos apmokymo kreivės, kurios pablogina arba nekeičia konvoliucinio tinklo rezultatų.

A. Papildomi kelių atpažinimo rezultatai

Šiame priede yra pateikiamos papildomos iliustracijos, vaizduojančios galutinius algoritmų rezultatus. Šie rezultatai yra perdengti virš originalių analizuojamų nuotraukų. Priede yra pateikiama po du atpažintus kaimo ir miesto kelius.

17 pav. Atpažinto kaimo kelio rezultatas.

Vaizdas, matomas 17 pav. pasižymi tokiais kokybiniais įverčiais:

- Išbaigtumas: 82,02%.
- Korektiškumas: 85,88%.
- Kokybė: 72,27%.

Vaizdas, matomas 18 pav. pasižymi tokiais kokybiniais įverčiais:

- Išbaigtumas: 83,17%.
- Korektiškumas: 58,94%.
- Kokybė: 52,66%.

Vaizdas, matomas 19 pav. pasižymi tokiais kokybiniais įverčiais:

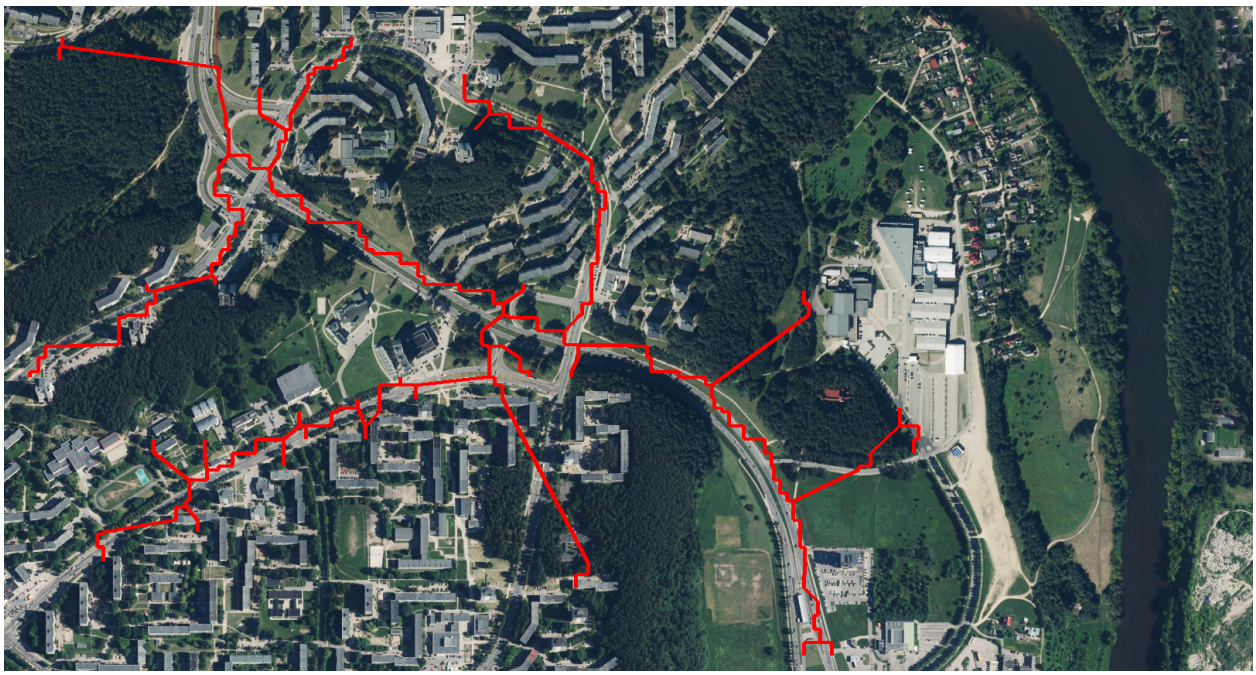
- Išbaigtumas: 50,34%.
- Korektiškumas: 80,66%.
- Kokybė: 44,92%.

Vaizdas, matomas 20 pav. pasižymi tokiais kokybiniais įverčiais:

18 pav. Atpažinto kaimo kelio rezultatas.

19 pav. Atpažinto miesto kelio rezultatas.

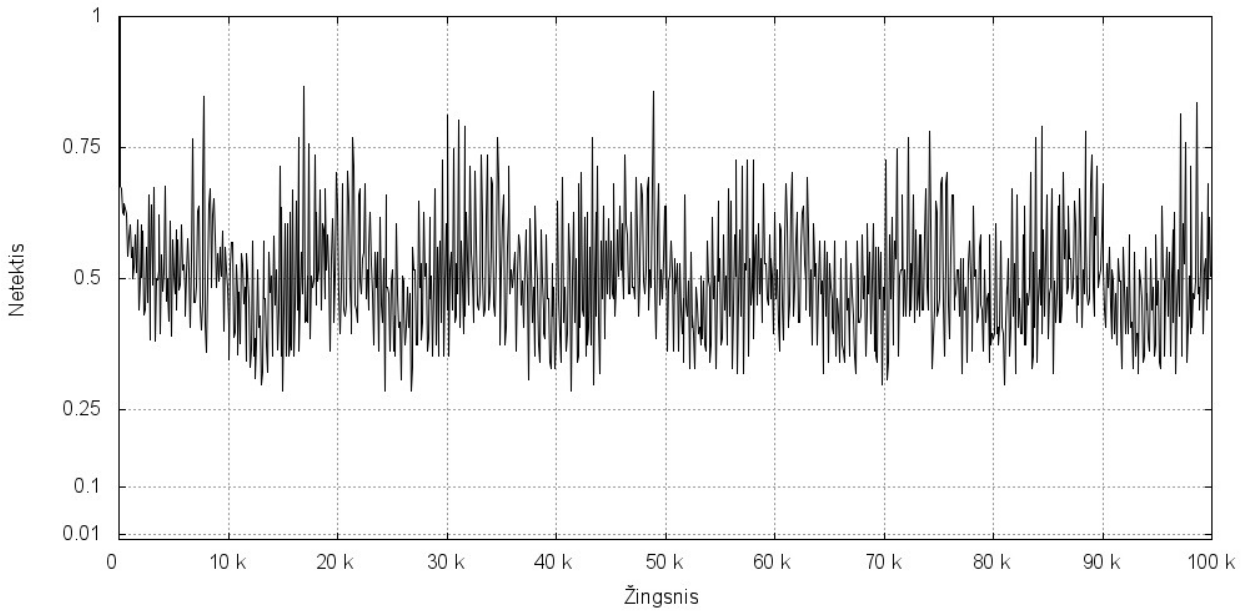
- Išbaigtumas: 35,44%.
- Korektiškumas: 83,10%.
- Kokybė: 33,06%.



20 pav. Atpažinto miesto kelio rezultatas.

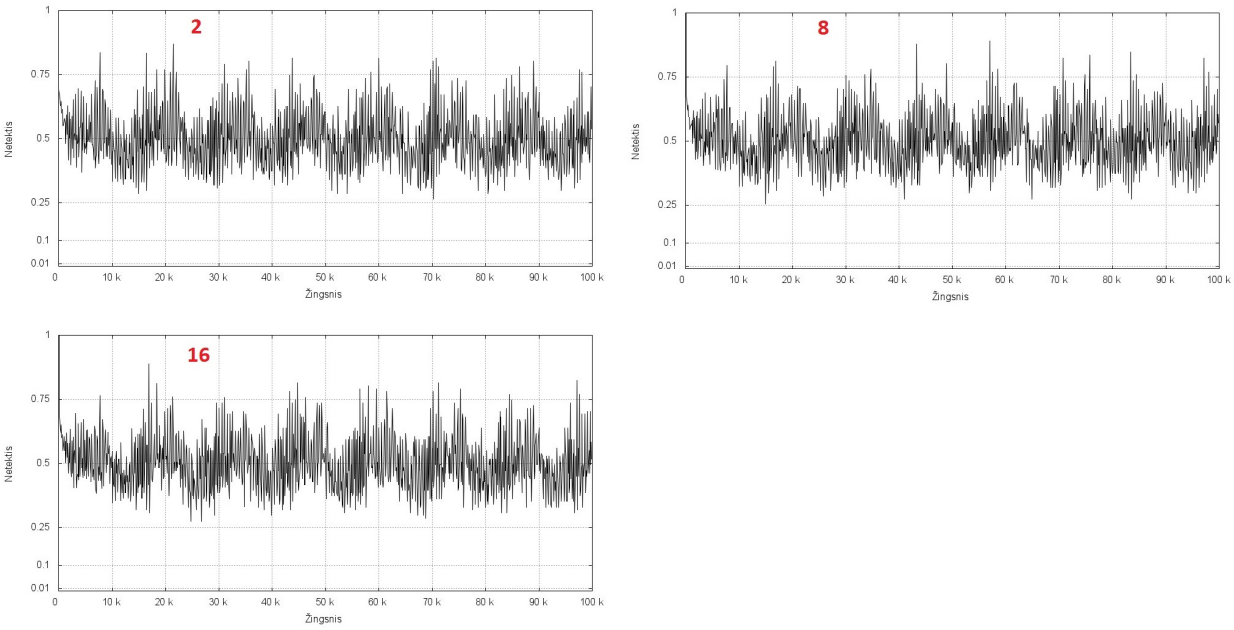
B. Papildomos apmokymų kreivės

Šiame priede yra pateikiamos keleto eksperimentų apmokymų kreivės. Šios kreivės daro arba neigiamą, arba nežymią įtaką konvoliucinio tinklo apmokymui ir neturi kokių nors ypatingų savybių. Dėl to, jos nėra pateikiamos pagrindinėje darbo dalyje.

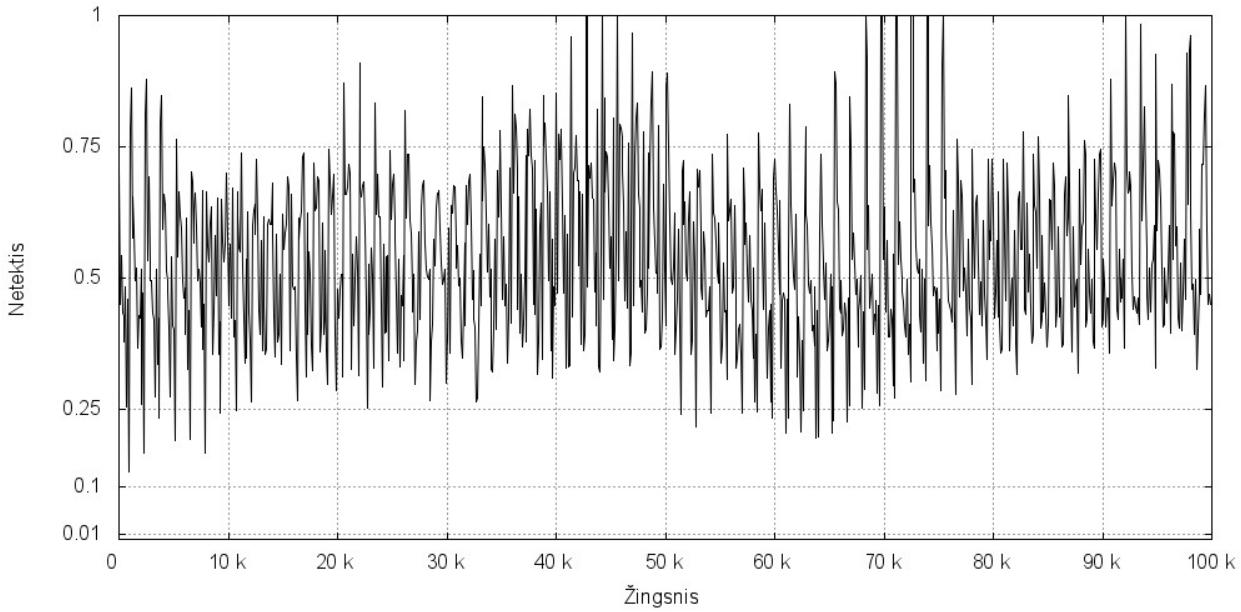


21 pav. Apmokymo kreivė, gauta pakeitus papildymo nuliais metodą.

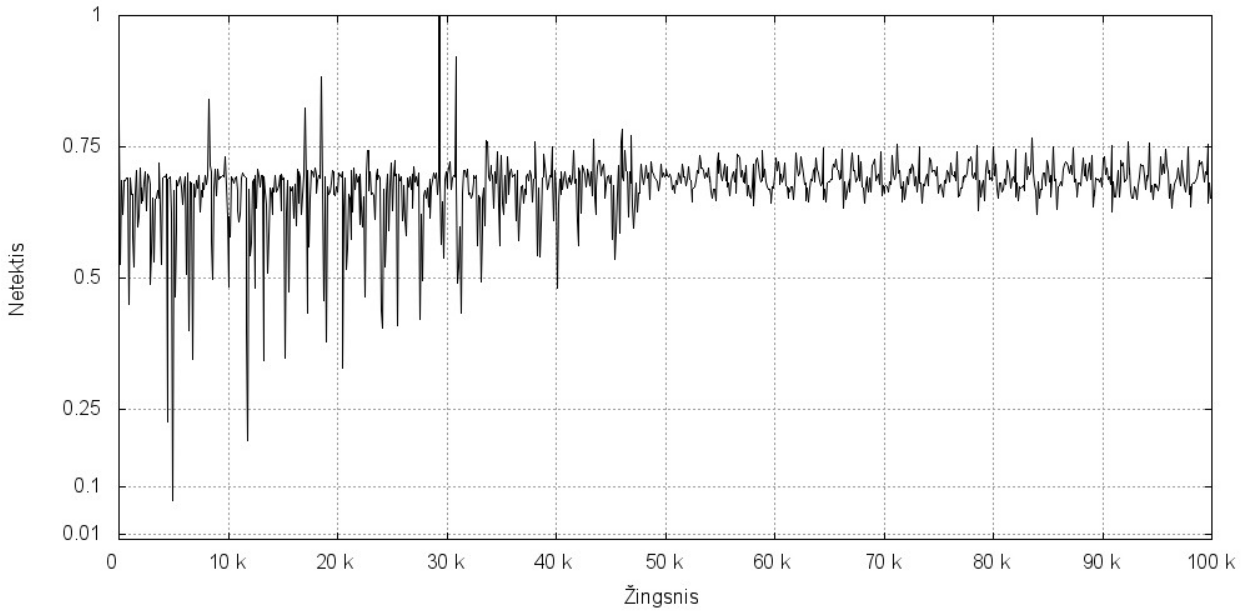
d



22 pav. Apmokymo kreivės, gautos pilnai sujungtame sluoksnyje panaudojus 2, 8 ir 16 neuronų.



23 pav. Apmokymo kreivė, pradiniam greičiui priskyrus reikšmę 10^{-2}.



24 pav. Apmokymo kreivė, pradiniam greičiui priskyrus reikšmę 10^{-4}.