

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

**Giliųjų neuroninių tinklų taikymas miškų ir vandens
telkinių satelitiniuose vaizduose analizei**

**Application of deep neural network for forests and water objects
in satellite images analysis**

Magistro baigiamasis darbas

Atliko:	Valdemar Karasevič	(parašas)
Darbo vadovas:	Prof. dr. Olga Kurasova	(parašas)
Recenzentas:	Prof. dr. Linas Laibinis	(parašas)

Vilnius – 2020

Santrauka

Šiame darbe nagrinėjamas giliųjų neuroninių tinklų taikymas satelitinių vaizdų analizei. Dėl įvairių natūralių procesų ir žmogaus veiksmų miškai nyksta, bei keičiasi vandens telkinių ribos. Šiuos pokyčius būtina stebėti, kad būtų galima sustabdyti kenksmingus procesus. Be to, tai leistų planuoti miškų atsodinimą ir vandens telkinių gaivinimą.

Šiai problemai spręsti gali būti pasitelkiami gilieji dirbtiniai neuroniniai tinklai, kurių pagalba būtų galima automatiškai segmentuoti satelitinius vaizdus bei stebėti pokyčius. Todėl šiame darbe aptariamos įvairių rūšių giliųjų neuroninių tinklų architektūros, pasirenkama detaliau išnagrinėti *U-Net* ir *SegCaps* architektūros tinklus. Šie tinklai pasirodė geriausiai medicininių vaizdų segmentavime, be to, *U-Net* architektūros gilieji neuroniniai tinklai ir praeityje buvo naudojamas satelitinių vaizdų segmentavime. Pagrindinė problema nagrinėjant satelitinius vaizdus yra tai, kad objektai juose yra maži ir nufotografuoti iš viršaus, todėl sunku išgauti didelį tikslumą segmentuojant atskirus objektus.

Lyginant *U-Net* ir *SegCaps* modelius jie parodė skirtingus rezultatus. *U-Net* išsiskyrė kaip tikslesnis dirbtinis neuroninis tinklas. *U-Net* architektūros tinklas tiksliau segmentavo vaizdus, ypač vandens telkinius. Tai galėjo nutikti dėl skirtingų priežasčių, pirmiausiai *SegCaps* buvo skirtas segmentuoti 3D medicininius vaizdus, kurie turėjo daugybę sluoksnių, susidedančių iš skirtingų vaizdų kiekviename, o satelitiniai vaizdai yra 2 dimensijų.

Summary

The application of deep neural networks for satellite analysis is researched in this work. Due to various natural processes and human behavior, deforestation is happening, and water boundaries are changing. These changes must be monitored in case harmful processes can be stopped. Also, it would help to plan reforestation and water bodies „recovery“.

This problem can be solved by using deep neural network which would help to segment satellite images automatically and monitor changes. There are also presented various types of deep neural network architectures in this work. *U-net* and *SegCaps* architecture networks are selected to be analyzed in more detail. These networks seem to be the best in medical images segmentation and U-Net architecture deep neural networks was also used in satellite images segmentation in the past. The main problem of analysis of satellite images is that objects in satellite images are small and photographed from above that is the reason why it is hard to obtain accuracy while doing segmentation separate objects.

U-Net and *SegCaps* deep neural networks have shown different results. *U-Net* has been distinguished as a more precise artificial neural network. U-Net architecture network was better in image segmentation than *SegCaps* architecture, especially water bodies.

It could happen for different reasons, first of all *SegCaps* was designed to segment 3D medical images which had many layers, and satellite images are 2-dimension images.

Turinys

Santrauka.....	2
Summary	3
1. Įvadas	6
1.1. Darbo aktualumas ir naujumas	6
1.2. Darbo tikslas	7
1.3. Darbo uždaviniai	7
1.4. Laukiami rezultatai	8
2. Literatūros analizė	9
2.1 Dirbtiniai neuroniniai tinklai	9
2.2. Gilieji neuroniniai tinklai.....	11
2.3. Be mokytojo apmokyti tinklai	12
2.3.1. Autoenkoderiai	12
2.3.2. Gilaus „tikėjimo“ tinklai	13
2.3.3. Generatyviniai besivaržantys tinklai	13
2.4. Rekurentiniai neuroniniai tinklai	14
2.5. Konvoliuciniai neuroniniai tinklai	14
2.6. Rekursyvūs neuroniniai tinklai	16
2.7. Kapsuliniai neuroniniai tinklai.....	17
2.8. Vaizdų segmentavimas taikant giliuosius neuroninius tinklus	18
2.9. Segmentavimo rezultatų vertinimui naudojamos metrikos	18
2.10. Pilnai konvoliucinis neuroninis tinklas.....	19
2.11. Atstatymo metodas	19
2.12. <i>U-Net</i>	19
2.13. Kapsuliniai neuroniniai tinklai objektų segmentavimui	20
2.14. Satelitinių vaizdų segmentavimas.....	21
2.15. Satelitinių vaizdų segmentavimas naudojant giliuosius neuroninius tinklus.....	22
2.16. Apibendrinimas	26

3.	Darbo rezultatai ir jų aptarimas	27
3.1.	Duomenų aibė	27
3.2.	Kompiuteriniai resursai.....	28
3.3.	Programinė įranga.....	28
3.4.	Duomenų paruošimas dirbtinio neuroninio tinklo mokymuisi	29
3.5.	<i>U-Net</i> architektūros dirbtinis neuroninis tinklas.....	30
3.5.1.	<i>U-Net</i> tinklo mokymasis.....	31
3.5.2.	<i>U-Net</i> rezultatai	31
3.6.	<i>SegCaps</i> architektūros dirbtinis neuroninis tinklas.....	35
3.6.1.	<i>SegCaps</i> modelio mokymasis.....	35
3.6.2.	<i>SegCaps</i> modelio rezultatai	36
3.7.	<i>SegCaps</i> ir <i>U-Net</i> modelių rezultatų palyginimas.....	39
4.	Išvados.....	40
	Literatūra	41
	Priedai	43

1. Įvadas

1.1. Darbo aktualumas ir naujumas

Miškai ir vandens telkiniai (šiąme darbe vandens telkiniai suprantami kaip ežerai, tvenkiniai, upės ir upeliai) yra vieni svarbiausių gamtos išteklių bei turtų.

Miškai – tai atskiros ekosistemos, kuriose daug įvairių gyvūnų bei augalų. Jie aprūpina pramonę mediena, o atmosferą - deguonimi. Miškai ir vandens telkiniai sąveikauja, todėl pakrantės zonose prie vandens telkinių esančiuose miškuose susidaro tam tikra savita ekosistema, kurioje gausu įvairių gyvūnų bei augalų. Šalia vandens telkinių esantys miškai turi tiesioginę įtaką vandens ekologijai bei biologinei įvairovei.

Deja, dėl įvairių natūralių procesų, reiškinių bei žmogaus neteisėtų veiksmų miškai nyksta, tai itin svarbu visai ekosistemai. Miškų plotų pokyčių stebėjimas būtinas, tai leistų planuoti miškų atsodinimą, vandens telkinių gaivinimą, taip pat leistų natūralius gamtos pokyčius.

Pažymėtina, kad vien tik Baltijos jūros regiono šalyse miškų plotas sudaro 68 mln. ha.¹ Tokį plotą apžiūrėti be satelitinių vaizdų ar dronų pagalbos yra ganėtinai sudėtinga, tai reikalauja itin daug tiek žmogiškųjų, tiek finansinių išteklių. Tačiau dronų taikymas nėra toks tikslus nes sudėtinga nustatyti tikslias drono buvimo koordinatas, todėl satelitinių vaizdų apdorojimas giliųjų neuroninių tinklų (toliau – GNT) pagalba leistų daug paprasčiau rasti nykstančius miškus bei besikeičiančius vandens telkinius, tai leistų optimizuoti kaštus bei greičiau reaguoti į pokyčius. Tačiau GNT čia susiduria su tokiais iššūkiais kaip prasta vaizdų kokybė, objektų dydis, skirtingų klasių objektų panašumai. Pavyzdžiui, miškų plotai gali atrodyti panašūs vizualiai į žolės ar pasėlių plotus. Upės gali būti panašios į kelius. Tai matyti ir žemiau pateiktame paveiksle (1 pav.).



1 pav. Satelitinis vaizdas

¹ Maisto ir žemės ūkio organizacijos (FAO) Pasaulinių miškų išteklių įvertinimas 2015; <http://www.fao.org/forest-resources-assessment/current-assessment/en/>.

Šių plotų analizei gali būti taikomi įvairūs algoritmai, o didėjant kompiuterių skaičiavimo pajėgumams populiarėja gilieji neuroniniai tinklai. Gilieji neuroniniai tinklai – tai dirbtiniai neuroniniai tinklai, kurių architektūroje yra daugiau nei 2 paslėpti sluoksniai [Sch15]. Anksčiau šie neuroniniai tinklai nebuvo itin populiarūs dėl didelių skaičiavimo resursų poreikio, tačiau didėjant skaičiavimo pajėgumams GNT sparčiai išpopuliarėjo. GNT gali modeliuoti sudėtingas netiesines priklausomybes. GNT tipo architektūros sukuria kompozicinius modelius, kuriuose objektas išreiškiamas kaip daugiasluoksnė paprastųjų duomenų tipų kompozicija [STE13]. Papildomi sluoksniai leidžia sudaryti požymių kompoziciją iš prieš tai buvusių sluoksnių, kas potencialiai leidžia modeliuoti sudėtingesnius duomenis [Ben09].

Egzistuoja skirtingi GNT tipai (pvz. konvoliuciniai neuroniniai tinklai, kapsuliniai neuroniniai tinklai ir kt.), kiekvienas šių tipų geriausiai susidoroja su tam tikra užduotimi, todėl svarbu nustatyti, koks GNT tipas labiausiai tinka miškų ir vandens telkinių satelitiniuose vaizduose analizei. Šie tinklai naudoja įvairias daugiasluoksnių perceptronų variacijas (perceptrono sąvoka detaliau analizuojama tolimesniuose skyriuose) ir yra sukurti taip, kad reikėtų kuo mažiau pirminio duomenų apdorojimo [LeC98]. Dėl šio ypatumo GNT gali modeliuoti daugiamačius duomenis, išskirti požymius, kas leidžia atsisakyti rankinio požymių išskyrimo, belieka tik GNT mokymo aibėje pažymėti norimus objektus nurodant jų klases [LeC98]. Vaizdų atpažinime ne tik svarbu atpažinti norimą objektą, tačiau ir pažymėti jį vaizde, tai yra vaizdą segmentuoti. Segmentavimas taip pat leidžia palyginti pirminį vaizdą ir gautą rezultatą, kas leidžia nustatyti GNT tikslumą.

Šio darbo eigoje bus siekiama palyginti įvairių tipų GNT architektūras ir išrinkti labiausiai tinkančią satelitinių miškų ir vandens telkinių vaizdų analizei.

1.2. Darbo tikslas

Nustatyti, koks gilusis neuroninis tinklas geba tiksliausiai atpažinti miškus bei vandens telkinius satelitiniuose vaizduose, lyginant skirtingų architektūrų giliuosius neuroninius tinklus.

1.3. Darbo uždaviniai

1. Atlikti mokslinės literatūros, susijusios su giliųjų neuroninių tinklų taikymu objektams vaizduose atpažinti, analizę.
2. Išnagrinėti įvairius giliuosius neuroninius tinklus ir jų taikymą objektams satelitiniuose vaizduose atpažinti.
3. Palyginti skirtingų architektūrų giliųjų neuroninių tinklų rezultatus miškams ir vandens telkiniams satelitiniuose vaizduose atpažinti.
4. Nustatyti, kokios architektūros gilusis dirbtinis neuroninis tinklas geba tiksliausiai nustatyti miškų ir vandens telkinių buvimo vietą satelitiniuose vaizduose.

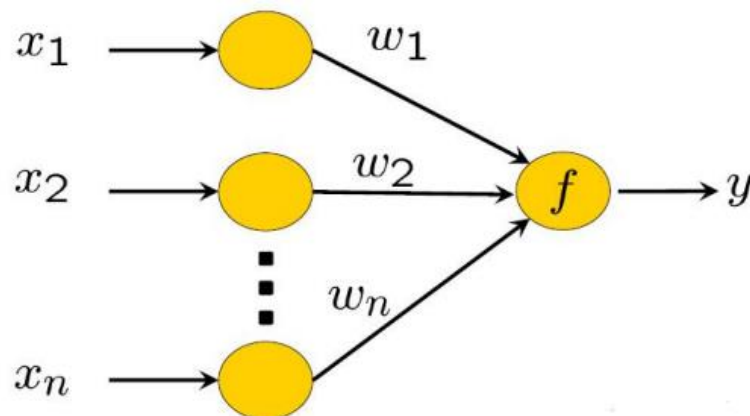
1.4. Laukiami rezultatai

1. Tarpusavyje palygintos ir aprašytos skirtingos giliųjų neuroninių tinklų architektūros.
2. Sukurtas gilusis neuroninis tinklas, gebantis tiksliai atpažinti miškus ir vandens telkinius satelitiniuose vaizduose.

2. Literatūros analizė

2.1 Dirbtiniai neuroniniai tinklai

Dirbtiniai neuroniniai tinklai – tai skaičiavimo modelis, kuris kai kuriais savo požymiais primena žmogaus smegenis, kuriose daug paprastų vienetų (neuronų) veikia lygiagrečiai be centralizuoto valdymo bloko. Pirmas toks modelis aprašytas dar 1943 metais Pitts ir McCulloch darbe. Vėliau 1958 metais Frank Rosenbalt aprašė dirbtinį neuroną – perceptroną. Perceptronas – tai binarinio klasifikavimo algoritmas. Dirbtinis neuronas (2 pav.) turi keletą įėjimų (kiekviena įėjimo jungtis turi savo svorį), įprastai šių įėjimų ir svorių reikšmės yra realieji skaičiai. Skaičiuojama reikšmių ir svorių sandaugų suma. Ši suma perduodama neurono aktyvacijos funkcijai. Atsižvelgiant į funkciją (jos gali būti skirtingos: tiesinė, sigmoidinė, hiperbolinis tangentas ir kt.) apskaičiuojamas neurono išėjimas, jis gali būti lygus 1 arba 0, priklausomai nuo to kokia yra slenksčio reikšmė (angl. *threshold*) [PG17]. Kiekvienas neuronas turi slenksčio reikšmę – w_0 . 2 pav. jis nėra pavaizduotas.



2 pav. Dirbtinis neuronas

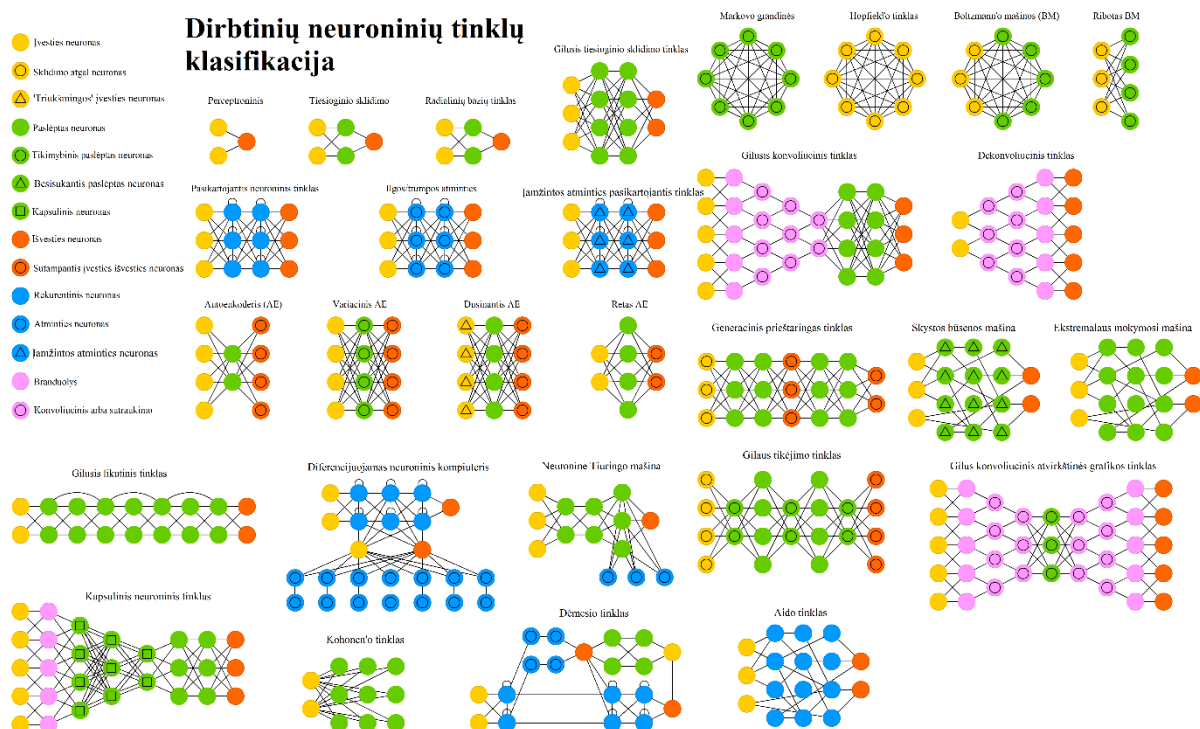
Šaltinis: [PG17]

Neurono mokymosi metu pagrindinis tikslas – rasti tokias svorių reikšmes, kurios būtų kuo artimesnės trokštamai reikšmei, tai yra, kad neurono veikimo paklaida būtų kiek galima mažesnė.

Dirbtiniai neuronai gali būti sujungti į dirbtinius neuroninius tinklus (toliau – DNT). Pagal jungimo būdą neuroniniai tinklai skirstomi į dvi pagrindines grupes:

- Tiesioginio sklidimo DNT, be grafo ciklų (angl. *feedforward*)
- Grįžtamojo ryšio DNT, su grafo ciklais (angl. *feedback*)

Sparčiai besiplečiant šiai dirbtinio intelekto šakai, skirtingų architektūrų dirbtinių neuroninių tinklų yra daug ir kasmet jų vis daugėja, 3 pav. pavaizduotos pagrindinės šiuo metu esančios DNT architektūros. Kiekviena iš šių architektūrų pritaikyta spręsti tam tikro tipo uždavinius.



3 pav. Dirbtinių neuroninių tinklų klasifikacija

Šaltinis: <http://www.asimovinstitute.org/neural-network-zoo/>

DNT – tai mokslo apie dirbtinį intelektą dalis, o pagrindinė intelekto savybė yra gebėjimas mokytis, todėl norint gauti tinkamus rezultatus DNT turi būti tinkamai apmokytas. Sėkmingą mokymosi procesą galima apibrėžti kaip tinklo struktūros ir jungčių svorio keitimo uždavinį, siekiant, kad tinklas gebėtų atlikti jam skirtą užduotį. DNT mokymosi procese, kiekviename žingsnyje keičiamos svorių reikšmės, atsižvelgiant į įėjimo ir išėjimo reikšmes, kurios gautos ankstesniame žingsnyje [PG17].

Kiekvienam skirtingam DNT būtinas kitoks mokymo algoritmas. Išskiriami trys pagrindiniai neuronų mokymo metodai:

1. Mokymas su mokytoju (angl. *supervised learning*).

Šiame metode naudojama sąvoka norima reikšmė (angl. *target*). Šios reikšmės iš anksto žinomos, tai gali būti klasės numeris.

2. Mokymas be mokytojo (angl. *unsupervised learning*)

Tais atvejais, kai nėra žinomos norimos reikšmės, naudojami mokymo be mokytojo metodai. Šiuo atveju tinklas mokomas ieškoti panašumų tarp mokymo aibės objektų. Šių metodų panaudojamumas yra mokymo duomenų suskirstymas į tam tikras klases, ypatumų arba tendencijų paieška.

3. Hibridinis mokymas

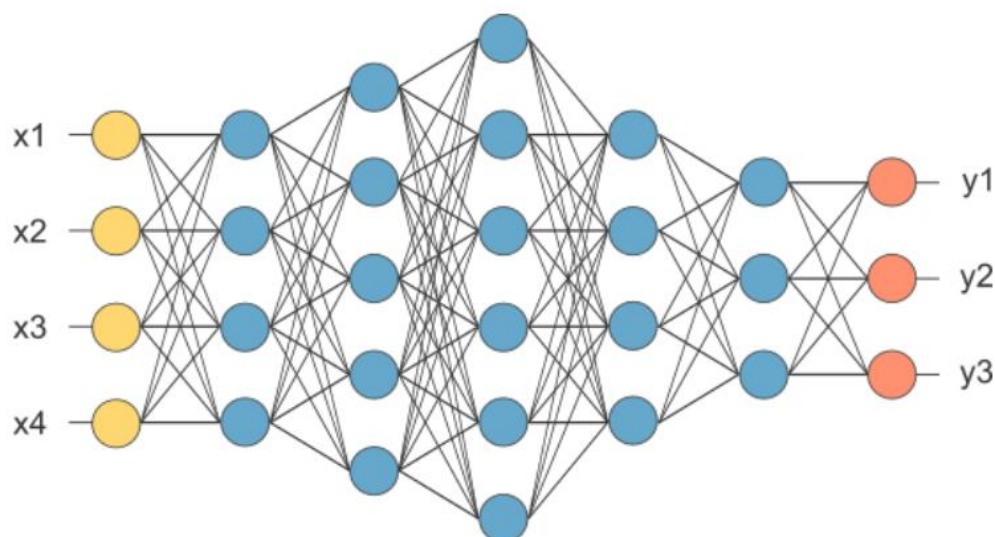
Kaip matyti iš metodo pavadinimo šis metodas apima tiek mokymo su mokytoju, tiek mokymo be mokytojo algoritmus. Dalis tinklo svorių nustatoma vienu metodu, kita dalis – kitu metodu [PG17].

2.2. Gilieji neuroniniai tinklai

Kaip jau minėta, GNT yra tokios architektūros dirbtiniai neuroniniai tinklai, kuriuose yra daugiau nei 2 paslėpti sluoksniai [Sch15]. Kadangi šio tipo tinklai reikalauja didelių skaičiavimo resursų, jie išpopuliarėjo pastaruoju metu, kai kompiuteriniai resursai tapo lengviau prieinami. Taip pat galima išskirti keletą savybių, skiriančių giliuosius neuroninius tinklus nuo įprastų tiesioginio sklaidimo neuroninių tinklų [PG17]:

- Įprastai turi daugiau neuronų
- Sudėtingesni sluoksnių jungimo būdai
- Reikalauja gerokai daugiau skaičiavimo resursų
- Automatinis požymių išskyrimas

Daugiau neuronų reiškia tai, kad jų skaičius padidintas tam, kad galėtų išreikšti sudėtingesnius modelius. Sluoksnių jungimo būdai taip pat skiriasi nuo pilnai sujungto daugiasluoksnio neuroninio tinklo, pavyzdžiui konvoliuciniame neuroniniame tinkle jungiami ne visi neuronai, o tik dalis, o rekurentiniuose tinkluose neuronas turi papildomą jungtį į save. 4 pav. pavaizduota giliųjų neuroninių tinklų schema.



4 pav. Giliųjų neuroninių tinklų schema

Šaltinis: [PG17]

Šiame darbe nagrinėtos keturios pagrindinės giliųjų neuroninių tinklų architektūros, siekiant suprasti, kurios jų geriausiai tinkamos vaizdams analizuoti [PG17].

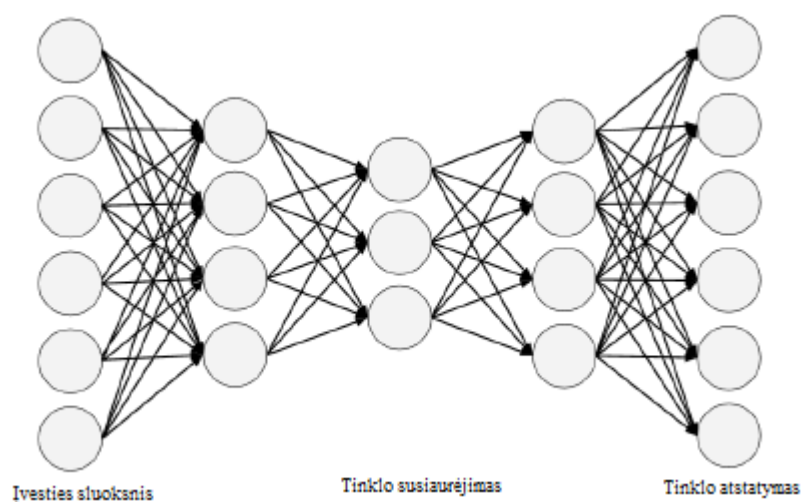
- Be mokytojo apmokyti tinklai
- Rekurentiniai neuroniniai tinklai
- Rekursyvūs neuroniniai tinklai
- Konvoliuciniai neuroniniai tinklai

2.3. Be mokytojo apmokyti tinklai

Į šią grupę patenka autoenkoderiai (angl. *autoencoders*), gilias „tikėjimo“ tinklai (angl. *deep belief networks*) ir generatyviniai bezivaržantys tinklai (angl. *generative adversarial networks*).

2.3.1. Autoenkoderiai

Paprastai šios architektūros tinklai naudojami duomenų aibės matmens sumažinimui. Autoenkoderiai savo architektūra (5 pav.) yra panašūs į daugiasluoksnį perceptroną, jie susideda iš įvesties sluoksnio, paslėptų sluoksnių ir išvesties sluoksnio. Pagrindinis skirtumas yra tai, kad autoenkoderių išvesties sluoksnių yra tiek pat kiek įvesties. Taip pat šios architektūros tinklai skiriasi tuo, kad jų mokymas vyksta be mokytojo, be to, šie tinklai grąžina suspaustą įvesties duomenų vaizdą. Dažniausiai šie tinklai naudojami atpažinti duomenų anomalijas [PG17]. Prie autoenkoderių architektūros būtų galima priskirti ir U-Net architektūros tinklą, kuris bus aptartas vėliau, tačiau pagrindinis skirtumas nuo autoenkoderių yra, tai, kad U-Net mokosi su mokytojo pagalba.

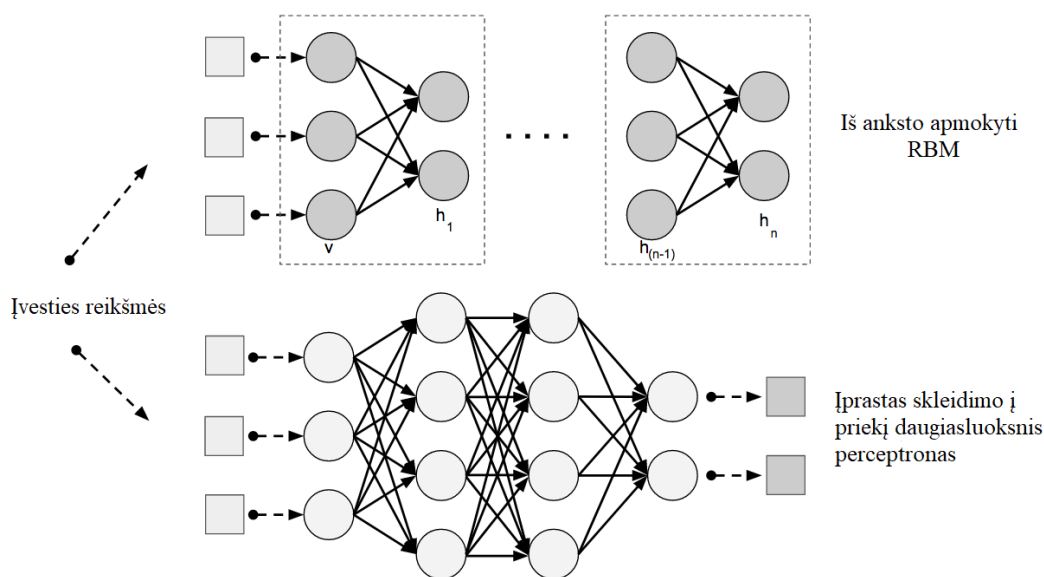


5 pav. Autoenkoderių architektūra

Šaltinis: [PG17]

2.3.2. Gilaus „tikėjimo“ tinklai

Šios architektūros tinklai (6 pav.) susideda iš apribotų Boltzmann mašinų (toliau – RBM) prieš mokymo fazėje ir tiesioginio sklaidimo neuroninio tinklo.



6 pav. Gilaus „tikėjimo“ neuroninių tinklų architektūra

Šaltinis: [PG17]

RBM naudojami išgauti aukštesnio lygio požymius iš neapdorotų įvesties vektorių. Norint tai padaryti paslėpto vieneto būseną ir svoriai nustatomi taip, kad kai RBM įvesties įrašas perduodamas ir prašoma RBM atkurti įrašą, šis sluoksnis sukuria kažką labai artimo pirminiam įvesties vektoriui. Hintonas apibūdina šį efektą kaip mašinos įsivaizduoja duomenis [PG17].

Pagrindinis RBM ir gilaus „tikėjimo“ tinklų tikslas gilaus mokymosi kontekste - išmokti tas duomenų aibės aukštesnio lygio atributus mokyme be mokytojo formoje. Išsiaiškinta, kad galima geriau mokyti neuroninius tinklus leidžiant RBM palaipsniui išsiaiškinti aukštesnio lygio požymius naudojant jau išmokus požymius iš žemesnio lygio prieš mokymosi (angl. *pretrain*) sluoksnio kaip įvestį į aukštesnio sluoksnio RBM prieš mokymosi sluoksnius [PG17].

Kadangi šios architektūros GNT nėra toks efektyvus vaizdų atpažinime kaip konvoliucinis neuroninis tinklas detaliau jis šiame darbe nėra nagrinėjamas.

2.3.3. Generatyviniai besivaržantys tinklai

Generatyviniai besivaržantys tinklai (toliau – GBT) tai tinklai, kurie naudoja mokymąsi be mokytojo mokydami du modelius lygiagrečiai. Pagrindinis GBT aspektas yra tai kaip jis naudoja parametrų skaičių, kuris yra gerokai mažesnis nei įprastai žvelgiant į kiekį duomenų, kuriais yra mokomas tinklas [KLA18].

Šių tinklų potencialas yra didžiulis, nes jie gali išmokti imituoti bet kokią duomenų paskirstymą ir tai jiems leidžia generuoti vaizdus, garsą ir kt. Tai ir yra šių tinklų pagrindinė paskirtis, todėl detaliau šiame darbe šie tinklai nėra analizuoti.

2.4. Rekurentiniai neuroniniai tinklai

Rekurentiniai neuroniniai tinklai priklauso tiesioginio sklidimo neuroninių tinklų šeimai. Jie skiriasi lyginant su kitais tiesioginio sklidimo neuroniniais tinklais savo gebėjimu siųsti informaciją per laiko žingsnį (angl. *time-step*). Įdomų rekurentinių tinklų paaiškinimą pateikė jų kūrėjas J. Schmidhuber:

„Rekurentiniai neuroniniai tinklai gali daryti lygiagrečius ir nuoseklius skaičiavimus, ir iš esmės gali apskaičiuoti viską, ką gali apskaičiuoti įprastas kompiuteris. Tačiau priešingai nei tradicinis kompiuteris, rekurentiniai neuroniniai tinklai yra panašūs į žmogaus smegenis, kurios yra didelis grįžtamojo ryšio tinklas jungiantis neuronus, kurie kažkaip gali išmokti versti nesibaigiantį jutiminį įvesties srautą į naudingą motorinio nervo išvesties seką. Smegenys yra nepaprastas modelis, kuris gali išspręsti daugybę dabartinių problemų, kurių kol kas negali išspręsti mašinos. [PG17]“

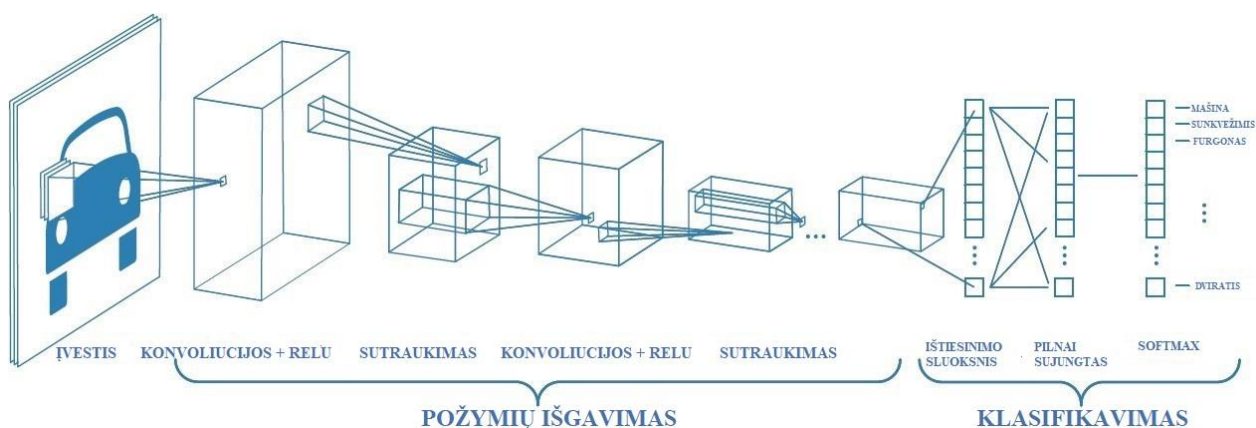
Tačiau pagrindinės problemos, kurias sprendžia rekurentiniai neuroniniai tinklai, tai kalbos atpažinimas, ranka rašytų tekstų atpažinimas, tačiau jie nėra tinkamai pritaikyti vaizdų analizei, todėl detaliau jų analizuoti nėra poreikio.

2.5. Konvoliuciniai neuroniniai tinklai

Konvoliuciniai neuroniniai tinklai (toliau – KNT) daugiausiai naudojami vaizdų analizei. KNT kūrėjus įkvėpė biologiniai procesai, nes ryšys tarp neuronų yra panašus į gyvūnų regėjimo žievės organizavimą. Atskiri žievės neuronai reaguoja į stimulus tik ribotame regėjimo lauko regione vadinamu imliu lauku. Šios architektūros tinklai gali identifikuoti veidus, asmenis, kelio ženklus ir daugelį kitų objektų [LeC98].

Tiesioginio sklidimo neuroninių tinklų įvestis yra vienas vienmatis vektorius, tinklas transformuoja duomenis paslėptuose sluoksniuose ir grąžina rezultatą išvesties sluoksnyje. Pagrindinė tradicinio tiesioginio sklidimo neuroninio tinklo problema yra ta, kad jie nėra pritaikyti dirbti su vaizdais ir mokymosi metu generuoja daug svorio reikšmių, tuo tarpu KNT yra pritaikyti daugiamachiams duomenims. KNT transformuoja įvesties duomenis per visus sluoksnius į klasių tikimybes, kurias grąžina išvesties sluoksniu. Egzistuoja daug skirtingų KNT variantų, tačiau šio darbo eigoje bus aptarti tik tie, kurie leidžia vaizdus ne tik klasifikuoti, bet ir segmentuoti. Kaip

matyti iš 7 pav. KNT susideda iš įvesties sluoksnio požymio išgavimo sluoksnių ir klasifikavimo sluoksnių [PG17].



7 pav. Konvoliucinis neuroninis tinklas

Šaltinis: [PG17]

Įvesties sluoksnis priima trimačius duomenis, dažniausiai tai vaizdo dydis (plotis x aukštis) ir spalvų kanalų gylis. Požymių išgavimo sluoksnyje dažniausiai kartojama ši sluoksnių seka:

1. Konvoliucijos sluoksnis.
2. Netiesiškumo sluoksnis.
3. Sujungimo sluoksnis (angl. *pooling*).

Konvoliucijos sluoksnyje vykdomos konvoliucijos operacijos. Šis sluoksnis yra pagrindinis KNT elementas.

Konvoliucija – tai matematinė operacija, kuri paima dvi funkcijas ir grąžina trečią, kuri tam tikra prasme parodo funkcijų persidengimo lygį. Įprastai viena funkcija imama kaip fiksuotas filtras, dar vadinamas branduoliu (angl. *kernel*). Panaudodama įvestį konvoliucija pritaiko branduolį ir pateikia mums požymių žemėlapi (angl. *feature map*) kaip išvestį.

Konvoliuciniai sluoksniai turi parametrus sluoksniams ir pridėtinius hiperparametrus. KNT mokomi minimizuojant paklaidos funkciją, įprastai taikant gradientiniu nusileidimu grįstus optimizavimo algoritmus.

Pagrindiniai konvoliucinio sluoksnio komponentai [PG17]:

- Filtrai – tai funkcija, kurios plotis ir aukštis mažesni nei įvesties plotis ir aukštis.
- Aktyvacijos žemėlapis. Kaip jau buvo minėta, aktyvacija – tai skaitinis rezultatas, pagal kurį neuronas nusprendžia, ar leisti duomenims pereiti į sekantį sluoksnį. Kai sakoma, kad filtras aktyvuojasi, turima omenyje, kad filtras leidžia informacijai pereiti iš įvesties į išvestį. Norint apskaičiuoti aktyvacijos žemėlapi filtrai slenkami per įvesties apimties gylį. Filtras vaizduoja svorius, kurie dauginami slenkant „langą“ per vaizdą. Tinklas moko filtrus, kad šie aktyvuotųsi, kai randa tam tikrą atributo tipą įvesties duomenyse.

- Parametrų padalijimas. KNT naudoja parametrų pasidalijimo schemą, tam, kad valdytų bendrą parametrų skaičių. Tai pagreitina mokymo laiką, nes naudojama mažiau resursų mokymosi metu.



8 pav. KNT filtrų pavyzdžiai

Šaltinis: [PG17]

Taikant paketo normalizaciją KNT modelio mokymasis pagreitėjo. Be to normalizuojant paketus galima naudoti kur kas didesnę mokymo tempą. Ši normalizacija sumažina mokymo jautrumą ir veikia kaip reguliatorius.

Sutelkimo sluoksniai įprastai įterpiami tarp konvoliucinių sluoksnių. Sutelkimo sluoksniai naudojami tam, kad palaipsniui sumažintų erdvės dydį duomenų vaizdavime. Sutelkimo sluoksniai palaipsniui mažina duomenų vaizdavimą tinkle ir padeda kontroliuoti persimokymą. Tinklo pabaigoje įprastai būna pilnai sujungtas sluoksnis, tam, kad apskaičiuoti tikimybes, kokios klasės ar klasė yra pavaizduota vaizde [PG17].

2.6. Rekursyvūs neuroniniai tinklai

Rekursyvūs neuroniniai tinklai, kaip ir rekurentiniai neuroniniai tinklai, gali apdoroti kintamojo ilgio įvestį. Pagrindinis skirtumas yra tas, kad rekurentiniai neuroniniai tinklai gali modeliuoti hierarchines struktūras mokymo duomenų rinkinyje. Vaizdus paprastai sudaro daugelio objektų kompozicija. Rekursinis pobūdis šių dekonstaravimo iššūkių verčia ne tik identifikuoti vaizde esančius objektus bet ir identifikuoti kaip objektai yra susiję su vaizdo formavimo, tai yra koks jų yra santykis [PG17].

Rekursyvių neuroninių tinklų architektūra susideda iš bendros svorių matricos ir dvejetainio medžio struktūros, kuri leidžia išmokti įvairias žodžių ar vaizdo dalių sekas. Tai naudinga analizuojant sakinius ir scenas. Šie tinklai naudoja klaidos sklidimo atgal algoritmą, vadinamą klaidos sklidimas atgal per struktūrą. Šiame tinkle perėjimas į priekį vyksta „iš apačios į viršų“ o klaidos sklidimas atgal - „žemyn“.

Rekursyvūs neuroniniai tinklai taikomi [PG17]:

1. Natūralios kalbos apdorojime (angl. *natural language processing*).

2. Vaizdo scenos dekompozicijai.

3. Garso į tekstą transponavime.

Kaip matyti iš aprašyto šių tinklų pritaikomumo, jie nėra taikytini vaizdų analizei todėl detaliau nenagrinėtini.

2.7. Kapsuliniai neuroniniai tinklai

Žmogaus regėjimas ignoruoja nereikšmingas detales, naudojant kruopščiai nustatytą fiksavimo taškų seką, taip užtikrinant, kad tik nedidelė optinės matricos dalis būtų apdorojama aukščiausia raiška. Tokiu pat būdu veikiančius neuroninius tinklus, vadinamus kapsuliniais, savo darbe 2017 metais aprašė S. Sabour ir N. Frosst [SF17].

Kapsulė – tai neuronų grupė, kurios aktyvumo vektorius atspindi konkrečios esybės pvz.: objekto ar objekto dalies momentinius parametrus. Aktyvumo vektoriaus ilgis naudojamas tam, kad parodyti tikimybę, kad esybė egzistuoja ir tos esybės padėtį, kad nurodyti atvaizdavimo parametrus. Aktyvios kapsulės viename lygyje daro prognozes transformuodamos matricas, kad perduotų atvaizdavimo parametrus aukštesnio lygio kapsulėms. Kai keletas prognozių sutampa aukštesnio lygio kapsulės tampa aktyvios [SF17]. Konvoliuciniuose neuroniniuose tinkluose neurono gražinamas rezultatas yra skaliarinė reikšmė, tuo tarpu kapsulės gražina vektorius, tai suteikia galimybę naudoti dinaminį maršrutizavimą užtikrinant, kad kapsulės išėjimo vektorius bus išsiųstas į tinkamą aukštesnio sluoksnio tėvinį elementą. Kapsuliniai neuroniniai tinklai neturi maksimalaus sujungimo sluoksnio (angl. *max pooling*), vietoje šito sluoksnio naudojamas dinaminis maršrutizavimas [SF17].

Šio tipo neuroniniuose tinkluose siekiama, kad kapsulės išvesties vektoriaus ilgis atspindėtų tikimybę, kad kapsulėje esanti esybė yra dabartinėje įvestyje. Todėl naudojama ne tiesinė suspaudimo (angl. *squashing*) funkcija užtikrinanti, kad trumpi vektoriai sumažėtų iki beveik nulinio ilgio, o ilgi vektoriai sumažėtų iki šiek tiek mažesnio nei 1 ilgio.

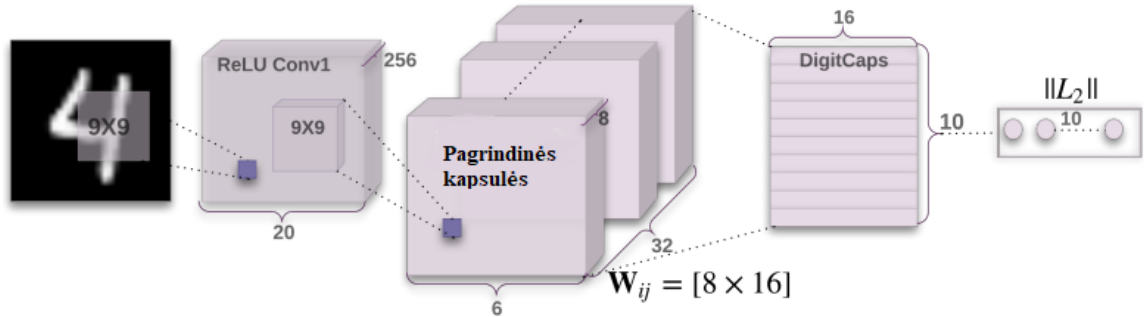
$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|}$$

Kur v_j – tai kapsulės j išvesties vektorius ir s_j – tai kapsulės įvesties suma. Visiems, išskyrus pirmąjį kapsulių sluoksnį, bendra kapsulės įvesties suma S_j – tai suma visų prognozavimo vektorių $\hat{u}_{j|i}$ iš kapsulių esančių prieš tai buvusiame sluoksnyje ir ji paskaičiuojama padauginus kapsulės išėjimo u_i prieš tai buvusiame sluoksnyje iš svorių matricos W_{ij} .

$$s_j = \sum_i c_{ij} \hat{u}_{j|i}, \quad \hat{u}_{j|i} = W_{ij} u_i$$

Kur c_{ij} – tai sujungimo koeficientas, kuri nustato iteracinis dinaminis maršrutizavimo procesas.

Paprasta kapsulinių neuroninių tinklų (toliau – *CapsNet*) architektūra pavaizduota 9 pav. Architektūra yra negili ir turi tik du konvoliucinius sluoksnius ir vieną visiškai sujungtą sluoksnį.



9 pav. Kapsulinių neuroninių tinklų architektūra

Šaltinis: [SF17]

2.8. Vaizdų segmentavimas taikant giliuosius neuroninius tinklus

Vaizdų segmentavimas – tai procesas, kurio metu skaitmeninis vaizdas padalinamas į tam tikrus segmentus. Vaizdų segmentavimas – tai vaizdo padalijimo į regionus su panašiais požymiais procesas [Pra91]. Objektų segmentavimo užduotį galima suformuluoti kaip bendrą objekto atpažinimo ir apibrėžimo problemą. Objektų atpažinimo tikslas yra surasti objekto buvimą vaizde, o objekto apibrėžimas (segmentavimas) bando „nupiešti“ objekto erdvinį plotą ir sudėtį [RFB15]. Vaizdų segmentavimas ypač svarbus mediciniuose vaizduose, kai siekiama nustatyti pažeistus ligos regionus (pvz. įvairūs navikai), taip pat analizuojant satelitinius vaizdus svarbu yra ne tik nustatyti objektų klases bet ir juos atitinkamai pažymėti vaizde.

2.9. Segmentavimo rezultatų vertinimui naudojamos metrikos

Vaizdų segmentavimo tikslumui nustatyti naudojamos dvi pagrindinės metrikos, tai *Jaccard* ir *Dice*. Šios abi metrikos skirtos tam, kad gali būtų nustatyti, kiek segmentuotas plotas sutampa su vaizde parodytu vaizdu.

Jaccard metrika gali būti interpretuojama kaip panašumo matas tarp baigtinio aibių skaičiaus. Dviems A ir B aibėms, ji gali būti apibrėžiama kaip:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Tuo tarpu *Dice* koeficientas apskaičiuojamas pagal formulę:

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|}$$

Kur X tai pikselių esančių vaizdo kaukėje aibė, o Y tai segmentavimo rezultato vaizde esančių pikselių aibė. Šis koeficientas parodo aibių panašumus, tai leidžia apskaičiuoti kiek panašus yra vaizdas su pažymėtais objektais ir tinklo pateiktais segmentuotais vaizdas.

2.10. Pilnai konvoliucinis neuroninis tinklas

Pagrindinis šio DNT išskirtinumas yra tai, kad pilnai sujungtą sluoksnį keičia pilnai konvoliucinis sluoksnis. Naudojant interpoliacijos sluoksnį, jis supranta, kad išėjimo dydis yra toks pat, kaip įvesties, kas yra būtina segmentuojant vaizdus.

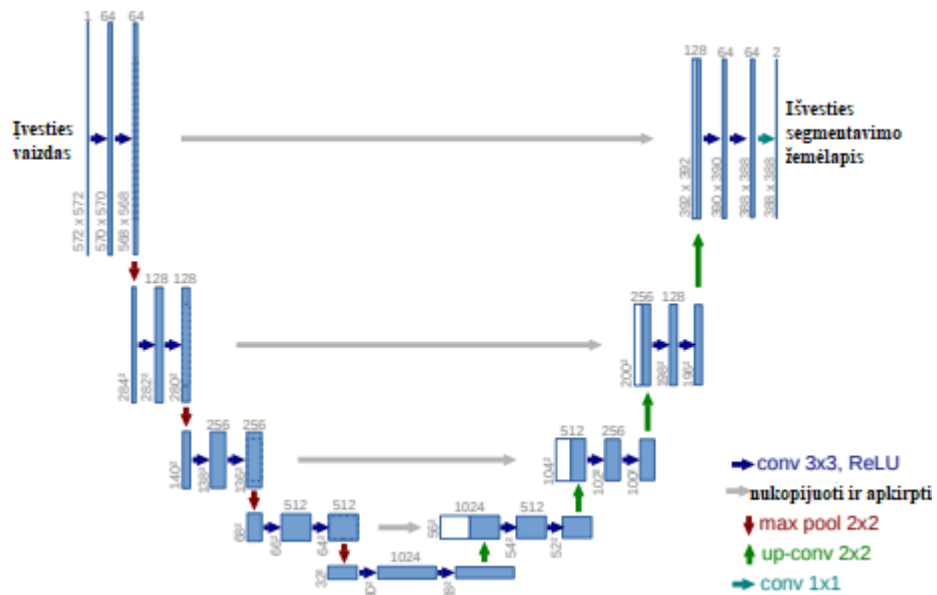
2.11. Atstatymo metodas

Papildomai prie pilnai konvoliucinio DNT gali būti naudojamas dekonvoliucijos sluoksnis, kuris taip pat pritaikomas semantiniame segmentavime. Dekonvoliucinį tinklą savo darbe naudojo Long 2015 m., kurį sudaro dekonvoliucija ir sujungimo sluoksniai, kurie identifikuoja kiekvieno pikselio klases ir prognozuoja segmentavimo kaukes. 2015 metais Ronnenberg pristatė alternatyvų vaizdų segmentavimo tinklą, vadinamą *U-Net*, kuris taip pat bandė atpažinti kiekvieno pikselio klasę [RFB15].

2.12. *U-Net*

U-Net tinklo architektūra pavaizduota 10 pav. Tai pilnai konvoliucinio neuroninio tinklo modifikacija [RFB15]. Šis tinklas modifikuotas tokiu būdu, kad jam reikia mažiau vaizdų mokymuisi ir jis yra tikslesnis nustatant kiekvieno pikselio klasę. Gerų rezultatų su mažiau vaizdų mokymuisi pasiekama padidinant duomenų kiekį (angl. *data augmentation*), tam gali būti naudojami skirtingi būdai: vaizdo pasukimas, mastelio keitimas ir kiti. Šiuo būdu didinant duomenų kiekį išvengiama tinklo persimokymo (angl. *overfitting*) [RFB15].

U-Net – specifinė pilnai konvoliucinių tinklų rūšis, kuria susidomėta dėl biomedicininų vaizdų segmentavimo naudojant mažą duomenų rinkinį, tačiau įrodyta, kad šis modelis labai efektyvus ir pikselių klasifikacijai satelitiniuose vaizduose [DKLP+18]. Kai kurie autoriai aprašo platesnį *U-Net* modelį, kuris naudojamas 3D medicininių vaizdų segmentacijai. Abiem atvejais tinklo tikslumas pasiekė 75% naudojant testinę duomenų aibę.



10 pav. U-Net architektūra

Šaltinis: [RFB15]

Šis tinklas sukurtas siekiant segmentuoti biomedicinius vaizdus, tačiau pasak šio tinklo autorių, jis gali plačiai taikomas ir kitoms problemoms spręsti [RFB15]. *U-Net* architektūros tinklai naudojami ir satelitinių vaizdų segmentavimui [RDN18] [IS18].

2.13. Kapsuliniai neuroniniai tinklai objektų segmentavimui

Kapsulinius neuroninius tinklus objektų segmentavimui pirmieji literatūroje pasiūlė R. LaLonde ir U. Bagci 2018 metais [LB18]. Savo darbe jie praplėtė kapsulinių neuroninių tinklų idėją, pridėję tinklam lokaliai sujungtą maršrutizavimą ir dekonvoliucinių kapsulių koncepciją. Pasiūlytas konvoliucinis-dekonvoliucinis kapsulinis tinklas vadinamas *SegCaps* parodė gerus rezultatus medicininių objektų segmentavime (10 pav.).

Metodas	Parametrai	Suskaid.-0	Suskaid.-2	Suskaid.-2	Suskaid.-3	Vidurkis (%)
U-Net	31,0 M	98,353	98,432	98,476	98,510	98,449
Tiramisu	2,3 M	98,394	98,358	98,543	98,339	98,410
Baseline Caps	1,7 M	82,287	79,939	95,121	83,608	83,424
SegCaps (R1)	1,4 M	98,471	98,444	98,401	98,362	98,419
SegCaps	1,4 M	98,499	98,523	98,455	98,474	98,479

1 lentelė. Kapsulinių tinklų rezultatai lyginant su kitais modeliais

Šaltinis: [LB18]

1 lentelėje pavaizduoti rezultatai apskaičiuoti pagal *Dice* koeficientą.

SegCaps kapsulinių neuroninių tinklų modifikacija leidžia į įvestį pateikti didelius vaizdus (512 x 512 pikselių), prieš tai aprašyta kapsulinių tinklų architektūra leido apdoroti tik gerokai mažesnius vaizdus (32 x 32 pikselių) [LB18].

Tačiau šio tipo architektūra nebuvo naudojama satelitinių vaizdų segmentavimui, todėl nėra aišku, kokius rezultatus parodytų, tačiau kadangi *U-Net* tipo architektūra su tam tikrais patobulinimais tiko satelitinių vaizdų segmentavimui, tikėtina, kad *SegCaps* gali parodyti taip pat ganėtinai gerus rezultatus, todėl darbo eigoje būtų naudinga įgyvendinti panašios architektūros tinklą ir satelitinių vaizdų segmentavimui.

2.14. Satelitinių vaizdų segmentavimas

Satelitinių vaizdų segmentavimas nėra nauja problema informatikos moksle. Satelitiniai vaizdai yra turtingas ir struktūrizuotas duomenų šaltinis, tačiau jie yra mažiau tiriami negu kiti labiau kasdieniai kompiuterio regos vaizdai.

Nors literatūros šia tema nėra ypatingai daug, tačiau įvairūs autoriai siūlo skirtingus sprendimus satelitinių vaizdų segmentavimo problemai spręsti, tokius kaip:

1. Naudojant „vandens baseino“ segmentavimo (angl. *watershed segmentation*) metodą originalūs duomenys transformuojami į vaizdą, kuriame kiekvienas pikselis rodo vaizde esantį pakilimą arba nusileidimą. Tamsesni pikseliai rodo mažesnę aukštį. Lokalus minimumas šiame vaizde, kuris yra iš anksto parinktas, vadinamas „gaudymo baseinu“ (angl. *catchment basins*). Visi taškai, kurie patenka į tą patį „gaudymo baseiną“, yra priskiriami tam pačiam baseinui. Segmentavimo proceso metu regionai, apsupantys šiuos baseinus, tampa „užtvindyti“ ir jie tokiu būdu užauga iki sienų su kitais regionais. Galiausiai, vaizdas segmentuojamas į atskirus baseinus, kurie ribojasi su kitais ir yra atskirti linijomis. Taikant šį metodą „baseinų“ skaičius yra lygus nurodytų klasių skaičiui. 2005 m. Wang [WSWC05] pagerino šį metodą, įtraukiant pradinio vaizdo tekstūrų informaciją kartu su gradiento vaizdu.

2. Regionų augimo metodas

Šį metodą pasiūlė L.S. Bins [BFEL96]. Naudojant šį metodą segmentavimo procedūra prasideda kiekvienam sėklos taške vaizde su vieno pikselio objektais, mažesni vaizdo objektai suliejami į didesnius, visų įmanomų porų klasterizavimo procesas rėmėsi trimis kriterijais: spalva, glotnumu ir kompaktiškumu. Balansas kada šie kriterijai yra taikomi priklauso nuo norimos išvesties. Jei mažiausias augimas viršija vartotojo nustatytą heterogeniškumo leistiną nuokrypį, šis procesas sustoja. Heterogeniškumo leistinas nuokrypis daro įtaką išvesties daugiakampių dydžiui. Pagrindinis šio metodo trūkumas, kad per daug segmentuoja homogenines vietas ir dažniausiai nepakankamai segmentuoja tekstūruotas vietas, tokias kaip miškai.

3. Imituoto atkaitinimo metodas

Tai iteracinė procedūra, prasidedanti su nuo pradžių segmentuotu vaizdu ir atsitiktinai keičianti dabartinę būseną. Šis metodas nusprendžia taikyti ar ne naują konfigūraciją, apskaičiuodamas skirtumą tarp tikimybės ar duomenys atitinka segmentavimą. Metodas naudojamas mažų struktūrų segmentavimui, bet jis dažnai klaidingai segmentuoja. Pagrindinės šio metodo problemos, tai prastas mažo kontrasto zonų segmentavimas bei segmento ribų nustatymas mažo kontrastingumo vietose, tokiose kaip miškai ir pievos [CMSO96].

4. Holderio eksponenčių metodas naudojamas didelės skiriamosios gebos satelitinių vaizdų segmentavimui. Holderio eksponentės skaičiavimas įvertina šiurkštumą ar glotnumą aplink kiekvieną pikselį vaizde. Holderio eksponenčių skaičiavimui naudojamas dispersijos matas. Remiantis šio metodo autoriais - šis metodas tinkamai segmentuoja sudėtingus didelės skiriamosios gebos satelitinius vaizdus. Taikant šį metodą nereikia iš anksto apibrėžti kiek klasterių turi būti. [CKH09]

Taip pat naudojami ir kiti metodai satelitinių vaizdų segmentavimui, pavyzdžiui klasterizavimo metodai, kurie sugrupuoja pikselius pagal jų reikšmes ir pozicijas. Šis metodas naudoja mokymąsi be mokytojo [Awa10].

Aptarti metodai nenaudoja GNT segmentavimo uždaviniams spręsti, bet atsiradus galimybėms naudoti giliuosius neuroninius tinklus buvo pradėta vykdyti daug tyrimų dėl jų pritaikomumo vaizdų segmentavimui, tame tarpe ir satelitinių vaizdų.

2.15. Satelitinių vaizdų segmentavimas naudojant giliuosius neuroninius tinklus

Šiuo metu įvairūs šaltiniai dažniausiai aprašo satelitinių vaizdų segmentavimą naudojant įvairias U-Net tinklo architektūros variacijas.

Viena iš tokių variacijų vadinasi *TernausNet* [IS18], šiame tinkle kodavimui (angl. *encoder*) naudojami VGG11 sluoksniai, kurie eina vienas po kito. VGG11 turi septynis konvoliucinius sluoksnius, po kiekvieno iš jų seka ReLu aktyvacijos funkcija ir penki maksimalaus sutelkimo sluoksniai (angl. *max pooling*), kurie mažina požymių žemėlapi (angl. *feature map*). Visi konvoliuciniai sluoksniai turi branduolį 3 x 3. Pirmas konvoliucinis sluoksnis sugeneruoja 64 kanalų ir tuomet tinklui gilėjant, kanalų skaičius dvigubėja kol pasiekia 512. Vėliau einančiuose sluoksniuose kanalų skaičius nesikeičia.

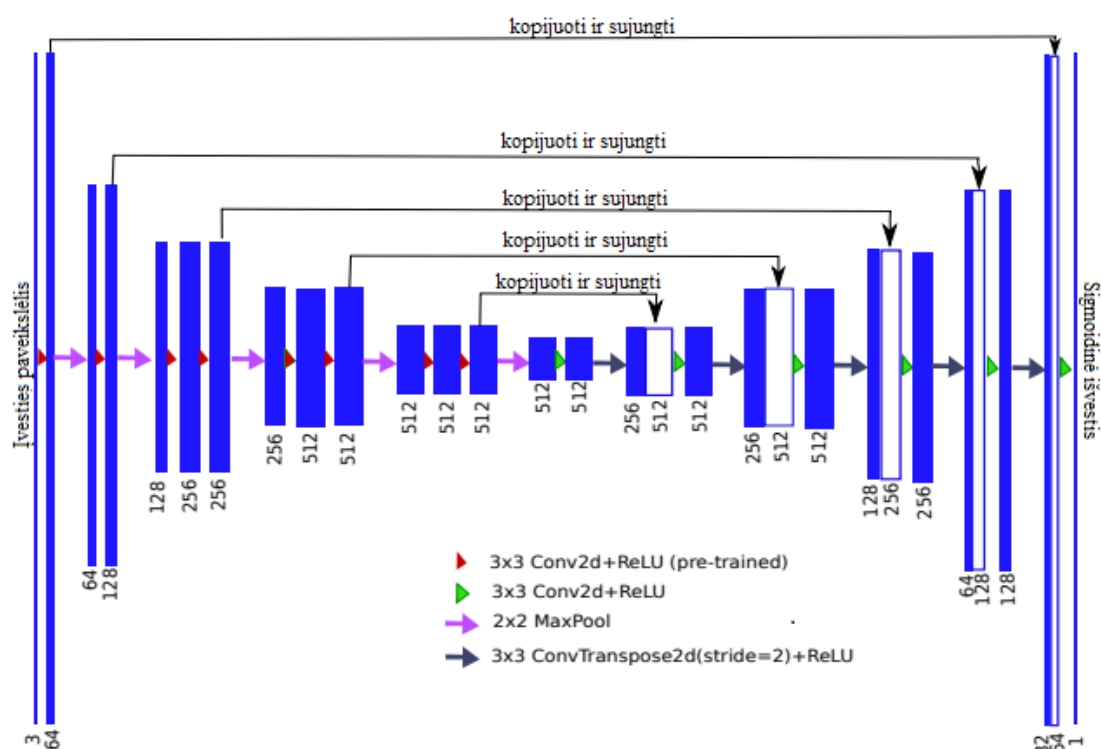
Tam, kad sukurti dekodavimo mechanizmą – pilnai išimami sujungti sluoksniai ir pakeičiami vienu konvoliuciniu sluoksniu su 512 kanalais, kuris yra centrinėje tinklo dalyje ir atskiria kodavimo sluoksnius nuo dekodavimo.

Kuriant šio tinklo dekodavimo sluoksnius naudojami transponuoti konvoliuciniai sluoksniai, kurie dvigubina požymių žemėlapi, o kanalų skaičių mažina per pusę. Transponuotas konvoliucijos išėjimas yra susietas su atitinkamos dekodavimo dalies išėjimu. Gautas požymių žemėlapis apdorojamas konvoliucijos operacija, kad kanalų skaičius būtų simetriškas kodavimo sluoksnyje esančių kanalų skaičiui. Ši atstatymo procedūra kartojama 5 kartus, kad suporuoti su 5 maksimalaus sujungimo sluoksniais.

TernausNet tinklui apmokyti ir validuoti buvo naudojamas Inria Aerial vaizdų ženklavimo duomenų rinkinys ir buvo išbandyti 3 vienodos architektūros (11 pav.) *U-Net* tinklai skyrėsi tik būdais kaip inicializuojami svoriai. Rezultatai buvo skaičiuojami naudojant *Jaacard* metriką kitaip vadinama IoU.

TernausNet pasiekti rezultatai su validavimo duomenų aibe [IS18]:

1. LeCun tolyginis iniciatorius: IoU = 0,593
2. Kodavimo sluoksniai iš anksto apmokyti su *ImageNet*: IoU = 0,686
3. Pilnai iš anksto apmokytas *U-Net* tinklas su *Carvana*: IoU = 0,687



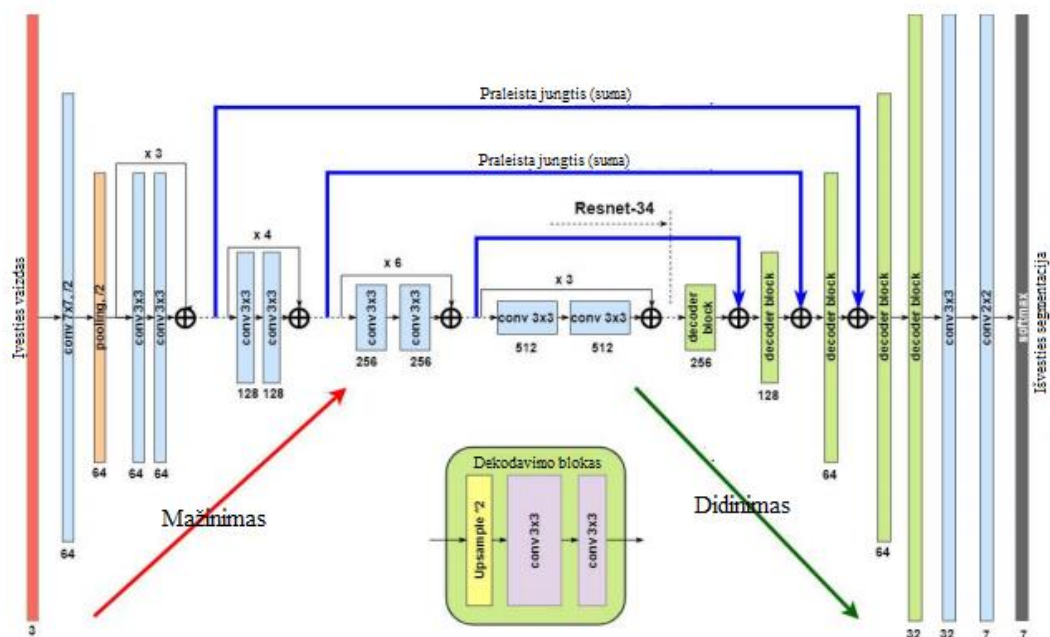
pav. 1 TernausNet architektūros schema

Šaltinis: [IS18]

Iš anksto apmokytos tinklo variacijos parodė geresnius rezultatus. Ir tokio tipo tinklai sumažina mokymosi laiką ir užkerta kelią persimokymui.

Sprendžiant žemės paviršiaus (vandens telkiniai, miškai, keliai it kt.) klasifikavimo uždavinį, taip pat buvo naudojamas *U-Net* architektūros neuroninis tinklas (12 pav.), tačiau tam tikromis modifikacijomis. Pirmiausia tyrimo metu buvo įvertinti trys skirtingi kodavimo sluoksniai [RDN18].

1. *VGG* pagrindu padaryta neuroninį tinklą *m46*
2. *Resnet-34*
3. *Inception Resnet*



2 pav. *U-Net* architektūra su *ResNet-34* kodavimu

Šaltinis: [RDN18]

Resnet architektūrose buvo pasiūlyti smulkūs pakeitimai. Pirmiausia buvo pakeista aktyvacijos funkcija iš *ReLU* į *ELU* ir pakeistas eiliškumas aktyvacijos ir paketų normalizacijos sluoksnių, bei naudojamas *Lovasz-Softmax* nuostolių funkcija. Žemiau pateiktoje lentelėje matyti skirtingų šios architektūros tinklo rezultatai, atsižvelgus į kodavimo sluoksnius ir nuostolių funkciją.

U-Net koduotojas	#params	Jaccard	Lovasz
m46	1M	0,624	0,619
Resnet-34	25M	0,615	0,641
Inception Resnet v2	61M	0,604	0,573

2 lentelė. Rezultatų palyginimas

Šaltinis: [RDN18]

Taigi, šių autorių nuomone, labiausiai prisidėjo prie bendros segmentavimo kokybės, tai *m46* kodavimo architektūra, skirta pašalinti trūkumams dėl duomenų stokos. Subalansuotas atrankos metodas (angl. *equibatch*), padedantis kovoti su klasių disbalansu, ir svarbiausia - *Lovasz-Softmax* nuostolių funkcija, specialiai sukurta optimizuoti IoU (angl. *intersection over union*) pagrįstas metrikas. Autorių nuomone neseniai sukurta *Lovasz-Softmax* nuostolių funkcija vaidins svarbų vaidmenį vaizdų segmentavime [RDN18].

Kitame tyrime sprendžiant pastatų atpažinimo problemą [CBB17], autoriai taip pat naudojo šiek tiek pakeistą U-Net architektūros tinklo versiją. Tyrimo pradžioje naudotas iš anksto apmokytas tinklas (įgyvendintas *ZFTurbo*), tačiau kadangi toks sprendimas nepadėjo tinklo mokymuisi jo buvo atsisakyta ir tinklas buvo apmokytas nuo nulio su keliais pirminės architektūros pakeitimais.

Pirmiausia buvo pakeistas stochastinio gradiento optimizavimo algoritmas į *Adam*. *Adam* algoritmas apjungia dviejų skirtingų metodų privalumus, tai *AdaGrad* (angl. *Adaptive Gradient Algorithm*) ir *RMSProp* (angl. *Root Mean Square Propagation*). Dažniausiai naudojant stochastinio gradiento nusileidimo metodą naudojama viena mokymo greičio reikšmė (angl. *Learning rate*) visų svorių keitime ir ji išlieka pastovi. *Adam* algoritmo taikymo atveju mokymo greitis derinamas mokymo procese.

Pakeitus optimizavimo algoritmą buvo pakeisti įvesties vaizdų matmenys, pirminė U-Net tinklo versija, kuri buvo skirta 572 x 572 x 3 dydžio vaizdams. Taip pat buvo naudojamas įprastas padidinimas (angl. *padding*), kad būtų išvengtas susitraukimas vykdant konvoliucijos operaciją. Tam, kad mokymas būtų pagerintas buvo pridėta paketo normalizacija po kiekvieno *ReLU* aktyvacijos sluoksnio. Parinkta nuostolių funkcija pagrįsta *dice* koeficientu vietoj kryžminės entropijos nuostolių funkcijos. *Dice* koeficientu paremta nuostolių funkcija dažniausiai naudojama vaizdų segmentavimui, kuri leidžia kovoti su klasių disbalansu. Išmetimo sluoksniai nebuvo naudojami, nes dėl duomenų padidinimo modelio mokymosi eigoje neatsirado persimokymo požymių. Galiausiai siekiant palengvinti optimizaciją ir sustabdyti nykstantį gradientą buvo nutarta išimti paskutinį sujungimo sluoksnį. Vertinant parengtą modelį su testavimo aibe, gautas *Dice* koeficientas 0,60 ir Jaccard'o koeficientas. Tačiau autoriai pastebi, kad gautas rezultatas yra labai priklausomas nuo vaizdų skiriamosios gebos [CBB17].

Ankstesni darbai daugiausiai buvo sutelkti į vaizdų klasifikavimą, kai kiekvienas vaizdas klasifikuojamas tik į vieną klasę. I. Demir ir kt. [DKLP+18] savo darbe apraše architektūrą, kurios pagalba segmentuojamos kelios klasės esančios vaizde. Jie sukūrė KNT *DeepLab* pagrindu naudodami *ResNet18* ir erdvinį piramidės sutraukimo (angl. ASPP) bloką ir paketo normalizaciją. Duomenų kiekį padidino naudojant vaizdų rotaciją. Šis metodas pasiekė *IoU*: 0,433 30-oje epochoje.

2.16. Apibendrinimas

Išanalizavus pateiktus šaltinius, darytina išvada, kad atsiradus pakankamiems skaičiavimo pajėgumams gilieji neuroniniai tinklai plačiai taikomi vaizdų analizei ir segmentavimui. Satelitinių vaizdų analizei dažniausiai pasitelkiami U-Net architektūros gilieji neuroniniai tinklai, tačiau tyrimų šia tema nėra pakankamai daug, tuo pačiu pastebėtina, kad išskirtinai miškų ir vandens telkinių analizės tema darbų neparašyta. Neseniai sukurti kapsuliniai neuroniniai tinklai dar nėra plačiai paplitę, tačiau demonstruoja gerus rezultatus biomedicininė vaizdų segmentavime, kadangi tokiai analizei ypatingai svarbu kiekvieną pikselį identifikuoti tinkamai, manytina, kad šio tinklo architektūra galbūt būtų tinkama ir satelitinių vaizdų analizei, todėl tyrimo eigoje būtų naudinga pritaikyti ir šios architektūros tinklus satelitinių vaizdų analizei.

3. Darbo rezultatai ir jų aptarimas

Mokslo tiriamojo darbo eigoje buvo nuspręsta sukurti du skirtingus GNT modelius, vieną - *U-Net* architektūros, kitą – *SegCaps* architektūros, ir palyginti šių modelių rezultatus. *U-Net* ir *SegCaps* modeliai rodo puikius rezultatus medicininių vaizdų segmentavime. *U-Net* modelis buvo naudojamas ir satelitinių vaizdų segmentavimui, ten taip pat rodė gerus rezultatus. Modeliams apmokyti ir rezultatams tikrinti buvo naudojama ta pati satelitinių vaizdų duomenų aibė.

3.1. Duomenų aibė

Satelitinių vaizdų duomenų bazių su kokybiškai pažymėtomis klasėmis nėra itin daug, todėl tyrimui buvo naudojama *Kaggle.com* satelitinių vaizdų duomenų aibė, skirta vienam iš konkursų. Šio konkurso metu buvo varžomasi satelitinių vaizdų segmentavime. Satelitinius vaizdus šiam konkursui pateikė Jungtinės Karalystės Gynybos mokslo ir technologijų laboratorija (angl. *DSTL*).

Tačiau šie vaizdai neturėjo paruoštų kaukių GNT apmokymui. Kaukės – tai yra paruošti vaizdai su pažymėtomis objektų klasėmis. Įprastai, kuriant kaukes, visi pikseliai, priklausantys klasei, žymimi viena spalva (įprastai balta), o nepriklausantys – juoda spalva. Tačiau šioje duomenų aibėje vietoje vaizdų su kaukėmis buvo pridėtas failas su pažymėtais klasių poligonais. Poligonai – tai klasės ribų aprašas vaizde (koordinatės), todėl norint pasinaudoti šia duomenų aibe reikėjo pirminio duomenų apdorojimo ir pritaikymo dirbtinio neuroninio tinklo mokymui (Kaukių kūrimo procesas aprašomas tolimesniuose skyriuose).

Ši duomenų aibė buvo sudaryta iš 4 tipų satelitinių vaizdų: įprasti 3-jų kanalų *RGB* vaizdai, tačiau labai didelės skiriamosios gebos, bei 16 ir 8 kanalų satelitiniai vaizdai, turintys multispektrinę bei trumpųjų infraraudonųjų spindulių diapazoną. Kadangi 3-jų kanalų *RGB* vaizdai užima daug vietos ir yra pakankamai dideli, buvo nuspręsta rinktis iš kitų tipų satelitinių vaizdų. Darbo eigoje naudojami 8 kanalų multispektriniai vaizdai, nes jie buvo pakankamai didelės skiriamosios gebos ir daugiakanaliai, kas leidžia jiems turėti pakankamai daug informacijos tinklo mokymuisi.

Iš šios vaizdų duomenų bazės buvo atrinktos 25 satelitinės nuotraukos, kuriose pavaizduoti miškai, pavieniai medžiai arba vandens telkiniai.

Vaizdų rinkinys susidėjo iš tokių vaizdų:

- 25 *.tiff formato satelitiniai vaizdai, kurių skiriamoji geba 849 x 837.
- 1 km x 1 km atvaizduojamas žemės paviršiaus plotas.
- 8 kanalų multispektriniai vaizdai.

3.2. Kompiuteriniai resursai

GNT atliekami skaičiavimai reikalauja didelių kompiuterinių resursų, ypač dirbant su vaizdų apdorojimu. Vaizdai užima disko vietą, todėl norint juos nuskaityti ir paversti į skaliarinį pavidalą, reikia juos perkelti į kompiuterio operatyviają atmintį. Taigi norint sėkmingai įgyvendinti išskeltus uždavinius buvo pasitelktos *Google Cloud Platform* (toliau – GCP) debesies kompiuterijos paslaugos. GCP leidžia pačiam susikonfigūruoti serverį su reikalinga konfigūracija.

Serverio parametrai:

- 8 branduolių *Intel Skylake* procesorius (*CPU*);
- 52 GB operatyvios atminties (*RAM*);
- 100 GB SSD diskas;
- 1 x *NVIDIA Tesla K80* vaizdo apdorojimo procesorius (*GPU*);

3.3. Programinė įranga

Šiuo metu labiausiai darbui su dirbtiniais neuroniniais tinklais, mašininio mokymusi pritaikyta kalba yra *Python*. Šia programavimo kalba parašyta daug bibliotekų, palengvinančių darbą su dirbtiniais neuroniniais tinklais, vaizdų apdorojimu. Be to, *Python* – tai dinaminė programavimo kalba, todėl jos nereikia kompiliuoti, tai supaprastina programinio kodo rašymą.

Tensorflow – tai *Google* sukurta atviro kodo platforma, skirta mašininiam mokymuisi. Ji turi išsamią ir lanksčią įrankių, bibliotekų bei bendruomenės išteklių ekosistemą, leidžiančią tyrėjams paspartinti dirbtinių neuroninių tinklų įgyvendinimą. Darbo su *Tensorflow* palengvinimui naudojama *Keras* biblioteka, kurios pagalba lengvai įgyvendinami konvoliuciniai ir kiti dirbtinių neuroninių tinklų sluoksniai, paskaičiuojamos įvairios metrikos, leidžiančios įvertinti sukurtą modelį. *Keras* – tai *API*, kurio pagrindas yra *Tensorflow*, o per šią sąsają yra paprasčiau įgyvendinti įvairių architektūrų dirbtinius neuroninius tinklus.

Darbui su vaizdų failais yra sukurta įvairių bibliotekų. Pirmiausia, *TiffFile*, skirta failų nuskaitymui ir įrašymui. Šios bibliotekos pasirinkimą lėmė tai, kad pasirinktoje duomenų aibėje vaizdai buvo išsaugoti *.tiff formatu. *Numpy* biblioteka buvo pasirinkta darbui su daugiamačiais masyvais, ši biblioteka taip pat skirta moksliniams skaičiavimams. Pavyzdžiui, *TiffFile* biblioteka nuskaitytas vaizdas iš karto paverčiamas į masyvą, kuris suprantamas *Numpy* bibliotekai ir leidžia manipuliuoti šiuo masyvu. Toks suderinamumas palengvina darbą su vaizdais ir leidžia labiau susikoncentruoti į modelio kūrimą.

Toliau pateikiamas sąrašas bibliotekų panaudotų darbo eigoje:

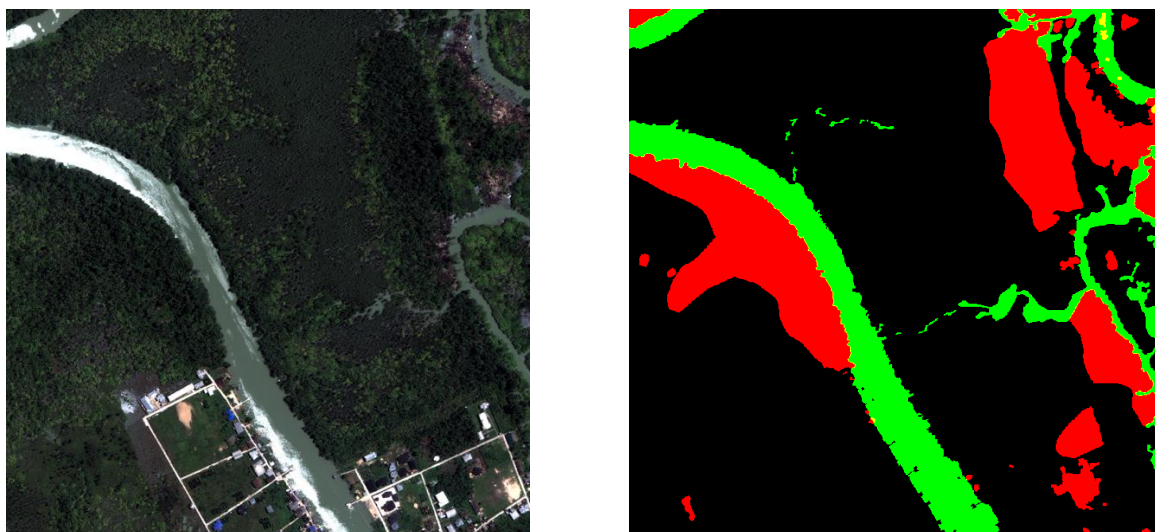
- *Python 3.6.9*,
- *TensorFlow 1.14.0*,
- *Keras 2.1.6*,
- *TiffFile 2019.7.26.2*,
- *Numpy 1.18.1*,
- *Skimage 0.16.2*,
- *GDAL 2.4.0*,
- *Pandas 0.23.4*,
- *Cv2*,

3.4. Duomenų paruošimas dirbtinio neuroninio tinklo mokymuisi

Kaip jau minėta, pasirinktoje duomenų aibėje vaizdai buvo pateikti be pažymėtų klasių kaukių, tačiau papildomai buvo pateiktas failas su klasių poligonų koordinatėmis, todėl pirmiausia prieš pradėdant kurti GNT modelį, reikėjo paruošti kaukes tinklo mokymuisi.

Pirmiausia buvo nuskaitomi *Pandas* bibliotekos *.csv failai ir GDAL bibliotekos pagalba nuskaitomas originalaus paveiksluko dydis, kad kaukių dydis nesiskirtų nuo originalaus vaizdo, taip daryti būtina nes esant bent mažiausiam skirtumui, galima tikėtis prastesnių tinklo mokymosi rezultatų.

Vėliau nuskaitomos poligonų koordinatės, apskaičiuojamos kaukių kraštinės ir išsaugomos į failą. Kadangi vienas iš šio darbo tikslų – atpažinti dviejų klasių objektus, tai yra miškus ir vandens telkinius, tam kiekvienos klasės kaukės buvo išsaugomos skirtinguose vaizdo sluoksniuose (13 pav.)



13 pav. Satelitinis vaizdas ir jo kaukė

13 pav. dešinėje paveikslėlio pusėje matoma kaukė su išskirtais pirmuoju ir antruoju spalvų kanalais, o kairėje pusėje matomas originalus vaizdas. Modeliui, darant klasių atpažinimą ir segmentuojant vaizdą, buvo saugomi 2 atskiri rezultatų failai, kiekviename jų pavaizduota skirtinga klasė.

Kadangi satelitinių vaizdų skiriamoji geba yra pakankamai didelė, dėl ko pritrūkdavo operatyvios atminties, prieš pateikiant juos į įvesties sluoksnį papildomai reikėjo „sukarpyti“ į mažesnes dalis. Taip pat „karpat“ vaizdą jis buvo dalinamas į mokymosi ir testavimo aibes santykiu 3 prie 1. Papildomai buvo pritaikyta duomenų augmentacija, tai yra vaizdai ir kaukės pasukti įvairiais kampais, tokiu būdu siekiant padidinti mokymosi aibę siekiant kokybiškesnio modelio apmokymo.

Prieš pateikiant vaizdus į modelio įvesties sluoksnį buvo pritaikytas duomenų normalizavimas. Normalizavimo metu kiekvienam vaizdo pikseliui buvo priskirta reikšmė tarp -1 ir 1 (formulė pateikta žemiau). Šis normalizavimas buvo taikomas tam, kad būtų galima optimaliau apmokyti modelį. Be to paminėtina, kad netaikant duomenų normalizavimo pirminiame etape modeliai rodė prastesnius rezultatus, ypač *SegCaps*.

$$x = 2 \frac{x - \min x}{\max x - \min x} - 1$$

3.5. *U-Net* architektūros dirbtinis neuroninis tinklas

Literatūros analizės skyriuje šios architektūros GNT buvo aprašytas kaip gerus rezultatus vaizdų segmentavime rodantis tinklas. Todėl pirmiausia darbo eigoje buvo bandyta sukurti šios topologijos modelį. Sukurto GNT architektūros diagrama pateikta prieduose.

Šį tinklą sudaro:

- 10 konvoliucinių sluoksnių (angl. *Convolutional*)
- 5 atstatymo sluoksniai (angl. *Up Sampling*)
- Sujungimo sluoksniai (angl. *MaxPooling*)
- Išmetimo sluoksniai (angl. *Dropout*) skirti tam, kad DNT nepersimokytų.
- Aktyvacijos funkcija: *ReLU*
- Optimizavimo algoritmas: *Adam*
- Nuostolių funkcija: *Jaacard*

Adam algoritmas buvo pasirinktas dėl savo efektyvumo ir gebėjimo greitai pasiekti gerų rezultatų.

ReLU aktyvacijos funkcijos privalumas yra tas, kad jos įgyvendinimas yra mažiau brangus kompiuterinių resursų prasme, o tai ypač svarbu turint gilius dirbtinius neuroninius tinklus, nes tai leidžia DNT greičiau mokytis.

Jaccard nuostolių funkcija nuo įprastos *Jaccard* metrikos skiriasi tuo, kad metrikos rezultatai reikia atimti iš 1. Tai yra padaryti skaičiavimą atvirkščiu, reiškiantį, kad kuo mažesnis rezultatas – tuo mokymasis vyksta geriau.

3.5.1. *U-Net* tinklo mokymasis

U-Net modelio mokymuisi buvo nustatyta 100 epochų, kiekvienos epochos metu modelis mokėsi iš 3000 pateiktų sukarpytų satelitinių vaizdų. Validavimo aibė buvo sudaryta iš 750 vaizdų.

Modelio mokymosi nustatymai:

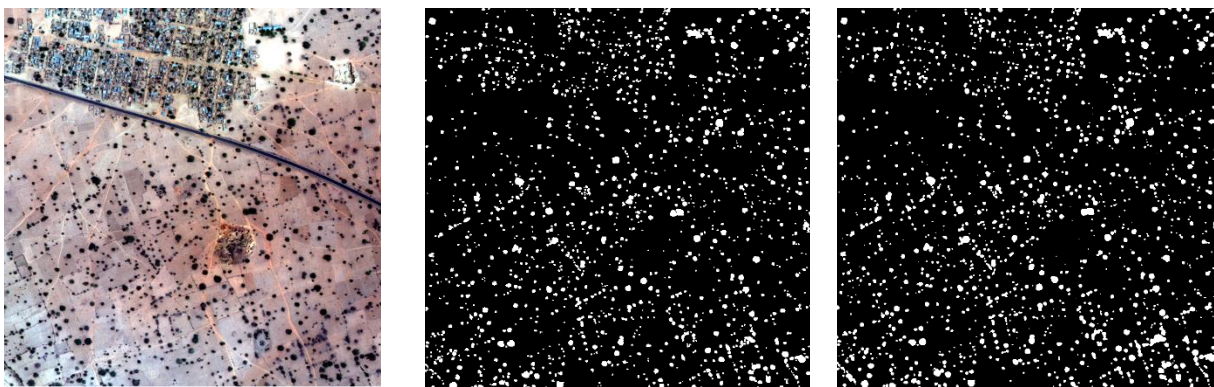
- 100 epochų
- Paketo dydis (angl. *batch size*) 50 (esant didesniai dydžiui pritrūkdavo atminties *GCP* serveryje).

Modelis mokymosi metu pakankamai greitai pradėjo gauti gerus rezultatus. Pirminiame etape modeliui apmokyti buvo pasirinkta 50 epochų (tarpiniai rezultatai pateikiami prieduose), tačiau mokymosi metu modelis ties paskutine epocha mokymosi rezultatus dar gerino, todėl sekančiame tyrimo etape epochų skaičius buvo padidintas iki 100, be to įvesties duomenys buvo normalizuoti ir tai leido pagerinti šio modelio rezultatus, lyginant su pirmuoju etapu.

3.5.2. *U-Net* rezultatai

U-Net tipo GNT parodė pakankamai neblogus rezultatus atpažįstant miškus ir vandens telkinius. Geriausio atvejo modelio rezultatas atpažįstant miškus siekė 0,7430, o vandens telkinius – 0,9302.

Žemiau pateiktame paveiksle, segmentuojant mišką, matyti skirtumas tarp kaukės ir gautų rezultatų. Kairėje pusėje pavaizduotas originalus vaizdas, viduryje - kaukė, o dešinėje - modelio segmentavimo rezultatas. Vizualiai skirtumus pastebėti yra ganėtinai sudėtinga, nors *Jaccard* metrika rodo, kad vaizdai nėra identiški.



14 pav. *U-Net* miškų segmentavimo geriausias rezultatas

Atpažįstant vandens telkinius modelio geriausio atvejo rezultatas yra netgi aukštesnis negu atpažįstant miškus, kaip matyti iš žemiau pateikto paveikslėlio šis modelis pakankamai tiksliai atpažino du vandens telkinius. Galbūt tam įtaką turėjo tai, kad vandens telkinių spalva šiame vaizde išsiskyrė nuo aplinkinių spalvų. Be to vandens telkiniai susidaro iš didesnio sujungtų pikselių kiekio.



15 pav. U-Net vandens telkinių segmentavimo geriausias rezultatas

Žemiau pateiktoje lentelėje parodyti U-Net modelio rezultatai. Žalia spalva pažymėti geriausi modelio rezultatai, o raudona blogiausi.

Vaizdo ID	Miškai	Vandens telkiniai
6010_1_2	0,5935	-
6010_4_2	0,7065	-
6010_4_4	0,6205	-
6040_1_0	0,5664	-
6040_1_3	0,6144	-
6040_2_2	0,6483	-
6040_4_4	0,5163	-
6060_2_3	0,7193	-
6070_2_3	0,5375	0,8048
6090_2_0	0,5708	0,8062
6100_1_3	0,6339	0,5359
6100_2_2	0,5088	0,8100
6100_2_3	0,6283	-
6110_1_2	0,6175	0,6146
6110_3_1	0,7042	0,9302
6110_4_0	0,7153	-

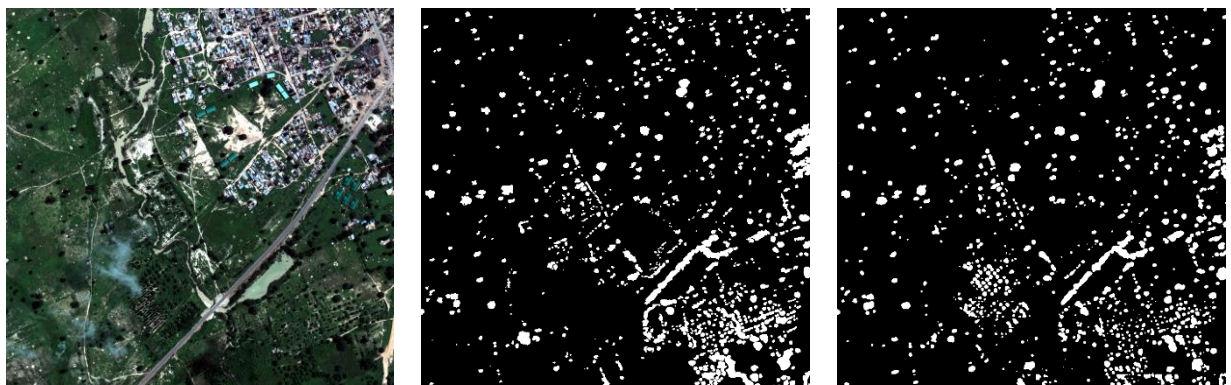
6120_2_0	0,7430	-
6120_2_2	0,6506	-
6140_1_2	0,5766	-
6140_3_1	0,7073	-
6150_2_3	0,7128	-
6160_2_1	0,5506	-
6170_0_4	0,6835	-
6170_2_4	0,5866	-
6170_4_1	0,5753	-

3 lentelė. U-Net segmentavimo rezultatai

Kaip matyti iš pateiktos 3 lentelės, modelio rezultatai, segmentuojant vandens telkinius, yra stabilesni negu atpažįstant miškus, tačiau abiem atvejais tarp geriausio ir blogiausio rezultatų yra nemažas skirtumas.

Pažymėtina tai, kad apmokant modelį per 50 epochų, rezultatai ženkliai skyrėsi ir modelis geresnius rezultatus rodė atpažindamas miškus, bei buvo stabilesnis, tačiau padidinus epochų skaičių ir pritaikius duomenų normalizavimą, modelis geresnius bei stabilesnius rezultatus rodo atpažindamas vandens telkinius, nors ir mokymosi aibė buvo mažesnė.

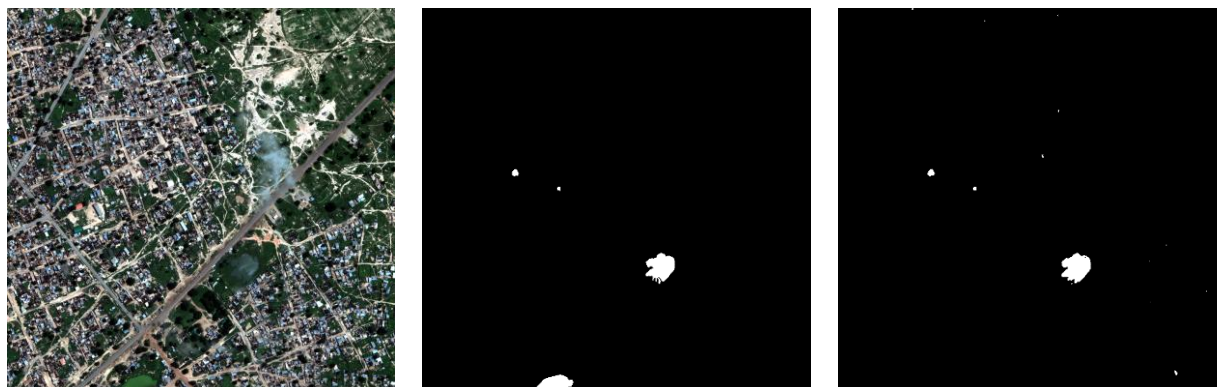
Žemiau pateikta kaukė vaizdo 6100_1_2 (viduryje) ir tinklo gautas rezultatas dešinėje, nors pagal *Jaacard* santykį šie vaizdai yra mažiausiai panašūs, tačiau vizualiai skirtumas nėra toks akivaizdus, galbūt dėl to, kad ši metrika rodo geresnius rezultatus su didesniais objektais, o šiame vaizde pateikti daugiausiai pavieniai smulkūs medžiai. Be to šiame vaizde yra daug žalios spalvos kuri nėra priskirta medžiams, tai galėjo suklaidinti modelį.



16 pav. U-Net miško segmentavimas (blogiausias atvejis)

Iš tinklo pateiktų rezultatų (17 pav.) matyti, kad tam tikrų vandens telkinių modeliui nepavyko atpažinti visiškai. Modelis nepažymėjo nei vieno pikselio esančio vandens telkinio plote. Sunku pasakyti, kodėl tai įvyko, bet vizualiai peržiūrint vaizdą matyti, kad vandens telkinio spalva yra

visiškai kitokia nei kitų pažymėtų vaizde, galbūt šis skirtumas ir lėmė, tai kad modelis suklydo. Nepaisant to modelis tiksliai atpažino aukščiau esantį vandens telkinį.



17 pav. U-Net vandens telkinių segmentavimas (blogiausias atvejis)

Siekiant objektyviai įvertinti *U-Net* tipo modelį, žemiau pateiktoje lentelėje pavaizduoti modelio geriausio, prasčiausio ir vidutinio spėjimo rezultatai. Kaip matyti iš pateiktos lentelės, modelis parodė geriausią rezultatą su vaizdu, kuriame pavaizduoti vandens telkiniai. Be to ir rezultatų vidurkis yra geresnis nei atpažįstant miškus. Kaip jau minėta modelis rodė visiškai priešingus rezultatus prie 50 mokymosi epochų, tad manytina, kad epochų skaičiaus padidinimas leido modeliui geriau išmokti atpažinti didesnio ploto objektus.

	Mišakai	Vandens telkiniai
Blogiausias	0,5088	0,5359
Geriausias	0,7430	0,9302
Rezultatų vidurkis	0,6275	0,7503

4 lentelė. U-Net rezultatų santrauka

Apibendrinant galima teigti, kad *U-Net* modelis, lyginant su literatūros analizėje pateiktais kitų autorių rezultatais, rodo gerus rezultatus. Pradiniame etape buvo pasirinktas modelis, turintis ne tokia gilią architektūrą, bei buvo pasirinktas mažesnis mokymosi epochų skaičius – 50. Padidinus epochų skaičių iki 100 bei pridėjus duomenų normalizavimą, tinklas ženkliai pagerino rezultatus. Rezultatų vidurkis analizuojant miškus nuo 0,6395 pakilo iki 0,7198, o vandens telkinius nuo 0,5632 iki 0,6090.

3.6. *SegCaps* architektūros dirbtinis neuroninis tinklas

SegCaps – palyginus neseniai įgyvendintas modelis, parodęs puikius rezultatus medicininių 3D vaizdų segmentavime, tačiau duomenų apie atliktus tyrimus satelitinių vaizdų segmentavime nėra. Todėl sunku buvo nuspėti kaip pavyks įgyvendinti šios architektūros modelį ir kokius rezultatus šis tinklas pademonstruos.

Šis tinklas susidaro iš konvoliucinio sluoksnio, konvoliucinių kapsulių sluoksnių ir dekonvoliucinių kapsulių sluoksnių. Kaip ir *U-Net* atveju buvo naudojamas *Adam* optimizavimo algoritmas, bei *ReLU* aktyvacijos funkcija. Šios architektūros tinklų ypatumas yra tas, kad tarp kapsulių vyksta dinaminis maršrutizavimas [SF17]. Dinaminis maršrutizavimas sugrupuoja kapsules tam, kad suformuotų tėvinę kapsulę ir suskaičiuoja kapsulės išėjimo reikšmę, kuri šiuo atveju nėra skaliarinė reikšmė kaip įprastinių konvoliucinių neuroninių tinklų atveju, o vektorius.

Pasak šių tinklų autorių geriausia naudoti 1 arba 3 maršrutizavimus. Esant didesniai skaičiui tinklas gali persimokyti. Atliekant šį tyrimą buvo pasirinktas 3-jų lygių maršrutizavimas.

Modelio architektūra:

- 1 x konvoliucinis sluoksnis;
- 1 x pirminis kapsulių sluoksnis;
- 4 x konvoliucinių kapsulių sluoksniai;
- 4 x dekonvoliucinių kapsulių sluoksniai;
- *Adam* optimizavimo algoritmas;
- *ReLU* aktyvacijos funkcija;

3.6.1. *SegCaps* modelio mokymasis

Pirminiame etape apmokant *SegCaps* architektūros modelį, buvo susidurta su problemomis ir pavyko pasiekti rezultatų tik naudojant pakeistus 3-jų kanalų vaizdus į nespaltvotus (angl. *greyscale*) vaizdus bei juos programiškai sumažinus iki 512 x 512 dydžio. Šiuo atveju tinklui užteko 20 epochų, kad mokymasis nebebūtų efektyvus ir tinklas nebegerintų rezultatų, be to modelis pasižymėjo nestabilumu (tarpiniai rezultatai pateikiami prieduose).

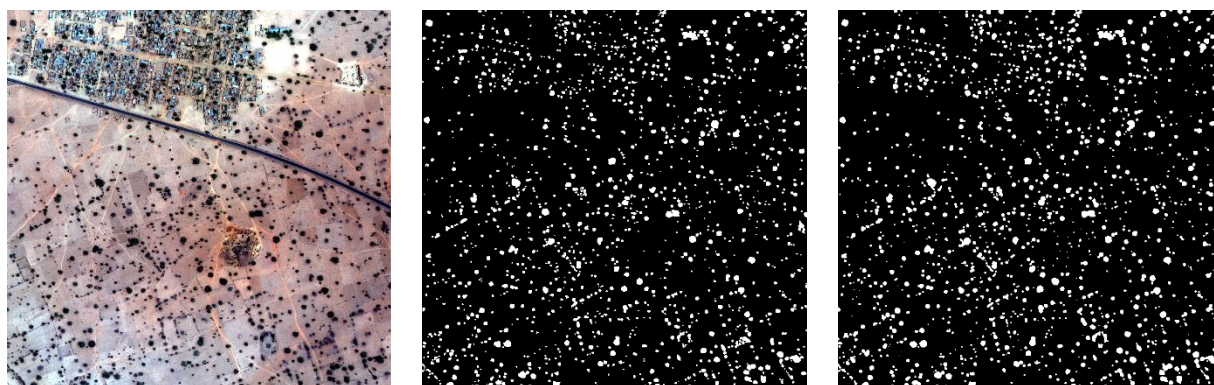
Pritaikius duomenis taip, kad tinklas sugebėtų dirbti su 8-ių kanalų vaizdais nekeičiant jų pirminio dydžio, o sukarplant į mažesnius 160 x 160 vaizdus, leido tinklo rezultatus pagerinti. Be to, tinklui prireikė 75 epochų bei 3000 sukarpytų mokymosi vaizdų bei 750 vaizdų validavimui.

Mažesni skaičiai negu *U-Net* atveju buvo pasirinkti todėl, nes atliekant bandymus modelis po 75 epochos pradėdavo rodyti nebegerėjančius rezultatus.

3.6.2. *SegCaps* modelio rezultatai

SegCaps rodė stabilėsius rezultatus atpažįstant miškus ir pavienius medžius negu vandens telkinius, tačiau didelius miško plotus *SegCaps* segmentavimo taip pat nesėkmingai. Galbūt taip įvyko dėl to, kad mokymui buvo naudojami sukarpyti vaizdai, bet tai yra tik spėjimas. Tačiau ateities bandymuose būtų naudinga tinklą apmokyti su didesniais vaizdais.

Kaip jau minėta modelis buvo mokomas su vaizdais kurių dydis 160×160 pikselių, o šios architektūros modelio autoriai naudojo didesnius 512×512 , todėl manytina, kad įvesties vaizdų padidinimas galėtų turėti įtakos modelio rezultatams.



18 pav. *SegCaps* miško segmentavimas (geriausias atvejis)

Kaip matyti iš palyginimui pateiktų paveikslų (18 pav.), modelis, bandydamas atpažinti medžius, pažymėdavo papildomus pikselius aplinkui, tai ir turėjo pagrindinės įtakos prastesniam modelio rezultatui, tačiau vizualiai skirtumai sunkiai pastebimi.

Įdomesnė situacija įvyko bandant segmentuoti vandens telkinius (pav. 19). Modelis pažymėjo ne tik upę, tačiau ir miške esančius medžius ir kitus objektus. Atskirai analizuojant vaizdus galima pastebėti, kad vandens telkiniai, upės turi skirtingas savybes, skiriasi spalva, forma. Todėl tikėtina, kad modelis gavo per mažai duomenų, kad galėtų tinkamai segmentuoti vandens telkinius. Vykdamas tolimesnius tyrimus, būtų naudinga pateikti modeliui daugiau pavyzdžių mokymuisi, galbūt tai leistų pagerinti rezultatą. Be to, galima pastebėti, kad kai kurie pikseliai pažymėti segmentavimo rezultate, realiame vaizde spalviškai panašūs į vandens telkinius.



19 pav. *SegCaps* vandens telkinių segmentavimas (geriausias atvejis)

SegCaps modelis, atpažindamas tame pačiame vaizde mišką, parodė blogiausią rezultatą, bet kaip matyti iš žemiau pateikto 20 pav. ir vėl pastebima, kad aplinkui mišką yra daug žalios spalvos, kuri nepriklauso miško klasei, tad manytina, kad ir šiuo atveju aplinkinių pikselių spalva daro įtaką modelio rezultatams.



20 pav. *SegCaps* miškų segmentavimas (blogiausias atvejis)

Tą patį galima pastebėti ir modeliui atpažįstant vandens telkinius. Nors modelis pažymėjo vieną vandens telkinį gan tiksliai, tačiau papildomai pažymėjo daugybę kitų pikselių, kurie nepriklauso vandens telkiniams, tačiau spalviškai gali juos priminti. Be to, šis modelis kaip ir *U-Net* nesugebėjo atpažinti vandens telkinio esančio apačioje.



21 pav. *SegCaps* vandens telkinių segmentavimas (blogiausias atvejis)

Kaip matyti iš žemiau pateiktos 5 lentelės - modelis nerodo stabilumo atpažįstant vandens telkinius, be to rezultatai yra žemesni palyginus su prieš tai nagrinėtų *U-Net* modelių. Žalia spalva pažymėti geriausi modelio rezultatai, o raudona blogiausi.

Apibendrinant, galima sakyti, kad *SegCaps* modelis prastesnius rezultatus rodė atpažįstant miškus tais atvejais kai vaizde vyravo daugiau žalios spalvos (tas pats buvo pastebima ir nagrinėjant *U-Net* modelį). O geriausius rezultatus modelis rodė tuomet, kai buvo aiškiai matyti spalvų skirtumas tarp miškų ir kitų aplinkinių objektų. Todėl manytina, kad nors ir 8-ių kanalų vaizdai pateikia daugiau informacijos modeliui, visgi didelę reikšmę turi pikselių spalvos.

Vaizdo ID	Mišškai	Vandens telkiniai
6010_1_2	0,5578	-
6010_4_2	0,6516	-
6010_4_4	0,5802	-
6040_1_0	0,4630	-
6040_1_3	0,5710	-
6040_2_2	0,6163	-
6040_4_4	0,5032	-
6060_2_3	0,6904	-
6070_2_3	0,3934	0,7147
6090_2_0	0,4801	0,6565
6100_1_3	0,53	0,1254
6100_2_2	0,4	0,663
6100_2_3	0,5141	-
6110_1_2	0,4246	0,2607
6110_3_1	0,6399	0,4934
6110_4_0	0,6308	-
6120_2_0	0,6934	-
6120_2_2	0,5763	-
6140_1_2	0,4448	-
6140_3_1	0,5962	-
6150_2_3	0,5919	-
6160_2_1	0,4824	-
6170_0_4	0,6036	-
6170_2_4	0,5154	-
6170_4_1	0,5233	-

5 lentelė. SegCaps modelio rezultatai

Pateiktoje rezultatų palyginimo 6 lentelėje matyti, kad šiam modeliui trūksta stabilumo atpažįstant vandens telkinius, o skirtumas tarp blogiausių ir geriausių rezultatų ganėtinai didelis. Todėl darytina išvada, kad šiuo atveju šis modelis nėra sėkmingas ir tobulintinas.

	Miškai	Vandens telkiniai
Blogiausias	0,3934	0,1254
Geriausias	0,6938	0,7147
Rezultatų vidurkis	0,5470	0,4856

6 lentelė. SegCaps rezultatų abibendrinimas

3.7. SegCaps ir U-Net modelių rezultatų palyginimas

Kaip matyti iš žemiau pateiktos lentelės, visais atvejais geresnius rezultatus rodė *U-Net* architektūros dirbtinis neuroninis tinklas. Tad galima teigti, kad taikant tokį patį pirminį duomenų apdorojimą ir karpant vaizdus iki 160 x 160 dydžio, *U-Net* rodo geresnius rezultatus. Tačiau savo moksliniame tyrime *SegCaps* kūrėjai naudojo didesnius – 512 x 512 dydžio vaizdus, tad galbūt padidinus modelio įvesties vaizdų dydį, rezultatus taip pat būtų įmanoma pagerinti. Taip pat pažymėtina, kad *U-Net* vandens telkinius atpažino geriau negu miškus, tuo tarpu *SegCaps* geresnius rezultatus rodė atpažįstant miškus. Tačiau sudėtinga pasakyti, kodėl šių tinklų gaunami rezultatai skiriasi.

Pažymėtina ir tai, kad mokymosi metu *SegCaps* modelis rodė geresnius rezultatus su mokymosi aibe negu *U-Net*, nors validavo aibės rezultatai buvo prastesni.

Atliekant šį tyrimą paaiškėjo, kad pagrindinis *SegCaps* privalumas, lyginant su *U-Net* architektūros tinklu, yra tas, kad tinklas pakankamai greitai mokosi. Jis nėra toks gilus kaip įprastas *U-Net* tinklas, todėl viena epocha trunka trumpiau negu *U-Net*. Todėl ateities tyrimuose būtų naudinga pagerinti šio modelio rezultatus, kadangi laiko ir kompiuterinių resursų atžvilgiu jis yra kur kas efektyvesnis. Be to, naudojant medicininius vaizdus rodė geresnius rezultatus. Tačiau tie medicininiai vaizdai buvo 3-jų dimensijų, o satelitiniai vaizdai - 2-jų, bet šie vaizdai daugiakanaliai.

Analizuojant šių abiejų modelių rezultatus, galima pastebėti, kad aplinkinių pikselių spalva turi įtakos modelių rezultatams. *U-Net* ir *SegCaps* modeliai esant didesniai kiekiui žalios spalvos pikselių rodė prastesnius rezultatus atpažįstant miškus.

	U-Net		SegCaps	
	Miškas	Vandens telkiniai	Miškas	Vandens telkiniai
Blogiausias	0,5088	0,5359	0,3934	0,1254
Geriausias	0,7430	0,9302	0,6933	0,7147
Vidurkis	0,6275	0,7503	0,5470	0,4856

7 lent. SegCaps ir U-Net modelių rezultatų palyginimas

4. Išvados

1. Mokslinės literatūros analizės metu paaiškėjo, kad geriausius rezultatus vaizdų segmentavime rodo *U-Net* ir *SegCaps* architektūros dirbtiniai neuroniniai tinklai. Abiejų šių tipų GNT buvo sukurti medicininiam vaizdams segmentuoti, o *U-Net* plačiai naudojamas ir satelitinių vaizdų segmentavime. *SegCaps* dar nėra taip plačiai paplitęs, todėl rasti duomenų apie bandymus, daromus su satelitiniais vaizdais, rasti nepavyko. Todėl buvo pasirinkta įgyvendinti šios architektūros modelius.
2. *U-Net* architektūros modelis geresnius rezultatus rodė atpažindamas vandens telkinius. Geriausio atvejo rezultatas skaičiuojant Jaacard'o metriką siekė 0,9302, o vidutinis 0,7503. Tai gali būti dėl to, kad vandens telkiniai susideda iš didesnių pikselių grupių. Tuo tarpu pateiktuose vaizduose miškai dažniausiai buvo vaizduojami kaip pavieniai medžiai ar nedidelės medžių grupės. Tad atpažindamas miškus *U-Net* modelis vidutiniškai pasiekdavo 0,6275 rezultatą, o geriausio atvejo 0,7430.
3. *SegCaps* modelis geresnius rezultatus rodė atpažindamas miškus. Tikėtina, taip gali būti dėl to, kad šis modelis buvo kuriamas atpažinti mažus objektus medicininuose 3D vaizduose, o kaip jau minėta pateiktuose vaizduose vyravo pavieniai medžiai ir nedidelės medžių grupės. Vidutiniškai modelis pasiekdavo 0,5470 rezultatą, o geriausio atvejo rezultatas siekė 0,6933. Atpažinti vandens telkinius modeliui buvo sudėtingiau, be to, rezultatai buvo nestabilūs. Geriausio atvejo rezultatas 0,7147, o vidutiniškai modelis pasiekdavo 0,4856 rezultatą.
4. Palyginus *U-Net* ir *SegCaps* modelių rezultatus darytina išvada, kad *U-Net* modelis geriau atpažįsta vandens telkinius ir miškus satelitiniuose vaizduose. Bet *SegCaps* modelis greitai mokosi ir maksimalių mokymosi rezultatų pasiekia ankstyvesnėje epochoje negu *U-Net*, todėl būtų naudinga ateityje pratęsti tyrimą ir pritaikyti modelį 512 x 512 vaizdams, galbūt tai leistų pagerinti šio modelio rodomus rezultatus.

Literatūra

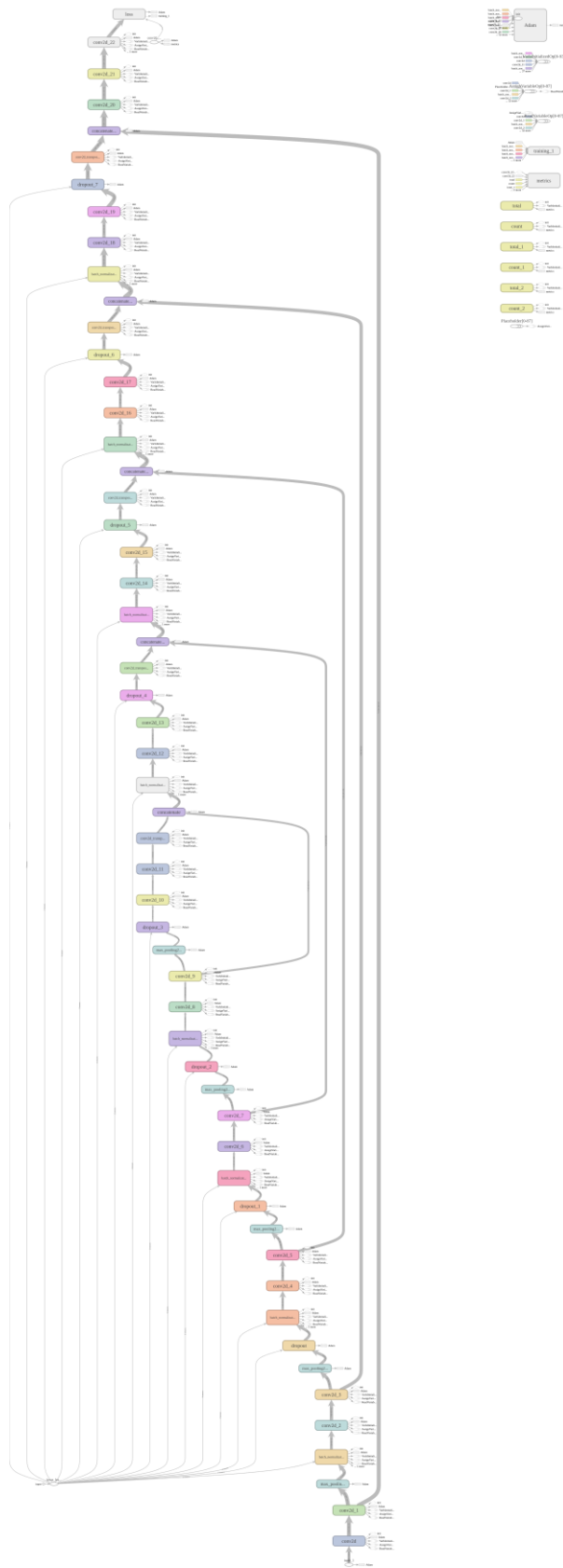
- [Sch15] J. Schmidhuber. Deep Learning in Neural Networks: An Overview. Neural Networks 61. 2015 pp. 85-117
- [LeC98] Y. LeCun. LeNet-5, convolutional neural networks [žiūrėta 2018-12-05] Prieiga per Internetą: <<http://yann.lecun.com/exdb/lenet/>>
- [STE13] Ch. Szegedy, A. Toshev, D. Erhan. Deep neural networks for object detection. Advances in Neural Information Processing Systems 26. 2013
- [Ben09] Y. Bengio. Learning Deep Architectures for AI. Foundations and Trends® in Machine Learning Vol. 2 No 1. 2009. pp. 1-127
- [CKH09] D. Chakraborty, G. Kumar Sen, S. Hazra. High-resolution satellite image segmentation using Hölder exponents. Journal of Earth System Science 118(5). 2009. pp. 609-617
- [WSWC05] Z. Wang, C. Song, Z. Wu ir X. Chen Improved watershed Segmentation algorithm for high resolution remote sensing images using texture. 2005
- [BFEL96] L.S. Bins, L. M. G. Fonseca, G. J. Ertha ir F. A. M. Ii. Satellite imagery segmentation: A region growing approach. INPE. 1996. p. 677-680
- [CMSO96] R. Cook, I. McConnell, D. Stewart and C. J. Oliver . Segmentation and simulated annealing. 1996
- [Pra91] W. Pratt. Satellite Imagery Segmentation: a region growing approach. 1991
- [RFB15] U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015
- [IS18] V. Iglovikov, A. Shvets. TerausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation. 2018
- [PG17] J. Patterson, A. Gibson. Deep Learning: Practitioner's Approach-O'Reilly Media. 2017
- [SF17] S. Sabour, N. Frosst. Dynamic Routing Between Capsules. 2017
- [KLA18] T. Karras, S. Laine, T. Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. 2018
- [RDN18] A. Rakhlin, A. Davydow, S. Nikolenko. Land Cover Classification from Satellite Imagery With U-Net and Lovasz-Softmax Loss. 2018
- [DKLP+18] I. Demir, K. Koperski, D. Lindenbaum, G. Pang, J. Huang, S. Basu, F. Hughes, D. Tuia, R. Raskar. DeepGlobe 2018: A Challenge to Parse the Earth through Satellite Images. 2018
- [CBB17]G. Chhor, C. Bartolome Aramburu, anis Bougdal-Lambert. Satellite Image Segmentation for Building Detection using U-net. 2017

[LB18] R. LaLonde, U. Bagci. Capsules for Object Segmentation. 2018

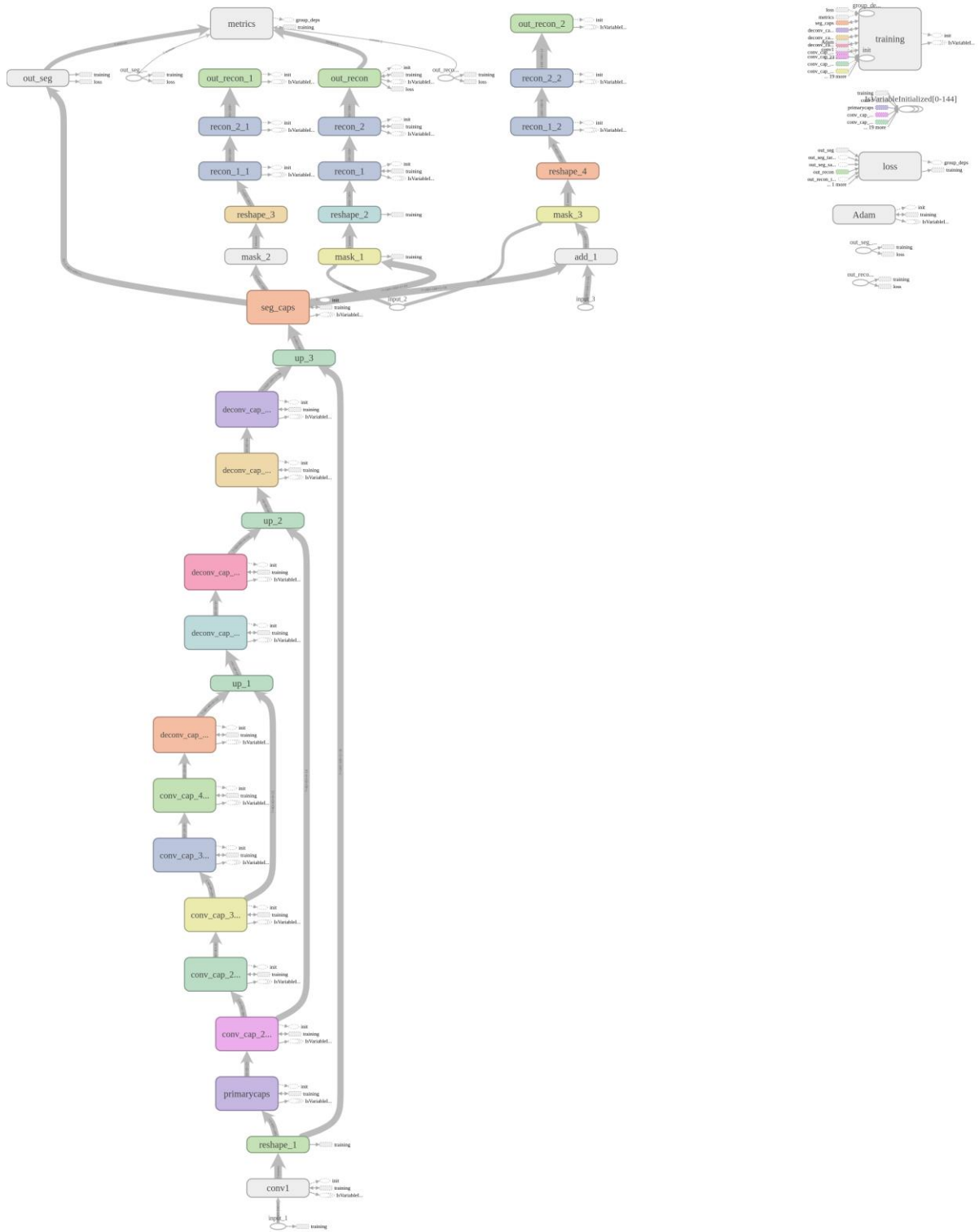
[Awa10] M. Awad. An Unsupervised Artificial Neural Network Method for Satellite Image Segmentation. 2010

Priedai

1 priedas. U-Net modelio architektūros diagrama



2 priedas. SegCaps modelio architektūros diagrama



3 priedas. *U-Net* modelio tarpinių rezultatų lentelė

Vaizdo ID	Mišakai	Vandens telkiniai
6010_1_2	0,5923	-
6010_4_2	0,7158	-
6010_4_4	0,6447	-
6040_1_0	0,5703	-
6040_1_3	0,6211	-
6040_2_2	0,6559	-
6040_4_4	0,5266	-
6060_2_3	0,7238	-
6070_2_3	0,7096	-
6090_2_0	0,5775	0,6473
6100_1_3	0,6340	0,2384
6100_2_2	0,5412	-
6100_2_3	0,6780	-
6110_1_2	0,5914	0,3596
6110_3_1	0,7089	0,8676
6110_4_0	0,7245	0,7932
6120_2_0	0,7462	-
6120_2_2	0,6459	0,7780
6140_1_2	0,5459	0,3082
6140_3_1	0,6853	0,8405
6150_2_3	0,7270	-
6160_2_1	0,5626	-
6170_0_4	0,6884	-
6170_2_4	0,5903	0,2365
6170_4_1	0,5813	-

4 priedas. SegCaps modelio tarpinių rezultatų lentelė

Vaizdo ID	Miškai	Vandens telkiniai
6010_1_2	0,2438	
6010_4_2	0,4948	
6010_4_4	0,4360	
6040_1_0	0,4976	
6040_1_3	0,4366	
6040_2_2	0,5366	
6040_4_4	0,3998	
6060_2_3	0,5627	
6070_2_3	0,0728	
6090_2_0	0,3710	0,2471
6100_1_3	0,2390	0,5809
6100_2_2	0,2163	
6100_2_3	0,1944	
6110_1_2	0,2995	0,1444
6110_3_1	0,4523	0,0797
6110_4_0	0,3885	0,2114
6120_2_0	0,4849	
6120_2_2	0,3698	0,0844
6140_1_2	0,3292	0,1853
6140_3_1	0,5181	0,1587
6150_2_3	0,5114	
6160_2_1	0,4240	
6170_0_4	0,5111	
6170_2_4	0,4087	0,0687
6170_4_1	0,4245	