Kauno technologijos universitetas

Informatikos fakultetas

Automatinis pastatų kontūrų segmentavimas palydoviniuose ir aeronuotraukų vaizduose

Baigiamasis magistro studijų projektas

Laurynas Buinauskas

Projekto autorius

Doc. dr. Eglė Butkevičiūtė

Vadovė

Kaunas, 2025

Kauno technologijos universitetas

Informatikos fakultetas

Automatinis pastatų kontūrų segmentavimas palydoviniuose ir aeronuotraukų vaizduose

Baigiamasis magistro studijų projektas

Programų sistemų inžinerija (6211BX011)

Laurynas Buinauskas

Projekto autorius

Doc. dr. Eglė Butkevičiūtė

Vadovė

Dr. Lina Narbutaitė

Recenzentė

Kaunas, 2025

Kauno technologijos universitetas

Informatikos fakultetas

Laurynas Buinauskas

Automatinis pastatų kontūrų segmentavimas palydoviniuose ir aeronuotraukų vaizduose

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;

2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąraše;

3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs (-usi);

4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Laurynas Buinauskas

Patvirtinta elektroniniu būdu

Kauno technologijos universitetas

Informatikos fakultetas

Baigiamojo magistro projekto užduotis

Projekto tema	Automatinis pastatų kontūrų segmentavimas palydoviniuose ir aeronuotraukų vaizduose
Reikalavimai ir sąlygos (tikslinti pavadinimą pagal poreikį)	
Vadovas / Vadovė	
	(vadovo pareigos, vardas, pavardė, parašas) (data)

Buinauskas Laurynas. Automatinis pastatų kontūrų segmentavimas palydoviniuose ir aeronuotraukų vaizduose. Magistro baigiamojo projekto vadovė doc. dr. Eglė Butkevičiūtė; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Programų sistemos.

Reikšminiai žodžiai: gilusis mokymasis, vaizdų segmentavimas, pastatų kontūrai, palydovinės nuotraukos, aeronuotraukos.

Kaunas, 2025. 110 p.

Santrauka

Šio darbo tikslas yra ištirti ir eksperimentuoti su pažangiais giliojo mokymosi algoritmais, skirtais automatiniam palydovinių nuotraukų segmentavimui, siekiant sudaryti tikslius pastatų kontūrus. Tyrimas koncentruojasi į semantinio segmentavimo metodus taikant konvoliucinius neuroninius tinklus, kad kiekvienas vaizdo pikselis būtų tiksliai priskirtas vienai iš dviejų klasių: pastatas arba ne pastatas. Eksperimentai ir tyrimai vykdomi naudojant *Python* programavimo kalbą, *PyTorch* karkasą ir juo paremtas giliojo mokymosi bibliotekas, kurios suteikia prieigą prie modernių modelių architektūrų bei palengvina jų pritaikymą. Darbo metu siekiama įvertinti skirtingų algoritmų tikslumą bei greitį sprendžiant šį specifinį vaizdų segmentavimo uždavinį ir ieškoti efektyvių būdų juos optimizuoti.

Tokių segmentavimo automatinių algoritmų poreikis yra didelis – jie leidžia efektyviai atnaujinti žemėlapius, padeda planuoti miestų plėtrą bei infrastruktūrą (pvz., kelių poreikį) bei gali būti naudojami nelegalios statybos stebėjimui. Siekiant geriausių rezultatų ir inovatyvių sprendimų, šiame darbe bus tiriami, lyginami ir taikomi įvairūs šiuolaikiniai modelių tipai, įskaitant plačiai naudojamas kodavimo-dekodavimo architektūras bei naujesnius, perspektyvius transformeriais pagrįstus modelius, pritaikytus vaizdų segmentavimo uždaviniams.

Buinauskas Laurynas. Automatic Building Contour Segmentation in Satellite and Aerial Imagery. Master's Final Degree Project supervisor Assoc. Prof. Dr. Eglė Butkevičiūtė; Faculty of Informatics, Kaunas University of Technology.

Study field and area (study field group): Program systems.

Keywords: Deep Learning, Image Segmentation, Building Contours, Satellite Images, Aerial Images.

Kaunas, 2025. 110.

Summary

The goal of this work is to investigate and experiment with advanced deep learning algorithms for the automatic segmentation of satellite and aerial images, aiming to create accurate buildings contours. The research focuses on semantic segmentation methods, particularly applying convolutional neural networks (CNNs), so that each image pixel is accurately assigned to one of two classes: building or non-building. Experiments and research are conducted using the Python programming language, PyTorch framework and deep learning libraries based on it, which provide access to modern model architectures and facilitate their application. During the work, the aim is to evaluate the accuracy and speed of different algorithms in solving this specific satellite image segmentation task and to search for effective ways to optimize them.

The need for such automated segmentation algorithms is significant – they allow for the efficient updating of maps, assist in planning urban development and infrastructure (e.g., road requirements), and can be used to monitor illegal construction. To achieve the best results and innovative solutions, this work will investigate, compare, and apply various contemporary model types, including widely used encoder-decoder architectures as well as newer, promising transformer-based models adapted for image segmentation tasks.

Turinys

Lentelių sąrašas

Paveikslų sąrašas

Santrumpų ir terminų sąrašas

Santrumpos

CNN – Konvoliuciniai neuroniniai tinklai (angl. *Convolutional Neural Networks*, *CNN*).

FCN – Pilnai konvoliuciniai tinklai (angl. *Fully Convolutional Networks*, *FCN*).

Terminai

Gilusis mokymasis (angl. *Deep Learning*) – Mašininio mokymosi metodas, naudojantis daugiasluoksnius neuroninius tinklus sudėtingiems dėsningumams mokytis.

Dirbtinis neuronų tinklas (angl. *Artificial Neural Network*) – Skaičiavimo modelis iš sujungtų neuronų sluoksnių, apdorojantis duomenis ir gebantis mokytis.

Konvoliucija (angl. *Convolution*) – Operacija, kurios metu filtras slenka per vaizdą ar duomenis, ieškodamas specifinių požymių.

Filtro branduoliai (angl. *Kernels*) – Mažos matricos (filtrai) konvoliuciniuose tinkluose, skirtos specifiniams požymiams aptikti.

Persimokymas (angl. *Overfitting*) – Modelio būsena, kai jis labai gerai išmoksta mokymo metu naudotus duomenis, bet prastai veikia su naujais.

Transformeris (angl. *Transformer*) – Giliojo mokymosi modelio architektūra, pagrįsta dėmesio (angl. *attention*) mechanizmais ryšiams tarp duomenų sekos elementų įvertinti.

Koduotuvas (angl. *Encoder*) – Modelio komponentas arba atskiras tinklas, kuris transformuoja įvesties duomenis (pvz., vaizdą) į žemesnės dimensijos reprezentaciją, vadinamą kodu arba vektoriumi, siekiant išgauti esminius požymius.

Dekoderis (angl. *Decoder*) – Modelio komponentas arba atskiras tinklas, kuris priima koduotą (žemos dimensijos) reprezentaciją ir transformuoja ją atgal į pradinį duomenų formatą (pvz., atkuriant vaizdą) arba į kitą pageidaujamą išvestį.

Bazinis modelis (arba fundamentas) (angl. *Backbone*) – Dažnai iš anksto apmokyta modelio dalis (pvz., kodavimo dalis kodavimo-dekodavimo architektūroje), kuri išgauna pagrindinius požymius iš įvesties duomenų, vėliau naudojamus specifinei užduočiai atlikti.

Įvadas

Informacijos išgavimas iš palydovinių nuotraukų, pavyzdžiui, pastatų ar kitų objektų identifikavimas, yra svarbus daugelyje sričių, įskaitant kartografiją, urbanistiką ir aplinkos stebėseną. Semantinis segmentavimas – kompiuterinės regos metodas, priskiriantis kiekvienam vaizdo pikseliui tam tikrą klasę (pvz., pastatas), tampa naudingu įrankiu detaliai geoerdvinei informacijai išgauti. Pastaraisiais metais gilieji neuroniniai tinklai, ypač konvoliuciniai neuroniniai tinklai (angl. *Convolutional Neural Networks, CNN*), pademonstravo išskirtinius gebėjimus automatizuoti šį procesą ir pasiekti aukštą tikslumą [1, 2, 3, 4].

Darbo naujumas ir aktualumas

Nuolat augant palydovinių duomenų kiekiui ir raiškai, automatizuoti analizės metodai tampa vis labiau aktualesni, nes rankinis žemėlapių sudarymas ar atnaujinimas yra gan lėtas procesas, kuris kartu yra brangus ir sunkiai pritaikomas dideliems plotams. Tikslūs ir laiku atnaujinami apstatymo ploto žemėlapiai yra svarbūs efektyviam miestų planavimui, infrastruktūros vystymui, aplinkosaugos stebėsenai bei net nelegalios statybos aptikimui [5, 6]. Nors giliojo mokymosi taikymas vaizdų segmentavimui nėra nauja sritis, šio darbo aktualumas kyla iš poreikio tirti ir adaptuoti naujausias giliojo mokymosi metodikas (pvz., transformerių modelius) specifiniam pastatų atpažinimo palydovinėse nuotraukose uždaviniui, siekiant ne tik aukšto tikslumo, bet ir skaičiavimų efektyvumo bei optimizavimo galimybių.

Tikslas ir uždaviniai

Tikslas: ištirti, apmokyti ir palyginti giliojo mokymosi modelius automatiniam pastatų kontūrų segmentavimui palydoviniuose bei aeronuotraukų vaizduose, analizuojant jų tikslumą pagal skirtingas vertinimo metrikas, hiperparametrų konfigūracijas ir įvairius duomenų rinkinius.

Uždaviniai:

1. atlikti egzistuojančių giliojo mokymosi metodų, taikomų semantiniam segmentavimui, analizę;
2. apibrėžti tyrimo objektą ir parengti tyrimo metodologiją, parenkant tinkamus įrankius, technologijas bei duomenų rinkinius ir nustatant rezultatų vertinimo kriterijus;
3. realizuoti ir apmokyti giliojo mokymosi modelius, atlikti eksperimentinius tyrimus siekiant įvertinti ir palyginti skirtingų modelių architektūrų bei konfigūracijų efektyvumą pastatų segmentavimo uždavinyje, analizuoti gautus rezultatus ir pateikti moksliškai pagrįstas išvadas apie tirtų metodų tinkamumą bei pagrindines optimizavimo kryptis.

Dokumento struktūra

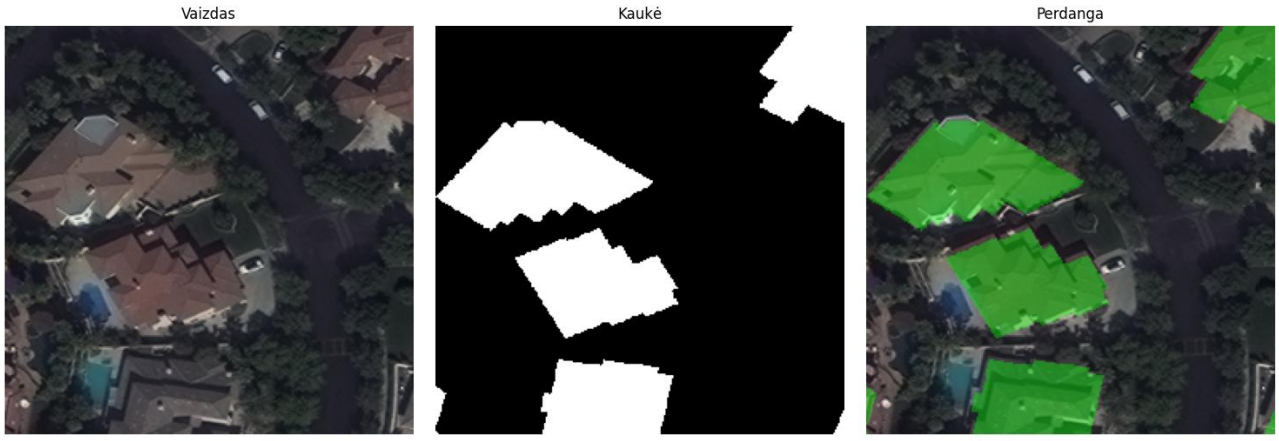
Šiame darbe pirmiausia yra apžvelgiama literatūra, susijusi su semantiniu segmentavimu ir taikomais giliojo mokymosi metodais. Toliau pristatoma tyrimo metodologija: aprašomi naudoti duomenys, taikyti metodai, vertinimo metrikos ir eksperimentų planas. Po to seka eksperimentinė dalis su tyrimo metu gautais rezultatais. Galiausiai, formuluojamos darbo išvados ir aptariamos tolimesnių tyrimų galimybės.

1. Literatūros apžvalga

1.1. Problema ir darbo tikslas

Pastatų kontūrų aptikimas yra semantinio segmentavimo uždavinys. Semantinis segmentavimas yra esminė kompiuterinės vizijos užduotis bei detalesnė vaizdų klasifikavimo forma, apimanti kiekvieno nuotraukos pikselio klasifikavimą į nustatytas kategorijas. Nuotolinio stebėjimo kontekste šis metodas leidžia išgauti tikslią geoerdvinę informaciją iš palydovinių nuotraukų ir aeronuotraukų. Skirtingai nuo vaizdų klasifikavimo ar objektų aptikimo, semantinis segmentavimas suteikia pikselių lygio prognozes, kurios yra svarbios užduotims, kuriose reikia tiksliai apibrėžti kontūrus.

Kadangi šio projekto tikslas yra automatiškai aptikti pastatų kontūrus, užduotis specifiškai formuluojama kaip dvejetainis semantinis segmentavimas, supaprastinant klasifikavimą iki dviejų pagrindinių klasių: pastatas ir ne pastatas (arba fonas). Ši dvejetainė klasifikacija yra pagrindinis reikalavimas pastatų kontūrų aptikimui. Nuotraukos ir joje segmentuotų pikselių (kaukės) pavyzdys pateiktas 1 pav.



1 pav. Nuotrauka su pažymėtais pikseliais, apibrėžiančiais pastatų kontūrus

1.2. Pamatiniai giliojo mokymosi metodai

Perėjimą nuo tradicinių metodų prie giliojo mokymosi semantiniam segmentavimui paskatino architektūrų, kurios iš pradžių buvo sukurtos vaizdų klasifikavimui, pritaikymas. CNN suteikė pagrindinius elementus, o FCN pasiūlė esminę architektūrinę modifikaciją, leidžiančią atlikti pikselių lygmens prognozes.

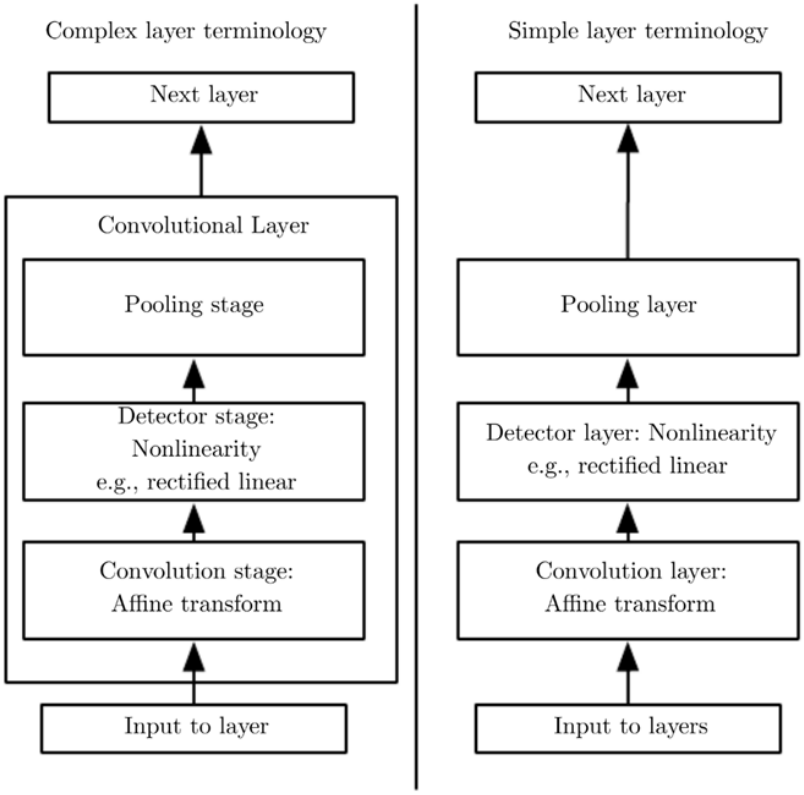
1.2.1. Konvoliuciniai neuroniniai tinklai

Gilusis mokymasis, konkrečiai konvoliuciniai neuroniniai tinklai (CNN), šiuo metu yra pažangiausia metodika semantinio segmentavimo užduotims spręsti. Jų stiprumas yra gebėjime automatiškai išmokti hierarchines požymių reprezentacijas tiesiogiai iš neapdorotų pikselių, taip pašalinant rankinės požymių inžinerijos (angl. *feature engineering*) poreikį [7, 8]. Tipiška CNN architektūra susideda iš tam tikrų sluoksnių sekos. Pagrindiniai komponentai yra šie:

- konvoliuciniai sluoksniai: jie taiko apmokomus filtrus (filtro branduolius, angl. *kernels*) įvesties vaizdui (arba praeito sluoksnio požymių žemėlapiui), kad aptiktų dėsningumus erdvėje, tokius kaip kampai, tekstūros, kraštinės bei taip pat galiausiai sudėtingesnes savybes ir tendencijas. Filtrų parametrai (svoriai) nustatomi apmokymo proceso metu;

- aktyvacijos funkcijos: jos taikomos kiekvienam konvoliucinio sluoksnio išvesties elementui (t. y. kiekvienam atskiram skaičiui gautame požymių žemėlapyje) individualiai (angl. *element-wise*). Pagrindinė jų užduotis – įvesti į modelį netiesiškumą (matematinę savybę, kai funkcijos išvestis nėra tiesiogiai proporcinga įvesčiai), kuris yra būtinas. Šis netiesiškumas leidžia visam neuroniniam tinklui mokytis sudėtingų priklausomybių bei hierarchinių požymių duomenyse, o ne apsiriboti tiktais paprastomis tiesinėmis kombinacijomis. Tokių funkcijų pavyzdys yra plačiai CNN naudojama *ReLU* (dalimis tiesinis vienetas, angl. *Rectified Linear Unit*);
- sutelkimo (angl. *pooling*) sluoksniai: esminė šių sluoksnių funkcija – sumažinti požymių žemėlapių erdvinius matmenis (plotį ir aukštį). Šiame sluoksnyje įvesties duomenys suskaidomi į mažus regionus, taip apibendrinant informaciją juose ir tokiu būdu išlaikant tiktais svarbiausius požymius. Dažniausiai naudojamas sutelkimas imant maksimalią reikšmę (angl. *Max Pooling*), kuris iš kiekvieno suskaidyto požymių žemėlapio langelio pasirenka didžiausią reikšmę. Tai padeda sumažinti skaičiavimo apimtį bei kontroliuoti modelio persimokymą (angl. *overfitting*). Taip pat, sutelkimas suteikia tinklui tam tikrą invariantiškumą postūmiui (angl. *translation invariance*) – t. y. savybę, dėl kurios požymio aptikimas mažiau priklauso nuo jo visiškai tikslios padėties vaizde – todėl modelis geriau atpažįsta požymius, net jei jie vaizde yra šiek tiek pasislinkę.

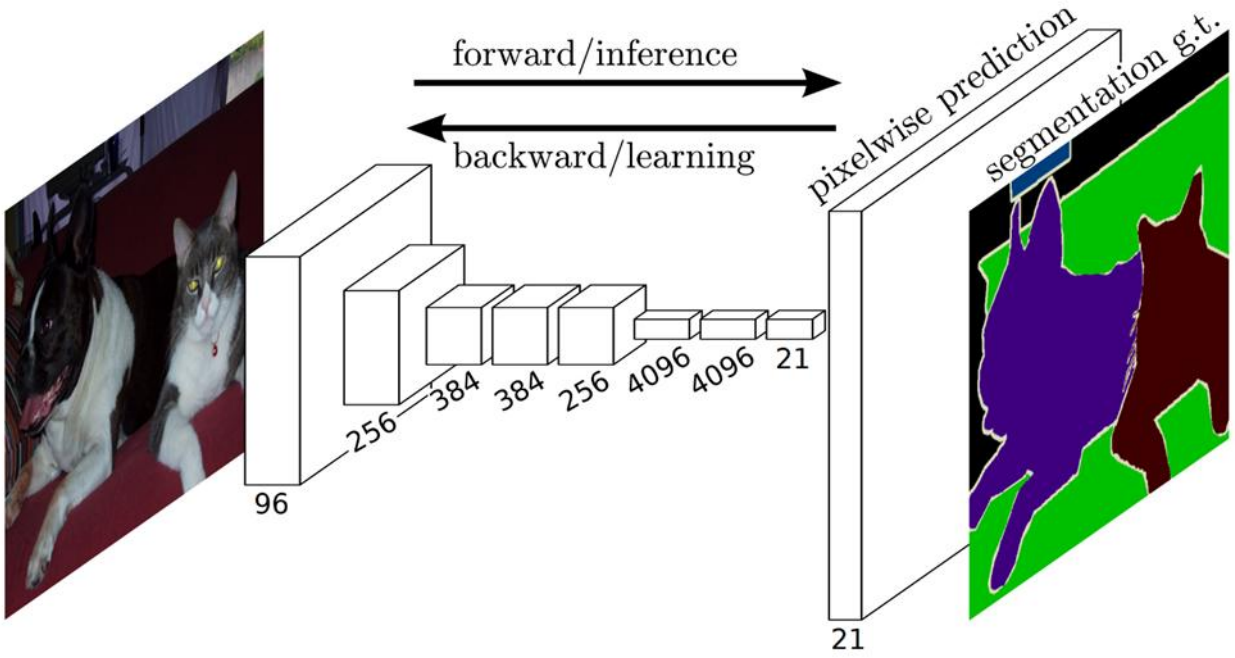
Dedant šiuos sluoksnius vieną po kito (angl. *stacking*), CNN laipsniškai kuria vis abstraktesnes ir sudėtingesnes reprezentacijas (t. y., skaitiniuose duomenyse išmoktų požymių ar šablonų aprašus). Ankstyvieji sluoksniai užfiksuoja žemo lygio požymius, o gilesni sluoksniai juos sujungia, kad reprezentuotų objektų dalis ar ištisus objektus. Tai yra būtina tankiai, pikselių lygio klasifikacijai, reikalingai semantiniame segmentavime [9]. Bendroji CNN architektūros schema (žr. 2 pav.) [8 p. 336].



2 pav. Tipinio CNN sluoksnio komponentai

1.2.2. Pilnai konvoliuciniai tinklai

Semantiniam segmentavimui yra sukurtos specializuotos CNN architektūros, tokios kaip pilnai konvoliuciniai tinklai (FCN) (angl. *Fully Convolutional Networks*, *FCN*). Šios architektūros yra CNN modifikacijos, pritaikytos būtent segmentavimo užduočiai. Jos vietoje įprastinių CNN pabaigoje esančių pilnai sujungtų sąryšių sluoksnių (kurie įprastai naudojami klasifikavimui ir praranda erdvinę informaciją apie požymių vietas) naudoja konvoliucinius sluoksnius (pvz., 1×1 konvoliucijas), kurie išsaugo erdvinę struktūrą. Kadangi ankstesni sutelkimo (angl. *pooling*) sluoksniai sumažino požymių žemėlapių raišką, FCN pasitelkia raiškos didinimo (angl. *up-sampling*) metodus, kad palaipsniui atkurtų pradinę vaizdo raišką. Šie pakeitimai leidžia FCN sugeneruoti galutinį išvesties žemėlapį, kurio raiška atitinka įvesties vaizdą ir kuriame kiekvienas pikselis turi priskirtą klasę [7]. FCN veikimo principas kategorizuojant pikselius (žr. 3 pav.) [7 p. 1].



3 pav. FCN veikimo principas

1.3. Kodavimo-dekodavimo architektūros

Remiantis pamatinėmis FCN idėjomis, ypač tokiomis koncepcijomis kaip: raiškos mažinimo (angl. *downsampling*), siekiant užfiksuoti kontekstą, ir raiškos didinimo (angl. *upsampling*) su praleidimo jungtimis (angl. *skip connections*), siekiant atkurti detales, kodavimo-dekodavimo architektūra tapo vyraujančia paradigma semantinio segmentavimo užduotims, taip pat įskaitant pastatų kontūrų išskyrimą.

1.3.1. Kodavimo-dekodavimo paradigma

Kodavimo-dekodavimo tinklai susideda iš dviejų pagrindinių komponentų, sujungtų siauro praėjimo sluoksniu (angl. *bottleneck layer*):

- koduotuvas (angl. *encoder*): ši dalis paprastai primena standartinį klasifikavimo CNN (pvz., *VGG* [10], *ResNet* [11]). Ji laipsniškai mažina įvesties vaizdo erdvinę skiriamąją gebą naudojant sutelkimo arba žingsnines konvoliucijas (angl. *strided convolutions*), tuo pačiu

metu didindama požymių kanalų (angl. *feature channels*) skaičių. Koduotuvo tikslas yra išmokti vis abstraktesnius ir aukštesnio lygio semantinius požymius, užfiksuojant vaizdo kontekstą [12];

- siaurasis praėjimas (angl. *bottleneck*): tai paprastai yra sluoksnis (-iai), turintis (-ys) mažiausią erdvinę skiriamąją gebą ir didžiausią požymių dimensiją (angl. *feature dimensionality*). Jis sujungia koduotuvą ir dekoderį. Jis atspindi labiausiai suspaustą, aukšto lygio įvesties požymių vaizdą [13];
- dekoderis (angl. *decoder*): ši dalis paima siauro praėjimo sluoksnio požymius ir laipsniškai didina jų raišką (angl. *upsamples*), paprastai naudodamas transponuotas konvoliucijas (angl. *transposed convolutions*) ar kitus raiškos didinimo metodus. Didėjant erdvinei skiriamajai gebai, požymių kanalų skaičius paprastai mažėja. Dekoderio tikslas yra rekonstruoti pilnos raiškos segmentavimo žemėlapį, perkeliant koduotuvo išmoktus aukšto lygio semantinius požymius atgal į pikselių lygmens klasifikacijas. Šuolinės jungtys (angl. *sskip connections*), jungiančios koduotuvo ir dekoderio sluoksnius atitinkamose erdvinėse skiriamosiose gebose, yra beveik visuotinai naudojamos, kad būtų susigrąžinamos koduojant prarastos didelės raiškos detalės [12].

Nors ir efektyvi, ši paradigma neišvengiamai turi informacijos siaurąjį praėjimą tarp koduotuvo ir dekoderio. Koduotuvas turi suspausti įvestį į žemesnės dimensijos, neišvengiamai prarasdamas dalį smulkios erdvinės informacijos. Dekoderio gebėjimas sukurti tikslias, ryškias segmentacijas labai priklauso nuo mechanizmų, tokių kaip praleidimo jungtys, ar sudėtingų suliejimo strategijų, kad būtų atkurtos šios prarastos detalės. Todėl praleidimo jungčių projektavimas ir pati dekoderio struktūra yra kritinės inovacijų sritys šioje paradigmoje.

1.3.2. Pagrindinės architektūros

Buvo pasiūlyta keletas įtakingų kodavimo-dekodavimo architektūrų, kurių kiekviena pristato specifines inovacijas, siekiant pagerinti segmentavimo tikslumą.

U-Net

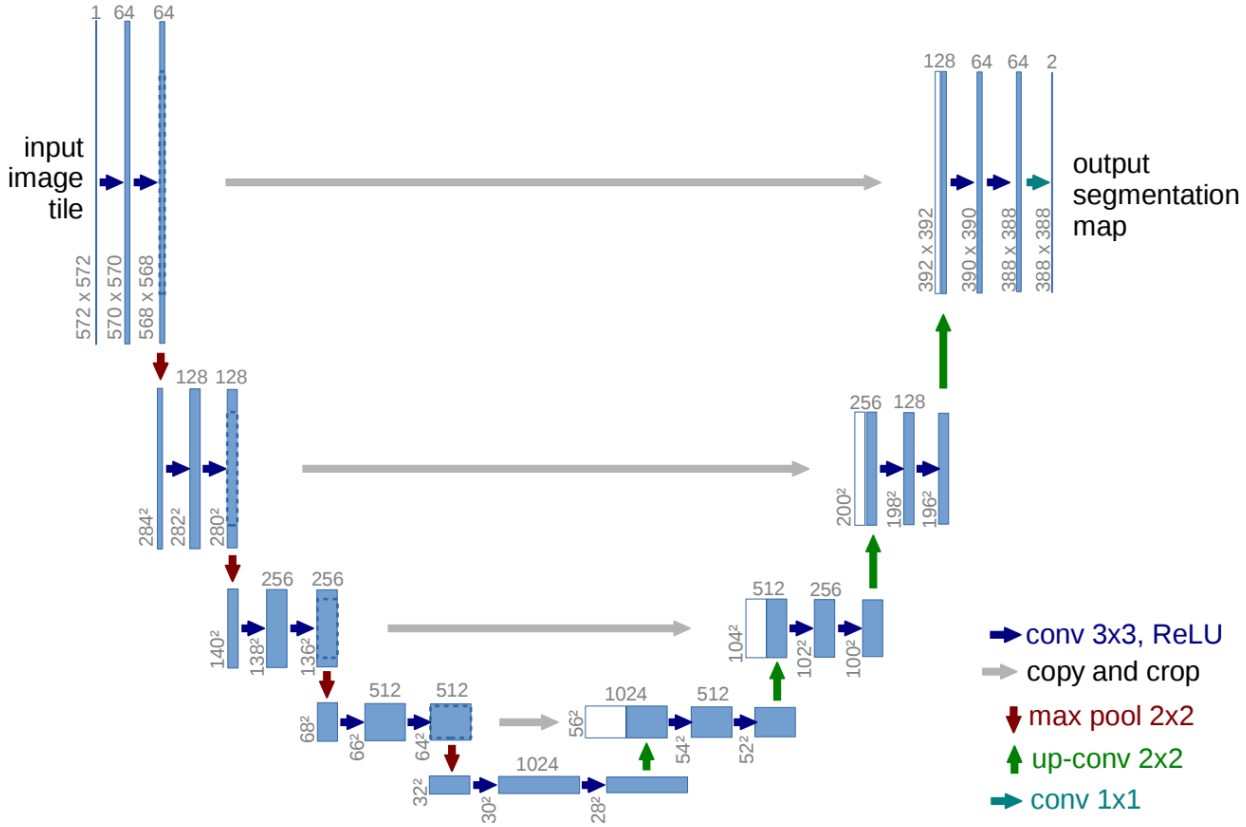
U-Net architektūra, kurią iš pradžių biomedicininių vaizdų segmentavimui sukūrė *Ronneberger* ir kt., [14] tapo itin populiari ir efektyvi įvairioms segmentavimo užduotims, įskaitant ir pastatų kontūrų išskyrimą [15]. Pagrindiniai jos bruožai apima:

- simetriška „U“ forma: simetriška architektūra, kurioje dekoderio kelias atspindi koduotuvo kelią;
- koduotuvas: susideda iš pasikartojančių blokų, kurių kiekvienas turi dvi 3×3 konvoliucijas be papildymo (angl. *unpadded convolutions*), po kurių seka *ReLU* aktyvacija, o tada 2×2 maksimalaus sutelkimo operacija kuri juda per 2 pikselius, taip perpus sumažinant raišką. Požymių kanalų skaičius dvigubėja kiekviename raiškos mažinimo žingsnyje;
- dekoderis: susideda iš pasikartojančių blokų, kurių kiekvienas prasideda raiškos didinimo žingsniu naudojant transponuotą konvoliuciją (angl. *transposed convolution*), kuri perpus sumažina kanalų skaičių, po kurio seka sujungimas (angl. *concatenation*) su atitinkamu požymių žemėlapiu iš koduotuvo kelio, o tada dvi 3×3 konvoliucijos su *ReLU*;
- praleidimo jungtys: sujungimu pagrįstos praleidimo jungtys (angl. *concatenation-based skip connections*) yra esminis bruožas, tiesiogiai perduodantis aukštos raiškos požymių žemėlapius

iš koduotuvo į dekoderį tame pačiame erdviniame lygmenyje. Tai leidžia dekoderiui sujungti tikslią lokalizacijos informaciją iš ankstyvųjų koduotuvo sluoksnių su kontekstine informacija, sklindančia per siaurąjį praėjimą, o tai yra labai svarbu generuojant detalias segmentacijas;

- galutinis sluoksnis: 1×1 konvoliucija susieja galutinį požymių vektorių su norimu išvesties klasių skaičiumi.

U-Net sėkmė dažnai siejama su efektyviu praleidimo jungčių naudojimu ir jos gebėjimu gerai veikti net su santykinai mažais duomenų rinkiniais, o tai yra įprasta specializuotose srityse, pavyzdžiui, medicininėje ar nuotolinio stebėjimo (angl. *remote sensing*) analizėje. Šis efektyvus duomenų išnaudojimas yra ypač vertingas pastatų kontūrų segmentavimui, kur didelius, įvairius ir tiksliai anotuotus duomenų rinkinius gali būti sudėtinga gauti. *U-Net* architektūros iliustracija pateikta 4 pav. [14 p. 2].



4 pav. *U-Net* architektūra

U-Net++

Zhou ir kt. pasiūlyta *U-Net++* [16] architektūra išplečia pradinę *U-Net* architektūrą, pertvarkydama praleidimo kelius (angl. *skip pathways*), kad sumažintų semantinį atotrūkį (angl. *semantic gap*) tarp koduotuvo ir dekoderio požymių.

Lizdiniai (angl. *nested*) ir tankūs praleidimo keliai: vietoje tiesioginių jungčių, *U-Net++* naudoja tankiai sujungtus konvoliucinius blokus praleidimo keliuose. Tai leidžia laipsniškai sulieti požymius iš skirtingų lygių, siekiant geresnio semantinio panašumo tarp koduotuvo ir dekoderio žemėlapių prieš galutinį suliejimą, o tai yra agresyvesnis metodas nei standartinės *U-Net* praleidimo jungtys.

Gilioji mokymo priežiūra (angl. *deep supervision*): segmentavimo išvestys generuojamos keliuose tarpiniuose dekoderio etapuose, o jų nuostolių funkcijos (angl. *loss functions*) yra sujungiamos. Tai padeda mokymo gradientams ir leidžia lanksčiai vykdyti išvedimą (angl. *inference*) – skaičiuoti išvesčių vidurkį tikslumui arba naudoti seklesnes išvestis greičiui (modelio genėjimas (angl. *model pruning*).

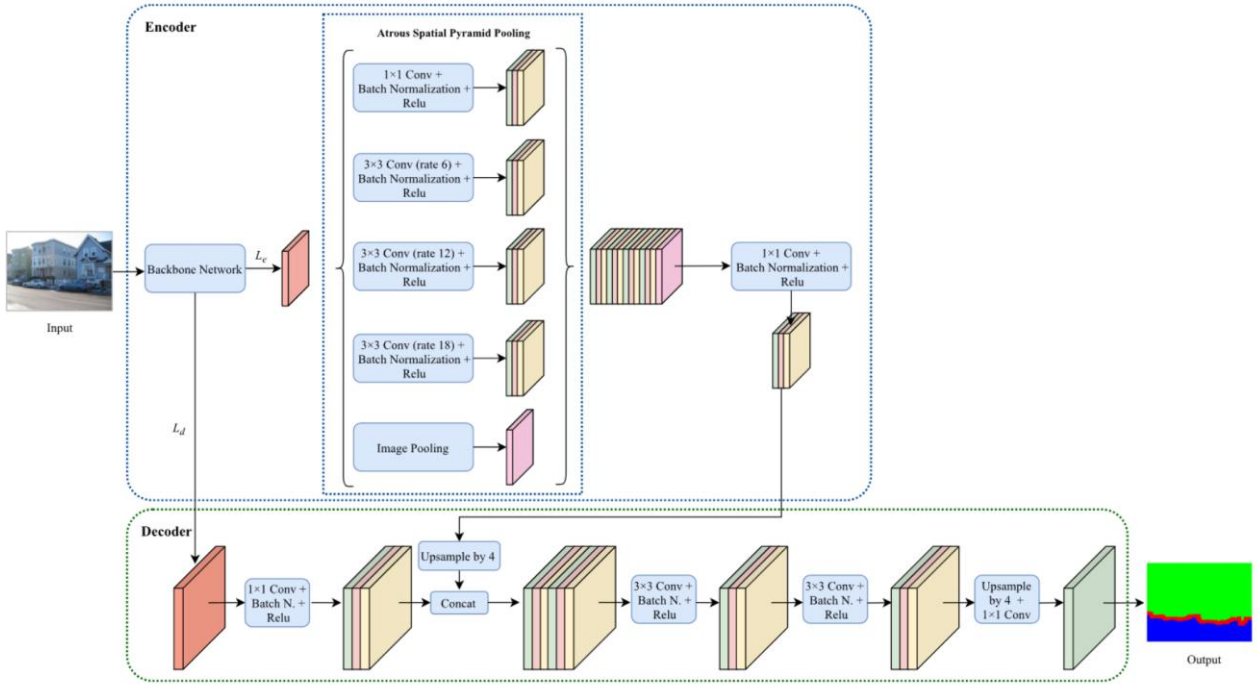
U-Net++ parodė reikšmingus susikirtimo padalinto iš sąjungos (angl. *Intersection over Union, IOU*) pagerėjimus, palyginti su standartiniu *U-Net*, medicininių vaizdų eksperimentuose.

***DeepLab* šeima (*DeepLabV3* ir *V3+*)**

DeepLab modelių šeima pristatė esmines inovacijas, skirtas daugiamačiam kontekstui (angl. *multi-scale context*) užfiksuoti ir kontūrams patikslinti bendroje kodavimo-dekodavimo struktūroje. Esminiai bruožai:

- praplėstoji konvoliucija (angl. *Atrous (Dilated) Convolution*): šis metodas įterpia nulius („skyles“) į filtro svorius, padidindamas priėmimo lauką (angl. *receptive field*) be papildomų parametrų ar skaičiavimų. Reguliuojant praplėtimo koeficientą (angl. *dilation rate*), leidžiama tiesiogiai valdyti požymių žemėlapio skiriamąją gebą ir daugiamačio mastelio konteksto tyrimą (angl. *multi-scale context probing*) [17];
- praplėstasis erdvinis piramidinis suliejimas (angl. *Atrous Spatial Pyramid Pooling*, *ASPP*): esminis nuo *DeepLabV2* versijos. *ASPP* taiko lygiagrečias atrozines konvoliucijas su kintančiais koeficientais (kartu su 1×1 konvoliucija, visuotiniu sutelkimu) aukščiausio lygio koduotuvo požymiams. Sujungtos išvestys yra patikslinamos galutine 1×1 konvoliucija, kad patikimai būtų užfiksuota daugiamačio mastelio informacija, taip pasiūlant aiškią konteksto agregavimo strategiją, skirtingą nuo *U-Net* praleidimo jungčių [17];
- *DeepLabV3+*: ši versija prideda dekoderio modulį, kad patikslintų kontūrų segmentavimą, sprendžiant ankstesnių, tik *ASPP* naudojančių, išvesčių trūkumus. Dekoderis sulieja turtingus semantinius požymius iš *ASPP* su aukštos raiškos žemo lygio požymiais iš koduotuvo bazinio modelio (fundamento) (angl. *backbone*). Šis suliejimas paprastai apima raiškos didinimą, sujungimą su sumažinto kanalų skaičiaus žemo lygio požymiais, tikslinimo konvoliucijas bei galutinį raiškos didinimą, kad būtų sugeneruota segmentacija [18];
- efektyvumas: *DeepLabV3+* naudoja gyliu atskiriamas konvoliucijas (angl. *depthwise separable convolutions*) savo *ASPP* ir dekoderio moduliuose, ženkliai sumažinant skaičiavimus ir parametrų skaičių, išlaikant tikslumą. Efektyvūs baziniai modeliai (angl. *backbones*), pavyzdžiui, modifikuotas *Xception*, naudoja šias konvoliucijas [18].

DeepLabV3+ architektūros schema naudojanti bazinio modelio (angl. *backbone*) tinklą pateikta 5 pav. [17 p. 7].



5 pav. *DeepLabV3+* architektūra su bazinio modelio tinklu

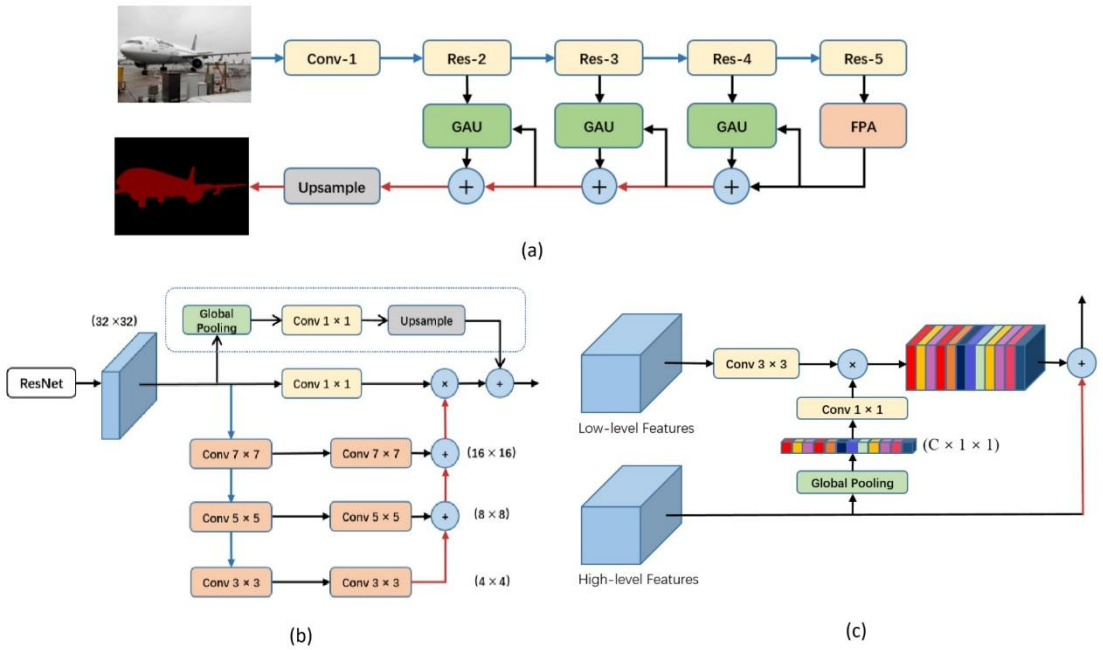
Piramidinio dėmesio tinklas (angl. *Pyramid Attention Network, PAN*)

PAN, pasiūlytas *Li* ir kt. [19], integruoja dėmesio mechanizmus (angl. *attention mechanisms*) su piramidinėmis struktūromis, kad gebėtų efektyviai užfiksuoti globalų kontekstą.

Požymių piramidės dėmesio (angl. *Feature Pyramid Attention, FPA*) modulis: taikomas aukšto lygio koduotuvo požymiams, *FPA* užfiksuoja daugiamačio mastelio kontekstą naudodamas erdvinę piramidę su kintančiais branduoliais. Jis skiriasi nuo *ASPP* sujungimo: pikselių lygmenyje padaugina išmoktą dėmesį, gautą iš šių piramidės požymių, su pradiniais požymiais.

Globalaus dėmesio raiškos didinimo (angl. *Global Attention Upsample, GAU*) modulis: šis dekoderio modulis priima aukšto lygio (*FPA* išvesties) bei žemo lygio koduotuvo požymius. Naudojant suliejimą jis išgauna globalų kontekstą iš aukšto lygio požymių, tada naudoja šį kontekstą sverti sumažinto kanalų skaičiaus žemo lygio požymius, nukreipdamas erdvinio detalumo parinkimą. Šie pasverti žemo lygio požymiai yra pridedami prie aukšto lygio požymių prieš raiškos didinimo veiksmus.

PAN rodo postūmį link sudėtingesnių, dėmesiu pagrįstų suliejimo mechanizmų (angl. *attention-based fusion mechanisms*) įtraukimo į kodavimo-dekodavimo struktūrą, derinant piramidinių struktūrų privalumus. Specializuotų modulių, tokių kaip *FPA* ir *GAU*, projektavimas atspindi tendenciją kurti aiškesnius bei žinoma potencialiai galingesnius komponentus konteksto agregavimui ir požymių suliejimui, o ne pasikliaujant vien standartinėmis konvoliucijomis ir paprastomis praleidimo jungtimis. *PAN* architektūra pateikta 6 pav. [19 p. 3, 4, 5].



6 pav. *PAN*. Visa architektūra (a), *FPA* modulis (b), *GAU* modulis (c)

1.3.3. Kitos paminėtinos architektūros

Be prieš tai minėtų pagrindinių architektūrų, prie semantinio segmentavimo srities prisidėjo ir keletas kitų reikšmingų modelių, kurie taip pat remiasi kodavimo-dekodavimo paradigma arba pristato naujas koncepcijas.

MAnet (daugiamačio mastelio dėmesio tinklas, angl. *Multi-Scale Attention Network*) [20]: sutelktas į požymių tikslinimą, įtraukiant dėmesio mechanizmus (angl. *attention mechanisms*), kurie veikia skirtingais masteliais dekoderyje – tai leidžia modeliui adaptyviu būdu pabrėžti svarbias erdvines sritis bei požymių kanalus.

LinkNet [21]: teikia pirmenybę efektyvumui, naudodamas tiesioginio perdavimo jungtis (angl. *passthrough connections*), kurios sujungia koduotuvo blokus su atitinkamais dekoderio blokais. Ši struktūra pagreitina skaičiavimus, kadangi išsaugo aukštos raiškos erdvinius duomenis iš koduotuvo.

FPN (požymių piramidės tinklas, angl. *Feature Pyramid Network*) [22]: nors iš pradžių sukurtas objektų aptikimui, *FPN* struktūra plačiai pritaikoma segmentavimui. Jis sukuria požymių piramidę dekoderyje, naudodamas šonines jungtis (angl. *lateral connections*) su atitinkamais koduotuvo sluoksniais, – tai leidžia atlikti daugiamačio mastelio prognozes.

PSPNet (piramidinis scenos analizės tinklas, angl. *Pyramid Scene Parsing Network*) [23]: pristato piramidinio sutelkimo modulį (angl. *pyramid pooling module*), taikomą galutiniam koduotuvo požymių žemėlapiui. Šis modulis užfiksuoja konteksto informaciją įvairiais masteliais, kuri vėliau sujungiama ir jos raiška padidinama galutinei segmentavimo prognozei sukurti.

UPerNet (vieningas suvokimo analizės tinklas, angl. *Unified Perceptual Parsing Network*) [24]: siekiama sukurti vieningą sistemą scenos supratimui, integruojant daugiamačio mastelio konteksto agregavimo privalumus tiek iš *FPN* [22], tiek iš piramidinio suliejimo modulių.

SegFormer [4]: naudoja hierarchinį vaizdų transformatorių (angl. *Vision Transformer, ViT*) kaip koduotuvą – tai pašalina pozicinių kodavimų (angl. *positional encodings*) poreikį. Jį suporuoja su paprastu visų sluoksnių *MLP* (daugiasluoksnis perceptronas, angl. *Multi-Layer Perceptron*) dekoderiu efektyviam segmentavimui.

DPT (tankių prognozių transformeris, angl. *Dense Prediction Transformer*) [25]: naudoja standartinį *ViT* bazinį modelį (angl. *backbone*) kaip koduotuvą ir konvoliucinį dekoderį, kuris laipsniškai sujungia požymius iš skirtingų transformerio etapų, kad būtų rekonstruojamos tankios, aukštos raiškos prognozės.

1.3.4. Bazinių modelių tinklų (angl. *backbone networks*) vaidmuo

Šių segmentavimo architektūrų koduotuvas dažnai įgyvendinamas naudojant iš anksto apmokytą klasifikavimo tinklą, kuris vadinasi baziniu modeliu arba fundamentu (angl. *backbone*). Bazinio modelio pasirinkimas ženkliai veikia modelio greitį kompiuterinių resursų reikalavimus bei žinoma tikslumą.

Klasikiniai baziniai modeliai (angl. *backbones*)

Pamatiniai ir vis dar plačiai naudojami baziniai modeliai (angl. *backbones*) apima *VGG* [10] ir ypač *ResNet* (liekanų neuroninis tinklas, angl. *Residual Network*) [11]. *ResNet* pristatė liekamąsias jungtis (angl. *residual connections*), kurios leido apmokyti daug gilesnius tinklus, palengvinant optimizavimą ir sumažinant nykstančių gradientų (angl. *vanishing gradients*) problemą. Vėlesni sprendimai, tokie kaip *ResNeXt* (naudojantis grupuotas konvoliucijas (angl. *grouped convolutions*) [26] ir *DenseNet* (sujungiantis kiekvieną sluoksnį su visais kitais sluoksniais tiesioginio sklidimo (angl. *feed-forward*) būdu) [27], taip pat yra naudojami.

Šiuolaikiniai baziniai modeliai

Naujesni tyrimai yra sutelkti į bazinių modelių kūrimą, kurie apima idėjas iš transformatorių arba praplečia CNN projektavimo ribas.

ConvNeXt: pasiūlytas *Liu* ir kt. [3], *ConvNeXt* modernizuoja standartinius CNN, palaipsniui įtraukdamas projektavimo elementus iš vaizdų transformerių (angl. *vision transformers*) (pvz., *Swin*). Tai apima etapų skaičiavimo santykio pakeitimus, lopų sudarymo kamieno (angl. *patchify stem*) naudojimą, gyliu atskiriamų konvoliucijų (angl. *depthwise separable convolutions*) pritaikymą, invertuoto siaurojo praėjimo (angl. *inverted bottleneck*) struktūros naudojimą, didesnių branduolių (pvz., 7×7) naudojimą, *GELU* aktyvacijos naudojimą vietoje *ReLU*, aktyvacijos ir normalizavimo sluoksnių skaičiaus mažinimą, *BatchNorm* pakeitimą į *LayerNorm*, ir atskirų raiškos mažinimo sluoksnių naudojimą. *ConvNeXt* pasiekia transformerių lygio našumą, išlaikydamas grynų konvoliucinių tinklų paprastumą ir efektyvumą. Vėliau buvo pristatyta *ConvNeXt V2* [28] versija , kuri dar labiau pagerino našumą, integruodama savarankiškai prižiūrimo mokymosi metodus, tokius kaip kaukių autoenkoderiai (angl. *masked autoencoders*), ir naują globalaus atsako normalizavimo (angl. *Global Response Normalization, GRN*) sluoksnį, siekiant pagerinti ypatybių konkurenciją tarp kanalų ir efektyviau pritaikyti savarankiško mokymosi metodus konvoliuciniams tinklams.

MaxViT: hibridinė architektūra, pasiūlyta *Tu* ir kt. [29], kuri sujungia konvoliuciją (naudojant *MBConv* blokus – *MobileNetV2* architektūroje pristatytus efektyvius konvoliucinius modulius,

pasižyminčius apversto siaurojo praėjimo (angl. *inverted bottleneck*) struktūra ir našumui optimizuotomis gylio krypties atskiriamosios konvoliucijomis (angl. *depthwise separable convolutions*) su nauju daugiaašio savidėmesio (angl. *Multi-Axis Self-Attention*, *Max-SA*) moduliu. *Max-SA* sujungia blokuotą vietinį dėmesį (angl. *blocked local attention*) ir praplėstą globalų dėmesį (angl. *dilated global attention*), kad pasiektų ir vietinę, ir globalią erdvinę sąveiką su linijiniu sudėtingumu. Jo hierarchinė struktūra leidžia naudoti globalų dėmesį net ir ankstyvuosiuose sluoksniuose.

CoAtNet: kitas hibridinis metodas, pasiūlytas *Dai* ir kt. [2], kuris sujungia konvoliuciją ir dėmesį. Jis suvienija gylio konvoliuciją (angl. *depthwise convolution*) ir savidėmesį (angl. *self-attention*) per santykinį dėmesį (angl. *relative attention*). Šis metodas siūlo strategiją vertikaliai sudėti konvoliucinius ir transformerių blokus, kad būtų subalansuotas apibendrinimas (angl. *generalization*) ir modelio talpa (angl. *model capacity*) skirtinguose duomenų masteliuose.

Tendencija link įvairesnių požymių ištraukėjų (angl. *feature extractors*)

Apart šių konkrečių pavyzdžių, dabartiniuose tyrimuose ryški tendencija yra derinti nusistovėjusias segmentavimo architektūras (pvz., *U-Net* [14]) su dar platesniu šiuolaikinių klasifikavimo tinklų spektru. Jie yra naudojami kaip požymius ištraukantys baziniai modeliai (angl. *feature-extracting backbones*). Šis metodas leidžia pasinaudoti galingais, bendrosios paskirties požymių ištraukėjais (angl. *feature extractors*), pritaikant architektūras, kurios originaliai buvo sukurtos klasifikavimui [30]. Šiai užduočiai naudojami baziniai modeliai (angl. *backbones*) yra iš įvairių architektūrinių šeimų, įskaitant:

- pažangūs CNN: *RegNet* [31], *ResNeSt* [32];
- vaizdų transformeriai: *SwinV2* [33], *PVT v2* [34] – jų architektūriniai principai, pagrįsti savidėmesiu (angl. *self-attention*);
- Dėmesio įkvėpti arba hibridiniai CNN (angl. *attention-inspired or hybrid CNNs*): *FocalNet* [1], *BotNet* [35], *LambdaResNet* [36].

Šių bazinių modelių (angl. *backbones*) našumui įtakos turi ir dideli duomenų rinkiniai, naudoti jų išankstiniam apmokymui (angl. *pre-training*) (pvz., *ImageNet*[1]), o tai prideda dar vieną aspektą, į kurį reikia atsižvelgti renkantis komponentus pastatų kontūrų segmentavimui. Nuolatinis bazinių modelių architektūrų tobulinimas atspindi nenutrūkstamas paieškas galingesnių ir efektyvesnių požymių ištraukėjų, tinkamų sudėtingoms užduotims, pavyzdžiui, aukštos raiškos pastatų kontūrų segmentavimui. Tendencija įtraukti dėmesio mechanizmus (angl. *attention mechanisms*) ar transformerių projektavimo principus, net į pagrinde konvoliucinius bazinius modelius, pabrėžia globalaus konteksto (angl. *global context*) svarbą.

1.4. Dėmesio mechanizmai ir transformeriai segmentavimo užduotyse

Nors kodavimo-dekodavimo architektūros su CNN baziniais modeliais tapo standartu, grynai konvoliucinių metodų būdingi trūkumai fiksuojant tolimas priklausomybes (angl. *long-range dependencies*) paskatino dėmesio mechanizmų integravimą ir vėliau – transformerių (angl. *transformer*) architektūrų pritaikymą, kurios iš pradžių buvo sukurtos natūraliosios kalbos apdorojimui (angl. *natural language processing*, *NLP*).

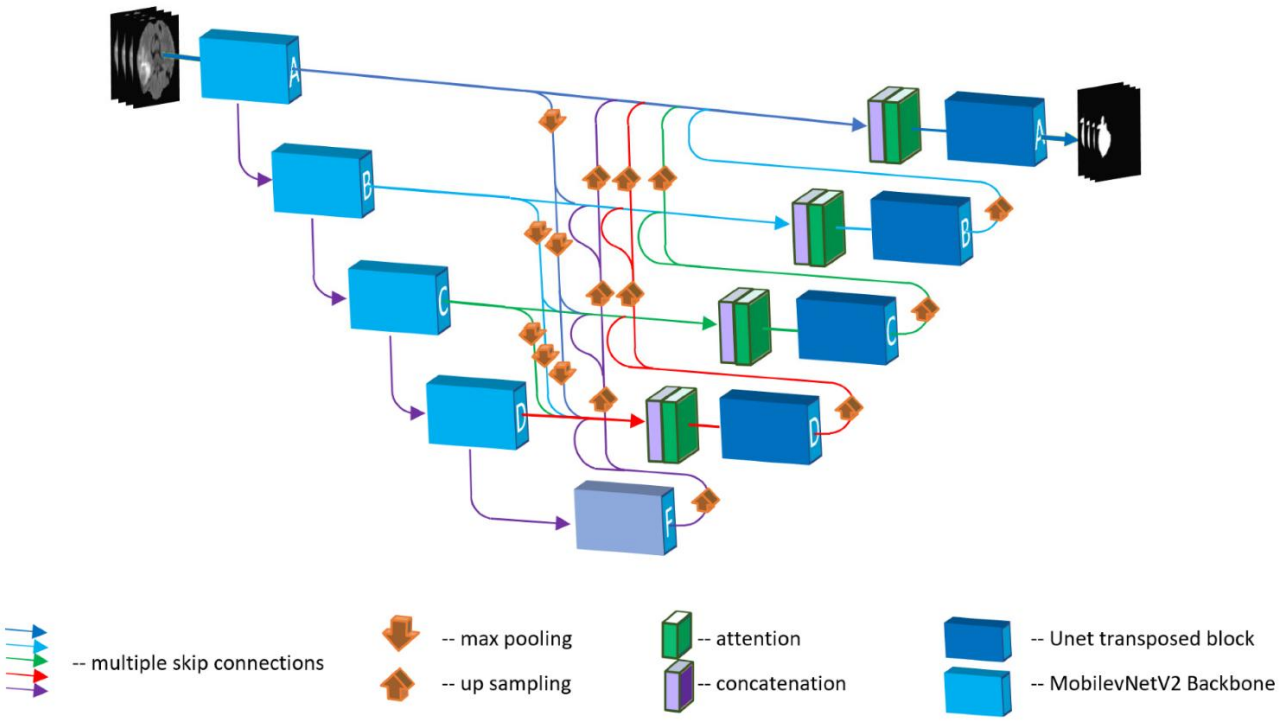
[1] https://www.image-net.org/

1.4.1. Dėmesio mechanizmai kompiuterinėje regoje (angl. *computer vision*)

Dėmesio mechanizmai (angl. *attention mechanisms*) imituoja žmogaus regos sistemos gebėjimą sutelkti dėmesį į svarbias vaizdo dalis, ignoruojant nesvarbias detales [37]. Giliajame mokymesi, jie veikia kaip dinaminiai svėrimo procesai, pagrįsti įvesties požymiais – tai leidžia modeliams adaptyviai pabrėžti svarbiausią informaciją. Kompiuterinėje regoje buvo pritaikyti keli dėmesio tipai:

- kanalų dėmesys (angl. *channel attention*): sutelkia dėmesį į tai, kas yra svarbu. Tai atlieka adaptyviai perkalibruodamas požymių atsakus kanalų lygmenyje. Suspaudimo ir sužadinimo (angl. *Squeeze-and-Excitation*, *SE*) blokas yra to pavyzdys, išmokstantis globalias kanalų tarpusavio priklausomybes ir priskiriantis svorius kiekvienam kanalui [38];
- erdvinis dėmesys (angl. *spatial attention*): sutelkia dėmesį į tai, kur yra svarbu. Tai atlieka generuodamas erdvinio dėmesio žemėlapį (angl. *spatial attention map*), kuris paryškina svarbias sritis požymių žemėlapyje [39];
- šakų dėmesys (angl. *branch attention*): sutelkia dėmesį į tai, kuri šaka ar kelias tinkle yra svarbesnė (-is) [37];
- hibridinis dėmesys (angl. *hybrid attention*): derina skirtingus dėmesio tipus, pavyzdžiui, kanalų ir erdvinį dėmesį, dažnai nuosekliai arba lygiagrečiai, kad būtų užfiksuota papildoma informacija [40].

Dėmesio mechanizmai gali būti integruoti į įvairias CNN architektūras, siekiant pagerinti jų požymių vaizdavimo (angl. *feature representation*) gebėjimus. Pavyzdžiui, *Attention U-Net* [41] įtraukia dėmesio vartus į *U-Net* [42] praleidimo jungtis, kad slopintų nesvarbias sritis koduotuvo požymiuose prieš suliejant juos su dekoderio požymiais. Šios architektūros pavyzdys pateiktas 7 pav. [43 p. 7].



7 pav. *Attention U-Net* koncepcijos įgyvendinimo pavyzdys

PAN naudoja dėmesį tiek savo *FPA*, tiek *GAU* moduliuose [19]. Specializuoti dėmesio moduliai taip pat buvo sukurti specifiniams nuotolinio stebėjimo (angl. *remote sensing*) iššūkiams, pavyzdžiui,

daugelio šakų kanalų dėmesys (angl. *multi-branch channel attention, MCAM*), daugiamačio mastelio dėmesio tinklai (angl. *multi-scale attention networks, MSANet*) [39], ir dėmesio žemėlapių suliejimo mechanizmai (angl. *attention map merging mechanisms, AMMM*) [43]. Šie mechanizmai leidžia modeliams išmokti labiau skiriančius požymius sutelkiant skaičiavimo išteklius į informatyviausias įvesties ir požymių žemėlapių dalis.

1.4.2. Vaizdų transformeris (angl. *Vision Transformer, ViT*) ir jo pritaikymas

Transformerio architektūros, visiškai pagrįstos savidėmesio (angl. *self-attention*) mechanizmais, pritaikytos kompiuterinės regos užduotyse. To puikus pavyzdys būtų vaizdų transformeris, kurį pasiūlė *Dosovitskiy* ir kt. [44]. *ViT* vaizdą traktuoja kaip lopų apdorojimo seką:

- lopų sudarymas ir įterpimas (angl. *patching and embedding*): įvesties vaizdas padalijamas į fiksuoto dydžio, nepersidengiančius lopus (pvz., 16×16 pikselių). Kiekvienas lopas yra ištiesinamas ir tiesiškai projektuojamas į įterpimo vektorių (angl. *embedding vector*);
- poziciniai įterpimai (angl. *position embeddings*): kadangi pati transformerių architektūra yra nepriklausoma nuo permutacijų (angl. *permutation-invariant*), prie lopų įterpimų pridedami išmokstami poziciniai įterpimai (angl. *position embeddings*), kad būtų išsaugota erdvinės vietos informacija;
- transformerio koduotuvas (angl. *transformer encoder*): įterptų lopų seka (kartu su poziciniais įterpimais) patenka į standartinį transformerio koduotuvą, sudarytą iš kelių daugiaagalvio savidėmesio (angl. *Multi-Head Self-Attention, MHSA*), *MLP* blokų, sluoksnių normalizacijos ir liekamųjų jungčių sluoksnių. *MHSA* mechanizmas palengvina globalias sąveikas tarp visų lopų, fiksuodamas tolimas priklausomybes visame vaizde;
- klasifikavimo galvutė (angl. *classification head*): vaizdų klasifikavimui, prie sekos pradžios pridedamas specialus išmokstamas klasės žetonas. Atitinkamas išvesties įterpimas, po transformerio koduotuvo, perduodamas į *MLP* galvutę klasifikavimui.

Nors *ViT* pasiekė aukštus rezultatus vaizdų klasifikavimo srityje, ypač kai buvo iš anksto apmokytas naudojant labai didelius duomenų rinkinius (pvz., *JFT-300M* [2]), jo tiesioginis taikymas tankių prognozių užduotims, pavyzdžiui, semantiniam segmentavimui, susidūrė su iššūkiais [3]:

- induktyviųjų poslinkių (angl. *inductive biases*) trūkumas: priešingai nei CNN, *ViT* neturi įgimtų induktyviųjų poslinkių (pvz., lokališkumo (angl. *locality*), transliacijos ekvivariantiškumo (angl. *translation equivariance*). Dėl to *ViT* dažnai reikalauja daugiau duomenų ir gali prasčiau veikti su mažais duomenų rinkiniais nei CNN;
- sudėtingumas pakeltas kvadratu (angl. *quadratic complexity*): savidėmesio mechanizmo skaičiavimo sudėtingumas yra pakeltas kvadratu lopų skaičiaus atžvilgiu. Dėl to jis tampa labai brangus didelės raiškos vaizdams (įprastiems nuotoliniame stebėjime), kurie generuoja ilgas lopų sekas;
- fiksuota požymių raiška (angl. *Fixed-Resolution Feature Maps*): standartinis *ViT* generuoja požymių įterpimus tik vienoje, žemoje raiškoje (lopų lygmenyje). Tai apsunkina modelio taikymą užduotims, reikalaujančioms aukštos raiškos išvesčių (pvz., segmentavimui).

Ankstyvieji *ViT* segmentavimo modeliai (pvz., *SETR* [45], *Segmenter* [46]) paprastai naudojo *ViT* koduotuvą kaip galingą požymių ištraukėją, po kurio sekė dekoderis, skirtas padidinti požymių raišką ir generuoti segmentavimo žemėlapį. Nors šie metodai demonstravo transformerių bazinių modelių

potencialą, jie išryškino poreikį kurti architektūras, labiau pritaikytas tankioms prognozėms. Pagrindinė motyvacija tyrinėti transformerius segmentavime buvo jų savidėmesio mechanizmo gebėjimas modeliuoti tolimas priklausomybes, taip sprendžiant standartiniams CNN būdingą riboto efektyvaus priėmimo lauko (angl. *effective receptive field*) problemą.

1.4.3. Hierarchiniai transformeriai

Siekiant pašalinti pradinio *ViT* trūkumus tankių prognozių užduotims ir pagerinti efektyvumą, buvo sukurtos hierarchinės transformerių architektūros, ypač *Swin* transformeris [47].

Hierarchinė struktūra: *Swin* įveda į CNN panašią hierarchiją: pradedant nuo mažų lopų, jie palaipsniui sujungiami gilesniuose sluoksniuose, mažinant raišką ir didinant kanalų skaičių. Tokiu būdu sukuriami daugiamačiai požymių žemėlapiai, tinkami standartinėms segmentavimo sistemoms (pvz., *FPN* [22], *U-Net* [14] dekoderiams). *SwinV2* [33] ir *PVT* [34] architektūros (paminėtos 1.3.4 skyriuje) iliustruoja šį efektyvų bazinių modelių (angl. *backbones*) metodą.

Paslinkto lango savidėmesys (angl. *Shifted Window Self-Attention, SW-MSA*): vietoje globalaus savidėmesio naudojamas dėmesys vietiniuose languose (angl. *Window-based Self-Attention, W-MSA*) (t. y. dėmesys skaičiuojamas tik lango viduje), sumažinant sudėtingumą iki tiesinio (vaizdo dydžio atžvilgiu) ir užtikrinant mastelio keitimą didelės raiškos vaizdams. Informacijos srautui tarp langų užtikrinti, *W-MSA* kaitaliojamas su paslinktų langų versija (*SW-MSA*) gretimuose blokuose, sukuriant jungtis tarp langų (angl. *cross-window connections*).

Swin transformeris pasiekė pažangius rezultatus daugelyje užduočių (įskaitant segmentavimą), tapdamas galingu universaliu pagrindu (angl. *backbone*). Dėl tiesinio sudėtingumo ir hierarchijos jis ypač tinka dideliems nuotolinio stebėjimo vaizdams.

Swin ir panašių architektūrų (pvz., *PVT* [34]) kūrimas atskleidžia transformerių projektavimo kompromisą: efektyvumui aukštos raiškos užduotyse dažnai aukojama globali savidėmesio aprėptis (ribojant ją iki vietinių langų), kas gali sumažinti globalaus modeliavimo pajėgumą lyginant su standartiniu *ViT*. Mechanizmai kaip paslinkti langai ir tinklelio dėmesys (angl. *grid attention*) *MaxViT* modelyje [29] yra inžineriniai sprendimai bandantys balansuoti šį kompromisą, derinant efektyvumą su platesniu informacijos sklidimu.

1.4.4. Hibridiniai CNN ir transformerio metodai

Įvertinus viena kitą papildančius CNN ir transformerių privalumus, buvo pasiūlyta daugybė hibridinių architektūrų, kurios aiškiai derina abiejų paradigmų modulius. Tikslas paprastai yra pasinaudoti stipriais CNN induktyviaisiais polinkiais (angl. *inductive biases*) ir efektyvumu išgaunant vietinius požymius, ypač ankstyvuosiuose sluoksniuose, ir tuo pat metu panaudoti transformerių pajėgumą globaliam kontekstui ir tolimoms priklausomybėms modeliuoti, dažnai gilesniuose sluoksniuose [48]. Toliau apžvelgsime įprastos strategijas.

CNN koduotuvas sujungtas su transformerio dekoderiu: naudojant CNN bazinį modelį daugiamačio mastelio požymiams išgauti ir perduodant juos į transformerio dekoderį, kad būtų atliekami konteksto agregavimas (angl. *context aggregation*) bei prognozavimas.

Transformerio koduotuvas sujungtas su CNN dekoderiu: naudojant transformerį (pvz., *ViT* [29] ar *Swin* [47]) kaip pagrindinį požymių ištraukėją ir naudojant CNN baziniu modeliu sukurtą dekoderį

(pvz., *U-Net* [42] stiliaus) raiškos didinimui ir tikslinimui. *TransUNet* [49] yra paminėtinas pavyzdys, perduodantis lopų įterpimus iš CNN į transformerio koduotuvą prieš naudojant kaskadinį raiškos didinimą (angl. *cascaded upsampler*) su praleidimo jungtimis.

Persipinantys blokai (angl. *Interleaved Blocks*): projektuojant architektūras, kuriose konvoliuciniai blokai ir transformerio arba dėmesio blokai yra persipynę įvairiuose tinklo etapuose. *CoAtNet* [2] ir *MaxViT* [29] priklauso šiai kategorijai, kruopščiai integruojant *MBConv* blokus su santykiniu dėmesiu arba daugiaašio dėmesio moduliais. Baziniai modeliai, pagrįsti šia hibridine filosofija, pavyzdžiui, anksčiau minėti *MaxViT* variantai, naudojami kaip požymių ištraukėjai segmentavimo konvejeriuose.

Suliejimo mechanizmai (angl. *Fusion Mechanisms*): naudojant specifinius mechanizmus, pavyzdžiui, kryžminį dėmesį (angl. *cross-attention*), arba specialius suliejimo modulius, kad būtų efektyviai integruoti požymiai iš CNN ir transformerio komponentų.

Pavyzdžiai, tokie kaip *BEFUnet* [50], *DDTransUNet* [51] ir *ICTANet* [48], yra specialiai skirti medicininių arba nuotolinio stebėjimo vaizdų segmentavimui, dažnai naudojant dviejų šakų koduotuvas (angl. *dual-branch encoders*) (vieną CNN, vieną transformerį), kad būtų aiškiai užfiksuota tiek vietinė, tiek globali informacija. Šių hibridinių metodų sėkmė ir įvairovė rodo, kad nusistovėjusių CNN privalumų derinimas su transformerių globalaus modeliavimo galimybėmis suteikia gana pragmatišką ir efektyvų kelią aukštam našumui sudėtingose segmentavimo užduotyse, pavyzdžiui, pastatų kontūrų išskyrimo (angl. *building footprint extraction*), potencialiai reikalaujant mažiau duomenų ar skaičiavimo išteklių nei gryni transformerio metodai, bei tuo pačiu įveikiant priėmimo lauko (angl. *receptive field*) apribojimus, būdingus gryniems CNN.

1.5. Pažangios segmentavimo paradigmos

Be pagrindinių architektūrų evoliucijos, tokių kaip kodavimo-dekodavimo ir transformeriai, naujausi tyrimai nagrinėjo naujas segmentavimo paradigmas ir galingus metodus, skirtus vaizdavimų mokymuisi (angl. *representation learning*) ir modelių tobulinimui.

1.5.1. Kaukių klasifikavimo metodai

Svarbus pasistūmėjimas segmentavime yra perėjimas nuo kiekvieno pikselio klasifikavimo atskirai prie dvejetainių kaukių rinkinio prognozavimo, kurių kiekviena susieta su viena klase. Šia kaukių klasifikavimo (angl. *mask classification*) paradigma siekiama segmentavimą traktuoti labiau kaip objektų aptikimą (angl. *object detection*), sutelkiant dėmesį į segmentų lygio prognozes.

MaskFormer

Pristatytas *Cheng* ir kt. [52], *MaskFormer* pritaiko *DETR* [53] objektų aptikimo sistemą segmentavimui. Jis naudoja transformerio dekoderį apdoroti išmokstamas užklausas (angl. *learnable queries*) (objektų užklausų). Kiekviena užklausa sąveikauja su vaizdo požymiais (iš bazinio modelio pvz., *ResNet* [11] ar *Swin* transformerio [47], ir pikselių dekoderio) per kryžminį dėmesį (angl. *cross-attention*) ir savidėmesį. Kiekvienos užklausos dekoderio išvestis pateikiama į dvi galvutes: klasifikavimo galvutę, prognozuojančią klasės etiketę, ir kaukės galvutę, prognozuojančią dvejetainės kaukės įterpimą (angl. *binary mask embedding*). Galutinės kaukės generuojamos derinant šiuos kaukių įterpimus su pikselių lygmens įterpimais iš pikselių dekoderio. *MaskFormer* parodė šio

metodo potencialą suvienyti semantinį ir pavyzdžių (angl. *instance*) segmentavimą, tačiau iš pradžių pasižymėjo lėtesne konvergencija, palyginti su pikselių lygmens metodais.

Mask2Former

Tiesiogiai remdamiesi *MaskFormer, Cheng* ir kt. [54] pasiūlė *Mask2Former*, kuris pasiekė aukštus rezultatus semantinio segmentavimo srityse, naudodamas vieną universalią architektūrą. Pagrindiniai patobulinimai, palyginti su *MaskFormer,* apima:

- kaukėtas dėmesys (angl. *Masked Attention*): standartinis kryžminis dėmesys (angl. *cross-attention*) transformerio dekoderyje pakeičiamas maskuotu dėmesiu. Kiekvienos užklausos dėmesio mechanizmas yra apribotas sutelkti dėmesį tik į lokalizuotą pirmojo plano sritį (angl. *foreground region*) kaukę, kurią prognozavo ta užklausa ankstesniame dekoderio sluoksnyje. Šis principas sutelkia skaičiavimus į svarbias sritis, išgauna geriau lokalizuotus požymius, lemia greitesnę konvergenciją bei geresnį našumą;
- daugiamačio mastelio aukštos raiškos požymiai (angl. *Multi-Scale High-Resolution Features*): siekiant geriau apdoroti mažus objektus arba sritis, *Mask2Former* perduoda daugiamačio mastelio požymius iš pikselių dekoderio požymių piramidės tinklo (*FPN* [22]) į skirtingus transformerio dekoderio sluoksnius, tai leidžia užklausoms atkreipti dėmesį į požymius tinkamose raiškose;
- mokymo efektyvumas: kaukės nuostolių (angl. *mask loss*) skaičiavimas tampa efektyvesnis, skaičiuojant juos tik mažam skaičiui atsitiktinai parinktų taškų kiekvienoje kaukėje, tai ženkliai sumažina atminties naudojimą, nekenkiant našumui.

Kaukių klasifikavimo metodas, ypač įgyvendintas *Mask2Former,* yra galinga alternatyva tradiciniam pikselių lygmens klasifikavimui. Šios pažangios paradigmos paprastai naudoja stiprius požymių išgavimo bazinius modelius, dažnai pasitelkdamos galingus hierarchinius transformerius, pavyzdžiui, *Swin* [47] transformerį (įskaitant *SwinV2* [33], kuris paminėtas 1.3.4, 1.4.3 ir 1.4.4 skyriuose), kad generuotų pradinius vaizdo požymius, reikalingus pikselių ir transformerio dekoderiams. Prognozuojant kaukes kaip vientisus vienetus, tai savaime skatina geresnį objektų vientisumą ir kontūrų apibrėžimą. Šis į objektą orientuotas požiūris atrodo iš esmės gerai tinka užduotims, susijusioms su atskirais objektais, pvz., pastatų kontūrais, atlikti. Galimai leidžia švariau atskirti gretimus pastatus ir tiksliau nubrėžti kontūrus, palyginti su metodais, kurie klasifikuoja pikselius atskirai. Taip pat tiriami efektyvūs variantai, pavyzdžiui, prototipais pagrįstas efektyvus *MaskFormer* (angl. *Prototype-based Efficient MaskFormer, PEM*) [55].

1.5.2. Išankstinis apmokymas (angl. *pre-training*) ir savarankiškas mokymasis (angl. *Self-Supervised Learning, SSL*)

Kaip minėta anksčiau, bazinio modelio tinklų išankstinis apmokymas (angl. *pre-training*) naudojant didelius duomenų rinkinius yra labai svarbus siekiant aukšto tikslumo segmentavime, ypač kai tikslinės užduoties duomenų yra nedaug. Nors prižiūrimas išankstinis apmokymas su duomenų rinkiniais, pvz., *ImageNet*, yra įprastas, savarankiškas mokymasis tapo galinga alternatyva, leidžiančia mokytis vaizdavimų (angl. *representations*) iš didelių neanotuotų duomenų (angl. *unlabeled data*) kiekių, kurių dažnai yra žymiai daugiau nei anotuotų duomenų (angl. *labeled data*) – tai ypač aktualu nuotoliniame stebėjime [56]. *SSL* veikia apibrėžiant paruošimąją užduotį (angl. *pretext task*), kurioje priežiūros signalas gaunamas iš pačių duomenų. Dvi dominuojančios

paradigmos vaizdais paremtame *SSL* yra kontrastinis mokymasis (angl. *contrastive learning*) ir maskuotas vaizdų modeliavimas (angl. *masked image modeling*).

Kontrastinio mokymosi metodai: moko vaizdavimus, sutraukiant augmentuotas to paties vaizdo versijas (teigiamas poras (angl. *positive pairs*) ir atstumiant skirtingų vaizdų versijas (neigiamas poras (angl. *negative pairs*) įterpimo erdvėje (angl. *embedding space*). Viso to tikslas – išmokti augmentacijoms nekintamus, bet skirtingus egzempliorius atskiriančius požymius. Pavyzdžiai: *SimCLR* [57], *MoCo* [58], *BYOL* [59] bei *SimSiam* [60]. Taikymas: iš anksto apmokyti koduotuvai tikslinami vėlesnėms užduotims, pvz., segmentavimui.

Maskuoto vaizdų modeliavimo metodai: įkvėpti *NLP* (pvz., *BERT* [61]), šie metodai užmaskuoja dalį vaizdo ir apmoko modelį rekonstruoti trūkstamą turinį iš matomo konteksto [62], tokiu būdu išmokstant turtingą kontekstinį supratimą. Pavyzdžiai: *MAE* [63], *BEiT* [64], *SimMIM* [65]. Taikymas: Efektyvus išankstinis apmokymas įvairioms modernioms architektūroms (pvz., *ViT* [44], *Swin* [47], *PVT* [34], *ConvNeXt* [3]). Išmokti vaizdavimai gerai perkeliami į vėlesnes užduotis (pvz., segmentavimą).

Abi *SSL* paradigmos leidžia išmokti galingus vizualinius požymius iš neanotuotų duomenų. Tai ypač aktualu pastatų kontūrų išskyrimui naudojant gausius neanotuotus nuotolinio stebėjimo vaizdus, nes *SSL* išankstinis apmokymas gali pagerinti našumą [66]. Unikalios nuotolinio stebėjimo duomenų savybės atveria galimybes kurti specializuotas *SSL* užduotis, galinčias duoti efektyvesnius vaizdavimus šioje srityje nei apmokant su bendriniais vaizdais.

1.5.3. Generatyvinių modelių (angl. *Generative Models*) galimybių pasitelkimas

Naujausi dideli generatyviniai modeliai, kaip vaizdo-kalbos *CLIP* ir vaizdų difuzijos (angl. *image diffusion*) *Stable Diffusion*, atveria naujas galimybes semantiniam segmentavimui, ypač nulinio bandymo (angl. *zero-shot*) (modelis turi gebėti atpažinti klases, kurių niekada nematė) ar atviro žodyno (angl. *open-vocabulary*) (modelis gali segmentuoti objektus pagal bet kokį tekstinį aprašymą) atvejais.

CLIP savybės segmentavimui

CLIP (kontrastinis kalbos ir vaizdo išankstinis apmokymas, angl. *Contrastive Language-Image Pre-training*) [67] išmoksta suderintus vaizdo ir teksto įterpimus. Jo gebėjimas susieti tekstą su vaizdo sritimis leidžia naudoti jį nulinio ar kelių bandymų (angl. *zero/few-shot*) segmentavimui, nors tiesioginis taikymas pikselių lygmeniui yra sudėtingas. Metodai: *CLIP* požymių ar teksto įterpimų naudojimas [68], užklausų inžinerija (angl. *prompt engineering*) [68] ar žinių distiliavimas (pvz., *CLIP-to-Seg*) [67]. Pagrindinis privalumas: galutinis segmentavimo modelis įgyja nulinio bandymo galimybes ir išvedimo metu jam nebereikia skaičiavimų požiūriu brangaus *CLIP* modelio.

Difuzijos modelių savybės segmentavimui

Difuzijos modeliai, ypač tekstą į vaizdą keičiantys kaip *Stable Diffusion*, geba generuoti itin tikroviškus vaizdus, kas rodo jų išmoktą gilų semantinį supratimą [69]. Būtent dėl to šių modelių užšaldytų vidinių sluoksnių (pvz., *U-Net* triukšmo šalintojo (angl. *U-Net denoiser*) požymiai, turintys daug semantinės informacijos, yra naudingi tankioms prognozėms. Pavyzdžiui, *MaskDiffusion* [69] metodas naudoja užšaldytą *Stable Diffusion* atviro žodyno segmentavimui be papildomo apmokymo, analizuodamas vidinius dėmesio ar požymių žemėlapius, reaguojančius į semantines sąvokas. Be

požymių naudojimo, tiriamas ir tiesioginis difuzijos modelių pritaikymas segmentavimo užduočiai [70] pvz., kaip sąlyginio generavimo, kai pagal vaizdą generuojamas jo segmentavimo žemėlapis.

Didelių modelių, tokių kaip *CLIP* ir *Stable Diffusion*, savybių tyrimas rodo ryškią tendenciją link atvirojo rinkinio atpažinimo (angl. *open-set recognition*) ir nulinio bandymo (angl. *zero-shot*) segmentavimo. Pasitelkus jų sukauptas didžiules žinias, galima segmentuoti nematytas klases, gerokai viršijančias prižiūrimo apmokymo (angl. *supervised fine-tuning*) ribas. Tai ypač naudinga didelio masto pastatų kartografavimui įvairiose vietovėse, nes mažina anotavimo poreikį ir didina metodo plečiamumą (angl. *scalabilty*) bei prisitaikymą.

1.5.4. Modelių sujungimo (ansambliavimo, angl. *Model Ensembling*) strategijos

Modelių sujungimas – tai kelių nepriklausomai apmokytų modelių prognozių derinimas siekiant didesnio tikslumo ir atsparumo nei pavienio modelio. Tai įprasta technika rezultatams gerinti (ypač mašininio mokymosi varžybose). Toliau bus aptariami dažniausi metodai semantiniam segmentavimui.

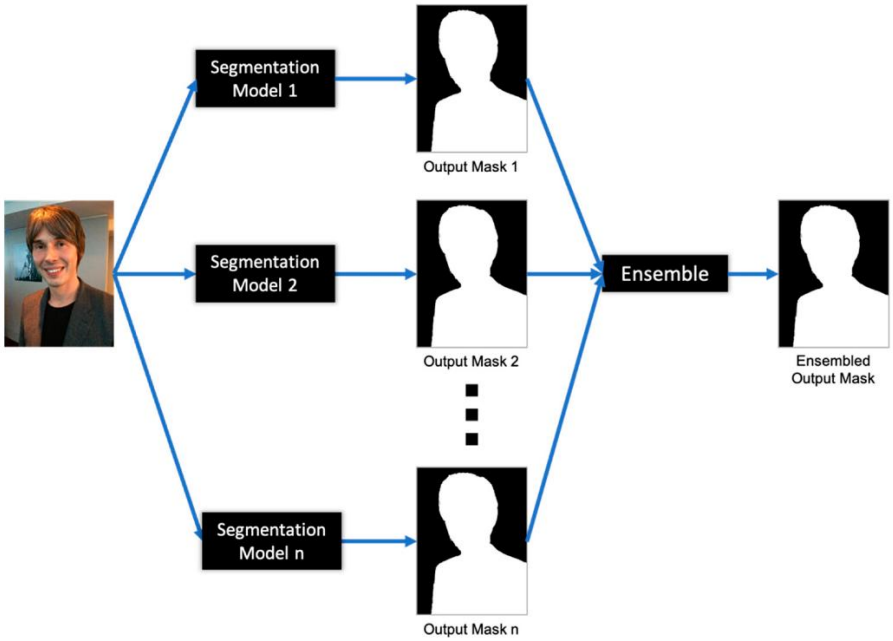
Tikimybių vidurkinimas (angl. *averaging probabilities*) [71]: paprasčiausias būdas – vidurkinti kelių modelių pateiktus pikselių lygmens tikimybių žemėlapius. Galutinė klasės etiketė kiekvienam pikseliui nustatoma pagal vidurkintas tikimybes taikant slenkstinę vertę (angl. *threshold*) (dvejetainiu atveju) arba imant *argmax* (atveju kai yra daug klasių).

Svertinis vidurkinimas (angl. *weighted averaging*) [72]: panašus į paprastą vidurkinimą, bet modelių prognozės sveriamos (pvz., pagal validavimo rezultatus ar įvairovę), suteikiant didesnį svorį geresniems ar labiau papildantiems modeliams.

Daugumos balsavimas (angl. *majority voting*) [73]: kiekvienas modelis prognozuoja diskrečią etiketę pikseliui; galutinė etiketė – ta, kurią prognozavo dauguma. Rečiau naudojama tikimybinėms išvestims (nebent po slenksčio taikymo).

Sukrovimas (angl. *stacking*) [74]: meta-modelis (angl. *meta-model*) apmokomas optimaliai derinti bazinių modelių prognozes, naudojant jų prognozes atidėtame rinkinyje (angl. *hold-out set*) kaip įvestį. Sudėtingesnis, bet potencialiai tikslesnis metodas.

Nors sujungimas gerina našumą (mažina dispersiją (angl. *variance*), gerina apibendrinimą (angl. *generalization*), jo trūkumas – didesnės skaičiavimo sąnaudos išvedimo (angl. *inference*) metu dėl kelių modelių vykdymo. Vis dėlto, kai svarbiausias maksimalus tikslumas, sujungimas išlieka vertinga technika. Modelių sujungimo veikimo pavyzdys pateiktas 8 pav. [75 p. 7].



8 pav. N segmentavimo modelių sujungimas

1.6. Praktiniai pastatų kontūrų segmentavimo aspektai

Norint efektyviai taikyti giliojo mokymosi modelius pastatų kontūrų išskyrimui, reikia atidžiai apsvarstyti praktinius aspektus, pradedant duomenų tvarkymu, baigiant nuostolių funkcijos (angl. *loss function*) pasirinkimu bei išvesties tikslinimu.

1.6.1. Aeronuotraukų ir palydovinių vaizdų išankstinis apdorojimas (angl. *Data Pre-processing*)

Aeronuotraukos ir palydoviniai vaizdai dėl savo unikalių savybių reikalauja specifinio išankstinio apdorojimo.

Dalijimas į lopus (arba plyteles) (angl. *tiling/patching*): kadangi neapdoroti vaizdai yra per dideli *GPU* (grafikos apdorojimo įrenginių (angl. *graphical processing unit*) atminčiai, jie dalijami į mažesnius lopus (pvz., 256×256, 512×512 pikselių), tinkamus neuroniniams tinklams apdoroti kartu su jų etikečių kaukėmis (angl. *label masks*) [76].

Persidengimo strategijos (angl. *Overlap Strategies*): siekiant išvengti artefaktų ties plytelių ribomis išvedimo metu, taikomas persidengimas: plytelės iškerpamos persidengiančiai, o prognozės persidengiančiose srityse sujungiamos (pvz., vidurkinant ar maišant) vientisam rezultatui gauti. Persidengimas mokymo metu taip pat gali veikti kaip duomenų augmentacija (angl. *data augmentation*) [77]. Didesnės plytelės ir persidengimas dažnai gerina rezultatus.

Normalizavimas (angl. *normalization*): būtinas stabiliam giliojo mokymosi apmokymui [78]. Dėl nuotolinio stebėjimo duomenų ypatumų (didesnis bitų gylis, specifiniai verčių diapazonai bei pasiskirstymai) reikalingi specialūs metodai. Įprastos strategijos:

- standartizavimas pagal juostas (angl. *Per-band Standardization*): kiekvienos juostos vidurkio atėmimas ir dalijimas iš standartinio nuokrypio;

- mastelio keitimas konstanta arba atspindžio konvertavimas (angl. *Scaling by Constant/Reflectance Conversion*): dalijimas iš konstantos (pvz., atspindžio aproksimavimui);
- nukirpimas pagal procentiles (angl. *percentile truncation*): ekstremalių verčių (išskirčių (*angl. outliers*) apribojimas pagal procentiles prieš keičiant mastelį (angl. *scaling*);
- logaritminis normalizavimas (angl. *log normalization*): logaritminės transformacijos taikymas, siekiant apdoroti sunkiasvorius (angl. *heavy-tailed*) skirstinius prieš keičiant mastelį (angl. *scaling*);
- *ImageNet* statistika: jei naudojamos tik *RGB* juostos ir bazinis modelis, iš anksto apmokytas su *ImageNet*, šių juostų normalizavimas naudojant *ImageNet* vidurkį ir standartinį nuokrypį gali būti būtinas efektyviam perkeliamajam mokymuisi (angl. *transfer learning*);
- santykinis normalizavimas (angl. *relative normalization*): metodai, tokie kaip histogramų suderinimas arba sklaidos diagramų suderinimas gali būti naudojami normalizuoti tikslinį vaizdą pagal etaloninį vaizdą taip siekiant užtikrinti nuoseklumą.

Daugiakanalių duomenų tvarkymas (angl. *Multi-band Handling*): labai aukštos raiškos (angl. *very high resolution, VHR*) palydoviniai vaizdai dažnai turi daugiau nei tris spektrines juostas (pvz., artimųjų infraraudonųjų spindulių juosta, trumpabangė infraraudonųjų spindulių juosta *WorldView* ar *Sentinel-2* jutikliuose). Šios papildomos juostos gali suteikti vertingos informacijos skiriant pastatus nuo augmenijos ar kito fono [79, 80]. Strategijos daugiakanalių duomenų naudojimui apima:

- juostų pasirinkimas: juostų poaibio pasirinkimas (pvz., *RGB* + artimųjų infraraudonųjų spindulių juosta), remiantis srities žiniomis ar požymių svarbos analize;
- įvesties sluoksnio modifikavimas: pirmojo konvoliucinio sluoksnio pritaikymas priimti specifinį įvesties kanalų skaičių, atitinkantį pasirinktas juostas;
- visų juostų naudojimas: modelio apmokymas naudojant visas prieinamas spektrines juostas.

Duomenų augmentacija (angl. *data augmentation*): standartinės augmentacijos, tokios kaip atsitiktiniai apvertimai ir pasukimai, yra dažnai naudojamos [81]. Atsižvelgiant į vaizdą iš viršaus, nekintamumas pasukimui dažnai yra didelis. Elastinės deformacijos (angl. *elastic deformations*) gali imituoti jutiklio geometrijos ar reljefo pokyčius. Mastelio drebėjimas (angl. *scale jittering*) (vaizdų dydžio keitimas atsitiktiniu faktoriumi) gali padėti modeliams tapti atsparesniems objektų mastelio pokyčiams [82]. Vaizdų iš skirtingų laiko momentų (pakartotinių stebėjimų (angl. *temporal revisits*) naudojimas tai pačiai vietai taip pat gali būti kaip augmentacijos forma, leidžiant modeliui susidurti su apšvietimo ir sezoniniais pokyčiais [83].

Unikalus nuotolinio stebėjimo duomenų pobūdis reiškia, kad išankstinio apdorojimo pasirinkimai ženkliai veikia modelio našumą. Skirtingai nuo standartinių kompiuterinės regos konvejerių, kurie dažnai numatytai keičia įvesčių dydį ir naudoja fiksuotą normalizavimo statistiką, nuotoliniam stebėjimui reikia atidžiai apsvarstyti dalijimo į plyteles strategijas raiškai išsaugoti, specializuotus normalizavimo metodus skirtingiems duomenų diapazonams ir pasiskirstymams apdoroti, ir apgalvotus metodus daugiaspektrės informacijos įtraukimui. Persidengiančios plytelės yra ypač svarbios išvedimo metu, kad būtų išvengta šachmatų lentos artefaktų (angl. *checkerboard artifacts*) ir užtikrinta erdviškai nuoseklių išvesčių dideliuose plotuose.

1.6.2. Nuostolių funkcijos (angl. *loss functions*)

Nuostolių funkcijos pasirinkimas yra kritiškai svarbus segmentavimo kokybei. Standartinė dvejetainė kryžminė entropija (angl. *Binary Cross-Entropy, BCE*) dažnai nepakankama segmentavimui dėl klasių disbalanso (pvz., pastatų kontūrų išskyrimo, kur fono pikselių daug daugiau nei pastatų) arba poreikio tiksliai apibrėžti kontūrus. Todėl kuriamos specializuotos nuostolių funkcijos, dažniausiai skirstomos į sritimi pagrįstas ir kontūru pagrįstas.

Sritimi pagrįstos nuostolių funkcijos (angl. *region-based Loss functions*)

Šios funkcijos vertina prognozuojamos ir etaloninės kaukės persidengimą ar pasiskirstymo panašumą [84, 85, 86].

Dice nuostolių (angl. *Dice Loss*) funkcija: išvesta iš *Dice* panašumo koeficiento (angl. *Dice Similarity Coefficient, DSC* arba *F1* įvertis (angl. *F1-score*). Ši funkcija (iš vieneto atėmus *Dice* koeficientą) iš prigimties yra atspari klasių disbalansui, nes normalizuoja pagal prognozuojamų ir etaloninių (angl. *ground truth*) pikselių sumą, numanomai vienodai traktuodama tikslinę klasę (angl. *foreground*) ir foną (angl. *background*).

IoU (*Jaccard*) nuostolių (angl. *IoU Loss, Jaccard Loss*) funkcija: pagrįsta susikirtimu padalintu iš sąjungos (angl. *Intersection over Union, IoU*) arba *Jaccard* indekso (angl. *Jaccard Index*) metrika. *IoU* nuostolių funkcija (iš vieneto atėmus *IOU* koeficientą) taip pat optimizuoja persidengimą, bet labiau baudžia už klaidingai teigiamus (angl. *false positives, FP*) rezultatus nei *Dice* nuostolių funkcija. *IoU* yra labai paplitusi segmentavimo vertinimo metrika.

Tversky (angl. *Tversky Loss*) nuostolių funkcija: *Dice* nuostolių funkcijos generalizacija, kuri įveda parametrus α ir β, kad skirtingai pasvertų klaidingai teigiamus (angl. *false negatives, FP*) ir klaidingai neigiamus (angl. *false negatives, FN*) rezultatus. Reguliuojant α ir β, galima kontroliuoti kompromisą tarp tikslumo ir aprėpties (angl. *recall*), todėl ji tinka labai nebalansuotiems duomenų rinkiniams, kur svarbu kontroliuoti *FN* ar *FP*. Nustačius $\alpha = \beta = 0{,}5$, gaunamas *Dice* koeficientas.

Židinio nuostolių (angl. *Focal Loss*) funkcija: iš pradžių pasiūlyta objektų aptikimui, židinio nuostolių funkcija modifikuoja standartinę *BCE* nuostolių funkciją, sumažindama lengvai klasifikuojamų pavyzdžių įtaką (dažnai gausių fono pikselių) ir sutelkdama mokymą į sunkiai klasifikuojamus pavyzdžius (dažnai priekinio plano ar kontūrų pikselius). Ji įveda moduliuojantį faktorių (angl. *modulating factor*).

Pagal sritį nuostolių funkcija (angl. *Region-wise Loss, RW*) [87]: bendra sistema, kuri priskiria kiekvienam pikseliui svorį *w(x)* (pvz., atstumą iki ribos), tuo pačiu spręsdama klasių disbalansą ir kontūrų tikslumą. Priklausomai nuo to, kaip konkrečiai apibrėžiama svorio funkcija *w(x)*, ši *RW* sistema veikia kaip specialus ribos (angl. *Boundary*), aktyvaus kontūro (angl. *Active-Contour*), *Hausdorff* ar atstumų žemėlapio (angl. *Distance-Map*) nuostolių funkcijos atvejis. Būtent dėl šių savybių – gebėjimo vienu metu efektyviai spręsti klasių disbalanso ir kontūrų tikslumo problemas – *RW* sistema yra laikoma perspektyvia specifinėms užduotims, tokioms kaip dvejetainė semantinė pastatų kontūrų segmentacija.

Kontūru pagrįstos nuostolių funkcijos (angl. *boundary-based loss functions*)

Šios funkcijos skirtos pagerinti prognozuojamų segmentavimo kontūrų (arba ribų) tikslumą. Tai labai svarbu, kai reikia išgauti tikslias formas (pvz., pastatų kontūrus), nes vien sritimi pagrįstos funkcijos kartais nepakankamai baudžia už kontūrų klaidas [84, 86].

Ribos nuostolių (angl. *Boundary Loss*) funkcija: skirtingai nei sritimi pagrįstos, ji vertina atstumą tarp prognozuojamos ir tikrosios ribos, o ne persidengiančius plotus. Dažnai naudoja atstumų transformacijas (angl. *distance transforms*), kad labiau baustų už klaidas arti tikrosios ribos, ji taip pat yra mažiau jautri objektų dydžio skirtumams.

Hausdorff atstumo nuostolių (angl. *Hausdorff Distance Loss*) funkcija: siekia sumažinti *Hausdorff* atstumą (angl. *Hausdorff Distance, HD*) – didžiausią atstumą tarp vienos ribos taško ir artimiausio taško kitoje riboje. Šis atstumas jautrus net mažiausioms kontūrų klaidoms ar išskirtims (angl. *outliers*). Kadangi tiesioginis optimizavimas sudėtingas, dažnai naudojami supaprastinimai (aproksimacijos), pvz., pasitelkiant atstumų transformacijas. Optimizuoti *GPU* įgyvendinimai siekia pagerinti efektyvumą [88].

Formą atpažįstanti nuostolių (angl. *Shape-aware Loss*) funkcija: įtraukia išankstines žinias apie tikėtiną objektų formą. Tai gali apimti atstumų žemėlapių naudojimą, gautų iš etaloninių kaukių (angl. *ground truth masks*), kaip baudų žemėlapius (angl. *penalty maps*), suteikiant didesnį svorį klaidoms arti kontūrų. Tikslas yra nukreipti tinklą link segmentacijų su realistiškesnėmis formomis generavimo.

Aktyvioji kontūro nuostolių (angl. *Active Boundary Loss*) funkcija: laipsniškai skatina suderinimą tarp prognozuojamų kontūrų (aptiktų iš dabartinės tinklo išvesties) ir etaloninių kontūrų, formuluojant suderinimą kaip diferencijuojamą krypties vektoriaus prognozavimo problemą.

Kontūro *DoU* nuostolių (angl. *Boundary DoU Loss,* kur *DoU* (angl. *Difference over Union*) reiškia skirtumo ir sąjungos santykį) [89] funkcija: naujas pasiūlymas, sutelktas į kontūro sritį. Ji apskaičiuojama kaip santykis tarp skirtumo aibės (angl. *difference set*) ir skirtumo aibės sąjungos su dalimi sankirtos aibės (angl. *intersection set*). Ji vengia aiškaus kontūrų aptikimo ar atstumų transformacijų.

Sudėtinės nuostolių funkcijos

Dažnai skirtingų nuostolių funkcijų derinimas gali duoti geresnių rezultatų, pasinaudojant jų vienas kitą papildančiais privalumais. Įprasti deriniai apima pridėjimą sritimi pagrįstos nuostolių funkcijos (pvz., *Dice*) prie pasiskirstymu pagrįstos nuostolių funkcijos (angl. *distribution-based loss*) (pvz., *BCE* ar *Focal* nuostolių funkcijos), arba sritimi pagrįstos nuostolių funkcijos derinimą su kontūru pagrįsta nuostolių funkcija. Tai leidžia subalansuoti siekinius, pavyzdžiui, disbalanso valdymą, persidengimo optimizavimą ir kontūrų tikslumo užtikrinimą.

Nuostolių funkcijos pasirinkimas labai priklauso nuo specifinių duomenų rinkinio savybių (pvz., klasių balanso, objektų dydžių, kontūrų sudėtingumo) ir nuo pageidaujamų galutinės segmentacijos savybių (pvz., didelis persidengimas ar tikslūs kontūrai). Specializuotų nuostolių funkcijų plitimas atspindi niuansus reikalavimams segmentavimo užduotims, tokioms kaip pastatų kontūrų išskyrimas, kur standartinės nuostolių funkcijos dažnai yra netinkamos. Sritimi pagrįstos nuostolių funkcijos puikiai tvarkosi su disbalansu ir maksimizuoja persidengimą, o kontūru pagrįstos nuostolių funkcijos

tiesiogiai orientuotos į geometrinį tikslumą, kuris yra esminis aspektas generuojant tikslius pastatų kontūrus.

1.6.3. Optimizatoriai (angl. *Optimizers*) ir aktyvacijos funkcijos (angl. *Activation Functions*)

Nors optimizatoriaus ir aktyvacijos funkcijų pasirinkimas yra mažiau specifiškas pačiai segmentavimo užduočiai, palyginti su architektūromis ar nuostolių funkcijomis, tai vis tiek išlieka svarbus aspektas sėkmingam mokymui.

Optimizatoriai

Stochastinis nusileidimas gradientais (angl. *Stochastic Gradient Descent, SGD*) buvo naudojamas ankstyvuosiuose FCN projektuose [7]. Tačiau adaptyvūs optimizatoriai, tokie kaip *Adam* [90, 91], ir ypač *AdamW* (*Adam* su atsietu svorių mažinimu (angl. *decoupled weight decay*) [92, 93, 94], dabar yra dažniau naudojami, ypač mokant transformeriais pagrįstus modelius ir siekiant greitesnės konvergencijos daugelyje scenarijų. Mokymosi greičio planavimas (angl. *learning rate scheduling*) (pvz., polinominis mažinimas, kosinusinis atkaitinimas (angl. *cosine annealing*) taip pat yra labai svarbus efektyviam mokymui.

Aktyvacijos funkcijos

ReLU išlieka įprastu pasirinkimu CNN pagrįstose architektūrose [10]. Tačiau *GELU* (Gauso paklaidos tiesinis vienetas, angl. *Gaussian Error Linear Unit*) [95], kuris yra įkvėptas transformerių, vis dažniau naudojama šiuolaikiniuose baziniuose modeliuose, tokiuose kaip *ConvNeXt* [3] ir *MaxViT* [29], kartais suteikdamas našumo pagerėjimą. Galutiniam išvesties sluoksniui dvejetainiam segmentavimui, paprastai naudojama sigmoidinė (angl. *Sigmoid*) aktyvacijos funkcija [96], kad gautų pikselių lygmens tikimybes tarp 0 ir 1. Daugiaklasiam segmentavimui (nėra šio darbo tikslas, bet yra svarbu kontekstui), naudojama *Softmax* funkcija [97], kad kiekvienam pikseliui būtų sukurtas tikimybių pasiskirstymas tarp klasių.

1.6.4. Papildomas apdorojimas (angl. *post-processing*) kontūrų tikslinimui

Giliojo mokymosi modeliai paprastai pateikia rastrinį tikimybių žemėlapį (angl. *raster probability map*) arba dvejetainę segmentavimo kaukę. Tačiau daugeliui *GIS* (Geografinių Informacinių Sistemų) taikymų, reikalingi vektoriniai poligonai, vaizduojantys pastatų kontūrus. Todėl reikalingi papildomo apdorojimo žingsniai, kad neapdorota modelio išvestis būtų paversta į švarius, naudojamus vektorinius duomenis.

Triukšmo šalinimas arba gludinimas (angl. *noise removal/smoothing*)

Neapdorotos segmentavimo kaukės gali turėti mažų klaidingų prognozių (druskos ir pipirų triukšmo (angl. *salt-and-pepper noise*) arba skylių pastatų srityse. Morfologinės operacijos, pavyzdžiui, atvėrimas (angl. *opening*) (erozija (angl. *erosion*), po kurios seka išplėtimas (angl. *dilation*), gali pašalinti mažus triukšmo telkinius, o užvėrimas (angl. *closing*) (išplėtimas, po kurio seka erozija) gali užpildyti mažas skyles [98]. Taip pat galima taikyti gludinimo filtrus (angl. *smoothing filters*), tačiau reikia saugotis, kadangi gali būti pernelyg sulieti ryškūs kontūrai.

Poligonizacija (angl. *polygonization*)

Tai yra pagrindinis žingsnis konvertuojant rastrinę kaukę į vektorinius poligonus. Įprasti algoritmai apima kontūrų radimą jungiesiems komponentams (angl. *connected components*) (prognozuotoms pastatų sritims) dvejetainėje kaukėje [99]. Šie kontūrai iš pradžių vaizduojami kaip pikselių koordinačių sekos.

Poligonų supaprastinimas arba gludinimas (angl. *polygon simplification/smoothing*)

Pradiniai poligonai, gauti tiesiai iš pikselių kontūrų, dažnai yra dantyti ir turi perteklinių viršūnių. Poligonų supaprastinimo algoritmai, pavyzdžiui, *Douglas-Peucker* algoritmas, yra naudojami sumažinti viršūnių skaičių, išsaugant esminę formą pagal nurodytą leistiną nuokrypį [99, 100]. Tokiu būdu gaunami lygesni ir kompaktiškesni vektoriniai vaizdai.

Reguliarizacija (angl. *regularization*)

Pastatų kontūrai dažnai pasižymi geometriniais reguliarumais (pvz., tiesiomis kraštinėmis, stačiais kampais). Dėl to galima taikyti papildomo apdorojimo metodus, kad šie reguliarumai būtų užtikrinti [101]. Tai gali apimti tiesių pritaikymą poligono kontūro atkarpoms, dominuojančių krypčių nustatymą, ir viršūnių padėčių koregavimą, kad jos atitiktų šias kryptis arba užtikrintų statmenumą. Kai kuriuose tyrimuose nagrinėjamas reguliarizacijos integravimas tiesiogiai į mokymosi procesą, pavyzdžiui, per specializuotas nuostolių funkcijas arba ištisinius (angl. *end-to-end*) modelius, kurie tiesiogiai prognozuoja vektorinius vaizdus.

Svarbu paminėti, kad papildomo apdorojimo efektyvumas labai priklauso nuo pradinės segmentavimo kaukės kokybės. Reikšmingų klaidų kaukėje (pvz., susiliejusių gretimų pastatų, didelių trūkstamų dalių) ne visada galima ištaisyti papildomo apdorojimo metodais. Tai pabrėžia svarbą pasiekti aukštą tikslumą pačiame giliojo mokymosi segmentavimo etape ir skatina tirti metodus, kurie tiesiogiai generuoja geometriškai tikslias išvestis arba įtraukia geometrinius apribojimus mokymo metu, sumažinant atotrūkį tarp pikseliais pagrįsto giliojo mokymosi segmentavimo pobūdžio ir vektorinių *GIS* taikymų reikalavimų.

1.7. Apibendrinimas

Literatūros analizė atskleidė, jog pastatų kontūrų segmentavimas iš palydovinių ir aeronuotraukų yra aktyvi tyrimų sritis, kurioje dominuoja giliojo mokymosi metodai. Pradedant nuo pamatinių konvoliucinių neuroninių tinklų (CNN) ir pilnai konvoliucinių tinklų (FCN) [7], vystymas vyko link sudėtingesnių kodavimo-dekodavimo architektūrų, tokių kaip *U-Net* [14], *DeepLabV3+* [17] ir *PAN* [19], kurios efektyviai sujungia skirtingų lygių požymius. Šios architektūros tapo plačiai taikomos dėl gebėjimo išgauti tiek aukšto lygio semantinę informaciją, tiek tikslias erdvines detales.

Naujausios tendencijos apima dėmesio mechanizmų ir transformerių architektūrų (pvz., *Vision Transformer* [25], *Swin Transformer* [47]) bei hibridinių CNN ir transformerio metodų taikymą, siekiant geriau modeliuoti globalų kontekstą ir tolimas priklausomybes vaizde. Taip pat integruojamos pažangios paradigmos, tokios kaip kaukių klasifikavimas (pvz., *Mask2Former* [54]), savarankiškas mokymasis (*SSL*) ir modelių sujungimas. Ne mažiau svarbus dėmesys skiriamas ir praktiniams aspektams, tokiems kaip specifinis nuotolinio stebėjimo vaizdų apdorojimas, specializuotų nuostolių funkcijų (pvz., *Dice, Focal, Hausdorff* [84, 85, 86]) parinkimas ir galutinių rezultatų tikslinimas vektoriniu formatu.

2. Tyrimo objektas ir metodologija

Šiame skyriuje apibrėžiamas tyrimo objektas – automatizuotas pastatų kontūrų išskyrimas iš aeronuotraukų ir palydovinių vaizdų – ir detaliai pristatoma tyrimo metodologija. Darbo metu buvo atrinkti ir paruošti įvairūs viešai prieinami duomenų rinkiniai bei specialiai šiam tyrimui sukurtas Lietuvos teritorijos duomenų rinkinys, skirti giliojo mokymosi modelių mokymui ir vertinimui. Eksperimentams vykdyti pasirinktos ir konfigūruotos specifinės konvoliucinių neuroninių tinklų (CNN) architektūros, tokios kaip *U-Net* ir *PAN*, bei atliktas jų komponentų (kodavimo dalių (bazinių modelių), nuostolių funkcijų, optimizatorių ir kt.) testavimas siekiant identifikuoti optimalias konfigūracijas pastatų segmentavimo uždaviniui. Taip pat šiame darbe sukurta ir ištirta nauja modelių sujungimo strategija, pagrįsta pastatų tankiu vaizde, kurios tikslas – padidinti segmentavimo tikslumą. Tolesniuose poskyriuose nuosekliai aprašomi naudoti duomenų rinkiniai, taikyti įrankiai ir technologijos bei išsamiai pristatomi tyrimo metodologijos etapai ir taip pat vertinimo metrikos.

Šis tyrimas bus tęsinys tyrimo, kuris buvo pristatytas DAMSS 2023 konferencijoje (žr. 1, 2 priedus).

2.1. Tyrimo objektas

Šio projekto tyrimo objektas yra giliojo mokymosi metodų, skirtų automatiniam pastatų kontūrų išskyrimui aeronuotraukų ir palydovinių nuotraukų vaizduose, kūrimas, vertinimas ir adaptavimas.

Šiame darbe pagrindinis dėmesys skiriamas dvejetainio semantinio segmentavimo uždaviniui spręsti naudojant giliojo mokymosi modelius. Semantinis segmentavimas šiuo atveju reiškia kiekvieno vaizdo pikselio klasifikavimą į vieną iš dviejų klasių: pastatas arba ne pastatas (fonas). Modelio įvestis yra aeronuotrauka arba palydovinė nuotrauka, o išvestis – dvejetainė kaukė, kurioje baltos spalvos pikseliai žymi pastatų plotus, o juodos – foną.

Siekiant išspręsti šį uždavinį, darbe nagrinėjami ir taikomi konvoliuciniai neuroniniai tinklai (CNN), specifiškai kodavimo-dekodavimo (angl. *encoder-decoder*) architektūros, tokios kaip *U-Net* [14] ir jos variantai bei kitos modernios architektūros (pvz., *PAN* [19]). Šios architektūros pasižymi gebėjimu efektyviai išgauti tiek aukšto lygio semantinius, tiek žemo lygio erdvinius požymius iš vaizdų, kas yra būtina tiksliam objektų kontūrų nustatymui.

Nepaisant giliojo mokymosi pažangos, automatizuotas pastatų pėdsakų išskyrimas išlieka sudėtingas dėl kelių veiksnių:

- duomenų įvairovė: skirtinga vaizdų raiška, apšvietimo sąlygos, sezoniniai pokyčiai;
- uždengimas: pastatai gali būti dalinai arba visiškai uždengti medžių lajos, šešėlių ar debesų;
- pastatų įvairovė: skirtingos pastatų formos, dydžiai, stogų tipai ir medžiagos.

Specifinis šio darbo tikslas yra ne tik pritaikyti esamus metodus, bet ir atlikti modelių komponentų (architektūrų, bazinių modelių, nuostolių funkcijų, optimizatorių) tyrimą, siekiant rasti optimalias konfigūracijas. Taip pat svarbus tyrimo aspektas yra modelių adaptavimas ir vertinimas naudojant Lietuvos geoerdvinius duomenis. Tam buvo sukurti ir paruošti duomenų rinkiniai, leidžiantys įvertinti, kaip skirtinguose duomenų rinkiniuose apmokyti modeliai veikia. Galiausiai, darbe tiriama nauja modelių sujungimo (ansambliavimo) strategija, pagrįsta pastatų tankiu vaizde, siekiant pagerinti segmentavimo tikslumą įvairiose situacijose.

2.2. Įrankiai ir technologijos

Tyrimas ir eksperimentai atlikti naudojant *Python*[2] programavimo kalbą.

Pagrindinis giliojo mokymosi karkasas: *PyTorch*[3]: atviro kodo mašininio mokymosi biblioteka, plačiai naudojama kompiuterinėje regoje ir natūralios kalbos apdorojime. Pasirinkta dėl lankstumo, gausios bendruomenės palaikymo ir dinaminių skaičiavimo grafikų. Toliau bus pateikiamos pagrindinės bibliotekos naudotos tyrime.

segmentation-models-pytorch (*smp*)[4]: aukšto lygio biblioteka, sukurta ant *PyTorch*, suteikianti prieigą prie daugelio populiarių semantinio segmentavimo architektūrų (*U-Net* [14], *PAN* [19], *FPN* [102], *LinkNet* [21], *PSPNet* [23], *DeepLabV3+* [18]) ir leidžianti lengvai keisti bei naudoti įvairius iš anksto apmokytus kodavimo bazinius modelius (angl. *encoders/backbones*).

timm (*PyTorch Image Models*)[5]: išsami biblioteka, teikianti didžiulį rinkinį modernių vaizdų klasifikavimo modelių su iš anksto apmokytais svoriais (įskaitant įvairius *ResNet* [11], *ConvNeXt* [3] ir *Vision Transformer* modelius). Šie modeliai buvo intensyviai naudojami kaip kodavimo dalys (baziniai modeliai) segmentavimo architektūrose, ypač atliekant bazinių modelių testavimą.

osmnx[6]: biblioteka, skirta gauti, apdoroti ir vizualizuoti gatvių tinklus ir kitus erdvinius duomenis (pvz., pastatų pėdsakus) iš *OpenStreetMap*[7].

Geopandas[8]: išplečia pandas bibliotekos funkcionalumą, leisdama dirbti su geoerdviniais vektoriniais duomenimis (pvz., skaityti arba rašyti *Shapefile*, *GeoJSON*, atlikti erdvines užklausas).

Shapely[9]: naudojama geometrinių objektų manipuliavimui ir analizei.

Rasterio[10]: pagrindinė biblioteka darbui su rastriniais geoerdviniais duomenimis (pvz., *GeoTIFF*). Leidžia skaityti, rašyti, transformuoti ir kitaip apdoroti rastrinius vaizdus ir kaukes.

Contextily[11]: biblioteka, leidžianti lengvai atsisiųsti ir pridėti žemėlapių fragmentus (angl. basemap tiles) iš įvairių viešų servisų (pvz., *OpenStreetMap*, *Esri*, *Google Maps*) į *matplotlib* ar *geopandas* vizualizacijas pagal nurodytas geografines ribas. Naudota palydoviniams vaizdams atsisiųsti.

OpenCV (*opencv-python*)[12]: plačiai naudojama kompiuterinės regos biblioteka, taikyta pagrindinėms vaizdų apdorojimo operacijoms (pvz., nuskaitymui, spalvų konvertavimui).

Pillow[13]: biblioteka vaizdų failų nuskaitymui ir įrašymui (pvz., *PNG* formatu).

---

[2] https://www.python.org/
[3] https://pytorch.org/
[4] https://github.com/qubvel-org/segmentation_models.pytorch
[5] https://github.com/pprp/timm
[6] https://github.com/gboeing/osmnx
[7] https://www.openstreetmap.org/
[8] https://github.com/geopandas/geopandas
[9] https://github.com/shapely/shapely
[10] https://github.com/rasterio/rasterio
[11] https://github.com/geopandas/contextily
[12] https://github.com/opencv/opencv-python
[13] https://github.com/python-pillow/Pillow

NumPy[14]: fundamentali biblioteka moksliniams skaičiavimams su *Python*, ypač darbui su masyvais.

Pandas[15]: biblioteka duomenų analizei ir manipuliavimui, ypač darbui su lentelių tipo duomenimis (pvz., rezultatų saugojimui ir analizei).

Scikit-learn[16]: populiari mašininio mokymosi biblioteka, naudota duomenų padalinimui į mokymo, validavimo ir testavimo aibes bei kai kurių papildomų metrikų skaičiavimui.

Matplotlib[17]: pagrindinė biblioteka grafikų ir vizualizacijų kūrimui *Python* aplinkoje.

Šių įrankių ir bibliotekų kombinacija leido efektyviai įgyvendinti visus tyrimo etapus – nuo duomenų rinkinių kūrimo ir paruošimo iki sudėtingų giliojo mokymosi modelių kūrimo, apmokymo, testavimo ir rezultatų vertinimo.

2.3. Duomenų rinkiniai

Giliojo mokymosi modelių efektyvumui ir gebėjimui generalizuoti didelę įtaką turi mokymui naudojamų duomenų kiekis, įvairovė ir kokybė. Šiame darbe modelių mokymui, derinimu ir vertinimui buvo naudojami keli viešai prieinami, plačiai šioje srityje taikomi aukštos raiškos aeronuotraukų ir palydovinių vaizdų duomenų rinkiniai, skirti pastatų pėdsakų segmentavimui, bei specialiai šiam tyrimui sukurtas Lietuvos teritorijos duomenų rinkinys. Modelių treniravimui bei vertinimui buvo naudojami 3 kanalų (*RGB*) vaizdai, kadangi tai labiausiai pasiekiami duomenų rinkiniai. Taip pat, vaizdų matmenys yra 256×256.

Viešai prieinami duomenų rinkiniai

Siekiant apmokyti modelius atpažinti įvairių tipų pastatus skirtingose aplinkose, šiame darbe buvo pasitelkti keli tarptautiniu mastu pripažinti ir plačiai naudojami duomenų rinkiniai. Naudoti rinkiniai:

- *SpaceNet*[18]: tai serija iššūkių ir su jais susijusių duomenų rinkinių, orientuotų į pastatų ir kelių išskyrimą iš palydovinių vaizdų. Naudotos kelios šio projekto versijos (1, 2, 6), apimančios aukštos raiškos *WorldView-2* ir *WorldView-3* palydovų nuotraukas. Rinkiniai dengia geografiškai ir urbanistiškai skirtingus miestus (pvz., Rio de Žaneirą, Las Vegasą, Šanchajų, Chartumą, Roterdamą), pateikiant pastatų pėdsakus *GeoJSON* formatu. Dėl didelės raiškos ir įvairovės šie rinkiniai yra vertingas resursas sudėtingiems modeliams mokyti ir testuoti;
- *WHU Building Dataset*[19]: tai dar vienas populiarus aeronuotraukų rinkinys, surinktas virš Kraistčerčo miesto Naujojoje Zelandijoje, pasižymintis 0,3 m raiška. Rinkinys yra patogiai padalintas į mokymo, validavimo bei testavimo aibes, kas palengvina standartizuotą modelių vertinimą. Pastatų kaukės pateikiamos rastriniu formatu;
- *Massachusetts Buildings Dataset*[20]: šį rinkinį sudaro 1 metro raiškos aeronuotraukos, apimančios Bostono miestą ir jo apylinkes (JAV). Jame yra virš 150 *TIFF* formato vaizdų (1500×1500), turinčių 3 (*RGB*) kanalus, ir atitinkamos dvejetainės pastatų kaukės (taip pat

[14] https://github.com/numpy/numpy
[15] https://github.com/pandas-dev/pandas
[16] https://github.com/scikit-learn/scikit-learn
[17] https://github.com/matplotlib/matplotlib
[18] https://spacenet.ai/
[19] https://www.kaggle.com/datasets/xiaoqian970429/whu-building-dataset
[20] https://www.kaggle.com/datasets/balraj98/massachusetts-buildings-dataset?select=png

TIFF formatu). Rinkinys dažnai naudojamas pastatų segmentavimo algoritmų testavimui ir palyginimui.

Toliau pateikiamoje 1 lentelėje apibendrinamos pagrindinės šių duomenų rinkinių charakteristikos ir jų panaudojimas šiame darbe.

1 lentelė. Pastatų segmentavimo duomenų rinkinių palyginamas

Duomenų rinkinys	**Šaltinio tipas**	**Raiška (m/pikselis)**	**Geografinė aprėptis**	**Vaizdų formatas ir dydis (pikseliai)**	**Vaizdų kanalai**	**Kaukių formatas**
SpaceNet 1	Palydovinis (*WV-2*)	0,5	Rio de Žaneiras	*GeoTIFF* ~439×406	3 (*RGB*) / 8 (*MS*)	*GeoJSON*
SpaceNet 2	Palydovinis (*WV-3*)	0,3	Las Vegasas, Paryžius, Šanchajus, Chartumas	*GeoTIFF* ~650×650	3 (*RGB*) / 8 (*MS*)	*GeoJSON*
SpaceNet 6	Palydovinis (*WV-3 + SAR*)	Optinė 0,3 – 0,5; *SAR* 0,5	Roterdamas	*GeoTIFF* 900×900	3/8 (optinė), N/A (*SAR*)	*GeoJSON*
WHU Building Dataset (NZ)	Aeronuotraukos ir palydovinis	0,075 – 0,3	Kraistčerčas (NZ)	*TIFF* 512×512	3 (*RGB*)	*TIFF / Shapefile*
Massachusetts Buildings	Aeronuotraukos	1,0	Bostonas (JAV)	*TIFF* 1500×1500	3 (*RGB*)	*TIFF*

Lietuvos duomenų rinkinio paruošimas

Siekiant įvertinti modelių pritaikomumą ir veikimo kokybę specifinėmis Lietuvos sąlygomis, buvo sukurtas Lietuvos gyvenviečių duomenų rinkinys. Toliau bus aptariami žingsniai sukurti rinkinį.

Duomenų šaltiniai:

- pastatų vektoriai: naudoti viešai prieinami *OpenStreetMap* (*OSM*) projekto vektoriniai duomenys (poligonai), reprezentuojantys pastatų kontūrus Lietuvos bei kitų valstybių teritorijose. Duomenys gauti naudojant *osmnx* biblioteką ir apdoroti naudojant *geopandas*;
- vaizdų šaltinis: kaip palydovinių vaizdų šaltinis pasirinktas *Esri World Imagery* servisas, teikiantis aukštos raiškos pasaulinę dangą per *contextily* biblioteką.

Vaizdų ir kaukių generavimas:

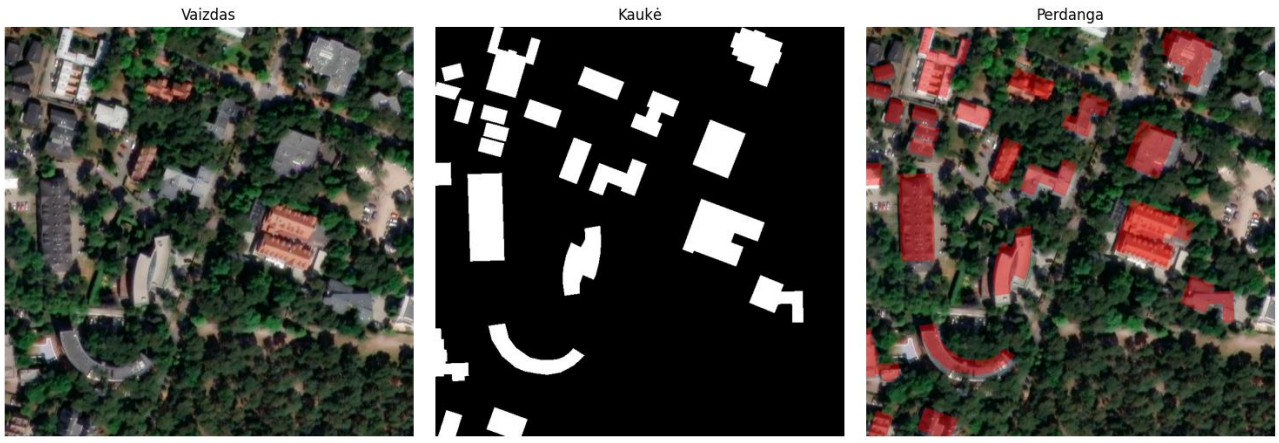
- naudojant *contextily* biblioteką, pagal *OSM* pastatų poligonų geografines ribas buvo atsisiunčiami atitinkami *Esri World Imagery* vaizdo fragmentai;
- *OSM* pastatų vektoriniai poligonai buvo transformuojami į tą pačią koordinačių sistemą kaip ir vaizdai (*EPSG:3857 – WGS 84/Pseudo-Mercator*);
- naudojant *rasterio* biblioteką, vektoriniai poligonai buvo rasterizuojami į dvejetaines kaukes, kurių matmenys, raiška ir geografinė padėtis tiksliai atitiko atsisiųstus vaizdo fragmentus. Kaukėse pastatų pikseliai žymimi viena reikšme (255), o fonas – kita (0);
- sugeneruotos didelės raiškos vaizdų ir kaukių poros buvo išsaugotos ir naudojant *pillow* biblioteką iškart padalintos į mažesnius, nepersidengiančius 256×256 pikselių fragmentus naudojant 12,5 % persidengimą.

Taip buvo sukurtas 3 kanalų (*RGB*) vaizdų Lietuvos rinkinys. Šis specializuotas rinkinys leidžia testuoti modelius Lietuvos kontekste ir analizuoti jų gebėjimą atpažinti pastatus, būdingus šiam regionui.

Reikia atkreipti dėmesį į iššūkius, kylančius dėl *OSM* duomenų prigimties ir jų derinimo su palydovine informacija. Kadangi *OSM* duomenys yra savanoriškai pildomi (angl. *crowdsourced*), jų tikslumas ir išsamumas gali skirtis priklausomai nuo vietovės. Dėl to galimi neatitikimai tarp OSM pastatų kontūrų ir faktinės padėties palydovinėse nuotraukose:

- neišsamumas: kai kurie realūs pastatai gali nebūti pažymėti *OSM* arba jų geometrija gali būti netiksli;
- laiko skirtumai: tai yra esminis neatitikimų šaltinis. *OSM* duomenys atspindi tam tikro momento būklę (kada objektas buvo sukurtas ar paskutinį kartą redaguotas), o palydovinės nuotraukos taip pat yra iš skirtingų, dažnai tiksliai nežinomų ar sunkiai nustatomų, datų. Užtikrinti, kad *OSM* duomenų ir palydovinio vaizdo fragmento datos sutaptų, naudojant standartinius įrankius yra praktiškai neįmanoma, todėl kaukės gali rodyti pastatus, kurių vaizde dar nėra, arba atvirkščiai;
- erdvinis nesutapimas: net ir esant teisingai koordinačių sistemai ir tariamam laiko sutapimui, gali pasitaikyti nedidelių *OSM* vektorių ir palydovinio vaizdo pikselių nesutapimų dėl abiejų šaltinių vidinių netikslumų ar skirtingų projekcijų interpretavimo.

Šie veiksniai, ypač neišvengiami laiko skirtumai ir erdviniai netikslumai, apsunkina modelių mokymąsi ir gali turėti įtakos galutiniam jų tikslumo vertinimui, lyginant su labiau prižiūrimais viešaisiais duomenų rinkiniais. Iš viso buvo sukurta 18000 256×256 matmenų vaizdų treniravimo imtis bei 163 įvairių matmenų (1024×1024, 768×1024, 1024×768) vaizdų testavimo imtis. Šiuo aprašytu metodu sukurto Lietuvos rinkinio pavyzdžio vaizdas ir kaukė pateikti 9 pav.



9 pav. Palangos vaizdas su kauke sukurtame Lietuvos duomenų rinkinyje

Šis pavyzdys (žr. 9 pav.) iliustruoja, kaip sukurta kaukė ne visiškai atitinka vaizde esančius pastatus. Viršutiniame kairiajame kampe (perdanga) matosi, kad du pastatai neturi atitinkamų kaukių. Taip pat, kaukėje esantys pažymėti pastatai ne iki galo atitinka vaizde esančių pastatų kontūrus. Šio duomenų rinkinio kokybė bus tiksliau įvertinta tyrimų dalyje treniruojant ir testuojant skirtingus modelius. Duomenų rinkinys sudarytas iš šių Lietuvos miestų ir miestelių: Kaunas, Vilnius, Klaipėda, Palanga, Birštonas, Šiauliai, Panevėžys, Alytus, Marijampolė, Utena, Druskininkai, Trakai, Anykščiai, Neringa, Telšiai, Kėdainiai, Molėtai, Ignalina, Zarasai ir Varėna.

Duomenų apdorojimo eiga

Kadangi skirtingi duomenų rinkiniai pateikiami nevienodais formatais ir struktūromis, buvo sukurta standartizuota duomenų apdorojimo eiga, skirta suvienodinti duomenis ir paruošti juos giliojo mokymosi modeliams. Toliau bus taikyti žingsniai duomenų apdorėjimui:

Kaukių paruošimas ir suvienodinimas: pirmasis žingsnis buvo užtikrinti, kad visos pastatų kaukės būtų pateiktos tinkamu rastriniu formatu, suderintu su atitinkamais vaizdais. Priklausomai nuo pradinio kaukių formato, taikyti skirtingi metodai:

- vektorinės kaukės (pvz., *GeoJSON, Shapefile*): naudojant *Python* programavimo kalbą ir *geopandas* ir *rasterio* bibliotekas, vektoriniai poligonai buvo konvertuojami į dvejetaines rastrines kaukes (*PNG* formato), kurių erdvinė raiška, aprėptis ir koordinačių sistema tiksliai atitiko susijusius vaizdus;
- rastrinės kaukės (pvz., *TIFF, PNG*): jei kaukės jau buvo pateiktos rastriniu formatu, jos buvo tiesiogiai nuskaitomos. Buvo tikrinama, ar jos yra dvejetainės (t. y., sudarytos iš dviejų reikšmių, žyminčių foną ir pastatus).

Po šio žingsnio visos kaukės buvo suvienodinto rastrinio dvejetainio formato.

Vaizdų dalijimas (angl. *tiling*) ir dydžio keitimas: kadangi modeliai paprastai priima fiksuoto dydžio įvesties vaizdus (pvz., 256×256 arba 512×512 pikseliai), o pradiniai duomenų rinkinių vaizdai yra įvairių dydžių (nuo 300×300 iki 5000×5000 pikseliai), reikėjo taikyti vaizdų dalijimo ir dydžio keitimo strategijas:

- pradinis metodas (taikytas *SpaceNet 2* duomenims): atliekant pradinius eksperimentus su *SpaceNet 2* duomenimis, kurių originalus dydis buvo 650×650 pikselių, buvo taikomas supaprastintas metodas: vaizdai pirmiausia sumažinami iki 512×512 pikselių, o tada padalinami į keturis nepersidengiančius 256×256 pikselių fragmentus. Nors tai paprasta įgyvendinti, toks metodas yra neoptimalus ir gali perskirti pastatus ties fragmentų ribomis;
- bendrasis metodas (persidengimas): vėlesniuose etapuose buvo taikomas lankstesnis ir dažnai efektyvesnis dalijimas su persidengimu (angl. *overlap*). Šiuo atveju didelis vaizdas dalijamas į mažesnius fragmentus (pvz., 512×512 ar 256×256 pikseliai) taip, kad gretimi fragmentai turėtų bendrą plotą (persidengtų). Žingsnis (angl. *stride*) tarp gretimų fragmentų pradžios yra mažesnis už paties fragmento dydį. Be to, tyrimai rodo, kad tam tikras persidengimo lygis (pvz., 12,5 %) gali reikšmingai pagerinti segmentavimo kokybę, lyginant su dalijimu be persidengimo, nes suteikia papildomo konteksto ties fragmentų ribomis ir sumažina prognozavimo artefaktus [103]. Šis metodas taip pat padeda išvengti objektų (pastatų) perkirtimo ties fragmentų ribomis, nes objektas, esantis ties riba, greičiausiai bus pilnai matomas bent viename iš persidengiančių fragmentų, suteikiant modeliui pilnesnį kontekstą. Dalijimas su persidengimu taip pat būtinas norint efektyviai apdoroti labai didelius vaizdus ribotos grafinės atminties (*GPU VRAM*) sąlygomis.

Visais atvejais, atitinkamai pagal vaizdų dalijimą buvo apdorojamos (dalijamos arba pritaikomos) ir paruoštos dvejetainės kaukės. Visi gauti vaizdo fragmentai (ir atitinkamos kaukės) buvo suvienodinto dydžio (256×256 pikseliai), kad atitiktų pasirinktų modelių architektūrų įvesties reikalavimus.

Kanalų atranka: buvo užtikrinama, kad visi vaizdai turėtų 3 spalvų kanalus (*RGB*), nes siekiama sukurti praktišką modelį ir daugiaspektriniai vaizdai nėra visada prieinami. Jei pradiniai duomenys turėjo daugiau kanalų (pvz., 8 kanalų daugiaspektriniai *SpaceNet* vaizdai), buvo atrenkami *RGB* kanalai.

Normalizavimas: vaizdų pikselių reikšmės buvo normalizuotos, siekiant paruošti jas giliojo mokymosi modeliui. Procesas vyko dviem pagrindiniais etapais:

- mastelio keitimas (angl. *rescaling*): iš pradžių, vaizdų pikselių reikšmės (paprastai 0-255 intervale) buvo pakeistos į [0, 1] intervalą, dalijant jas iš 255;
- specifinis modelio fundamento (angl. *backbone*) normalizavimas: po pirminio mastelio keitimo, buvo taikoma papildoma, konkrečiam modelio baziniui modeliui (angl. *backbone*) būdinga normalizavimo funkcija. Ši funkcija gaunama iš modelį kuriančios bibliotekos ir paprastai įgyvendina standartizavimą pagal *ImageNet* duomenų rinkinio statistiką – atimamas vidurkis ir dalijama iš standartinio nuokrypio kiekvienam kanalui. Normalizavimas yra standartinis žingsnis, padedantis pagreitinti ir stabilizuoti giliojo mokymosi modelių mokymą, ypač naudojant iš anksto apmokytus (angl. *pre-trained*) modelių pagrindus.

Duomenų padalinimas ir augmentacija: galiausiai, visas paruoštas vaizdų ir kaukių porų rinkinys buvo padalintas į mokymo, validavimo ir testavimo aibes. Taip pat apmokant modelį buvo visada naudota augmentacija naudojant iš anksto apmokytus parametrus iš skirtingų duomenų bazių (pvz., *ImageNet*).

2.4. Eksperimentinė aplinka

Šiame darbe aprašyti eksperimentai ir modelių mokymas buvo vykdomi naudojant du skirtingus skaičiavimo resursus. Dalis darbo atlikta asmeniniu nešiojamuoju kompiuteriu, turinčiu *Intel Core i7-14700HX* procesorių, 32 *GB* operatyviosios atminties (*RAM*) ir *NVIDIA GeForce RTX 4070 Laptop* grafinį procesorių. Šiame kompiuteryje naudota *Windows* 11 operacinė sistema. Kita, skaičiavimams imlesnė, eksperimentų ir mokymo dalis buvo atlikta *Google Colaboratory* (*Colab Pro*) debesijos platformoje, naudojant *NVIDIA A100* grafinį procesorių ir *Linux* operacinę sistemą.

Programinė eksperimentų aplinka abiem atvejais buvo pagrįsta *Python* programavimo kalba (naudotos 3.11.12 *Google Colab* ir 3.10.17 asmeniniame kompiuteryje versijos) bei *PyTorch* giliojo mokymosi karkasu (atitinkamai *2.6.0+cu124 Google Colab* ir *2.6.0+cu118* asmeniniame kompiuteryje versijos) ir *Cuda*[21] bei *cuDNN*[22]. Asmeninio kompiuterio aplinkoje, dirbant su *RTX 4070* vaizdo plokšte, naudotos *CUDA* 11.8.0 ir *cuDNN* 8.9.2.26 įrankių versijos. *Google Colab* aplinkoje naudotos *CUDA* 12.4 ir *cuDNN* 9.1.0 versijos.

2.5. Metodologija

Tyrimo metodologija pagrįsta giliojo mokymosi principais, taikant dvejetainį semantinį segmentavimą pastatų pėdsakams išskirti.

[21] https://developer.nvidia.com/cuda-toolkit

[22] https://developer.nvidia.com/cudnn

2.5.1. Modelių architektūros

Naudojamos kodavimo-dekodavimo architektūros, tokios kaip *U-Net* [42] ir *PAN* [19], kurios yra plačiai taikomos kompiuterinės regos segmentavimo uždaviniams. Šios architektūros susideda iš kodavimo dalies, kuri ištraukia požymius iš įvesties vaizdo palaipsniui mažindama erdvinę raišką, ir dekodavimo dalies, kuri palaipsniui atkuria erdvinę raišką iki pradinio vaizdo dydžio, generuodama pikselių lygio segmentavimo kaukę. Jų efektyvumas remiasi gebėjimu sujungti aukšto lygio semantinius požymius su žemo lygio erdviniais požymiais per praleidimo jungtis (angl. *skip connections*). Dauguma testuotų architektūrų buvo panaudotos iš *segmentation-models-pytorch* bibliotekos. Išimtis buvo *Attention U-Net* [41] architektūra, kuri buvo įgyvendinta tiesiogiai, naudojant *PyTorch* karkaso standartinius modulius.

2.5.2. Komponentų testavimas

Siekiant optimizuoti modelių veikimą, taikytas skirtingų modelio komponentų vertinimas. Buvo atliekami išsamūs eksperimentai, keičiant vieną komponentą ir kitus paliekant nekintamus. Pavyzdžiui, buvo testuojama apie 140 skirtingų kodavimo dalių (bazinių modelių, angl. *backbones*), 198 skirtingų bazinių modelių ir iš anksto apmokytų svorių kombinacijų *U-Net* [14] architektūroje, naudojant *timm* bibliotekos modelius su *imagenet* bei kitais svoriais, tuo tarpu kiti parametrai (pvz., nuostolių funkcija, aktyvacijos funkcija, optimizatorius, mokymosi greitis) buvo palikti pastovūs. Analogiški testai buvo atlikti ir kitoms architektūroms (pvz., *PAN* [19], *FPN* [102], *DeepLabV3+* [18]), nuostolių funkcijoms (*Jaccard, Dice, Focal* ir kt.) bei optimizatoriams (pvz. *AdamW* [93]). Šis metodas leido identifikuoti optimaliausias komponentų kombinacijas konkrečiam uždaviniui.

2.5.3. Modelių konstravimas ir eksperimentavimas įvairiuose duomenų rinkiniuose

Identifikavus perspektyviausius modelio komponentus – kodavimo bazinius modelius, architektūras, nuostolių funkcijas ir optimizatorius – buvo pereita prie segmentavimo modelių konstravimo naudojant šiuos komponentus. Šie modeliai, sudaryti iš geriausiai ankstesniame etape įvertintų komponentų derinių, buvo apmokomi ir jų našumas tiriamas naudojant platų spektrą skirtingų duomenų rinkinių. Eksperimentai apėmė tiek plačiai naudojamus tarptautinius etaloninius duomenų rinkinius (pvz., *SpaceNet, Massachusetts Buildings, WHU*), tiek specialiai šiam darbui sukurtą Lietuvos teritorijos vaizdų duomenų rinkinį. Toks metodas leido įvertinti sukonstruotų modelių tikslumą, gebėjimą adaptuotis prie skirtingų geografinių bei vaizdo charakteristikų.

2.5.4. Nauja modelių sujungimo (ansambliavimo) strategija

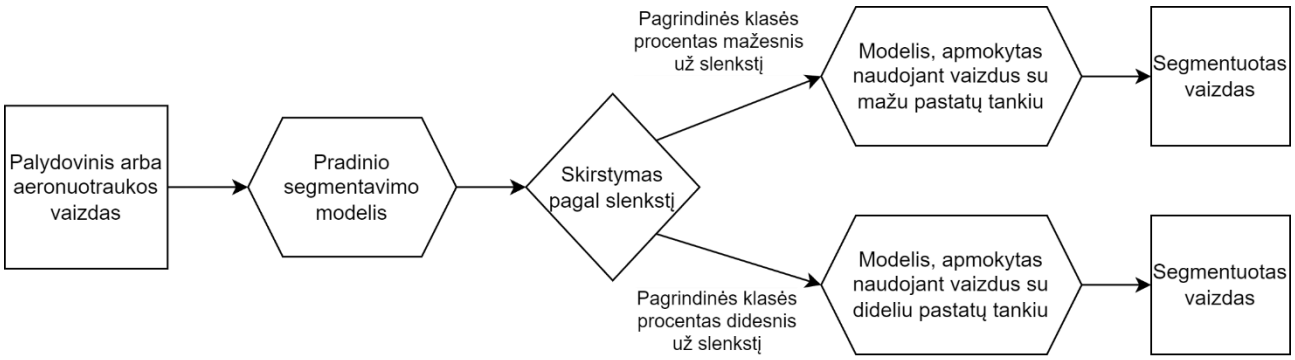
Darbe pasiūlyta ir eksperimentiškai įvertinta nauja modelių ansambliavimo (sujungimo) (angl. *ensembling*) strategija, pagrįsta pastatų tankiu vaizduose. Strategijos esmė:

1. apmokomi trys atskiri modeliai:
    a. modelis 1 (bendras): apmokytas naudojant visą duomenų rinkinį;
    b. modelis 2 (retas): apmokytas naudojant tik tuos vaizdo fragmentus, kuriuose pastatų pikselių procentas yra mažas (pvz., mažiau nei 25 % pagal kaukę);
    c. modelis 3 (tankus): apmokytas naudojant tik tuos vaizdo fragmentus, kuriuose pastatų pikselių procentas yra didelis (pvz., daugiau nei 25 % pagal kaukę).
2. išvedimo (angl. *inference*) procesas:
    a. pirmiausia, segmentavimas atliekamas naudojant bendrą modelį (modelį 1);

b. apskaičiuojamas prognozuotų pastatų pikselių procentas bendro modelio gautoje kaukėje;
c. pagal nustatytą slenkstį (pvz., 25 %), parenkamas arba retas modelis (modelis 2), arba tankus modelis (modelis 3);
d. galutinis segmentavimas atliekamas naudojant pasirinktą specializuotą modelį (modelį 2 arba modelį 3).

3. hipotezė: tikimasi, kad ši tankiu pagrįsta specializuotų modelių parinkimo ir optimizavimo strategija leis pasiekti aukštesnį segmentavimo tikslumą, palyginti su vieno bendro modelio naudojimu, nes specializuoti modeliai geriau prisitaiko prie skirtingo tankio sričių specifikos ir klasių disbalanso problemų.

Šio modelių sujungimo diagrama yra pateikta 10 pav.



10 pav. Modelių sujungimo strategijos pagal pastatų tankumą vaizde veikimo diagrama

2.5.5. Vertinimas

Modelių veikimo kokybei ir palyginimui įvertinti naudotos kelios standartinės semantinio segmentavimo bei dvejetainio klasifikavimo metrikos, apskaičiuotos testavimo duomenų aibėje.

Pagrindinės segmentavimo metrikos

Susikirtimo padalinto iš sąjungos vidurkis (angl. *Mean Intersection over Union,* m*IoU*): vidutinė *IoU* reikšmė per atskirus testo vaizdus. Tyrimų ir jų rezultatų dalyje (3 skyrius) ši metrika bus vadinama tiesiog *IOU*. Tai viena iš plačiausiai naudojamų ir svarbiausių metrikų semantiniam segmentavimui. Ji matuoja prognozuotos kaukės ir tikrosios (etaloninės) kaukės persidengimo laipsnį. Aukštesnė *IoU* reikšmė rodo, kad modelis tiksliau identifikuoja pastato plotą ir formą. Ji yra intuityvi ir griežta, nes baudžia tiek už klaidingai teigiamus, tiek už klaidingai neigiamus pikselius. Toliau bus apžvelgiama formulė (1) [104].

$$mIoU = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i+FP_i+FN_i}; \quad (1)$$

čia *TP* (angl. *True Positives*) – tikrai teigiamų pikselių skaičius (pastato pikseliai, teisingai klasifikuoti kaip pastatas), *FP* (angl. *False Positives*) – klaidingai teigiamų pikselių skaičius (fono pikseliai, klaidingai klasifikuoti kaip pastatas), *FN* (angl. *False Negatives*) – klaidingai neigiamų pikselių skaičius (pastato pikseliai, klaidingai klasifikuoti kaip fonas), *N* – bendras vaizdų skaičius testavimo duomenų rinkinyje, *i* – individualaus vaizdo indeksas ($i = 1,\ldots,N$).

Dice koeficiento vidurkis (angl. *Mean Dice Coefficient*): ši metrika taip pat matuoja prognozuotos ir tikrosios kaukės panašumą (persidengimą) ir yra glaudžiai susijusi su *IoU*. *Dice* koeficientas yra tapatus *F1* Įverčiui (angl. *F1 Score*) ir yra vidurkis tarp preciziškumo ir jautrumo. Jis ypač naudingas esant klasių disbalansui (pvz., kai pastatai užima mažą vaizdo dalį), nes įvertina teisingai klasifikuotų teigiamų pikselių dalį atsižvelgiant į klaidingus klasifikavimus. Tyrimų ir jų rezultatų dalyje (3 skyrius) ši metrika taip pat bus vadinama *Dice*. Toliau bus apžvelgiama formulė (2)[23].

$$mDice = \frac{1}{N}\sum_{i=1}^{N}\frac{2\cdot TP_i}{2\cdot TP_i+FP_i+FN_i}; \quad (2)$$

čia *N* – bendras vaizdų skaičius testavimo duomenų rinkinyje, *i* – individualaus vaizdo indeksas (*i* = 1,…,*N*), *TP* – tikrai teigiamų pikselių skaičius (pastato pikseliai, teisingai klasifikuoti kaip pastatas), *FP* – klaidingai teigiamų pikselių skaičius (fono pikseliai, klaidingai klasifikuoti kaip pastatas), FN – klaidingai neigiamų pikselių skaičius (pastato pikseliai, klaidingai klasifikuoti kaip fonas).

Hausdorff atstumo vidurkis (angl. Mean *Hausdorff Distance*): skirtingai nuo *IoU* ar *Dice*, kurie vertina regionų persidengimą, *Hausdorff* atstumas matuoja kontūrų panašumą. Jis parodo didžiausią atstumą nuo vienos kontūro taško iki artimiausio taško kitame kontūre. Vidutinis *Hausdorff* atstumas per duomenų rinkinį parodo, kiek vidutiniškai prognozuoti pastatų kontūrai skiriasi nuo etaloninių blogiausiu atveju. Mažesnė vidutinė reikšmė rodo geresnį bendrą kontūrų atitikimą. Tai ypač svarbu, kai reikalingas tikslus pastatų ribų apibrėžimas. Tyrimų ir jų rezultatų dalyje taip pat bus vadinamas tiesiog *Hausdorff* arba *Hausd*. Toliau bus apžvelgiama formulės[24]. Pirma bus apžvelgiama formulė vienam vaizdui (3).

$$H(A_i, B_i) = max\left(sup_{a\in A_i} inf_{b\in B_i} d(a,b), sup_{a\in B_i} inf_{b\in A_i} d(a,b)\right); \quad (3)$$

dabar bus apžvelgiamas vidurkis per daug vaizdų (4):

$$mHaussdorff = \sum_{i=1}^{N} H(A_i, B_i); \quad (4)$$

čia *N* – bendras vaizdų skaičius testavimo duomenų rinkinyje, *i* – individualaus vaizdo indeksas (*i* = 1,…,*N*), *A, B* – taškų aibės, apibrėžiančios prognozuotą ir etaloninį kontūrus, *a* – taškas aibėje *A*, *b* – taškas aibėje *B*, *d(a,b)* – atstumas (pvz., *Euklido*) tarp taškų *a* ir *b*.

Įvertinti modelio tikslumą taip pat papildomai bus naudojama *ROC AUC* (plotas po *ROC* kreive): *ROC* kreivė vaizduoja modelio gebėjimą atskirti teigiamą (pastatas) ir neigiamą (fonas) klases esant įvairiems klasifikavimo slenksčiams. Ši metrika skaičiuojama vieną kartą visam duomenų rinkiniui, surenkant visas pikselių prognozes (tikimybes) ir tikrąsias klases. Plotas po šia kreive (*AUC*) yra viena skaitinė reikšmė nuo 0 iki 1. *AUC* parodo modelio klasifikavimo kokybę nepriklausomai nuo pasirinkto sprendimo slenksčio. Kaip teigiama *Google Machine Learning Crash Course*[25], *ROC* kreivė sudaroma apskaičiuojant tikrųjų teigiamų atvejų dažnį (*TPR*) ir klaidingų teigiamų atvejų dažnį (*FPR*) esant kiekvienam įmanomam slenksčiui. *AUC* interpretuojamas kaip tikimybė, kad modelis atsitiktinai parinktą teigiamą pavyzdį įvertins aukščiau (kaip labiau tikėtiną teigiamą) nei atsitiktinai parinktą neigiamą pavyzdį. *AUC* = 1 reiškia idealų klasifikatorių, o *AUC* = 0,5 – atsitiktinį spėjimą. Formulė (5).

[23] https://torchmetrics.readthedocs.io/en/v0.10.2/classification/dice.html

[24] https://cgm.cs.mcgill.ca/~godfried/teaching/cg-projects/98/normand/main.html

[25] https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

$$\widehat{AUC} = \sum_{k=1}^{N-1}(FPR_{k+1} - FPR_k)\frac{TPR_{k+1}+TPR_k}{2}; \quad (5)$$

čia N – bendras taškų skaičius (atitinkantis skirtingas slenksčio reikšmes), iš kurių sudaryta *ROC* kreivė, k – indeksas, žymintis konkretų tašką (arba intervalą tarp taškų) *ROC* kreivėje, *FPR* ir *TPR* – atitinkamai klaidingų teigiamų atvejų dažnis ir tikrųjų teigiamų atvejų dažnis.

Papildomo metrikos

Pikselių tikslumas (angl. *Pixel Accuracy*): bendras teisingai klasifikuotų pikselių procentas. Ši metrika gali būti klaidinanti esant dideliam klasių disbalansui. Modelis, daugumą pikselių priskyręs dominuojančiai fono klasei, gali turėti aukštą pikselių tikslumą, net jei prastai segmentuoja pastatus. Tyrimų ir jų rezultatų dalyje taip pat bus vadinamas tiesiog *PA*. Formulė (6) [105].

$$mPA = \frac{\sum_{i=1}^{N}(TP_i+TN_i)}{\sum_{i=1}^{N}(TP_i+TN_i+FP_i+FN_i)}; \quad (6)$$

čia *N* – bendras vaizdų skaičius testavimo duomenų rinkinyje, *i* – individualaus vaizdo indeksas (*i* = 1,…,*N*), *TP* – tikrai teigiamų pikselių skaičius, *TN* – tikrai neigiamų pikselių skaičius, *FP* – klaidingai teigiamų pikselių skaičius, *FN* – klaidingai neigiamų pikselių skaičius.

Preciziškumas (angl. *Precision*): dalis teisingai prognozuotų pastato klasės pikselių iš visų pikselių, kuriuos modelis priskyrė pastato klasei. Ši metrika vertina tik teigiamų prognozių tikslumą (kiek iš modelio identifikuotų pastatų pikselių iš tiesų yra pastatai), tačiau neatsižvelgia į tai, kiek pastatų pikselių modelis praleido (klaidingai neigiamų atvejų). Dėl to modelis su aukštu preciziškumu gali praleisti daug realių pastatų. Tyrimų ir jų rezultatų dalyje taip pat bus vadinama *Prec*. Formulė (7) [106].

$$mPrecision = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N}(TP_i+FP_i)}; \quad (7)$$

čia *i* – individualaus vaizdo indeksas, *TP* – tikrai teigiamų pikselių skaičius, *FP* – klaidingai teigiamų pikselių skaičius, N – bendras vaizdų skaičius testavimo duomenų rinkinyje.

Jautrumas (angl. *Recall*): dalis teisingai prognozuotų pastato klasės pikselių iš visų faktinių pastato klasės pikselių. Ši metrika rodo kokią dalį visų faktinių pastatų pikselių modelis sugebėjo teisingai atrasti, bet nebaudžia už klaidingai teigiamus atvejus (kai fonas klaidingai priskiriamas pastatams). Modelis su aukštu jautrumu gali identifikuoti daugumą pastatų, bet kartu klaidingai pažymėti ir daug fono sričių kaip pastatus. Tyrimų ir jų rezultatų dalyje taip pat bus vadinama *Rec*. Formulė (8) [106].

$$mRecall = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N}(TP_i+FN_i)}; \quad (8)$$

čia *i* – individualaus vaizdo indeksas, *N* – bendras vaizdų skaičius testavimo duomenų rinkinyje, *TP* – tikrai teigiamų pikselių skaičius, *FN* – bendras klaidingai neigiamų pikselių skaičius visame duomenų rinkinyje.

Mokymo laikas (sekundėmis): laikas, per kurį modelis buvo apmokytas nurodytą epochų skaičių.

Šios metrikos buvo skaičiuojamos naudojant specialiai tam sukurtas bibliotekas pvz., *scikit-learn*.

Dice koeficientas ir *IoU* yra laikomi pagrindinėmis segmentavimo metrikomis, nes jie abu atsižvelgia tiek į klaidingai teigiamus, tiek į klaidingai neigiamus atvejus, suteikdami labiau subalansuotą regionų persidengimo įvertį, ypač esant klasių disbalansui. *ROC AUC* įvertina bendrą modelio gebėjimą atskirti klases nepriklausomai nuo pasirinkto slenksčio, o *Hausdorff* atstumas yra itin svarbus vertinant kontūrų tikslumą, kas yra esminis aspektas uždavinyje. Preciziškumas ir jautrumas yra *Dice* koeficiento (arba *F1* įverčio, kuris yra jam tapatus) komponentai, todėl *Dice* jau integruoja jų balansą.

3. Tyrimai ir jų rezultatai

Šiame skyriuje projekto metu atlikti eksperimentiniai tyrimai ir gauti rezultatai, siekiant pagrindinio darbo tikslo – sukurti ir įvertinti giliojo mokymosi modelius, skirtus automatizuotam pastatų kontūrų išskyrimui, ypatingą dėmesį skiriant komponentų analizei ir metodų pritaikomumui. Skyriuje nuosekliai nagrinėjami atliktų testų rezultatai, lyginant skirtingus modelių kodavimo bazinius modelius, nuostolių funkcijas, optimizatorius ir architektūras. Toliau vertinama pasiūlyta tankiu pagrįsta modelių sujungimo (ansambliavimo) strategija, lyginant jos efektyvumą su vieno modelio metodu.

Reikia pažymėti, kad didžioji dalis lyginamųjų komponentų testų (pvz., kodavimo bazinių modelių, nuostolių funkcijų, optimizatorių), siekiant efektyvumo ir rezultatų palyginamumo, buvo atlikta naudojant standartizuotą ir plačiai naudojamą *SpaceNet 2* duomenų rinkinio Chartumo miesto poaibį, kuris aprašytas 2.3 skyriuje. Modelių veikimo kokybė šiame skyriuje bus vertinama ir lyginama remiantis pagrindinėmis 2.5.5 skyriuje apibrėžtomis metrikomis, tokiomis kaip vidutinė susikirtimo padalinto iš sąjungos reikšmė (*Mean IoU*) ir *Dice* koeficientas (angl. *Dice Coefficient*), o tam tikrais atvejais bus atsižvelgiama ir į mokymosi laiką, vertinant skaičiavimo resursų poreikį.

3.1. Modelio komponentų testavimas

Giliojo mokymosi modelių efektyvumas labai priklauso nuo pasirinktos architektūros ir jos hiperparametrų. Siekiant rasti optimalią konfigūraciją pastatų pėdsakų segmentavimo uždaviniui, buvo taikomas skirtingų komponentų testavimas. Jo esmė – nuosekliai keisti vieną modelio komponentą (pvz., kodavimo bazinį modelį, nuostolių funkciją, optimizatorių ar visą architektūrą), tuo tarpu kitus svarbius parametrus (pvz., mokymo epochų skaičių, mokymosi greitį, vaizdų dydį) paliekant pastovius. Toks metodas leidžia izoliuoti ir įvertinti kiekvieno komponento įtaką galutiniam modelio tikslumui bei efektyvumui konkrečiame duomenų rinkinyje. Buvo pasirinkta naudoti *SpaceNet 2* Chartumo poaibį šiam tyrimui atlikti. Šis rinkinys buvo dalinamas į 80 % treniravimo duomenis, 15 % validacijos ir 5 % testavimo. Visi tyrimai buvo atlikti *naudojant Google Colab* su *A100* grafiniu procesoriumi.

Kodavimo bazinių modelių (angl. *Backbones*) palyginimas

Šiame etape buvo atliktas tyrimas, siekiant nustatyti, kuris bazinis modelis, naudojamas požymių ištraukimui, duoda geriausius rezultatus pastatų pėdsakų segmentavimo užduočiai. Eksperimento metu buvo laikomasi testavimo principo: keičiami buvo tik kodavimo baziniai modeliai ir iš anksto apmokyti svoriai iš įvairių vaizdų duomenų rinkinių, o visi kiti svarbūs modelio ir mokymo parametrai buvo fiksuoti. Kaip buvo minėta anksčiau, šioje tyrimo dalyje buvo naudotas S*paceNet 2* Chartumo miesto duomenų poaibis.

Statiniai parametrai, naudoti kodavimo bazinių modelių tyrime:

- architektūra: *U-Net* [14];
- nuostolių funkcija: sujungta *Focal* (γ = 2,0) (50 %) ir *Jaccard* (*IOU*) (50 %) nuostolių funkcija;
- optimizatorius: *AdamW* [93] su 0,00001 parametrų slopinimu (angl. *weight decay*);
- mokymo epochų skaičius: 12;
- mokymosi greitis: 0,0001;

- vienu metu paduodamo duomenų rinkinio dydis (angl. *Batch size*): 32;
- įvesties vaizdo dydis: 256×256 pikseliai.

Kiekvieno eksperimento metu, laikantis šių pastovių parametrų, buvo keičiamas tik *U-Net* [14] architektūros bazinis modelis (koduotuvas), naudojant platų spektrą modelių iš *segmentation-models-pytorch* ir *timm* bibliotekų. Iš viso buvo ištirta 140 skirtingų kodavimo bazinių modelių bei 198 skirtingų bazinių modelių ir iš anksto apmokytų svorių kombinacijų. 15 geriausių bazinių modelių ir iš anksto apmokytų svorių kombinacijų pateiktos 2 lentelėje.

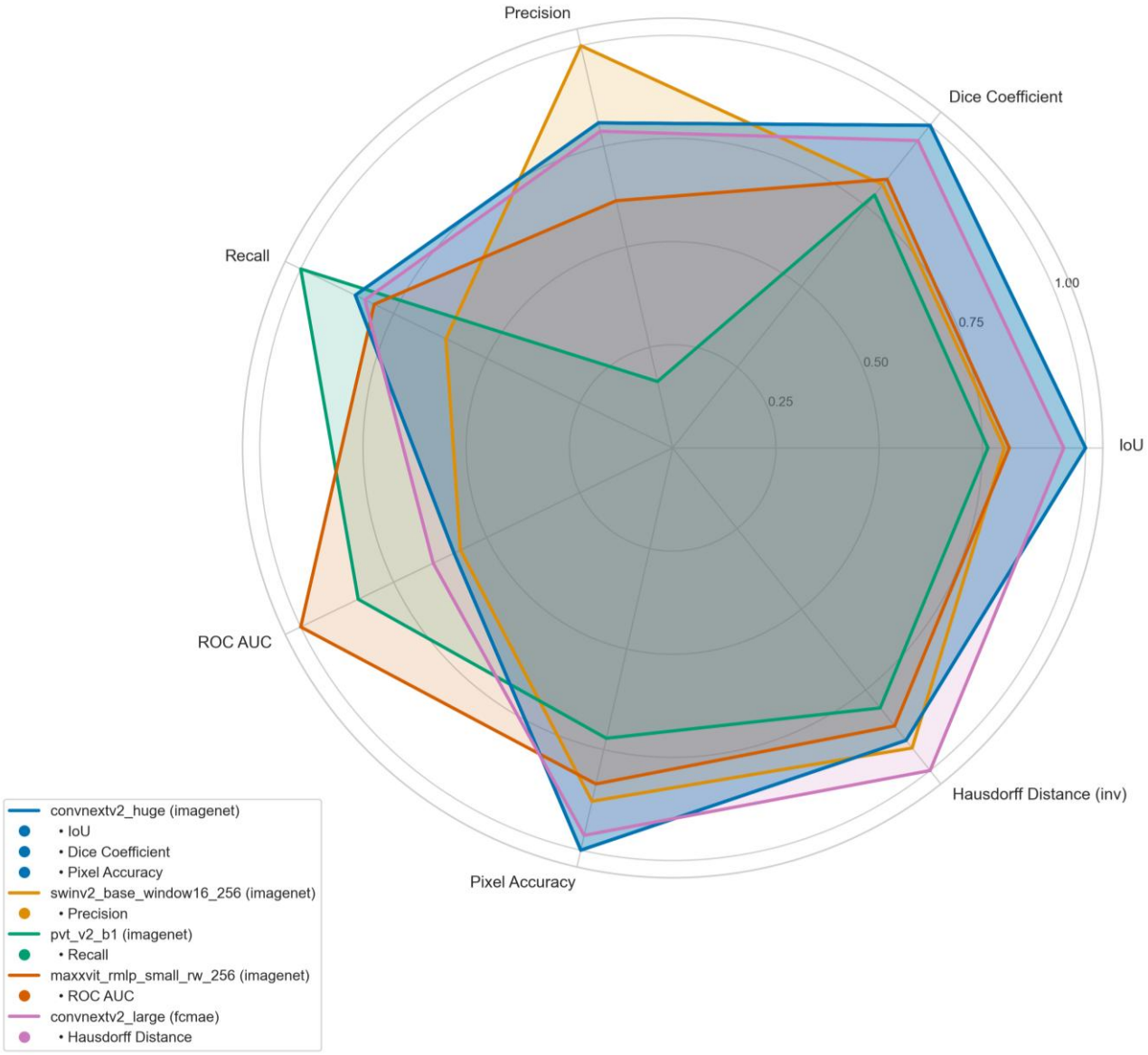
2 lentelė. Pagal *IoU* metriką 15 geriausiai pasirodžiusių bazinių modelių ir svorių kombinacijų

Bazinis modelis	**Duomenų rinkinys ant kurio buvo apmokytas**	**Laikas (s)**	**IoU**	**PA**	**Dice**	**Prec**	**Rec**	**ROC AUC**	**Hausd**
convnextv2_huge	imagenet	616,301	0,772	0,92	0,868	0,87	0,88	0,95	39,461
convnextv2_large	imagenet	377,212	0,767	0,918	0,865	0,868	0,877	0,956	39,114
convnext_xxlarge	imagenet	617,641	0,767	0,918	0,864	0,873	0,869	0,948	38,635
convnextv2_large	fcmae_ft_in22k_in1k	444,374	0,766	0,918	0,864	0,871	0,872	0,957	39,276
convnextv2_large	fcmae	436,222	0,766	0,918	0,864	0,869	0,876	0,953	37,393
focalnet_huge_fl4	imagenet	766,007	0,765	0,916	0,863	0,854	0,891	0,96	41,349
focalnet_huge_fl3	imagenet	520,329	0,762	0,917	0,862	0,878	0,862	0,951	40,597
convnextv2_base	imagenet	282,941	0,762	0,916	0,862	0,863	0,877	0,953	41,542
focalnet_xlarge_fl4	imagenet	575,479	0,762	0,917	0,861	0,871	0,869	0,953	43,171
resnext101_32x16d	ssl	385,653	0,762	0,916	0,861	0,865	0,874	0,959	43,223
regnety_640	imagenet	405,392	0,759	0,916	0,86	0,872	0,866	0,954	39,004
resnext101_32x8d	instagram	256,869	0,759	0,914	0,859	0,857	0,879	0,95	39,966
resnext101_32x8d	swsl	275,214	0,758	0,914	0,858	0,856	0,88	0,955	43,107
maxxvitv2_nano_rw_256	imagenet	194,073	0,757	0,915	0,858	0,862	0,874	0,961	42,341
pvt_v2_b2_li	imagenet	203,732	0,757	0,914	0,858	0,862	0,873	0,962	41,398

Kaip matyti 2 lentelėje, aukščiausius rezultatus pagal *Mean IoU* metriką demonstravo modernios konvoliucinių tinklų architektūros, ypač *ConvNeXt V2 [28]*, *FocalNet* [1] ir *ConvNeXt* [3] šeimų modeliai. Geriausiai pasirodė *convnextv2_huge* bazinis modelis, pasiekęs 0,772 *IoU*. Taip pat tarp top 10 pateko ir *ResNeXt* [26] bei *RegNetY* [31] šeimų atstovai, taip pat mažesnis *MaxxVitV2* [29](*maxxvitv2_nano_rw_256*) modelis, kuris pasižymėjo santykinai trumpu mokymo laiku (194,073 s) ir geru tikslumu (0,757 *IoU*). Tai rodo, kad naujesnės architektūros, tokios kaip *ConvNeXt* ir *FocalNet*, pasižymi dideliu potencialu šiai užduočiai, tačiau renkantis pagrindą svarbu atsižvelgti ir į modelio dydžio bei mokymo laiko kompromisą.

Pilna visų 198 skirtingų bazinių modelių ir iš anksto apmokytų svorių kombinacijų rezultatų lentelė pateikiama 3 priede.

Toliu bus palyginami geriausiai kiekvienoje tikslumo metrikoje pasirodę modeliai. Šių modelių tarpusavio palyginimo spindulinė diagrama pateikta 11 pav.



11 pav. Pagal skirtingas metrikas rastų bazinių modelių tarpusavio palyginimas

Pagal spindulinę diagramą (žr. 11 pav.), kurioje lyginamas kelių bazinių modelių ir apmokytų svorių kombinacijų tikslumas, matoma, kad *ConvNeXt V2* [28] architektūros, konkrečiai *convnextv2_huge* su *imagenet* ir *convnextv2_large* su *fcmae*, rodo itin aukštą bendrą tikslumą. Šie modeliai ypač pasižymi aukštais *Dice* koeficiento, *IoU* bei pikselių tikslumo rodikliais. Nors *swinv2_base_window16_256* [33] su *imagenet* taip pat konkurencingas, ypač preciziškumo (angl. *Precision*) srityje, jo bendras tikslumas diagramoje atrodo šiek tiek mažesnis. Galima daryti prielaidą, kad *ConvNeXt V2* baziniai modeliai yra vieni efektyviausių pasirinkimų šiai užduočiai. Taip pat reikia paminėti, kad diagramoje reikšmės yra normuotos (0-1 skalėje, kur 1 – geriausias rezultatas, o *Hausdorff* atstumas normuotas atvirkščiai). Tikslesnė informacija yra pateikta 3 lentelėje.

3 lentelė. Pagal skirtingas metrikas rastų bazinių modelių tarpusavio palyginimo lentelė

Geriausios metrikos	Bazinis modelis	Duomenų rinkinys ant kurio buvo apmokytas	Laikas (s)	IoU	PA	Dice	Prec	Recall	ROC AUC	Hausd
IoU, Dice, PA	convnextv2_huge	imagenet	616,301	0,772	0,920	0,868	0,870	0,880	0,950	39,461
Precision	swinv2_base_window16_256	imagenet	231,368	0,749	0,914	0,853	0,884	0,843	0,950	38,938
Recall	pvt_v2_b1	imagenet	196,9	0,744	0,907	0,850	0,824	0,903	0,960	41,684
ROC AUC	maxxvit_rmlp_small_rw_256	imagenet	214,092	0,750	0,912	0,854	0,857	0,873	0,966	40,445
Hausd	convnextv2_large	fcmae	436,222	0,766	0,918	0,864	0,869	0,876	0,953	37,393

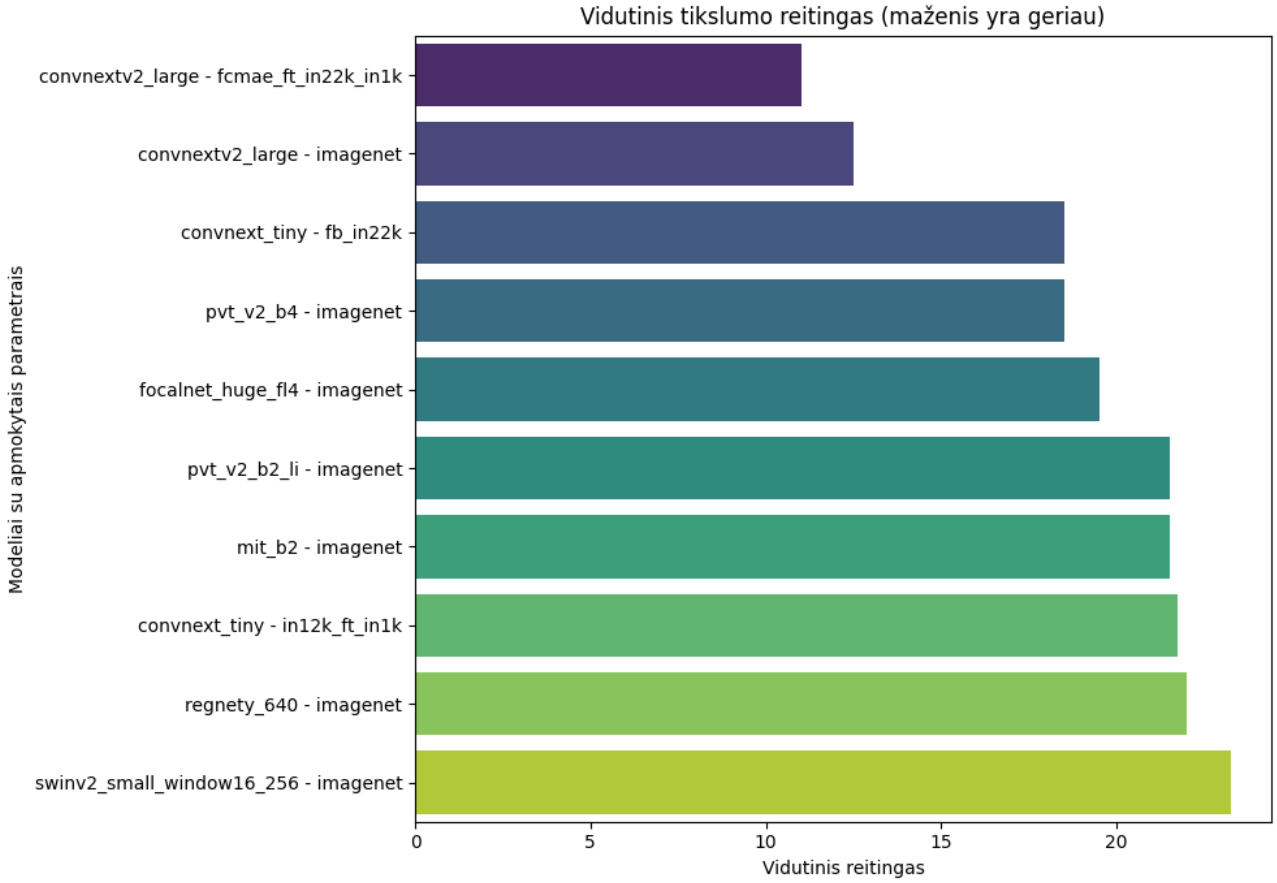
Toliau bus apžvelgtas geriausių bazinių modelių ir apmokytų parametrų kombinacijų vidutinis reitingas naudojant keturias esmines segmentavimo metrikas (*IoU, Dice* koeficientas, *ROC AUC*, *Hausdorff* atstumas). Bus naudojamos ne metrikų vertės, bet vieta kurią modelis užėmė vertinant metrikų rezultatus nuo geriausio iki prasčiausio. Visų minėtų metrikų vietos bus sudedamos ir padalinamos iš metrikų skaičiaus. Tokiu būdu siekiama įvertinti bazinių modelių ir apmokytų parametrų kombinacijų bendrą tikslumą. Kitos metrikos taip pat bus pateiktos, bet nebus naudojamos vertinime. Ši informacija yra pateikta 4 lentelėje.

4 lentelė. 10 geriausių bazinių modelių ir apmokytų svorių kombinacijų vidutinis reitingas

Bazinis modelis ir apmokyti parametrai	Vidutinis	Geriausia metrika	IoU	Dice	ROC AUC	Hausd	PA	Prec	Rec
convnextv2_large (fcmae_ft_in22k_in1k)	11	IoU (4)	4	4	25	11	4	8	52
convnextv2_large (imagenet)	12,5	IoU (2)	2	2	37	9	3	18	35
convnext_tiny (fb_in22k)	18,5	Recall (11)	18	20	21	15	35	135	11
pvt_v2_b4 (imagenet)	18,5	ROC AUC (2)	23	24	2	25	24	56	44
focalnet_huge_fl4 (imagenet)	19,5	IoU (6)	6	6	14	52	8	77	10
pvt_v2_b2_li (imagenet)	21,5	ROC AUC (5)	15	13	5	53	14	42	50
mit_b2 (imagenet)	21,5	ROC AUC (3)	17	17	3	49	37	141	7
convnext_tiny (in12k_ft_in1k)	21,75	Hausdorff (4)	16	16	51	4	13	38	54
regnety_640 (imagenet)	22	Precision (7)	11	11	58	8	9	7	78
swinv2_small_window16_256 (imagenet)	23,25	Hausdorff (3)	40	40	10	3	28	37	76

Analizuojant lentelėje (žr. 4 lentelę) pateiktus našumo reitingus matyti, kad *convnextv2_large* [28] tipai, ypač tie, kurie naudoja *fcmae* ir *imagenet* apmokytus parametrus, pasiekia žemiausius vidutinius reitingus – tai rodo jų geriausią bendrą tikslumą tarp vertintų geriausių modelių. Šie modeliai nuolat užima aukštas vietas pagal svarbiausias segmentavimo metrikas, įskaitant *IoU, Dice* koeficientą. Nors kiti modeliai pirmauja pagal atskiras specifines metrikas, pavyzdžiui, *ROC AUC*, būtent

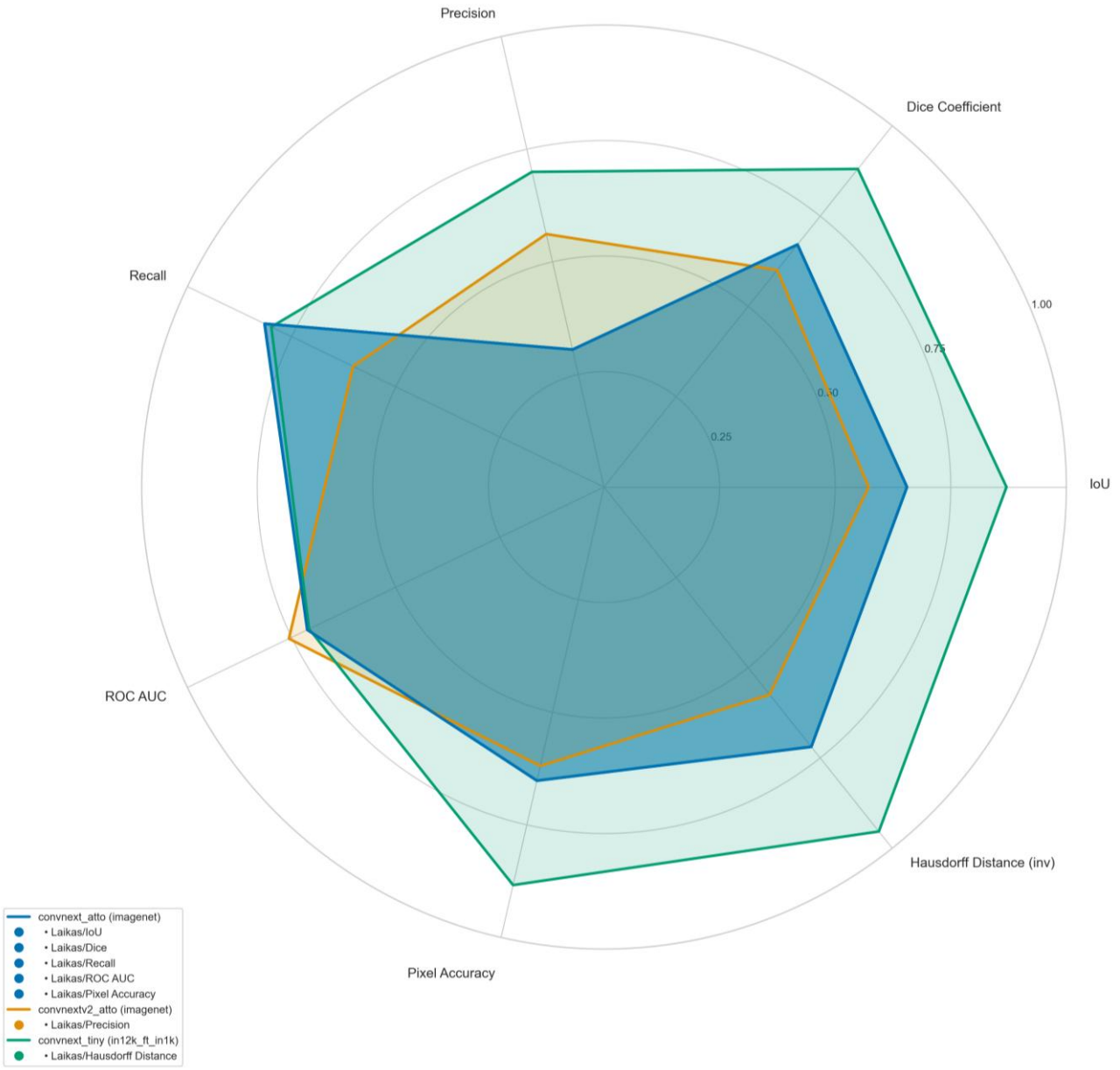
convnextv2_large modeliai pasižymi patikimiausiu ir labiausiai subalansuotu našumu. Ši informacija stulpelinės diagramos formoje pateikta 12 pav.



12 pav. 10 geriausių bazinių modelių vidutiniai tikslumo metrikų reitingai pateikti stulpelinėje diagramoje

Išanalizavus lentelės (žr. 4 lentelę) ir diagramų (žr. 12 pav.) rezultatus matyti, kad *convnextv2_large* [28] bazinis modelis su *fcmae_ft_in22k_in1k* duomenų bazės apmokytais parametrais pasirodė geriausiai. Dėl šios priežasties ši kombinacija pasirinkta tolimesniems eksperimentams su kitais duomenų rinkiniais.

Atsižvelgiant į skirtingą modelių apmokymo trukmę, tikslinga įvertinti ir bazinių modelių mokymosi efektyvumą. Šis efektyvumas bus vertinamas pagal apmokymo laiko santykį su pasiekta tikslumo metrikos reikšme (t. y., laikas padalintas iš metrikos reikšmės). Mažesnė šio santykio reikšmė rodys didesnį efektyvumą – gebėjimą greičiau pasiekti aukštus tikslumo rezultatus. Toliu bus palyginami geriausiai kiekvienoje efektyvumo ir tikslumo santykyje pasirodę modeliai. Bus atrenkami modeliai, kurie skirtingose efektyvumo metrikose pasirodė geriausiai, ir tarpusavyje lyginami jų metrikų tikslumo rezultatai. Šių modelių tarpusavio palyginimo spindulinė diagrama pateikta 13 pav.



13 pav. Pagal skirtingas efektyvumo metrikas rastų bazinių modelių tarpusavio palyginimas

Pagal spindulinę diagramą (žr. 13 pav.), kurioje pateikti bazinių modelių, atrinktų pagal mokymosi efektyvumą (gebėjimą greitai pasiekti aukštus rezultatus), tikslumo metrikų rezultatai, matyti, kad *convnext_atto* (*imagenet*) [28] demonstruoja aiškiai pranašiausią bendrą efektyvumą. Šis modelis pasiekia itin aukštus įverčius pagal daugumą efektyvumo metrikų, įskaitant *Dice* koeficientą, *IoU*, pikselių tikslumą, *ROC AUC* ir *Recall*. Modelis *convnext_tiny* su *in12k_ft_in1k* apmokytais parametrais buvo efektyviausias *Precision* metrikos atžvilgiu ir palyginus su kitais efektyviais modeliais demonstruoja geriausius rezultatus tikslumo metrikų atžvilgiu. Reikia paminėti, kad diagramoje reikšmės yra normuotos (0-1 skalėje, kur 1 – geriausias rezultatas, o *Hausdorff* atstumas normuotas atvirkščiai). Šių rezultatų detalesnė informacija pateikta 5 lentelėje.

5 lentelė. Pagal skirtingas efektyvumo metrikas rastų bazinių modelių tarpusavio palyginimas

Efektyvumo metrika	Bazinis modelis	Parametrai	IoU	Dice	Prec	Recall	ROC AUC	PA	Haus d	Laikas (s)
Laikas/IoU, Laikas/Dice, Laikas/Recall, Laikas/ROC AUC, Laikas/PA	convnext_atto	imagenet	0,732	0,84	0,834	0,875	0,955	0,904	43,608	180,45
Laikas/Precision	convnextv2_atto	imagenet	0,722	0,834	0,853	0,842	0,957	0,903	46,803	182,739
Laikas/Hausdorf	convnext_tiny	in12k_ft_in1k	0,757	0,858	0,863	0,872	0,955	0,915	38,421	201,052

Iš šių rezultatų (žr. 5 lentelę ir 13 pav.), matoma, kad *ConvNeXt* [3, 28] šeimos modeliai efektyviausiai išnaudoja mokymosi laiką. Jei yra poreikis gana greitai apmokyti modelius ir gauti aukštus rezultatus, tai *convnext* šeimos modeliai turėtų geriausiai atlikti tokio tipo užduotį.

Architektūrų palyginimas

Kita esminė modelio dalis yra architektūra. Šiam tyrimui atlikti buvo naudojamos *segmentation-models-pytorch* bibliotekos teikiamos architektūros, bei naudojant *pytorch* mašininio mokymosi karkasą buvo įgyvendinta *Attention U-Net* architektūra. Iš viso buvo ištestuota dvylika architektūrų. Kaip ir praeitame bazinių modelių tyrime, visi, išskyrus architektūra, modelio parametrai buvo statiniai, kad būtų galima įvertinti architektūros poveikį skirtingoms metrikoms.

Statiniai parametrai, naudoti kodavimo bazinių modelių tyrime:

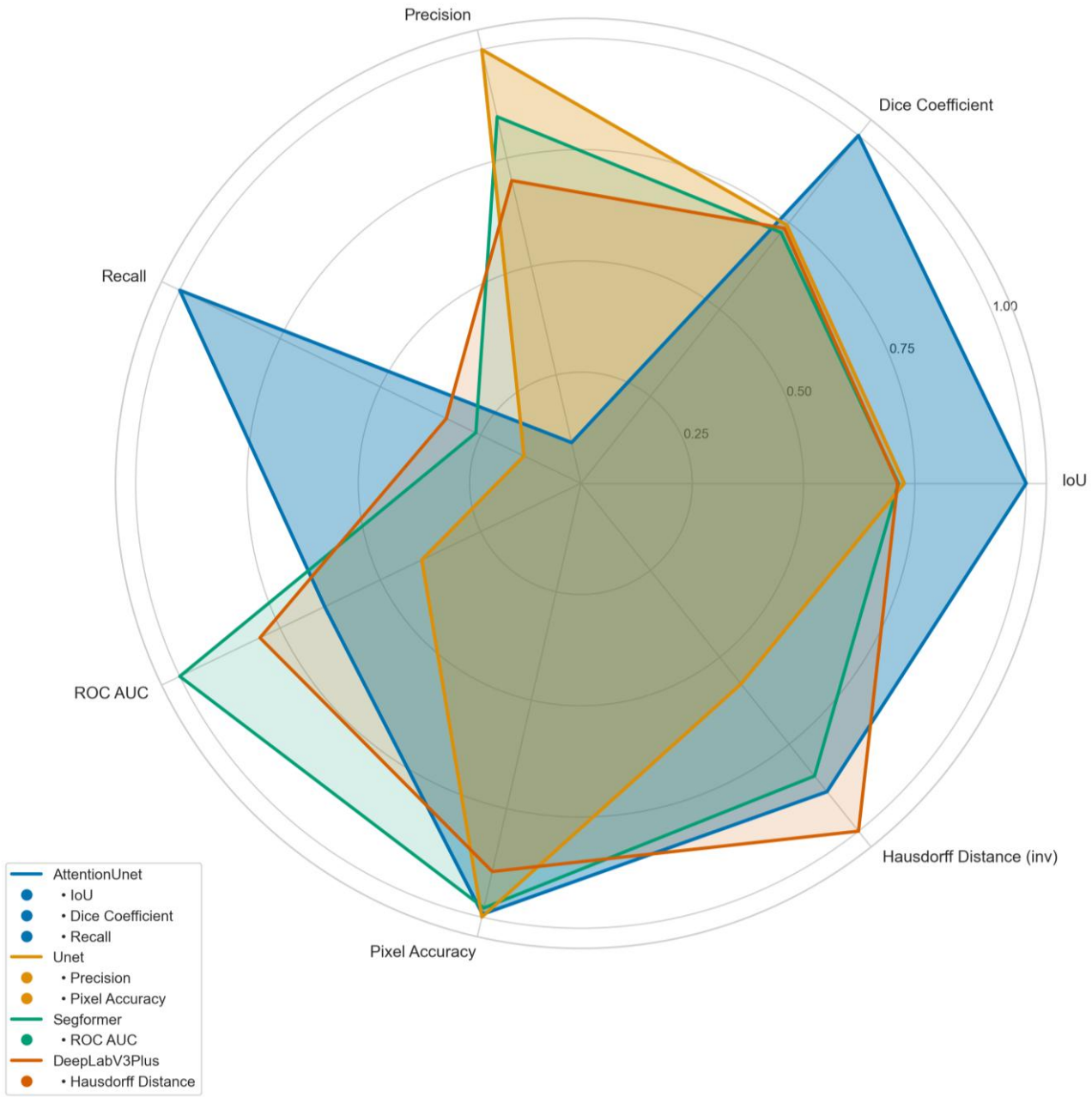
- bazinis modelis: *convnext_tiny* [3] su *in12k_ft_in1k* apmokytais parametrais;
- nuostolių funkcija: sujungta *Focal* (γ=2) (50 %) ir *Jaccard* (*IOU*) (50 %) nuostolių funkcija;
- optimizatorius: *AdamW* [93] su 0,00001 parametrų slopinimu (angl. *weight decay*);
- mokymo epochų skaičius: 12;
- mokymosi greitis: 0,0001;
- vienu metu paduodamo duomenų rinkinio dydis (angl. *Batch size*): 32;
- įvesties vaizdo dydis: 256×256 pikseliai.

Kiekvieno eksperimento metu, laikantis šių pastovių parametrų, buvo keičiamas tik architektūros. Iš viso buvo ištestuota 12 skirtingų architektūrų. Visų bandytų architektūrų rezultatai pateikti 6 lentelėje.

6 lentelė. Architektūrų testavimo metu gautos metrikos

Achitektūra	Laikas (s)	IoU	Pixel Accuracy	Dice	Precision	Recall	ROC AUC	Hausdorff Distance
Attention U-Net	217,619	**0,756**	0,910675	**0,857**	0,831	**0,904**	0,956	40,795
U-Net	206,726	0,745	**0,910756**	0,85	**0,87**	0,847	0,951	43,25
U-Net++	204,788	0,719	0,899	0,832	0,845	0,838	0,943	45,594
LinkNet	197,807	0,716	0,898	0,83	0,84	0,843	0,943	47,82
FPN	210,121	0,743	0,909	0,849	0,863	0,853	0,96	40,041
PSPNet	182,061	0,724	0,9	0,835	0,827	0,869	0,957	47,294
MANet	214,369	0,744	0,908	0,849	0,844	0,873	0,958	41,058
PAN	198,482	0,742	0,91	0,848	0,865	0,851	0,957	40,692
DeepLabV3	198,488	0,741	0,907	0,847	0,854	0,857	0,96	42,574
DeepLabV3+	199,155	0,744	0,909	0,85	0,857	0,86	0,959	**39,897**
UPerNet	200,275	0,744	0,91	0,849	0,856	0,863	0,958	41,713
SegFormer	193,324	0,744	0,91	0,849	0,864	0,855	**0,964**	41,153

Analizuojant gautus rezultatus (žr. 6 lentelę), išsiskiria *Attention U-Net* [41] architektūra, pasiekianti aukščiausius *IoU* (0,756) bei *Dice* (0,857) rodiklius ir puikų *Recall* (0,904), tačiau jos apmokymas trunka ilgiausiai (217,6 s), o *Precision* yra sąlyginai žemesnis (0,831). Tuo tarpu *DeepLabV3+*[18] pasižymi geriausiu *Hausdorff* atstumu (39,9) ir greitu apmokymu (199,2 s), išlaikydamas gerus kitus rodiklius. *SegFormer* [4] rodo aukščiausią *ROC AUC* reikšmę (0,964) ir yra greitai apmokomas (193,3 s), o *FPN* [102] bei *PAN* [19] siūlo gerą bendrą balansą tarp greičio ir įvairių tikslumo metrikų. Galime daryti išvadą, kad nėra vienos architektūros, kuri absoliučiai dominuotų visose metrikose. Toliau vaizdinėmis priemonėmis bus žiūrima kaip kiekvienoje metrikoje geriausios architektūros tarpusavyje lyginasi. Šių architektūrų tarpusavio palyginimo spindulinė diagrama pateikta 14 pav.



14 pav. Pagal skirtingas metrikas rastų architektūrų tarpusavio palyginimas

Pagal spindulinę diagramą (žr. 14 pav.), matoma, kad *Attention U-Net* [41] architektūra (mėlynas plotas) demonstruoja ryškiai geriausią bendrą tikslumą. Šis modelis ypač pasižymi aukštais *Recall*, *ROC AUC*, *IoU* bei *Dice* koeficiento rodikliais. *U-Net* [14] (oranžinis plotas) taip pat rodo gerus rezultatus, ypač *Precision* srityje, tačiau nusileidžia *Attention U-Net* pagal daugumą kitų metrikų. *SegFormer* [4] (žalias plotas) ir *DeepLabV3+* (rudas plotas) atrodo stiprūs pikselių tikslumo (angl. *Pixel Accuracy*) ir *Hausdorff* atstumo srityse, bet jų bendras našumas pagal kitas svarbias metrikas yra mažesnis. Remiantis šiuo vizualiu palyginimu, *Attention U-Net* atrodo kaip universaliausiai pajėgi architektūra iš šių keturių. Reikia paminėti, kad diagramoje reikšmės yra normuotos (0-1 skalėje, kur 1 – geriausias rezultatas, o *Hausdorff* atstumas normuotas atvirkščiai). Šios diagramos detalesnė informacija taip yra pateikta 7 lentelėje.

7 lentelė. Pagal skirtingas metrikas rastų architektūrų tarpusavio palyginimo lentelė

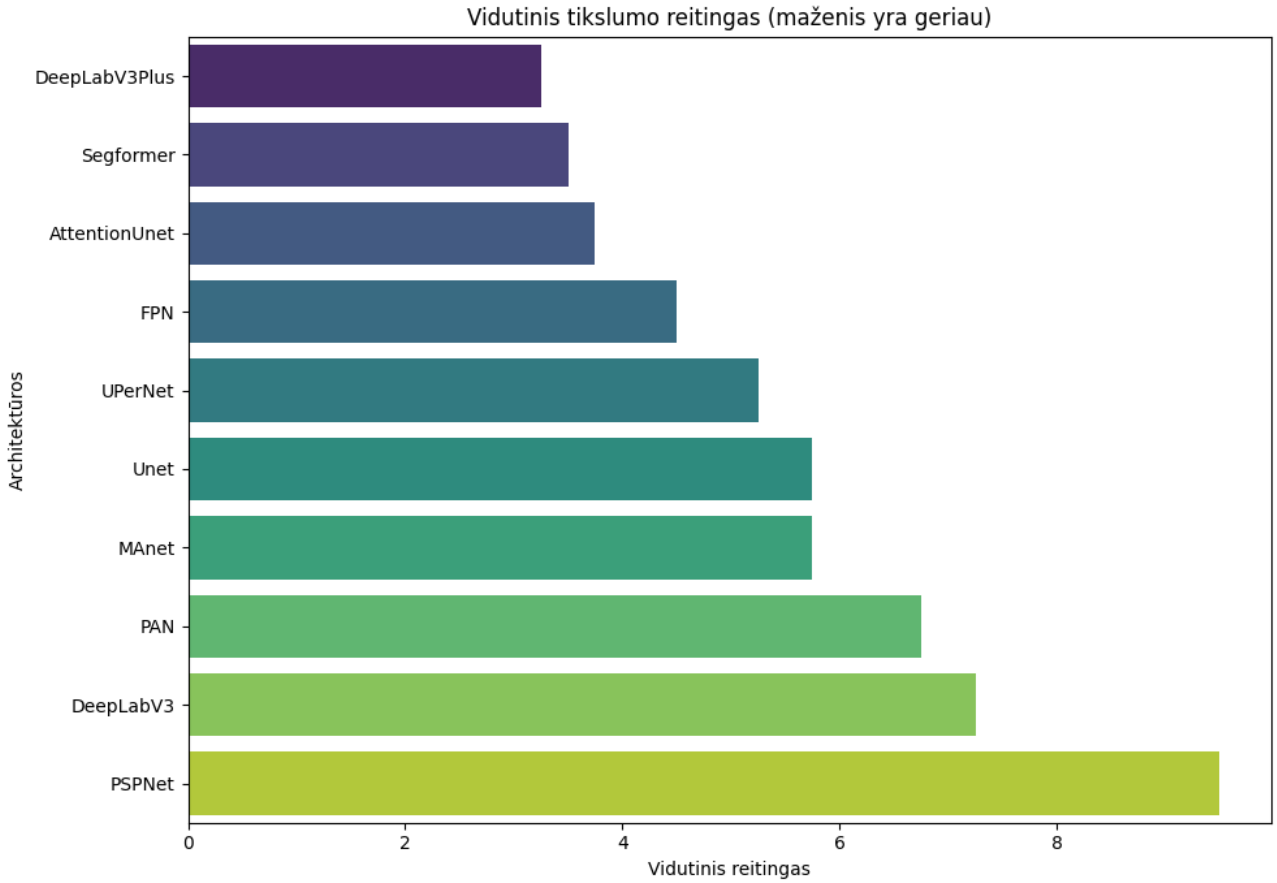
Geriausia metrika	Architektūra	Laikas (s)	IoU	PA	Dice	Prec	Recall	ROC AUC	Hausd
IoU, Dice Coefficient, Recall	AttentionUnet	217,619	0,756	0,911	0,857	0,831	0,904	0,956	40,795
Precision, Pixel Accuracy	Unet	206,726	0,745	0,911	0,85	0,87	0,847	0,951	43,25
ROC AUC	Segformer	193,324	0,744	0,91	0,849	0,864	0,855	0,964	41,153
Hausdorff Distance	DeepLabV3Plus	199,155	0,744	0,909	0,85	0,857	0,86	0,959	39,897

Toliau bus apžvelgtas geriausių 10 geriausių architektūrų vidutinis reitingas naudojant keturias esmines segmentavimo metrikas (*IoU*, *Dice* koeficientas, *ROC AUC*, *Hausdorff* atstumas). Bus naudojamos ne metrikų vertės, bet vieta kurią architektūra užėmė vertinant metrikų rezultatus nuo geriausio iki prasčiausio. Visų minėtų metrikų vietos bus sudedamos ir padalinamos iš metrikų skaičiaus. Tokiu būdu bus siekiama įvertinti architektūrų bendrą tikslumą. Kitos metrikos taip pat bus pateiktos, bet nebus naudojamos vertinime. Ši informacija yra pateikta 8 lentelėje.

8 lentelė. 10 geriausių architektūrų vidutinis reitingas

Architektūra	Vidutnis reitingas	Geriausia metrika	IoU	Dice	ROC AUC	Hausdorff	PA	Precision	Recall
DeepLabV3+	3,25	Hausdorff Distance (1)	5	3	4	1	6	5	5
SegFormer	3,5	ROC AUC (1)	3	4	1	6	3	3	7
Attention U-Net	3,75	IoU (1)	1	1	9	4	2	11	1
FPN	4,5	ROC AUC (2)	7	7	2	2	7	4	8
UPerNet	5,25	IoU (4)	4	5	5	7	5	6	4
U-Net	5,75	Pixel Accuracy (1)	2	2	10	9	1	1	10
MANet	5,75	Recall (2)	6	6	6	5	8	9	2
PAN	6,75	Precision (2)	8	8	8	3	4	2	9
DeepLabV3	7,25	ROC AUC (3)	9	9	3	8	9	7	6
PSPNet	9,5	Recall (3)	10	10	7	11	10	12	3

Matoma (žr. 8 lentelę), kad pagal vidutinį reitingą, geriausiai pasirodė *DeepLabV3+* (vid. reitingas 3,25) ir *SegFormer* (vid. reitingas 3,5) architektūros. *DeepLabV3+* išsiskiria geriausiu *Hausdorff* atstumo reitingu (1 vieta), o *SegFormer* – aukščiausiu *ROC AUC* reitingu (1 vieta) tarp vidurkiui naudotų metrikų. Nors *Attention U-Net* (vid. reitingas 3,75) užima pirmąsias vietas pagal *IoU* ir *Dice* metrikas, jos žemesnis *ROC AUC* reitingas (9 vieta) lėmė aukštesnį vidutinį reitingą. Tai rodo, kad *DeepLabV3+*, *SegFormer* bei *Attention U-Net* siūlo geriausią subalansuotą našumą pagal pagrindines persidengimo, klasifikavimo ir atstumo metrikas, kurios buvo naudotos vidurkiui apskaičiuoti. Ši informacija stulpelinės diagramos formoje pateikta 15 pav.



15 pav. 10 geriausių architektūrų vidutiniai tikslumo metrikų reitingai pateikti stulpelinėje diagramoje

Išanalizavus pateiktų lentelių (žr. 6, 7, 8 lenteles) ir diagramų (žr. 14, 15 pav.) rezultatus, matyti skirtingų architektūrų privalumai ir trūkumai. Atsižvelgiant į tai, tolimesniems tyrimams pasirinktos trys perspektyviausios architektūros: *Attention U-Net* [41], *DeepLabV3+* [18] bei *SegFormer* [4]. *Attention U-Net* pasirinkta dėl jos demonstruojamų aukščiausių rezultatų pagal pagrindines segmentavimo kokybės metrikas – *IoU* ir *Dice* koeficientą – kas rodo didžiausią potencialą tiksliam objektų išskyrimui. Tuo tarpu *DeepLabV3+* ir *SegFormer* atrinktos kaip geriausiai subalansuotą tikslumą parodžiusios architektūros, užėmusios aukščiausias vietas pagal vidutinį reitingą (skaičiuotą pagal *IoU*, *Dice*, *ROC AUC*, *Hausdorff* metrikas). *DeepLabV3+* papildomai pasižymi geriausiu ribų nustatymo tikslumu (*Hausdorff* atstumas), o *SegFormer* – puikiu klasių atskyrimu (*ROC AUC*). Šių trijų architektūrų pasirinkimas leis tolimesniuose etapuose konstruoti segmentavimo modelius. Kadangi visų architektūrų mokymosi laikas gana panašus, nebus vertinamos jų efektyvumas.

Nuostolių funkcijų palyginimas

Kita svarbi modelio dalis yra nuostolių funkcija. Šiam tyrimui atlikti buvo naudojamos *segmentation-models-pytorch* bibliotekos teikiamos nuostolių funkcijos ir pačio *pytorch* mašininio mokymosi karkaso teikiamos nuostolių funkcijos bei įgyvendintos funkcijos, kurios nėra pasiekiamos šiose įrankiuose. Iš viso buvo ištestuotos 26 nuostolių funkcijos. Kaip ir praeituose tyrimuose, visi, išskyrus nuostolių funkcija, modelio parametrai buvo statiniai, kad būtų galima įvertinti nuostolių funkcijų poveikį skirtingoms metrikoms.

Statiniai parametrai, naudoti nuostolių funkcijų tyrime:

- bazinis modelis: *mit-b0* [4] su *imagenet* apmokytais parametrais;
- architektūra: *U-Net* [14];
- optimizatorius: *AdamW* [93] su 0,00001 parametrų slopinimu (angl. *weight decay*);
- mokymo epochų skaičius: 12;
- mokymosi greitis: 0,0001;
- vienu metu paduodamo duomenų rinkinio dydis (angl. *Batch size*): 32;
- įvesties vaizdo dydis: 256×256 pikseliai.

Eksperimentuojant su nuostolių funkcijomis, rezultatų analizėje ir grafikuose buvo naudoti jų trumpiniai (metodų pavadinimai). Toliau pateikiami šių trumpinių paaiškinimai ir juos atitinkančios nuostolių funkcijos [84, 85, 86, 87, 89]:

- *bce_jaccard_loss*: *BCE* (50 %) ir *Jaccard* (*IoU*) (50 %);
- *focal_dice_loss*: *Focal* (50 %) ir *Dice* (50 %);
- *binary_focal_jaccard_loss_0.7j_0.3f*: *Focal* (30 %, $\gamma = 2{,}0$) ir *Jaccard* (*IoU*) (70 %);
- *weighted_bce_dice_loss*: *BCE* (teigiamos klasės svoris yra 2,0) (50 %) ir Dice (50 %);
- *bce_dice_loss*: *BCE* (50 %) ir *Dice* (50 %);
- *binary_focal_jaccard_loss*: *Focal* ($\gamma = 2{,}0$) (50 %) ir *Jaccard* (*IoU*) (50 %);
- *weighted_bce_loss*: *BCE* (teigiamos klasės svoris yra 2,0);
- *focal_jaccard_boundary_loss*: *Focal* ($\gamma = 2{,}0$) (20 %), *Jaccard* (*IoU*) (50 %) ir kontūrų *DoU* (30 %);
- jaccard_loss: *Jaccard* (*IoU*);
- *tversky_focal_loss*: *Tversky* ($\beta = 0{,}6$) (50 %) ir *Focal* ($\gamma = 2{,}0$) (50 %);
- *shape_aware_loss*: formą atpažįstantys nuostoliai (*BCE* pagrindu, sverti pagal aproksimuotą atstumą iki kontūrų);
- *dice_loss*: *Dice* nuostoliai;
- *bce_loss*: *BCE* nuostoliai;
- *boundary_aware_loss*: kontūrus pabrėžiantys nuostoliai (*BCE* pagrindu, sverti pagal Laplaso filtru išgautas kraštines);
- *tversky_loss*: *Tversky* nuostoliai ($\beta = 0{,}6$);
- *adaptive_weighted_jaccard_loss*: adaptyviai sverti *Jaccard* (*IoU*) nuostoliai ($\gamma = 1{,}0$);
- *dice_tversky_loss*: *Dice* (50 %) ir *Tversky* ($\beta = 0{,}6$) (50 %);
- *focal_tversky_loss*: *Focal* ir *Tversky* nuostoliai ((1−*Tversky* indeksas) γ su $\alpha = 0{,}3$, $\beta = 0{,}7$, $\gamma = 1{,}5$);
- *combo_region_boundary_loss*: *Dice* (regioniniai) (60 %) ir kontūrų *DoU* (40 %);
- *lovasz_loss*: *Lovasz-Softmax*;
- *focal_lovasz_loss*: *Focal* (50 %) ir Lovasz (50 %);
- *boundary_dice_loss*: *Dice* (50 %) ir kontūrų *DoU* (50 %);
- *active_contour_loss*: aktyvių kontūrų nuostoliai (kontūro ilgio (50 %) ir regiono atitikimo (50 %) junginys);
- *focal_loss*: *Focal* nuostoliai ($\gamma = 2{,}0$, $\alpha = 0{,}25$);
- *boundary_dou_loss*: *Boundary DoU* nuostoliai;
- *sparse_focal_loss*: *Focal* nuostoliai ypač retiems duomenims ($\gamma = 4{,}0$, $\alpha = 0{,}25$).

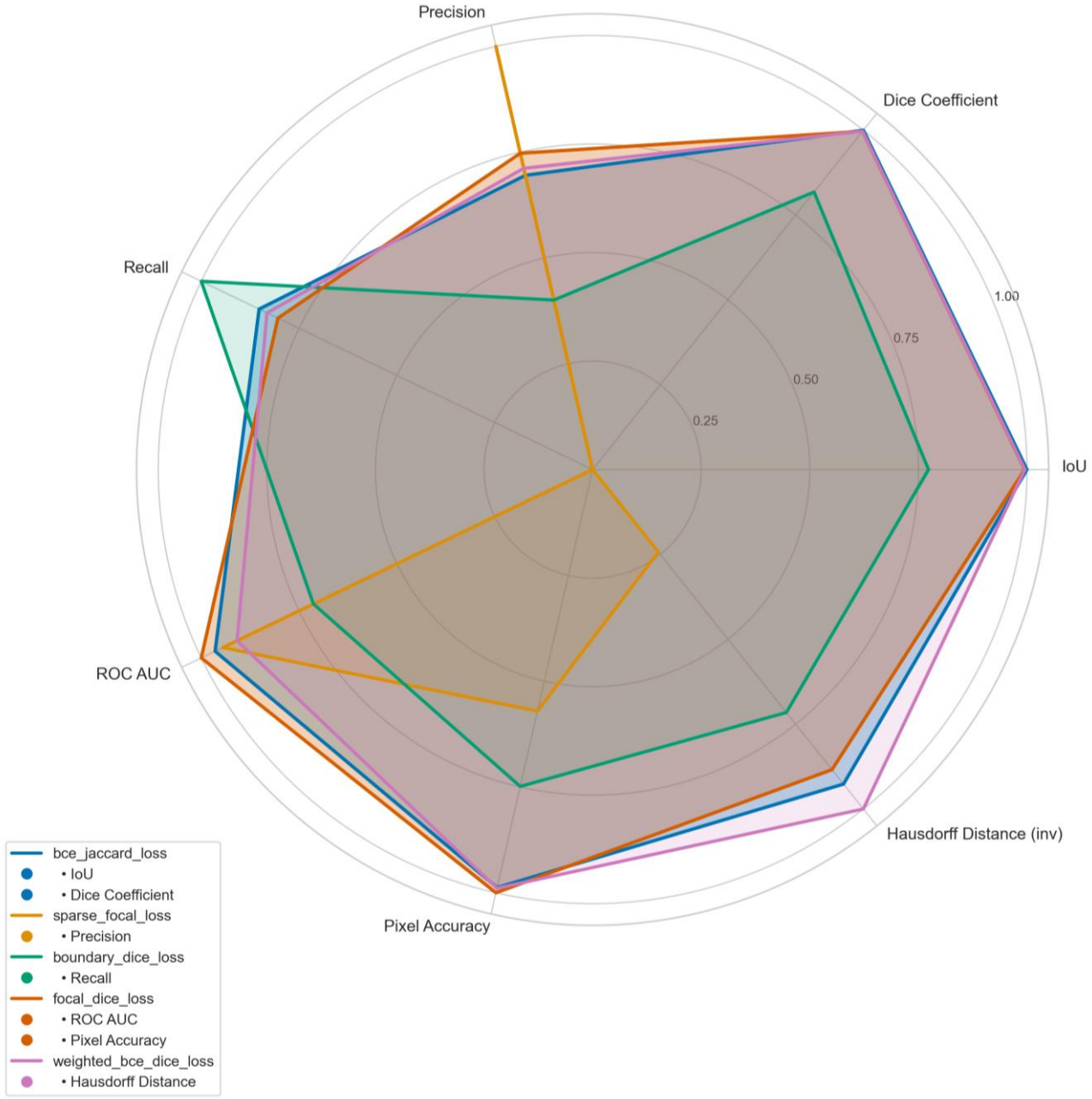
Kiekvieno eksperimento metu, laikantis šių pastovių parametrų, buvo keičiamas tik nuostolių funkcijos. Visų bandytų architektūrų rezultatai pateikti 9 lentelėje.

9 lentelė. Nuostolių funkcijų testavimo metu gautų metrikų rezultatai, surikiuoti pagal *IoU*

Nuostolių funkcija	**Laikas (s)**	**IoU**	**PA**	**Dice**	**Prec**	**Recall**	**ROC AUC**	**Hausd**
bce_jaccard_loss	186,256	**0,741**	0,906	**0,848**	0,83	0,889	0,956	44,155
focal_dice_loss	184,819	0,74	**0,909**	0,846	0,856	0,862	**0,962**	46,321
binary_focal_jaccard_loss_0.7j_0.3f	182,831	0,74	0,908	0,846	0,842	0,877	0,957	43,839
weighted_bce_dice_loss	189,546	0,74	0,907	0,847	0,838	0,878	0,948	**40,414**
bce_dice_loss	187,838	0,739	0,907	0,846	0,841	0,875	0,954	41,813
binary_focal_jaccard_loss	191,536	0,739	0,908	0,846	0,855	0,859	0,96	43,902
weighted_bce_loss	187,508	0,738	0,905	0,845	0,832	0,883	0,957	43,67
focal_jaccard_boundary_loss	182,225	0,73	0,9	0,84	0,807	0,903	0,947	41,856
jaccard_loss	187,285	0,729	0,897	0,838	0,794	0,917	0,945	42,072
tversky_focal_loss	180,3	0,729	0,903	0,839	0,851	0,845	0,95	42,308
shape_aware_loss	180,471	0,722	0,902	0,834	0,856	0,836	0,957	43,295
dice_loss	187,125	0,721	0,891	0,832	0,773	0,934	0,944	40,825
bce_loss	184,805	0,719	0,9	0,832	0,838	0,853	0,954	45,825
boundary_aware_loss	189,632	0,717	0,904	0,831	0,878	0,814	0,955	45,365
tversky_loss	187,768	0,716	0,888	0,83	0,764	0,941	0,936	41,206
adaptive_weighted_jaccard_loss	181,149	0,716	0,892	0,83	0,779	0,925	0,939	43,156
dice_tversky_loss	187,918	0,711	0,884	0,826	0,754	0,945	0,939	44,012
focal_tversky_loss	172,652	0,7	0,877	0,818	0,739	0,955	0,942	44,833
combo_region_boundary_loss	180,659	0,688	0,868	0,81	0,718	0,968	0,93	49,49
lovasz_loss	186,191	0,688	0,898	0,809	0,911	0,755	0,932	50,128
focal_lovasz_loss	183,973	0,676	0,895	0,8	0,91	0,746	0,935	52,205
boundary_dice_loss	182,707	0,661	0,85	0,789	0,688	**0,972**	0,918	54,911
active_contour_loss	180,788	0,624	0,869	0,755	0,809	0,774	0,931	91,412
focal_loss	184,282	0,598	0,867	0,74	0,878	0,678	0,932	61,499
boundary_dou_loss	179,887	0,473	0,674	0,634	0,496	0,947	0,808	85,118
sparse_focal_loss	185,276	0,386	0,808	0,529	**0,977**	0,411	0,953	79,023

Pateiktoje lentelėje (žr. 9 lentelę), kurioje nuostolių funkcijos surikiuotos pagal *IoU* rodiklį, matyti skirtingų funkcijų įtaka galutiniam segmentavimo tikslumui. Aukščiausias *IoU* (0,741) ir *Dice* (0,848) reikšmes pasiekė kombinuota *bce_jaccard_loss* funkcija, tačiau labai panašius rezultatus demonstravo ir kitos kombinuotos funkcijos, tokios kaip *focal_dice_loss*, *binary_focal_jaccard_loss* bei *weighted_bce_dice_loss* (visų *IoU* ~0,74, *Dice* ~0,846-0,848). Tarp šių lyderių *weighted_bce_dice_loss* išsiskyrė ne tik aukštais *IoU* ar *Dice* rodikliais, bet ir geriausiu *Hausdorff* atstumu (40,414). Palyginus, pavienės ar labiau specializuotos nuostolių funkcijos pagal *IoU* ir *Dice* metrikias atsiliko nuo geriausių kombinuotų funkcijų. Taip pat verta paminėti, kad mokymosi laikas visoms funkcijoms buvo labai panašus, dėl to nebus vykdoma efektyvumo analizė. Apibendrinant, kombinuotos nuostolių funkcijos, ypač *weighted_bce_dice_loss*, šioje analizėje pasiūlė geriausią kompromisą tarp regionų ir ribų segmentavimo tikslumo. Toliau vaizdinėmis priemonėmis bus

žiūrima kaip kiekvienoje metrikoje geriausios nuostolių funkcijos tarpusavyje lyginasi. Šių nuostolio funkcijų tarpusavio palyginimo spindulinė diagrama pateikta 16 pav.



16 pav. Pagal skirtingas metrikas rastų geriausių nuostolių funkcijų tarpusavio palyginimas

Spindulinėje diagramoje (žr. 16 pav.) matoma, kad kombinuotos nuostolių funkcijos *weighted_bce_dice_loss* (rožinis plotas) ir *bce_jaccard_loss* (mėlynas plotas) demonstruoja aukščiausius bendrus rezultatus. Abi šios funkcijos pasiekia labai gerus *IoU*, *Dice* koeficiento, pikselių tikslumo bei *ROC AUC* rodiklius, o *weighted_bce_dice_loss* papildomai išsiskiria geriausiu *Hausdorff* atstumu bei aukštu *Recall*. *focal_dice_loss* (raudonas plotas) taip pat konkurencinga, ypač *Precision* srityje, tačiau šiek tiek nusileidžia pagal kitas metrikas. Tuo tarpu *boundary_dice_loss* (žalias plotas) ir ypač *sparse_focal_loss* (oranžinis plotas) rodo ženkliai prastesnius rezultatus. Diagrama parodo, kad tinkamai parinktos kombinuotos nuostolių funkcijos, kaip *weighted_bce_dice_loss*, turėtų leisti pasiekti gerai subalansuotą ir aukštą tikslumą pagal įvairius vertinimo kriterijus. Reikia paminėti, kad diagramoje reikšmės yra normuotos (0-1 skalėje, kur 1 –

geriausias rezultatas, o *Hausdorff* atstumas normuotas atvirkščiai). Šios diagramos detalesnė informacija taip yra pateikta 10 lentelėje.

10 lentelė. Pagal skirtingas metrikas rastų geriausių nuostolio funkcijų tarpusavio palyginimas

Geriausia metrika	**Nuostolio funkcija**	**Laikas (s)**	**IoU**	**PA**	**Dice**	**Prec**	**Recall**	**ROC AUC**	**Hausd**
IoU	bce_jaccard_loss	186,256	0,741	0,906	0,848	0,83	0,889	0,956	44,155
Dice Coefficient	bce_jaccard_loss	186,256	0,741	0,906	0,848	0,83	0,889	0,956	44,155
Precision	sparse_focal_loss	185,276	0,386	0,808	0,529	0,977	0,411	0,953	79,023
Recall	boundary_dice_loss	182,707	0,661	0,85	0,789	0,688	0,972	0,918	54,911
ROC AUC	focal_dice_loss	184,819	0,74	0,909	0,846	0,856	0,862	0,962	46,321
Pixel Accuracy	focal_dice_loss	184,819	0,74	0,909	0,846	0,856	0,862	0,962	46,321
Hausdorff Distance	weighted_bce_dice_loss	189,546	0,74	0,907	0,847	0,838	0,878	0,948	40,414

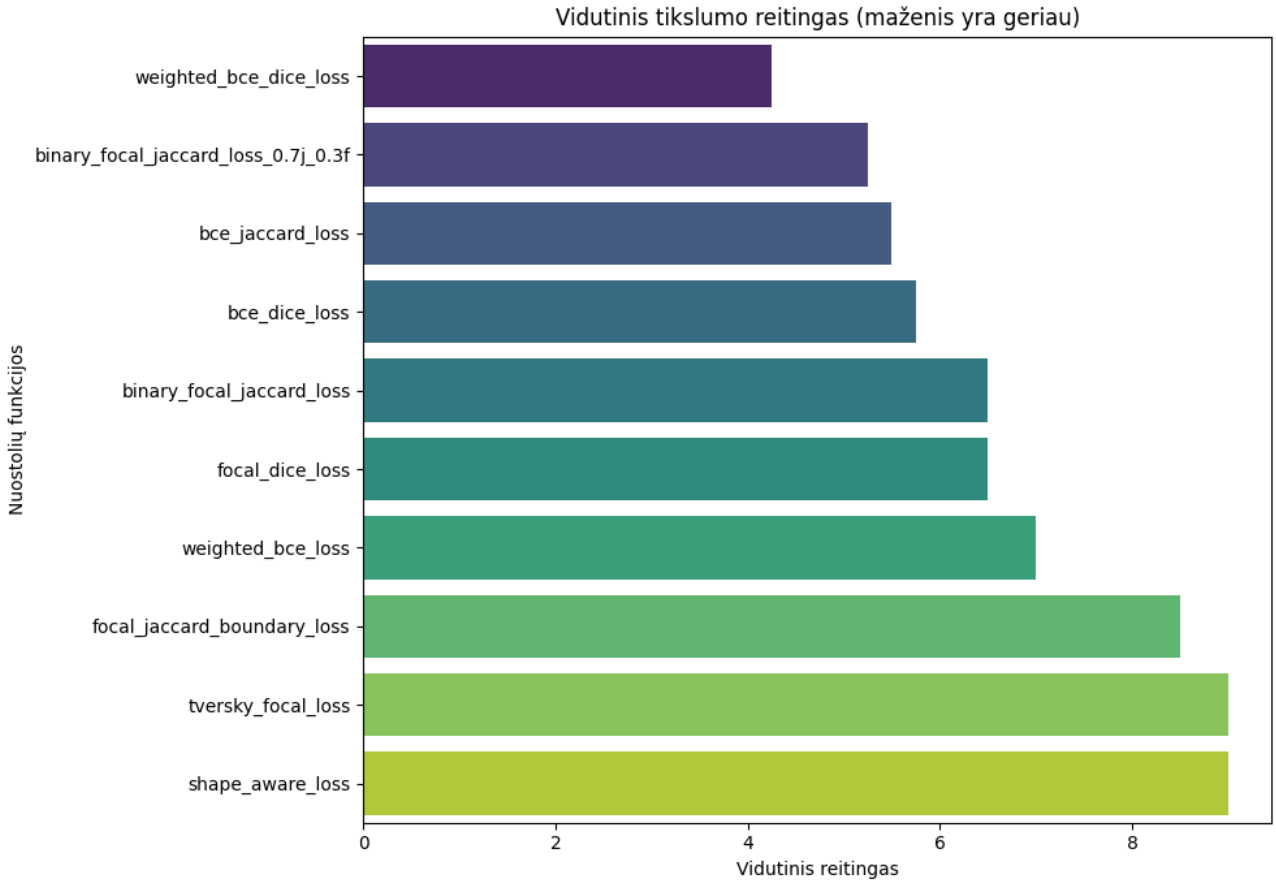
Toliau bus apžvelgtas geriausių 10 geriausių nuostolių funkcijų vidutinis reitingas naudojant keturias esmines segmentavimo metrikas (*IoU*, *Dice* koeficientas, *ROC AUC*, *Hausdorff* atstumas). Bus naudojamos ne metrikų vertės, bet vieta kurią nuostolių funkcija užėmė vertinant metrikų rezultatus nuo geriausio iki prasčiausio. Visų minėtų metrikų vietos bus sudedamos ir padalinamos iš metrikų skaičiaus. Tokiu būdu bus siekiama įvertinti nuostolių funkcijų bendrą tikslumą. Kitos metrikos taip pat bus pateiktos, bet nebus naudojamos vertinime. Ši informacija yra pateikta 11 lentelėje.

11 lentelė. 10 geriausių nuostolių funkcijų vidutinis reitingas

Nuostolių funkcija	**Vidutinis reitingas**	**Geriausia metrika**	**IoU**	**Dice**	**ROC AUC**	**PA**	**Hausd**	**Prec**	**Recall**
weighted_bce_dice_loss	4,25	Hausdorff Distance (1)	2	2	12	5	1	12	13
binary_focal_jaccard_loss_0.7j_0.3f	5,25	IoU (3)	3	4	3	3	11	10	14
bce_jaccard_loss	5,5	IoU (1)	1	1	6	6	14	15	11
bce_dice_loss	5,75	Pixel Accuracy (4)	5	5	9	4	4	11	15
binary_focal_jaccard_loss	6,5	Pixel Accuracy (2)	6	6	2	2	12	8	17
focal_dice_loss	6,5	Pixel Accuracy (1)	4	3	1	1	18	7	16
weighted_bce_loss	7	ROC AUC (4)	7	7	4	7	10	14	12
focal_jaccard_boundary_loss	8,5	Hausdorff Distance (5)	8	8	13	12	5	17	10
tversky_focal_loss	9	Hausdorff Distance (7)	9	9	11	9	7	9	19
shape_aware_loss	9	ROC AUC (5)	11	11	5	10	9	6	20

Pateiktoje nuostolių funkcijų reitingų lentelėje (žr. 11 lentelę) matoma, kad *weighted_bce_dice_loss* pasiekė geriausią (žemiausią) vidutinį reitingą (4,25). Ši funkcija ypač išsiskyrė aukščiausiu reitingu pagal *Hausdorff* atstumą (1 vieta) ir užėmė antrąsias vietas pagal *IoU* bei *Dice* metrikas, nors pagal

ROC AUC, *Precision* ir *Recall* metrikas jos reitingai buvo žemesni. Kitos aukštai reitinguotos funkcijos, kaip *binary_focal_jaccard_loss_0.7j_0.3f* (vid. reitingas 5,25) ir *bce_jaccard_loss* (vid. reitingas 5,5), demonstravo skirtingus privalumus: pirmoji pasižymėjo geru balansu tarp kelių metrikų, o antroji užėmė pirmąsias vietas pagal *IoU* ir *Dice*, bet prasčiau pasirodė pagal *Hausdorff* atstumą. Apibendrinant, pagal pateiktą vidutinį reitingą, *weighted_bce_dice_loss* funkcija laikytina optimaliausiu pasirinkimu tarp tirtų variantų, siūlančiu gerą kompromisą tarp regionų ir ribų tikslumo. Ši informacija stulpelinės diagramos formoje pateikta 17 pav.



17 pav. 10 geriausių nuostolių funkcijų vidutiniai tikslumo metrikų reitingai pateikti stulpelinėje diagramoje

Išanalizavus nuostolių funkcijų palyginimo lentelių (žr. 9, 10, 11 lenteles) ir diagramų (žr. 16 ir 17 pav.) rezultatus matoma, kad sujungta *weighted_bce_dice_loss* nuostolių funkcija parodė geriausius subalansuotus rezultatus. Ji ne tik užėmė aukščiausią vietą pagal vidutinį reitingą (11 lentelė), bet ir pasiekė antrąsias vietas pagal *IoU* bei *Dice* metrikas ir pirmąją vietą pagal *Hausdorff* atstumą (11 lentelė), kas rodo gerą gebėjimą tiksliai atpažinti tiek regionus, tiek jų ribas – tai vizualiai patvirtina ir spindulinė diagrama (žr. 16 pav.). Dėl šios priežasties *weighted_bce_dice_loss* funkcija pasirinkta tolimesniems eksperimentams su skirtingomis architektūromis ir duomenų rinkiniais. Taip pat *binary_focal_jaccard_loss_0.7j_0.3f* funkcija buvo pasirinkta tolimesniems tyrimams, kadangi ji buvo antra pagal vidutinį reitingą.

Optimizatorių palyginimas

Kita modelio dalis, kurią apžvelgsime yra optimizatoriai. Šiam tyrimui atlikti buvo naudojami *pytorch* mašininio mokymosi karkaso teikiami optimizatoriai. Iš viso buvo ištestuoti 7 optimizatoriai.

Kaip ir praeituose tyrimuose, visi, išskyrus optimizatoriai, modelio parametrai buvo statiniai, kad būtų galima įvertinti optimizatorių poveikį sekamoms skirtingoms metrikoms.

Statiniai parametrai, naudoti optimizatorių tyrime:

- bazinis modelis: *mit_b0* [4] su *imagenet* apmokytais parametrais;
- architektūra: *U-Net*;
- nuostolių funkcija: sujungta *Focal* (50%) ir *Jaccard* (*IOU*) (50%) nuostolių funkcija;
- mokymo epochų skaičius: 12;
- mokymosi greitis: 0,0001;
- vienu metu paduodamo duomenų rinkinio dydis (angl. *Batch size*): 32;
- įvesties vaizdo dydis: 256×256 pikseliai.

Kiekvieno eksperimento metu, laikantis šių pastovių parametrų, buvo keičiamas tik nuostolių funkcijos. Testuoto optimizatoriai ir jų parametrai:

- *AdamW* [93] – mokymosi žingsnis: 0,0001, svorių nykimas: 0,00001;
- *SGD* [107] – mokymosi žingsnis: 0,001, inercija: 0,9, svorių nykimas: 0,0001;
- *Adam* [90] – mokymosi žingsnis: 0,0001;
- *RMSprop* [108] – mokymosi žingsnis: 0,0001, *alpha*: 0,99;
- *RAdam* [109] – mokymosi žingsnis: 0,001, svorių nykimas: 0,00001;
- *Lion* [110] – mokymosi žingsnis: 0,0001, svorių nykimas: 0,001;
- *Ranger* [111] – mokymosi žingsnis: 0,001, svorių nykimas: 0,00001.

Kiekvienam optimizatoriui buvo bandoma nustatyti parametrus, kad būtų ekvivalentūs kitų optimizatorių parametrams. Visų bandytų optimizatorių rezultatai pateikti 12 lentelėje.

12 lentelė. Optimizatorių testavimo metu gautų metrikų rezultatai, surikiuoti pagal *IoU*

Optimizatorius	**Laikas (s)**	**IoU**	**PA**	**Dice**	**Precision**	**Recall**	**ROC AUC**	**Hausdorff Distance**
AdamW	307,601	**0,737**	**0,907**	**0,845**	0,85	0,863	0,959	**41,002**
Adam	182,217	0,736	0,907	0,844	0,851	0,861	**0,96**	43,369
Lion	184,059	0,735	0,904	0,843	0,837	**0,87**	0,957	45,722
RMSprop	185,91	0,733	0,904	0,842	0,843	0,862	0,952	45,965
RAdam	184,419	0,722	0,897	0,835	0,835	0,845	0,945	46,005
Ranger	179,883	0,716	0,897	0,829	**0,866**	0,804	0,943	45,532
SGD	180,994	0,117	0,665	0,208	0,431	0,141	0,589	85,54

Pateiktoje lentelėje (žr. 12 lentelę), pagal daugumą tikslumo metrikų geriausiai pasirodė *AdamW* optimizatorius, pasiekęs aukščiausius *IoU* (0,737), *Dice* (0,845), pikselių tikslumo (0,96 kartu su *Adam*) ir *Hausdorff* (41,002) įverčius. Labai panašius tikslumo rezultatus demonstravo ir Adam optimizatorius (*IoU* 0,736, *Dice* 0,844). *Lion* ir *RMSprop* rezultatai buvo šiek tiek žemesni, o *RAdam* ir *Ranger* atsiliko labiau. *SGD* optimizatorius su naudotais parametrais šiuose bandymuose nesugebėjo efektyviai apmokyti modelio ir jo rezultatai buvo itin prasti. Galima atkreipti dėmesį į apmokymo trukmę: *AdamW* apmokymas užtruko ženkliai ilgiau (apie 308 s), palyginti su kitais adaptyviais optimizatoriais (kurių trukmė svyravo apie 180-186 s). Apibendrinant, *AdamW* pasiekė

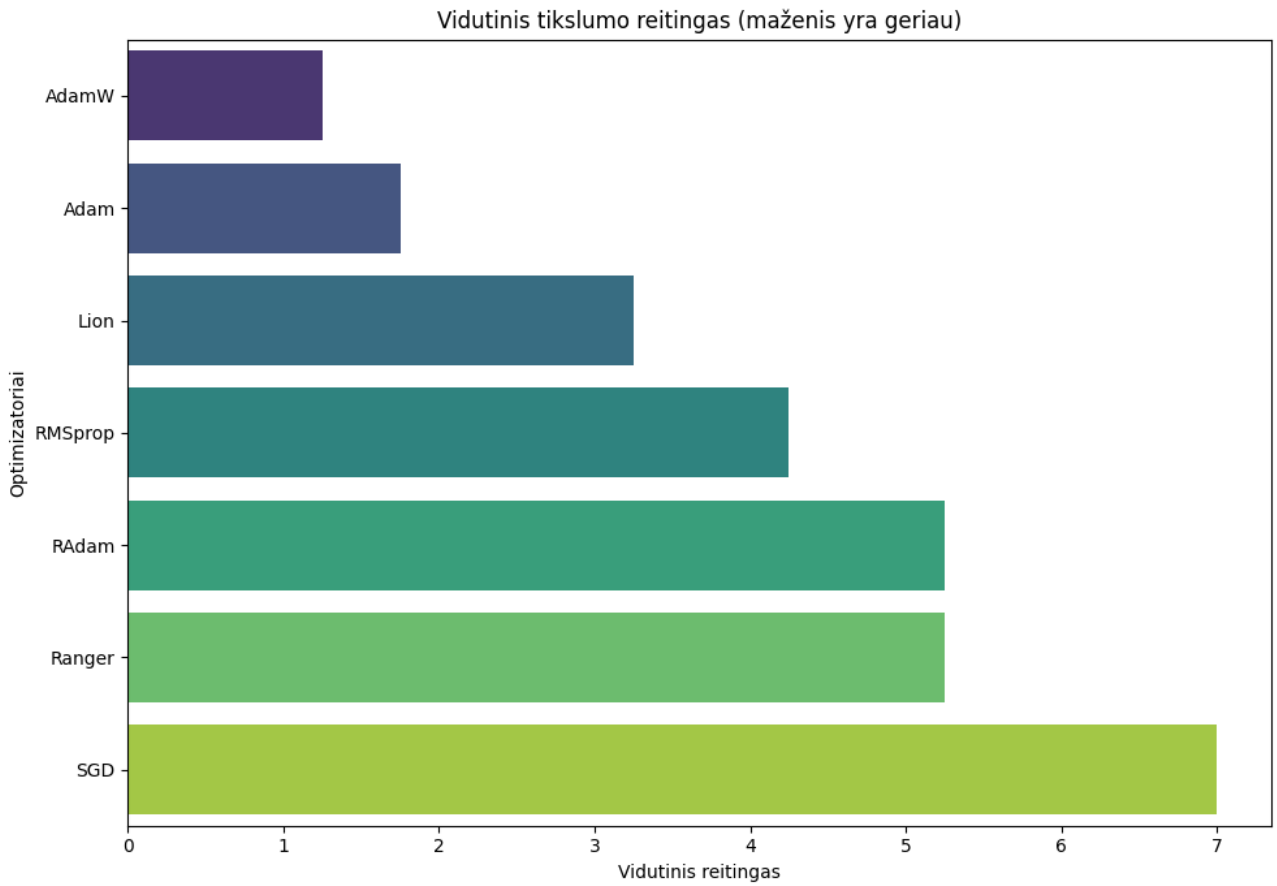
geriausią tikslumą, tačiau pagal pateiktus duomenis reikalavo gerokai daugiau laiko, tuo tarpu *Adam* pasiūlė beveik identišką tikslumą per žymiai trumpesnį laiką, kas jį galėtų daryti praktiškesniu pasirinkimu priklausomai nuo laiko resursų.

Toliau bus apžvelgti optimizatorių reitingai naudojant keturias esmines segmentavimo metrikas (*IoU, Dice* koeficientas, *ROC AUC, Hausdorff* atstumas). Bus naudojamos ne metrikų vertės, bet vieta kurią nuostolių funkcija užėmė vertinant metrikų rezultatus nuo geriausio iki prasčiausio. Visų minėtų metrikų vietos bus sudedamos ir padalinamos iš metrikų skaičiaus. Tokiu būdu bus siekiama įvertinti optimizatorių bendrą tikslumą. Kitos metrikos taip pat bus pateiktos, bet nebus naudojamos vertinime. Ši informacija yra pateikta 13 lentelėje.

13 lentelė. Optimizatorių vidutinis reitingas

Optimizatorius	**Vidutinis reitingas**	**Geriausia metrika**	**IoU**	**Dice**	**ROC AUC**	**PA**	**Hausd**	**Precision**	**Recall**
AdamW	1.25	IoU (1)	1	1	2	1	1	3	2
Adam	1.75	ROC AUC (1)	2	2	1	2	2	2	4
Lion	3.25	Recall (1)	3	3	3	4	4	5	1
RMSprop	4.25	Pixel Accuracy (3)	4	4	4	3	5	4	3
RAdam	5.25	IoU (5)	5	5	5	6	6	6	5
Ranger	5.25	Precision (1)	6	6	6	5	3	1	6
SGD	7	IoU (7)	7	7	7	7	7	7	7

Pateiktoje optimizatorių reitingų lentelėje (žr. 13 lentelę) matomi aiškūs lyderiai. *AdamW* optimizatorius užėmė pirmąją vietą pagal vidutinį reitingą (1,25) ir demonstravo išskirtinai gerus bei subalansuotus rezultatus, užimdamas pirmąsias arba antrąsias vietas pagal daugumą individualių metrikų (įskaitant *IoU, Dice, PA, Hausdorff, ROC AUC, Recall*). Antroje vietoje pagal vidutinį reitingą liko Adam (1,75), taip pat rodęs labai aukštus ir stabilius reitingus per įvairias metrikas, įskaitant pirmą vietą pagal *ROC AUC*. Kiti optimizatoriai, kaip *Lion* (vid. reitingas 3,25) ar *Ranger* (vid. reitingas 5,25), nors ir pasiekė aukščiausią reitingą pagal vieną metriką (atitinkamai *Recall* ir *Precision*), bendrai atsiliko. *SGD* optimizatorius (vid. reitingas 7,0) pagal visas metrikas užėmė paskutinę vietą. Akivaizdu, kad pagal šiuos reitingus *AdamW* ir *Adam* optimizatoriai yra žymiai pranašesni už kitus tirtus variantus. Ši informacija stulpelinės diagramos formoje pateikta 18 pav.



18 pav. Optimizatorių vidutiniai tikslumo metrikų reitingai pateikti stulpelinėje diagramoje

Atlikus 7 skirtingų optimizatorių palyginimą (žr. 12 lentelę), esant fiksuotiems kitiems modelio parametrams (architektūrai, nuostolių funkcijai ir kt.), nustatyta, kad *AdamW* pasiekė nežymiai aukštesnius pagrindinių metrikų (*IoU, Dice, Hausdorff*) įverčius. Tačiau beveik identiškus tikslumo rezultatus parodė ir *Adam* optimizatorius, kurio apmokymas užtruko ženkliai trumpiau (atitinkamai ~308 s ir ~182 s). Kiti adaptyvūs optimizatoriai (*Lion*, *RMSprop*, *RAdam*, *Ranger*) pasirodė prasčiau, o *SGD* su naudotais parametrais visiškai nepasiteisino.

Atlikus detalesnę reitingų analizę (žr. 13 lentelę ir 18 pav.), *AdamW* (vid. reitingas 1,25) ir Adam (vid. reitingas 1,75) aiškiai išsiskyrė kaip lyderiai, užimdami aukščiausias vietas pagal daugumą metrikų ir ženkliai pralenkdami kitus tirtus optimizatorius. Nors *AdamW* demonstruoja nežymų pranašumą pagal reitingus, *Adam* optimizatorius šioje analizėje pasiūlė labai panašų tikslumą su gerokai didesniu laiko efektyvumu, kas rodo jį esant praktišku ir pajėgiu pasirinkimu.

3.2. Tankiu pagrįstos modelių sujungimo strategijos vertinimas ir modelių testavimas

Standartinių giliojo mokymosi modelių tikslumas segmentuojant vaizdus gali priklausyti nuo objektų tankio juose – modelis, gerai veikiantis tankiai užstatytose teritorijose, gali prasčiau atpažinti retus objektus, ir atvirkščiai. Siekiant pagerinti pastatų pėdsakų segmentavimo kokybę esant skirtingam užstatymo tankiui, šiame skyriuje siūloma ir eksperimentiškai vertinama nauja modelių ansambliavimo strategija, pagrįsta būtent pastatų tankiu vaizdo fragmentuose. Strategijos esmė – apmokyti kelis specializuotus modelius (bendrą, pritaikytą retam ir pritaikytą tankiam užstatymui) ir išvedimo etape dinamiškai parinkti tinkamiausią modelį pagal pradinės prognozės tankį, taip pat leidžiant tikslinį jų optimizavimą. Tyrimo tikslas – įvertinti, ar ši tankiu pagrįsta modelių sujungimo strategija leidžia pasiekti aukštesnį segmentavimo tikslumą, palyginti su įprastu vieno bendro modelio naudojimu. Eksperimentai buvo atlikti naudojant keletą duomenų rinkinių: *SpaceNet 1*, *SpaceNet 2*, *SpaceNet 6*, Masačusetso, Uhano universiteto bei sukurto Lietuvos duomenų rinkinio. Skaičiavimai atlikti naudojant asmeninį kompiuterį (*RTX 4070 Laptop GPU*) ir *Google Colab* su *A100* grafiniu procesoriumi.

Pirmas hipotezės bandymas

Pirmam testui atlikti buvo naudojamas asmeninis kompiuteris bei sujungtasis *SpaceNet 2* duomenų rinkinys. Visų miestų vaizdai ir kaukė buvo sujungti į atitinkamas bendras vaizdų ir kaukių direktorijas. Sukurti 256×256 dimensijų nuotraukas buvo naudojama ne persidengiančių vaizdų strategija: originalios 650×650 nuotraukos buvo sumažintos iki 512×512 dimensijų ir padalintos į keturias 256×256 dimensijų nuotraukas. Tokiu būdu buvo sukurta 21,472 vaizdų ir kaukių porų. Šis rinkinys buvo dalinamas į 80 % treniravimo duomenis, 15 % validacijos ir 5 % testavimo. Buvo naudotas asmeninis kompiuteris.

Pirmam testui visiems modeliams buvo naudota ši konfigūracija:

- bazinis modelis: *resnet18* su *imagenet* apmokytais parametrais;
- architektūra: *U-Net*;
- optimizatorius: *AdamW* su 0,00001 parametrų slopinimu (angl. *weight decay*);
- mokymo epochų skaičius: 20;
- mokymosi greitis: 0,0001;
- vienu metu paduodamo duomenų rinkinio dydis (angl. *Batch size*): 32;
- įvesties vaizdo dydis: 256×256 pikseliai;
- tankumo slenkstis: 25 %, duomenų rinkinio pasiskirstymas: 26,95 % balti pikseliai, 73,05 % juodi pikseliai;
- mokymosi metu buvo išsaugomas geriausią *IoU* rezultatą pasiekęs modelis;
- mokymo žingsnio planuotojas: *ReduceLROnPlateau*, kuris mažina optimizatoriaus mokymosi žingsnį (daugindamas ją iš koeficiento 0,5) po 3 epochų be pagerėjimo.

Testavimui buvo panaudotos 1074 vaizdų ir kaukių poros, modeliui vienu metu buvo paduodama po 32 poras. Pirmo bandymo testavimo imties įvertinimo rezultatai pateikti 14 lentelėje.

14 lentelė. Pirmo tankiu pagrįstos modelių sujungimo strategijos bandymo rezultatai

Modelių metodas	IoU	Dice	PA	Precision	Recall	Hausdorff	Įvertinimo laikas (s)
Bendras modelis	0,783306	0,872838	0,940117	0,907817	0,86475	45,674454	17,957469
Modeliai pagal tankumą	0,851778	0,914704	0,960153	0,939850	0,90980	36,490598	22,482193

Matoma (žr. 14 lentelę), kad strategija naudoti modelį pagal duomenų tankį pagerino tikslumo rezultatus. Palyginti su bendru modeliu, tankiu pagrįsta strategija padidino *IoU* rodiklį nuo 0,783 iki 0,852 (+0,069) ir *Dice* koeficientą nuo 0,873 iki 0,915 (+0,042). Taip pat pagerėjo pikselių tikslumas (*PA*), *Precision* ir *Recall*. Ženkliai sumažėjo (pagerėjo) ir *Hausdorff* atstumas – nuo 45,7 iki 36,5 (-9,2). Vis dėlto, ši strategija pareikalavo šiek tiek ilgesnio įvertinimo laiko: 22,5 s, palyginti su 18,0 s naudojant bendrą modelį. Šie rezultatai kol kas patvirtina hipotezę, kad specializuotų modelių sujungimas pagal tankumą leidžia pasiekti aukštesnį segmentavimo našumą. Tolimesni eksperimentai turi būti atlikti su skirtingais duomenų rinkiniais ir naudojant pajėgesnes konfigūracijas.

Hipotezės testavimas ir geriausio modelio ieškojimas Masačusetso duomenų rinkinyje

Vykdant tyrimą šiame naudojant šį duomenų rinkinį, bus naudojamos dvi šio treniravimo rinkinio versijos: visi vaizdai ir kaukės, vaizdų ir kaukių poros kuriose yra daugiau nei 0 pastatų klasės pikselių skaičius. Treniravimo duomenų rinkinys buvo skaidytas į 256×256 dimensijų nuotraukas naudojant 12,5% persidengimo strategiją. Tokiu būdu buvo sukurta 6909 vaizdų ir kaukių poros rinkinyje, kuriame nebuvo panaikintos poros su visiškai juodomis kaukėmis, ir 6150 vaizdų ir kaukių poros rinkinyje, kuriame buvo panaikintos poros su visiškai juodomis kaukėmis. Sukurtų duomenų rinkinių detalesnė informacija:

- rinkinys be juodų kaukių panaikinimo: 13,13 % pastatų klasės (baltų) pikselių, 86,87 % ne pastatų klasės (juodų) pikselių, 41,55 % vaizdų ir kaukių porų su <10 % pastatų klasės pikselių;
- rinkinys su panaikintomis juodomis kaukėmis: 14,75 % pastatų klasės (baltų) pikselių, 85,25 % ne pastatų klasės (juodų) pikselių, 34,34 % vaizdų ir kaukių porų su <10 % pastatų klasės pikselių.

Testavimo rinkinys Masačusetso duomenų rinkinyje yra atskiras. Šį duomenų rinkinį sudaro 10 1500×1500 dimensijų vaizdai ir tiek pat kaukių su tokiomis pačiomis dimensijomis. Testavimo rinkinyje yra 18,61 % pastatų klasės (baltų) pikselių ir 81,39 % ne pastatų klasės (juodų) pikselių bei yra tiktais dvi kaukės, kurios turi mažiau nei 10 % pastatų klasės (baltų) pikselių. Modelių testavimo metu šie vaizdai bus skaidomi į 256×256 dimensijų vaizdus naudojant 12,5 % persidengimą kaip ir treniravimo imtyje. Segmentuoti 256×256 vaizdai bus sujungiame į 1500×1500 vaizdus taip pat naudojant 12,5 % persidengimą. Toks originalių testavimo vaizdų suskaidymas sukuria 490 256×256 dimensijų testavimo vaizdų ir kaukių porų.

Abejų duomenų rinkinių bandymuose visada buvo naudotas *AdamW* [93] optimizatorius su 0,00001 parametrų slopinimu (angl. *weight decay*) ir *ReduceLROnPlateau* mokymo žingsnio planuotojas,

kuris mažina optimizatoriaus mokymosi žingsnį (daugindamas ją iš koeficiento 0,5) po 3 epochų be pagerėjimo.

Šiems tyrimams atlikti buvo naudota *Google Colab* aplinka su *A100* grafiniu procesoriumi.

Pirma bus apžvelgiami rezultatai nefiltruotame duomenų rinkinyje. Pradžioje bendrų ir tankiu pagrįstų metodų rezultatai bus pateikiami kartu – pirma eis bendro metodo konfigūracija ir po to jai atitinkama tankiu pagrįsta konfigūracija. 5 geriausios metodų poros pateiktos 15 lentelėje.

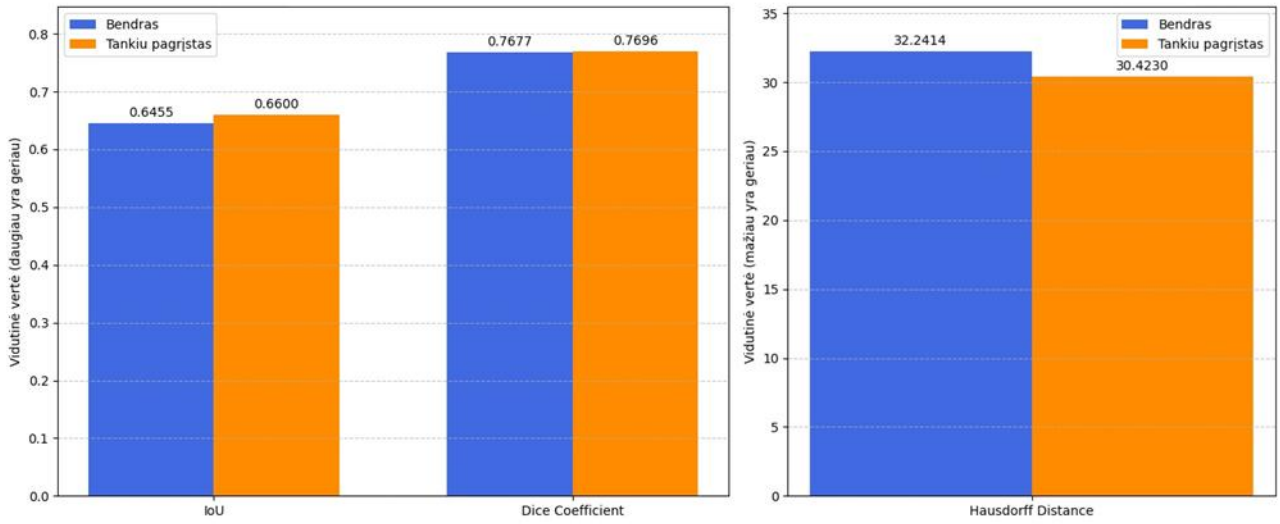
15 lentelė. Pagal *IoU* 5 geriausios metodų poros nefiltruotame Masačusetso rinkinyje

Modelių konfigūracija	**Treniravimo konfigūracija**	**Metodas**	**IoU**	**Dice**	**Hausd**	**PA**	**Prec**	**Rec**
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32	Bendras	0,724	0,83	27,34	0,942	0,861	0,827
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,10	Pagal tankį	0,704	0,813	30,65	0,939	0,85	0,825
DeepLabV3+ convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32	Bendras	0,703	0,814	29,24	0,935	0,83	0,827
DeepLabV3+ convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,10	Pagal tankį	0,705	0,814	29,64	0,939	0,846	0,826
DeepLabV3+ convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 48	Bendras	0,708	0,818	26,6	0,937	0,837	0,831
DeepLabV3+ convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 48, Slenkstis: 0,10	Pagal tankį	0,695	0,807	30,96	0,936	0,838	0,823
SegFormer convnextv2_large, fcmae_ft_in22k_in1k, 0,5 bce dice loss	LR: 0,0001, E: 50, BS: 32	Bendras	0,697	0,809	27,79	0,935	0,83	0,823
SegFormer convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,20	Pagal tankį	0,684	0,801	29,59	0,931	0,815	0,819
SegFormer convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 48	Bendras	0,696	0,81	29,34	0,933	0,816	0,83
SegFormer convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 48, Slenkstis: 0,20	Pagal tankį	0,682	0,799	29,55	0,93	0,816	0,812

Išnagrinėjus pagal *IoU* metriką geriausias modelių konfigūracijas (žr. 15 lentelę) matoma, kad bendras metodas šiuose eksperimentuose dažniausiai pranoko pagal tankį metodą. Tiksliau, vieno modelio metodas parodė geresnius rezultatus pagal įvairius vertinimo rodiklius keturiuose iš penkių

tiesioginių palyginimų. Vienintelis atvejis, kai tankiu pagrįstas metodas parodė nedidelį pranašumą, buvo su DeepLabV3+ modeliu, naudojant 32 paduodamu rinkinio dydį (*BS*) ir 0,10 tankio slenkstį. Šie rezultatai parodė, kad sudėtingesnė dviejų modelių sistema ne visada užtikrino pranašesnį našumą, palyginti su vienu, bendru modeliu. Pilna lentelės versija yra pateikta 4 priede.

Toliau bus apžvelgiama abiejų metodų metrikų vidurkius – ši informacija pateikta 19 pav.



19 pav. Nefiltruotame Masačusetso rinkinyje visų naudotų konfigūracijų metrikų vidurkio palyginimas

Pateiktoje diagramoje (žr. 19 pav.) lygina bendro ir tankiu pagrįsto metodų vidutines metrikų vertes, apskaičiuotas pagal visas naudotas konfigūracijas nefiltruotame Masačusetso duomenų rinkinyje. Analizė rodo, kad tankiu pagrįstas metodas vidutiniškai pasirodė šiek tiek geriau: jis pasiekė aukštesnes vidutines *IoU* ir *Dice* koeficiento vertes. Be to, šis metodas taip pat parodė pranašumą pagal *Hausdorff* atstumą, o tai rodo geresnę kokybę. Nors skirtumai nėra dideli, tankiu pagrįstas metodas nuosekliai lenkė bendrą metodą pagal visas tris vidutines metrikas.

Taip pat buvo ištirta kiek kartų vienas ar kitas metodas buvo pranašesnis (5 priedas). Pagal šį tyrimą buvo atrasta, kad *IoU* metrikoje 8 iš 15 eksperimentų (53,3 %) tankiu pagrįstas metodas buvo geresnis už vieną (bendrą) modelį, *Dice* koeficiento metrikoje 8 iš 15 kartų (53,3 %) bendras modelis buvo pranašesnis už tankiu pagrįstą metodą ir *Hausdorff* atstumo atveju 8 iš 15 eksperimentų (53,3 %) tankiu pagrįstas metodas rodė geresnius rezultatus.

Nors tiriant Masačusetso duomenų rinkinį tankumu pagrįstas metodas per visus eksperimentus pasirodė šiek tiek geriau, bet, vertinant geriausius rezultatus iš abiejų metodų (žr. 6 priedą), vieno modelio konfigūracijos buvo pranašesnės. Iš šio Masačusetso duomenų rinkinio tyrimo galima daryti išvadą, kad optimali modelio konfigūracija yra pranašesnė už duomenų rinkinio tankumu pagrįstą metodą, kuriame modeliai naudoja tokią pačią konfigūraciją kaip ir optimalus modelis.

Toliau bus apžvelgiamos geriausiai ir blogiausiai pasirodžiusios abiejų metodų konfigūracijos pagrindinėse trijose metrikose bei bendrai. Geriausiai pasirodžiusios konfigūracijos yra pateiktos 16 lentelėje.

16 lentelė. Abiejų metodų geriausiai pasirodžiusios konfigūracijos

Metodas	Metrika	Modelio konfigūracijos	Treniravimo konfigūracijos	IoU	Dice	Hausd
Bendras	IoU, Dice Coefficient, bendrai	Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32	0,724	0,83	27,34
Bendras	Hausdorff Distance	DeepLabV3+, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 48	0,708	0,818	26,6
Tankiu pagrįstas	IoU, Dice Coefficient	Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,10	0,709	0,818	30,06
Tankiu pagrįstas	Hausdorff	SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 64, Slenkstis: 0,18	0,334	0,39	15,66
Tankiu pagrįstas	Bendrai	DeepLabV3+, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 64, Slenkstis: 0,10	0,699	0,813	29,12

Matoma (žr. 16 lentelę), kad bendro modelio metodas pasiekė aukščiausią *IoU* (0,724) ir *Dice* (0,83) su *Attention U-Net* [41] modeliu, o geriausią Hausdorff atstumą (26,6) su *DeepLabV3+*. Tankiu pagrįstas metodas geriausiai pasirodė su *SegFormer* [4] modeliu pagal *Hausdorff* atstumą (15,66), o kitos jo geriausios konfigūracijos su *Attention U-Net* ir *DeepLabV3+* [18] parodė šiek tiek žemesnius *IoU* ir *Dice* rodiklius nei bendro metodo geriausi rezultatai, bet su skirtingomis nuostolių funkcijomis ir slenksčiais.

Prasčiausiai pasirodžiusių abiejų metodų konfigūracijų rezultatai yra pateikti 17 lentelėje.

17 lentelė. Abiejų metodų prasčiausiai pasirodžiusios konfigūracijos

Metodas	Metrika	Modelio konfigūracijos	Treniravimo konfigūracijos	IoU	Dice	Hausd
Bendras	IoU, Dice Coefficient	DeepLabV3+, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 32	0,548	0,685	35,97
Bendras	Hausdorff, Bendrai	SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 32	0,569	0,704	39,99
Tankiu pagrįstas	IoU, Dice Coefficient	SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 64, Slenkstis: 0,18	0,334	0,39	15,66

Tankiu pagrįstas	Hausdorff, bendrai	SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,15	0,582	0,702	45,44

Masačusetso duomenų rinkinio geriausių ir blogiausių abiejų metodų konfigūracijų klasifikuotų pikselių rezultatai yra pateikti 20 pav.

20 pav. Masačusetso rinkinio geriausių ir blogiausių abiejų metodų konfigūracijų rezultatai

Nors gautos kaukės (žr. 20 pav.) labai panašios, geriausias bendras modelis (metodas) mažiausiai prognozuoja neegzistuojančius (apatinis dešinys kampas) pastatus. Žiūrint vizualiai galima teigti, kad bendras modelis šioje užduotyje (nefiltruotame Masačusetso rinkinyje) pasirodė geriausiai. Geriausio bendro modelio ir jo atitinkamo tankiu pagrįsto metodo palyginimas pateiktas 18 lentelėje.

18 lentelė. Geriausio bendro modelio ir jo atitinkamo tankiu pagrįsto metodo palyginimas

Metodas	**IoU**	**Dice koeficientas**	**Hausdorff atstumas**	**Įvertinimo laikas (s)**
Bendras modelis	0,724	0,830	27,335	19,093
Pagal tankumą	0,704	0,813	30,654	19,395

Nors pagal tankį pagrįstame metode naudojami iš viso trys modeliai, įvertinimo laikas (žr. 18 lentelę) nėra žymiai didesnis. Galima daryti prielaidą, kad galimai didžiausią laiko dalį sudaro modelio inicijavimas. Modeliu sujungimas pagal tankumą turėjo tokį pasiskirstymą: retesnių duomenų modelis segmentavo 119 (24,3 %) vaizdų, tankesnių vaizdų modelis segmentavo 371 (75,7 %).

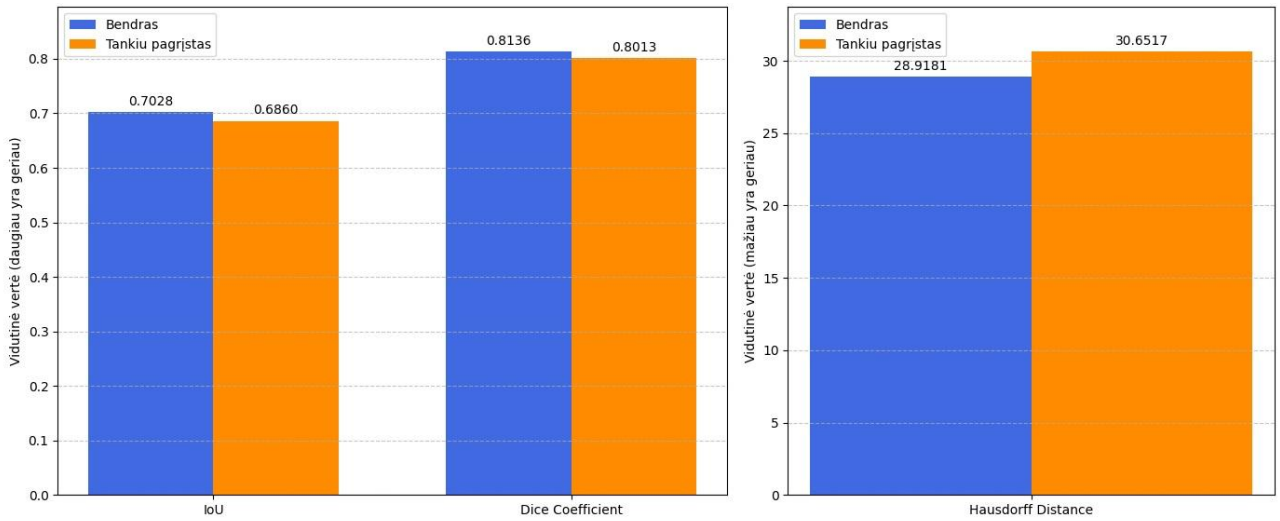
Dabar bus apžvelgiama kaip rezultatus filtruotame Masačusetso duomenų rinkinyje, kuriame buvo panaikintos visiškai juodos kaukės ir atitinkami vaizdai kuriems priklausė tos kaukės. Vykdant eksperimentus filtruotame rinkinyje buvo dažniau naudotos optimalesnės konfigūracijos, kurios buvo atrastos nefiltruoto rinkinio eksperimentuose. 5 geriausios metodų poros filtruotame duomenų rinkinyje pateiktos 19 lentelėje.

19 lentelė. Pagal vidutinį *IoU* 5 geriausios abiejų metodų poros filtruotame Masačusetso rinkinyje

Modelio konfigūracija	**Treniravimo konfigūracija**	**Metodas**	**IoU**	**Dice**	**Hausd**	**PA**	**Prec**	**Rec**
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32	Bendras	0,723	0,829	28,28	0,942	0,855	0,832
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,15	Pagal tankį	0,711	0,822	29,88	0,939	0,851	0,821
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 48	Bendras	0,728	0,834	26,9	0,942	0,854	0,834
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 48, Slenkstis: 0,15	Pagal tankį	0,706	0,816	28,73	0,938	0,846	0,821
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 48	Bendras	0,721	0,828	27,35	0,941	0,853	0,829
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 48, Slenkstis: 0,15	Pagal tankį	0,705	0,816	29,35	0,938	0,85	0,817

Išnagrinėjus pateiktas metodų konfigūracijas matoma (žr. 20 lentelę), kad bendras metodas šiuose konkrečiuose eksperimentuose nuosekliai pranoko metodą pagal tankį. Tiksliau, vieno metodo bendras metodas parodė geresnius rezultatus pagal visas metrikas (*IoU*, *Dice* koeficientas, *Hausdorff* atstumas, pikselių tikslumas, *Precision*, *Recall*) visuose trijuose tiesioginiuose palyginimuose, kai buvo naudojama ta pati modelio architektūra, nuostolių funkcija ir pagrindiniai treniravimo parametrai (išskyrus slenkstį, taikomą tik metodams pagal tankumą). Nė vienu iš šių atvejų pagal tankumą metodas, neparodė pranašumo prieš paprastesnį bendrą modelį. Taip pat galima pastebėti, kad *Attention U-Net* [41] architektūros parodė geriausius rezultatus. Apibendrinant, ši lentelė parodo, kad sudėtingesnė dviejų trijų modelių sistema neužtikrino geresnio tikslumo. Pilna lentelės versija yra pateikta 7 priede.

Toliau bus apžvelgiama abiejų metodų visų eksperimentų metrikų vidurkius, kurie yra pateikti 21 pav.



21 pav. Filtruotame Masačusetso rinkinyje visų naudotų konfigūracijų metrikų vidurkio palyginimas

Vertinant bendrojo ir tankiu pagrįsto metodų vidutinius rezultatus filtruotame Masačusetso duomenų rinkinyje, atsižvelgiant į visas taikytas konfigūracijas (žr. 21 pav.), nustatyta, kad tankiu pagrįstas metodas vidutiniškai demonstravo prastesnį tikslumą šį kartą. Bendras modelis pasiekė aukštesnes vidutines *IoU* bei *Dice* koeficiento vertes, taip pat parodė pranašumą pagal *Hausdorff* atstumą, kas rodo geresnę segmentavimo kokybę. Tankiu pagrįstas metodas neišlaikė pranašumo prieš bendrąjį modelį visose trijose vidutinėse metrikose.

Taip pat pastebėtina, kad bendri rezultatai reikšmingai pagerėjo, kai iš duomenų rinkinio buvo pašalintos visiškai juodos kaukės kartu su atitinkamais vaizdais. Šis pagerėjimas rodo, kad vaizdų ir kaukių porų, pasižyminčių itin dideliu klasių disbalansu, filtravimas gali būti efektyvi strategija siekiant tikslesnių rezultatų.

Įvertinant kiek kartų vienas ar kitas metodas buvo pranašesnis (žr. 8 priedą), buvo atrasta, kad filtruotame duomenų rinkinyje 15 atliktų eksperimentų vieno modelio sprendimas *IoU* ir *Dice* koeficiento metrikose visada buvo pranašesnis už trijų modelių sprendimą, kurie buvo apmokyti pagal tankumą. *Hausdorff* atstumo metrikos rezultatuose tankumu pagrįstas sprendimas buvo pranašesnis tik 2 iš 15 kartų. Tai rodo, kad duomenų rinkinyje, kuriame klasių disbalansas yra mažesnis, bei naudojant optimalesnes modelių konfigūracijas vieno optimalaus modelio sprendimas yra pranašesnis už pagal tankumą treniruotų modelių sujungimą.

Geriausiai pasirodžiusios konfigūracijos yra pateiktos 20 lentelėje.

20 lentelė. Abiejų metodų geriausiai pasirodžiusios konfigūracijos filtruotame Masačusetso rinkinyje

Metodas	**Metrika**	**Modelio konfigūracija**	**Treniravimo konfigūracija**	**IoU**	**Dice**	**Hausdorff**
Bendras	IoU, Dice koeficientas, Hausdorff atstumas, Bendrai	Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 48	0,728	0,834	26,9
Tankiu pagrįstas	IoU, Dice koeficientas	Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,15	0,711	0,822	29,88
Tankiu pagrįstas	Hausdorff atstumas, Bendrai	Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 48, Slenkstis: 0,15	0,706	0,816	28,73

Matoma (žr. 20 lentelę), kad bendro modelio metodas pasiekė aukščiausią *IoU* (0,728), *Dice* (0,834) ir *Hausdorff* atstumą (26,9) su *Attention U-Net* [41] architektūra. Tankumu pagrįstas metodas geriausiai pasirodė taip pat su *Attention U-Net* pagal *IoU* (0,711) ir *Dice* koeficientą (0,822), o naudojant tuos pačius parametrus, išskyrus didesnį duomenų padavimo kiekį (48), pasiekė geriausią *Hausdorff* atstumą (29,88) bei buvo geriausias tankumu pagrįstas modelis bendrai.

Geriausio modelio iš nefiltruoto duomenų rinkinio ir filtruoti duomenų rinkinio pikselių rezultatai pateikti 22 pav.

22 pav. Geriausių bendrų modelių iš filtruoto ir nefiltruoto rinkinio palyginimas

Matoma (žr. 22 pav.), kad gautos kaukės yra labai panašios, Nors modelio kuris naudoja filtruotus duomenis tikslumo metrikų rezultatai yra geresni (0,728 *IoU*, 0,834 *Dice* koeficientas, 26,90 *Hausdorff* atstumas) negu modelio kuris naudoja nefiltruotus duomenis (0,724 *IoU*, 0,830 *Dice* koeficientas, 27,335 *Hausdorff* atstumas), nefiltruotų duomenų modelis prognozuoja mažiau neegzistuojančių pastatų (apatinis dešinys kampas).

Geriausių bendrų modelių iš filtruotų ir nefiltruotų rinkinių ir jų atitinkamo tankiu pagrįstų metodų palyginimai pateikti 21 lentelėje.

21 lentelė. Geriausių abiejų metodų naudojant filtruotus ir nefiltruotus duomenis palyginimas

Filtravimas	Metodas	IoU	Dice koeficientas	Hausdorff atstumas	Įvertinimo laikas (s)
Filtruota	Bendras modelis	0,728	0,834	26,899	21,262
Filtruota	Pagal tankumą	0,706	0,816	28,727	19,306
Nefiltruota	Bendras modelis	0,724	0,830	27,335	19,093
Nefiltruota	Pagal tankumą	0,704	0,813	30,654	19,395

Lentelėje (žr. 21 lentelę) matoma, kad filtravimas ir pasirinktas metodas (bendras arba pagal tankumą) turi nedidelę pranašumą segmentavimo metrikoms (*IoU*, *Dice* koeficientas, *Hausdorff* atstumas). Įdomu tai, kad bendro modelio kuris naudoja filtruotus duomenis įvertinimo laikas yra didžiausias. Pagal tankumą apmokytų modelių testavimo vaizdų pasiskirstymas: 180 (36,7 %) vaizdų segmentavo mažiau tankių vaizdų modelis ir 310 (63,3 %) segmentavo tankesnių vaizdų modelis.

Modelių testavimas Uhano universiteto duomenų rinkinyje

Tikrinti modelio efektyvumą buvo naudotas Uhano duomenų rinkinys, kuris taip pat turi atskirą testavimo duomenų rinkinį, kuris bus naudojamas įvertinti modelio tikslumą. Šis rinkinys turi 6960 512×512 dimensijų vaizdų ir kaukių poras treniravimo rinkinyje ir 1228 testavimo rinkinyje. Kaip ir su Masačusetso rinkiniu, buvo naudojama 12,5 % persidengimas, taip pat buvo išfiltruojamos vaizdų ir kaukių poros kuriose pastatų klasės pikselių procentas buvo mažesnis už 10 %. Iš šio sukurto duomenų rinkinio buvo atsitiktinai atrinkta 9000 vaizdų ir kaukių porų, kurios ir buvo naudotas treniravimui. Paduodant testavimo rinkinio duomenis modeliams buvo taip pat naudotas 12,5 % persidengimas suskaidyti nuotraukas į 256×256 vaizdus – taip padalinus nuotraukas susikuria 11052 vaizdai, kuriuos galima paduoti modeliui. Gauti modelio rezultatai sujungiame taip pat naudojant 12,5 % persidengimą. Sukurto treniravimo duomenų rinkinio klasių pasiskirstymo informacija: 25,85 % pastatų klasės (baltų) pikselių ir 74,15 % ne pastatų klasės (juodų) pikselių. Testavimo rinkinio klasių pasiskirstymo informacija: 15,52 % pastatų klasės (baltų) pikselių, 84,48 % ne pastatų klasės (juodų) pikselių ir 471 (38,36 %) vaizdų su <10 % pastatų klasės (baltų) pikselių. Testų metu buvo naudojama viena konfigūracija:

- architektūra: *Attention U-Net* [41];
- bazinis modelis: *ConvNext V2 Large* [28];
- apmokytų parametrų duomenų bazė: *fcmae_ft_in22k_in1k*;
- nuostolių funkcija: jungtinė *Focal* (30 %) ir *Jaccard* (70 %) nuostolių funkcija;
- optimizatorius: *AdamW* [93] su 0,00001 parametrų slopinimu (angl. *weight decay*);
- mokymo epochų skaičius: 50;
- mokymosi žingsnis: 0,0001;
- vienu metu paduodamo duomenų rinkinio dydis (angl. *Batch size*): 32;
- įvesties vaizdo dydis: 256×256 pikseliai;
- mokymo žingsnio planuotojas: *ReduceLROnPlateau*, kuris mažina optimizatoriaus mokymosi žingsnį (daugindamas ją iš koeficiento 0,5) po 3 epochų be pagerėjimo.

Šiuose bandymuose vienintelis parametras, kuris keitėsi buvo slenkstis, kuris nustato kaip padalinti duomenų rinkinį į tankias vaizdų ir kaukių poras ir į retas. Buvo išbandytos dvi slenksčio vertės: 0,25 ir 0,30. Jų rezultatų palyginimai su vieno modelio metodu yra pateikti 22 lentelėje.

22 lentelė. Uhano universiteto duomenų rinkinio bandymų rezultatai

Metodas	IoU	Dice	Hausdorff atstumas	Pikselių tikslumas	Precision	Recall	Įvertinimo laikas
Bendras	0,835	0,868	24,02	0,985	0,951	0,951	181,94
Pagal tankumą (0,25 slenkstis)	0,738	0,773	27,05	0,983	0,945	0,949	385,552
Pagal tankumą (0,30 slenkstis)	0,799	0,834	26,18	0,977	0,906	0,949	381,677

Matoma (žr. 22 lentelę), kad ir Uhano universiteto duomenų rinkinyje bendras modelis, kuris buvo apmokytas ant visų duomenų, pasirodė geriau nei modeliai apmokyti pagal tankumą. Rezultatai yra bendrai geresni nei Masačusetso rinkinyje nors buvo naudota ne pilna duomenų imtis. Taip pat galima pastebėti, kad tinkamai pasirinktas slenkstis gali reikšmingai pakelti rezultatų modelių sujungime, kuris naudoja treniruotus modelius pagal rinkinio tankumą. Šiam tyrimui atlikti buvo taip pat naudota *Google Colab* aplinka su *A100* grafiniu procesoriumi.

Modelių testavimas Lietuvos duomenų rinkinyje

Šiam tyrimui buvo sukurtas Lietuvos vietovių duomenų rinkinys. Iš viso buvo sukurta 18000 256×256 dimensijų vaizdų ir kaukių porų skirtų treniravimui ir 163 vaizdų ir kaukių poros testavimui. Testavimo rinkinyje yra trijų matmenų vaizdai: 1024×1024, 768×1024 ir 1024×768. Kaip ir su Masačusetso ir Uhano universiteto rinkiniais, buvo naudojama 12,5 % persidengimas, taip pat buvo išfiltruojamos vaizdų ir kaukių poros kuriose pastatų klasės pikselių procentas buvo mažesnis už 10 %. Testavimo rinkinys pateikiamas modeliui tokiu pat būdu kaip ir buvo naudota Uhano universiteto ar Masačusetso duomenų rinkinių atvejais – suskaidžius testavimo rinkinį į 256×256 vaizdus sukuriamos 3387 vaizdų ir kaukių poros. Eksperimentų metu bus naudojami du treniravimo rinkiniai: pilnas 18000 dydžio rinkinys ir 9000 dydžio rinkinys. Sukurto pilno (18000) treniravimo duomenų rinkinio klasių pasiskirstymo informacija: 25,94 % pastatų klasės (baltų) pikselių ir 74,06 % ne pastatų klasės (juodų) pikselių. Ne pilno informacija: 25,85 % pastatų klasės (baltų) pikselių ir 74,15 % ne pastatų klasės (juodų) pikselių. Testavimo rinkinio klasių pasiskirstymo informacija: 15,87 % pastatų klasės (baltų) pikselių, 84,13 % ne pastatų klasės (juodų) pikselių. Eksperimentų metu naudoti statiniai parametrai:

- optimizatorius: *AdamW* [93] su 0,00001 parametrų slopinimu (angl. *weight decay*);
- mokymo žingsnio planuotojas: *ReduceLROnPlateau*, kuris mažina optimizatoriaus mokymosi žingsnį (daugindamas ją iš koeficiento 0,5) po 3 epochų be pagerėjimo;
- mokymosi žingsnis: 0,0001;
- epochų skaičius: 50.

Kai kuriose eksperimentuose buvo naudojami ne tik Lietuvos rinkinio duomenys, bet ir kitų rinkinių duomenys ar net tiktais kitų rinkinių duomenys – taip buvo siekiama įvertinti ar galima pasiekti gerus rezultatus nenaudojant Lietuvos duomenų rinkinio, kadangi, kaip prieš tai buvo minėta, paruoštas

Lietuvos rinkinys negarantuoja aukšto atitikimo tarp vaizdų ir kaukių. Eksperimentai kaip ir kitų duomenų rinkinių atveju buvo vykdomi abiems metodams vienu metu – kiekvienas bendras modelis turi atitinkamą pagal tankumą naudotą ansambliavimą. Šiam tyrimui atlikti buvo naudota *Google Colab* aplinka su *A100* grafiniu procesoriumi. Pagal vidutinį poros *IoU* rezultatą 3 geriausios metodų poros yra pateiktos 23 lentelėje – pilna versija yra 9 priede.

23 lentelė. 3 geriausių konfigūracijų porų pagal vidutinį *IoU* rezultatai Lietuvos rinkinyje

Modelio konfigūracija	**Treniravimo konfigūracija**	**Treniravimo duomenys**	**Metodas**	**IoU**	**Dice**	**Hausdorff**
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32	Lietuvos rinkinys (18000)	Bendras	0,528	0,623	95,3
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,30	Lietuvos rinkinys (18000)	Pagal tankumą	0,525	0,622	94,46
DeepLabV3+, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32	Masačusetso, Lietuvos rinkinys (9000)	Bendras	0,526	0,625	92,92
DeepLabV3+, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,25	Masačusetso, Lietuvos rinkinys (9000)	Pagal tankumą	0,52	0,62	95,28
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32	Lietuvos rinkinys (18000)	Bendras	0,527	0,622	94,78
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,20	Lietuvos rinkinys (18000)	Pagal tankumą	0,516	0,614	98,43

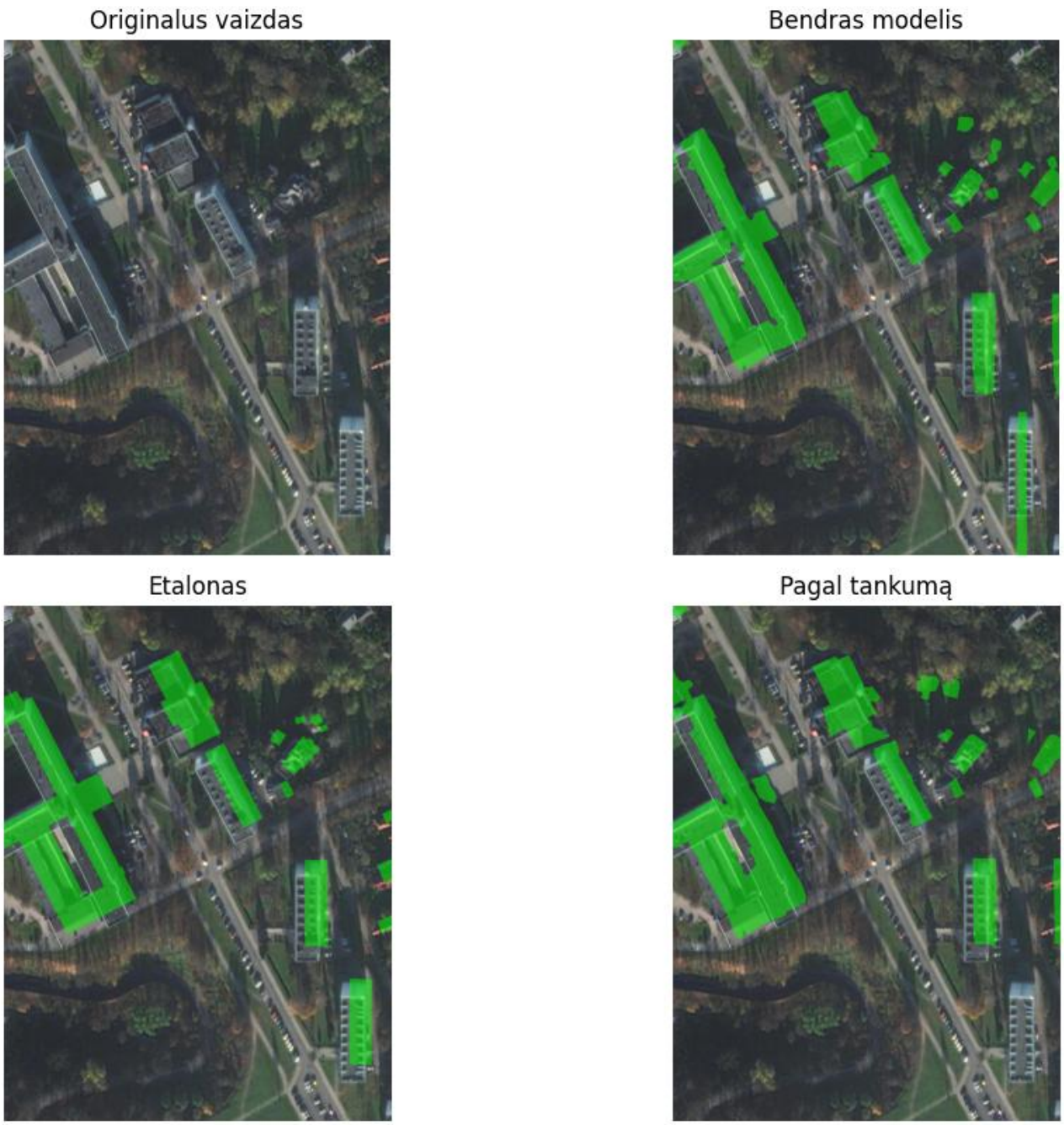
Pateiktuose rezultatuose (žr. 23 lentelę) matoma, kad geriausi rezultatai (bent pagal *IoU*) yra pasiekiami naudojant pilną Lietuvos duomenų rinkinį arba ne pilną, bet įtraukiant kitą duomenų rinkinį. Kad būtų galima susidaryti aiškesnį vaizdą, bus žiūrima kokios konfigūracijos pasirodė geriausiai skirtingose metrikose. Geriausios konfigūracijos pagal skirtingas metrikas yra pateiktas 24 lentelėje.

24 lentelė. Abiejų metodų geriausiai pasirodžiusios Lietuvos rinkinyje konfigūracijos

Metodas	**Metrikos**	**Modelio konfigūracija**	**Treniravimo konfigūracija**	**Treniravimo duomenys**	**IoU**	**Dice**	**Hausdorff**
Bendras	IoU	Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32	Lietuvos rinkinys (18000)	0,528	0,623	95,3
Bendras	Dice koeficientas, Hausdorff atstumas, Bendrai	DeepLabV3+, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32	Masačusetso, Lietuvos rinkinys (9000)	0,526	0,625	92,92
Pagal tankumą	IoU, Dice koeficientas, Hausdorff atstumas, Bendrai	Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,30	Lietuvos rinkinys (18000)	0,525	0,622	94,46

Šiose rezultatuose (žr. 24 lentelę) matoma, kad bendras *DeepLabV3+* [18] architektūros modelis, kuris naudoja *ConvNext V2 Large* [28] kaip bazinį modelį ir sujungtą *Focal* (30 %) *Jaccard* (70 %) nuostolių funkciją bei buvo apmokytas ant mišrių Lietuvos (9000) ir Masačusetso treniravimo duomenų, pasiekė geriausius rezultatus *Dice* ir *Hausdorff* atstumo metrikose. *IoU* metrikoje iš vieno modelio metodų geriausiai pasirodė *Attention U-Net* [41] architektūros modelis, kuris naudojo tuos pačius parametrus, bet kitą duomenų rinkinį (pilnas Lietuvos). Ansambliavime pagal tankumą geriausiai pasirodė ir buvo pirma visose tikslumo metrikose konfigūracija, kuri modeliams naudojo *Attention U-Net* architektūrą, tuos pačius parametrus kaip prieš tai minėtos konfigūracijos ir pilną Lietuvos duomenų treniravimo rinkinį. Taip pat, pagal tankumą modelių treniravimo metodas ir sujungimas *Hausdorff* atstumo metrikoje pranoko tokių pačių konfigūracijų modelį, kuris buvo treniruotas naudojant visus duomenis. Verta paminėti, kad tikslumo metrikų rezultatai yra žemesni nei prieš tai naudotuose duomenų rinkiniuose. Galima daryti prielaidą, kad tam įtakos turi prieš tai minėti Lietuvos duomenų rinkinio trūkumai. Pagal tankumą modelių naudojimo metodo testavimo rinkinio prognozavimo pasiskirstymas: 2814 (83,1 %) vaizdų buvo paduota modeliui treniruotam ant mažiau tankių vaizdų ir 573 (16,9 %) vaizdų buvo paduota modeliui treniruotam ant tankesnių vaizdų.

Kad būtų galima susidaryti geresnį supratimą, kaip gerai pasirodė sukurti modeliai, bus apžvelgiamos jų sukurtos pikselių klasifikacijos. Prognozuotų kaukių ir etalono palyginimas yra pateiktas 23 pav.



23 pav. Geriausių abiejų metodų konfigūracijų prognozuotų pikselių palyginimas (KTU miestelis)

Vaizde (žr. 23 pav.) matome segmentuotą KTU studentų miestelį. Taip pat galima pastebėti etalono kaukės trūkumus ir kaip ji ne visai atitinka vaizde esančių pastatų: kaukės yra šiek tiek pasislinkusios lyginant su pastatais. Galima daryti prielaidą: kadangi vaizdai nėra visiškai paimti iš viršaus, bet labiau kampu, *OpenStreetMap* poligonai, kurie buvo naudoti sukurti etalonines kaukes, neatitiks vaizdų, nes jie yra sukurti pagal pastatų kontūrus žiūrint 90 laipsnių iš viršaus. Paruoštas Lietuvos regionų rinkinys nėra aukštos kokybės lyginant su labiau prižiūrimais rinkiniais ir tai parodo poreikį kurti aukštos kokybės prižiūrimus duomenų rinkinius, kuriuose būtų Lietuvos vaizdai.

3.3. Rezultatų aptarimas ir išvados

Atlikti eksperimentiniai tyrimai parodė, kad modernios konvoliucinių tinklų architektūros, tokios kaip *ConvNeXt* [3] šeimos atstovai ir *FocalNet* [1], kai naudojami kaip kodavimo baziniai modeliai, pasižymi aukštu tikslumu – specifiškai *ConvNeXtV2 Large* [28] parodė geriausius bendrus rezultatus pagrindinėse segmentavimo kategorijose. Tarp segmentavimo architektūrų geriausią tikslumą demonstravo *DeepLabV3+* [18] ir *SegFormer* [4], o *Attention U-Net* [41] išsiskyrė aukštais *IoU* ir *Dice* rodikliais. Kombinuotos nuostolių funkcijos, pavyzdžiui, *BCE* ir *Dice*, ir adaptyvūs optimizatoriai, tokie kaip *AdamW* [93] bei *Adam* [90], taip pat parodė, kad gali prisidėti prie geresnių rezultatų. Pastebėta, kad kruopščiau paruošti ir kuruojami duomenų rinkiniai, tokie kaip Masačusetso ir Uhano universiteto, demonstravo geresnius segmentavimo rezultatus, kas pabrėžia duomenų kokybės svarbą.

Vertinant tankiu pagrįstą modelių sujungimo strategiją, pirminiai bandymai su *SpaceNet 2* duomenų rinkiniu, naudojant paprastesnį *ResNet18* [11] bazinį modelį, parodė reikšmingą tikslumo pagerėjimą lyginant su bendru modeliu. Tai leidžia daryti prielaidą, kad tokia strategija gali būti ypač naudinga siekiant pagerinti rezultatus, kai naudojami mažiau galingi kodavimo baziniai modeliai. Tačiau tolimesni eksperimentai su galingesniais baziniais modeliais Masačusetso ir Uhano universiteto duomenų rinkiniuose atskleidė, jog optimaliai sukonfigūruotas vienas bendras modelis dažnai pasiekdavo panašius ar net geresnius rezultatus, ypač kai duomenų rinkiniai buvo filtruojami siekiant sumažinti klasių disbalansą. Tyrimai su šiam tyrimui sukurtu Lietuvos duomenų rinkiniu išryškino aukštos kokybės, regionui specifinių anotuotų duomenų svarbą, kadangi dėl galimų neatitikimų tarp etaloninių kaukių ir faktinių vaizdų, pasiekti tikslumo rodikliai buvo žemesni nei su kitais, labiau prižiūrimais, duomenų rinkiniais. Taip pat verta paminėti, kad testuotos modelių sujungimo strategijos rezultatai iš dalies priklauso nuo klasių pasiskirstymo testavimo rinkinyje ir treniravimo rinkinio klasių pasiskirstymas nebūtinai nusako testavimo rinkinio pasiskirstymą.

Apibendrinant, giliojo mokymosi modelių komponentų parinkimas yra svarbus veiksnys siekiant aukšto pastatų kontūrų segmentavimo tikslumo. Nors modelių sujungimo strategijos, tokios kaip tankiu pagrįstas modelių sujungimas, gali pasiūlyti pranašumų, ypač su mažiau galingais baziniais modeliais, gerai optimizuotas vieno modelio sprendimas dažnai išlieka konkurencingas. Esminę reikšmę galutiniams rezultatams turi ir naudojamų duomenų kokybė bei jų atitikimas tiriamai geografinei vietovei.

Išvados

1. Atlikta egzistuojančių giliojo mokymosi metodų analizė parodė platų spektrą sprendimų, taikomų semantiniam segmentavimui ir specifiškai pastatų kontūrų išskyrimui. Išnagrinėtos pagrindinės konvoliucinių neuroninių tinklų architektūros (pvz., *U-Net* [14], *DeepLabV3+* [18], *SegFormer* [4]), transformerių modeliai, dėmesio mechanizmai, pažangios segmentavimo paradigmos ir praktiniai jų taikymo aspektai, kas sudarė teorinį pagrindą šio darbo eksperimentinei daliai.
2. Apibrėžtas tyrimo objektas ir parengta tyrimo metodologija. Šiame etape parinkti reikiami įrankiai (*Python*, *PyTorch*) bei technologijos, taip pat eksperimentinė aplinka. Atrinkti bei paruošti viešai prieinami ir specialiai Lietuvos teritorijai skirti vaizdų duomenų rinkiniai. Sudarytas eksperimentinių tyrimų planas ir nustatyti rezultatų vertinimo kriterijai bei pagrindinės metrikos.
3. Realizavus ir apmokius įvairius giliojo mokymosi modelius bei atlikus eksperimentinius tyrimus, nustatyti geriausi modelių komponentai (pvz., *ConvNeXt V2 Large* [28] bazinis modelis, *Attention U-Net* [41], *DeepLabV3+* [18], *SegFormer* [4] architektūros, *Focal* (30%) Jaccard (70%) kombinuota nuostolių funkcija [86], *AdamW* [93] ir *Adam* [90] optimizatoriai). Pasiūlyta ir ištirta tankiu pagrįsta modelių sujungimo strategija pademonstravo reikšmingą tikslumo pagerėjimą naudojant paprastesnį bazinį modelį (*ResNet18* [11]) su *SpaceNet 2* duomenimis. Tačiau, naudojant galingesnius bazinius modelius, gerai optimizuotas vienas bendras modelis dažnai pasiekdavo panašius ar net geresnius rezultatus, ypač kai duomenys buvo filtruojami siekiant sumažinti klasių disbalansą. Tyrimai su Lietuvos duomenų rinkiniu parodė žemesnius tikslumo rodiklius, kas sietina su iššūkiais derinant *OpenStreetMap* duomenis su palydovine informacija (pvz., laiko neatitikimai, erdviniai netikslumai), ir tai pabrėžė aukštos kokybės, regionui specifinių duomenų svarbą.
4. Rekomendacijos ir tolimesnių tyrimų kryptys: rekomenduotina išsamiau ištirti tankiu pagrįstos sujungimo strategijos potencialą su įvairesniais, ypač mažiau galingais, kodavimo baziniais modeliais, kur specializacijos nauda galėtų būti ryškesnė. Taip pat svarbu analizuoti tankio slenksčio parinkimo metodikas bei specializuotų ansamblio modelių tikslinio optimizavimo (pvz., individualių nuostolių funkcijų ar mokymosi greičių parinkimo) įtaką bendram efektyvumui. Kalbant apie Lietuvos duomenų rinkinį, esminė rekomendacija yra kurti aukštos kokybės, tiksliai anotuoto nacionalinio duomenų rinkinį. Tuo pačiu, *SSL* taikymas leistų išnaudoti neanotuotus Lietuviškus duomenis modelių išankstiniam apmokymui, taip potencialiai gerinant rezultatus, ypač su ribotu anotuotų duomenų kiekiu. Galiausiai, perspektyvu gilintis į pažangesnių architektūrų, tokių kaip naujausi transformeriai (pvz., *Mask2Former* [54]) ir generatyviniai modeliai, taikymo galimybes bei tobulinti perėjimo nuo rastrinių kaukių prie vektorinių poligonų metodus.

Vykdant šį darbą, tarpiniai tyrimo rezultatai ir sistemos koncepcija, kuri naudotų realizuotus modelius, buvo pristatyti DAMSS 2023 konferencijoje (žr. 1 ir 2 priedus).

Literatūros sąrašas

1. YANG, Jianwei ir kt. Focal Modulation Networks. *Advances in Neural Information Processing Systems*. 2022, 35, 4203–4217. Prieiga per: Focal Modulation Networks - NeurIPS Proceedings.
2. DAI, Zihang ir kt. CoAtNet: Marrying Convolution and Attention for All Data Sizes. *Advances in Neural Information Processing Systems*. 2021, 34, 3965–3977. Prieiga per: doi: https://doi.org/10.48550/arXiv.2106.04803.
3. LIU Zhuang ir kt. A ConvNet for the 2020s. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, 11966–11976. Prieiga per: doi: https://doi.org/10.1109/CVPR52688.2022.01167.
4. XIE, Enze ir kt. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. *Advances in Neural Information Processing Systems*. 2021, 34, 12077–12090. Prieiga per: SegFormer - NeurIPS Proceedings.
5. MEHTA, Yashasvi, Abdullah BAZ ir Shobhit K. PATEL. Semantic segmentation of optical satellite images for the illegal construction detection using transfer learning. *Results in Engineering*. 2024, 24, 103383. Prieiga per: doi: https://doi.org/10.1016/j.rineng.2024.103383
6. LI, Qingyuir kt. Detection of Undocumented Building Constructions from Official Geodata Using a Convolutional Neural Network. *Remote Sensing*. 2020, 12(21), 3537. Prieiga per: doi: https://doi.org/10.3390/rs12213537.
7. SHELHAMER, Evan, Jonathan LONG ir Trevor DARRELL. Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2017, 39(4), 640–651. Prieiga per: doi: https://doi.org/10.1109/TPAMI.2016.2572683
8. GOODFELLOW, Ian, Yoshua Bengio ir Aaron Courville. *Deep Learning.* Cambridge, MA: MIT Press, 2016. ISBN 978-0262035613.
9. GARCIA-GARCIA, Alberto ir kt. A Review on Deep Learning Techniques Applied to Semantic Segmentation. *arXiv*. 2017, 1704.06857. Prieiga per: doi: https://doi.org/10.48550/arXiv.1704.06857.
10. SIMONYAN, Karen ir Andrew ZISSERMAN. Very Deep Convolutional Networks for Large-Scale Image Recognition. *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015).* 2015, 1–14. Prieiga per: doi: https://doi.org/10.48550/arXiv.1409.1556.
11. HE, Kaiming ir kt. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, 770–778. Prieiga per: doi: https://doi.org/10.1109/CVPR.2016.90.
12. IBM. *What is an encoder-decoder model?* [interaktyvus]. 2024 [žiūrėta 2025-05-14]. Prieiga per: https://www.ibm.com/think/topics/encoder-decoder-model.
13. IBM. *What Is an Autoencoder?* [interaktyvus]. 2023 [žiūrėta 2025-05-14]. Prieiga per: https://www.ibm.com/think/topics/autoencoder.
14. RONNEBERGER, Olaf, Philipp FISCHER ir Thomas BROX. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Lecture Notes in Computer Science*. 2015, 9351, 234–241. Prieiga per: doi: https://doi.org/10.1007/978-3-319-24574-4_28.
15. LI, Weijia ir kt. Semantic Segmentation-Based Building Footprint Extraction Using Very High-Resolution Satellite Images and Multi-Source GIS Data. *Remote Sensing*. 2019, 11(4), 403. Prieiga per: doi: https://doi.org/10.3390/rs11040403.

16. ZHOU, Zongwei ir kt. UNet++: A Nested U-Net Architecture for Medical Image Segmentation. *Lecture Notes in Computer Science*. 2018, 11045, 3–11. Prieiga per: doi: https://link.springer.com/chapter/10.1007/978-3-030-00889-5_1.
17. CHEN, Liang-Chieh ir kt. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2018, 40(4). Prieiga per: doi: https://doi.org/10.1109/TPAMI.2017.2699184.
18. CHEN, Liang Chieh ir kt. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2018, 11211 LNCS, 833–851. Prieiga per: doi: http://dx.doi.org/10.1007/978-3-030-01234-2_49.
19. LI, Hanchao ir kt. Pyramid Attention Network for Semantic Segmentation. *arXiv*. 2018, 1805.10180. Prieiga per: doi: https://doi.org/10.48550/arXiv.1805.10180.
20. FAN, Tongle ir kt. Ma-net: A multi-scale attention network for liver and tumor segmentation. *IEEE Access*. 2020, 8, 179656–179665. Prieiga per: doi: http://dx.doi.org/10.1109/ACCESS.2020.3025372.
21. CHAURASIA, Abhishek ir Eugenio CULURCIELLO. LinkNet: Exploiting Encoder Representations for Efficient Semantic Segmentation. *2017 IEEE Visual Communications and Image Processing, VCIP 2017*. 2017, 1–4. Prieiga per: doi: https://doi.org/10.48550/arXiv.1707.03718.
22. KIRILLOV, Alexander ir kt. Panoptic Feature Pyramid Networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, 5804–5813. Prieiga per: doi: https://doi.org/10.1109/CVPR.2019.00656.
23. ZHAO, Hengshuang ir kt. Pyramid Scene Parsing Network. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, 2881–2890. Prieiga per: doi: https://doi.org/10.1109/CVPR.2017.660.
24. XIAO, Tete ir kt. Unified Perceptual Parsing for Scene Understanding. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2018, 11209 LNCS, 432–448. Prieiga per: doi: https://doi.org/10.48550/arXiv.1807.10221.
25. RANFTL, René, Alexey BOCHKOVSKIY ir Vladlen KOLTUN. Vision Transformers for Dense Prediction. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, 12159–12168. Prieiga per: doi: https://doi.org/10.1109/ICCV48922.2021.01196.
26. XIE, Saining ir kt. Aggregated Residual Transformations for Deep Neural Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, 5987–5995. Prieiga per: doi: https://doi.org/10.1109/CVPR.2017.634.
27. HUANG, Gao ir kt. Densely Connected Convolutional Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, 2261–2269. Prieiga per: doi: https://doi.org/10.1109/CVPR.2017.243.
28. WOO, Sanghyun ir kt. ConvNeXt V2: Co-designing and Scaling ConvNets with Masked Autoencoders. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, 16133–16142. Prieiga per: doi: https://doi.org/10.48550/arXiv.2301.00808
29. TU, Zhengzhong ir kt. MaxViT: Multi-Axis Vision Transformer. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2022, 13684 LNCS, 459–479. Prieiga per: doi: https://doi.org/10.48550/arXiv.2204.01697.

30. HOLDER, Christopher J ir Muhammad SHAFIQUE. On Efficient Real-Time Semantic Segmentation: A Survey. 2022. Prieiga per: doi: https://doi.org/10.48550/arXiv.2206.08605.
31. RADOSAVOVIC, Ilija ir kt. Designing Network Design Spaces. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, 10425–10433. Prieiga per: doi: https://doi.org/10.1109/CVPR42600.2020.01044.
32. ZHANG, Hang ir kt. ResNeSt: Split-Attention Networks. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2022, 2735–2745. Prieiga per: doi: https://doi.org/10.1109/CVPRW56347.2022.00309.
33. LIU, Ze ir kt. Swin Transformer V2: Scaling Up Capacity and Resolution. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, 11999–12009. Prieiga per: doi: https://doi.org/10.1109/CVPR52688.2022.01170.
34. WANG, Wenhai ir kt. PVT v2: Improved Baselines with Pyramid Vision Transformer. *Computational Visual Media*. 2021, 8(3), 415–424. Prieiga per: doi: https://doi.org/10.1007/s41095-022-0274-8.
35. SRINIVAS, Aravind ir kt. Bottleneck Transformers for Visual Recognition. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, 16514–16524. Prieiga per: doi: https://doi.org/10.1109/CVPR46437.2021.01625.
36. BELLO, Irwan. LambdaNetworks: Modeling Long-Range Interactions Without Attention. *ICLR 2021 - 9th International Conference on Learning Representations*. 2021. Prieiga per: doi: https://doi.org/10.48550/arXiv.2102.08602.
37. GUO, Meng-Hao ir kt. Attention Mechanisms in Computer Vision: A Survey. *Computational Visual Media*. 2021, 8(3), 331–368. Prieiga per: doi: https://doi.org/10.48550/arXiv.2111.07624.
38. ZHANG, Cheng ir kt. Transformer and CNN Hybrid Deep Neural Network for Semantic Segmentation of Very-High-Resolution Remote Sensing Imagery; Transformer and CNN Hybrid Deep Neural Network for Semantic Segmentation of Very-High-Resolution Remote Sensing Imagery. *IEEE Transactions on Geoscience and Remote Sensing*. 2022, 60, 4408820. Prieiga per: doi: https://doi.org/10.1109/TGRS.2022.3144894.
39. CHANG, Jing ir kt. Multi-Scale Attention Network for Building Extraction from High-Resolution Remote Sensing Images. *Sensors 2024*. 2024, 24(3), 1010. Prieiga per: doi: https://doi.org/10.3390/s24031010.
40. CHEN, Xiangyu ir kt. HAT: Hybrid Attention Transformer for Image Restoration. 2023. Prieiga per: doi: https://doi.org/10.48550/arXiv.2309.05239.
41. OKTAY, Ozan ir kt. Attention U-Net: Learning Where to Look for the Pancreas. 2018. *arXiv*. Prieiga per: doi: https://doi.org/10.48550/arXiv.1804.03999.
42. WENG, Weihao ir Xin ZHU. U-Net: Convolutional Networks for Biomedical Image Segmentation. *IEEE Access*. 2015, 9, 16591–16603. Prieiga per: doi: https://doi.org/10.1109/ACCESS.2021.3053408.
43. NODIROV, Jakhongir, Akmalbek Bobomirzaevich ABDUSALOMOV ir Taeg Keun WHANGBO. Attention 3D U-Net with Multiple Skip Connections for Segmentation of Brain Tumor Images. *Sensors*. 2022, 22(17). Prieiga per: doi: https://doi.org/10.3390/s22176501.
44. DOSOVITSKIY, Alexey ir kt. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *ICLR 2021 - 9th International Conference on Learning Representations*. 2020. Prieiga per: doi: https://doi.org/10.48550/arXiv.2010.11929.
45. ZHENG, Sixiao ir kt. Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers. *2021 IEEE/CVF Conference on Computer Vision and Pattern*

Recognition (CVPR). 2021, 6877–6886. Prieiga per: doi: https://doi.org/10.1109/CVPR46437.2021.00681.
46. STRUDEL, Robin ir kt. Segmenter: Transformer for Semantic Segmentation. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, 7242–7252. Prieiga per: doi: https://doi.org/10.1109/ICCV48922.2021.00717.
47. LIU, Ze ir kt. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, 9992–10002. Prieiga per: doi: https://doi.org/10.1109/ICCV48922.2021.00986.
48. YU, Xizi, Shuang LI ir Yu ZHANG. Incorporating convolutional and transformer architectures to enhance semantic segmentation of fine-resolution urban images. *European Journal of Remote Sensing*. 2024, 57(1). Prieiga per: doi: https://doi.org/10.1080/22797254.2024.2361768.
49. CHEN, Jieneng ir kt. TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation. *arXiv*. 2021, 2102.04306. Prieiga per: doi: https://doi.org/10.48550/arXiv.2102.04306.
50. MANZARI, Omid Nejati ir kt. BEFUnet: A Hybrid CNN-Transformer Architecture for Precise Medical Image Segmentation. 2024. Prieiga per: doi: https://doi.org/10.48550/arXiv.2402.08793.
51. ZHANG, Chong ir kt. A dual-branch and dual attention transformer and CNN hybrid network for ultrasound image segmentation. *Frontiers in Physiology*. 2024, 15, 1432987. Prieiga per: doi: https://doi.org/10.3389/fphys.2024.1432987.
52. CHENG, Bowen, Alexander G. SCHWING ir Alexander KIRILLOV. Per-Pixel Classification is Not All You Need for Semantic Segmentation. *Advances in Neural Information Processing Systems*. 2021, 22, 17864–17875. Prieiga per: https://doi.org/10.48550/arXiv.2107.06278.
53. CARION, Nicolas ir kt. End-to-End Object Detection with Transformers. *Lecture Notes in Computer Science*. 2020. Prieiga per: doi: https://doi.org/10.1109/ICCV48922.2021.00298.
54. CHENG, Bowen ir kt. Masked-attention Mask Transformer for Universal Image Segmentation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, 1280–1289. Prieiga per: doi: https://doi.org/10.1109/CVPR52688.2022.00135.
55. CAVAGNERO, Niccoì O ir kt. PEM: Prototype-based Efficient MaskFormer for Image Segmentation. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024, 15804–15813. Prieiga per: doi: https://doi.org/10.1109/CVPR52733.2024.01496.
56. GUI, Jie ir kt. A Survey on Self-supervised Learning: Algorithms, Applications, and Future Trends. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2023, 14(8). Prieiga per: doi: https://doi.org/10.1109/TPAMI.2024.3415112.
57. CHEN, Ting ir kt. A Simple Framework for Contrastive Learning of Visual Representations. *37th International Conference on Machine Learning, ICML 2020*. 2020, PartF168147-3, 1575–1585. Prieiga per: doi: https://doi.org/10.48550/arXiv.2002.05709.
58. HE, Kaiming ir kt. Momentum Contrast for Unsupervised Visual Representation Learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, 9726–9735. Prieiga per: doi: https://doi.org/10.1109/CVPR42600.2020.00975.
59. GRILL, Jean Bastien ir kt. Bootstrap your own latent: A new approach to self-supervised Learning. *Advances in Neural Information Processing Systems*. 2020. Prieiga per: doi: https://doi.org/10.48550/arXiv.2006.07733.

60. NAKAMURA, Hiroki, Masashi OKADA ir Tadahiro TANIGUCHI. Representation Uncertainty in Self-Supervised Learning as Variational Inference. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2022, 16438–16447. Prieiga per: https://doi.org/10.1109/ICCV51070.2023.01511.
61. DEVLIN, Jacob ir kt. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2019, 1, 4171–4186. Prieiga per: doi: https://doi.org/10.18653/v1/N19-1423.
62. LI, Yangyang ir kt. Masked Image Modeling Boosting Semi-Supervised Semantic Segmentation. *arXiv*. 2024, 2411.08756. Prieiga per: doi: https://doi.org/10.48550/arXiv.2411.08756.
63. HE, Kaiming ir kt. Masked Autoencoders Are Scalable Vision Learners. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, 15979–15988. Prieiga per: doi: https://doi.org/10.1109/CVPR52688.2022.01553.
64. BAO, Hangbo ir kt. BEiT: BERT Pre-Training of Image Transformers. *ICLR 2022 - 10th International Conference on Learning Representations*. 2022. Prieiga per: https://doi.org/10.48550/arXiv.2106.08254.
65. XIE, Zhenda ir kt. SimMIM: A Simple Framework for Masked Image Modeling. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, 9643–9653. Prieiga per: doi: https://doi.org/10.1109/CVPR52688.2022.00943.
66. WANG, Yuan ir kt. Fully Self-Supervised Learning for Semantic Segmentation. 2022. *arXiv*. Prieiga per: doi: https://doi.org/10.48550/arXiv.2202.11981.
67. CHEN, Jialei ir kt. CLIP Is Also a Good Teacher: A New Learning Framework for Inductive Zero-shot Semantic Segmentation. *arXiv*. 2023, 2310.02296. Prieiga per: doi: https://doi.org/10.48550/arXiv.2310.02296.
68. GUO, Shi Cheng ir kt. CLIP-Driven Prototype Network for Few-Shot Semantic Segmentation. *Entropy*. 2023, 25(9), 1353. Prieiga per: doi: https://doi.org/10.3390/e25091353.
69. KAWANO, Yasufumi ir Yoshimitsu AOKI. MaskDiffusion: Exploiting Pre-trained Diffusion Models for Semantic Segmentation. *IEEE Access*. 2024, 12, 127283–127293. Prieiga per: doi: https://doi.org/10.1109/ACCESS.2024.3456442.
70. SHI, Yaqing ir kt. Diffusion Models for Medical Image Computing: A Survey. *Tsinghua Science and Technology*. 2025, 30(1), 357–383. Prieiga per: doi: https://doi.org/10.26599/TST.2024.9010047.
71. JU, Cheng, Aurélien BIBAUT ir Mark VAN DER LAAN. The Relative Performance of Ensemble Methods with Deep Convolutional Neural Networks for Image Classification. *Journal of Applied Statistics*. 2017, 45(15), 2800–2818. Prieiga per: doi: https://doi.org/10.48550/arXiv.1704.01664.
72. SHAGA DEVAN, Kavitha ir kt. Weighted average ensemble-based semantic segmentation in biological electron microscopy images. *Histochemistry and Cell Biology*. 2022, 158(5), 447–462. Prieiga per: doi: http://dx.doi.org/10.1007/s00418-022-02148-3.
73. ASGHARI OSKOEI, Mohammadreza ir Huosheng HU. Myoelectric control systems-A survey. *Biomedical Signal Processing and Control*. 2007, 2(4), 275–294. Prieiga per: doi: https://doi.org/10.1016/j.bspc.2007.07.009.
74. CAI, Yuanzhi, Lei FAN ir Yuan FANG. SBSS: Stacking-Based Semantic Segmentation Framework for Very High Resolution Remote Sensing Image. *IEEE Transactions on*

Geoscience and Remote Sensing. 2022, 61. Prieiga per: doi: https://doi.org/10.1109/TGRS.2023.3234549.

75. KIM, Yong Woon, Yung Cheol BYUN ir Addapalli V.N. KRISHNA. Portrait Segmentation Using Ensemble of Heterogeneous Deep-Learning Models. *Entropy*. 2021, 23(2), 197. Prieiga per: doi: https://doi.org/10.3390/e23020197.
76. HUANG, Bohao ir kt. Tiling and Stitching Segmentation Output for Remote Sensing: Basic Challenges and Recommendations. *arXiv*. 2018, 1805.12219. Prieiga per: doi: https://doi.org/10.48550/arXiv.1805.12219.
77. ABRAHAMS, Ellianna ir kt. A Concise Tiling Strategy for Preserving Spatial Context in Earth Observation Imagery. *arXiv*. 2024, 2404.10927. Prieiga per: doi: https://doi.org/10.48550/arXiv.2404.10927.
78. CORLEY, Isaac ir kt. Revisiting pre-trained remote sensing model benchmarks: resizing and normalization matters. *arXiv*. 2023, 2305.13456. Prieiga per: doi: https://doi.org/10.48550/arXiv.2305.13456.
79. CHENG, Xiaohui ir kt. MMDL-Net: Multi-Band Multi-Label Remote Sensing Image Classification Model. *Applied Sciences*. 2024, 14(6), 2226. Prieiga per: doi: https://doi.org/10.3390/app14062226.
80. CHEN, Peimin ir kt. A benchmark GaoFen-7 dataset for building extraction from satellite images. *Scientific Data*. 2024, 11(1), 187. Prieiga per: doi: https://doi.org/10.1038/s41597-024-03009-5.
81. GHAFFAR, Mohamed A. A ir kt. Data Augmentation Approaches for Satellite Image Super-Resolution. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. 2019, 4(2/W7), 47–54. Prieiga per: http://dx.doi.org/10.5194/isprs-annals-IV-2-W7-47-2019.
82. SHAN, Zhe ir kt. ROS-SAM: High-Quality Interactive Segmentation for Remote Sensing Moving Object. *arXiv*. 2025, 2503.12006. Prieiga per: doi: https://doi.org/10.48550/arXiv.2503.12006.
83. AYALA, Christian ir kt. Multi-temporal data augmentation for high frequency satellite imagery: a case study in Sentinel-1 and Sentinel-2 building and road segmentation. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*. 2022, 43(B3-2022), 25–32. Prieiga per: doi: http://dx.doi.org/10.5194/isprs-archives-XLIII-B3-2022-25-2022.
84. JADON, Shruti. A survey of loss functions for semantic segmentation. *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, CIBCB 2020*. 2020, 9277638. Prieiga per: doi: https://doi.org/10.1109/CIBCB48159.2020.9277638.
85. ELHARROUSS, Omar ir kt. Loss Functions in Deep Learning: A Comprehensive Review. *arXiv*. 2025, 2504.04242. Prieiga per: doi: https://doi.org/10.48550/arXiv.2504.04242.
86. AZAD, Reza ir kt. Loss Functions in the Era of Semantic Segmentation: A Survey and Outlook. *arXiv*. 2023, 2312.05391. Prieiga per: doi: https://doi.org/10.48550/arXiv.2312.05391.
87. VALVERDE, Juan Miguel ir Jussi TOHKA. Region-wise loss for biomedical image segmentation. *Pattern Recognition*. 2023, 136, 109208. Prieiga per: doi: https://doi.org/10.1016/j.patcog.2022.109208.
88. BLACK, Dale, Wenbo LI ir Sabee MOLLOI. Optimized Hausdorff Distance Loss Function Based on a GPU-Accelerated Distance Transform. *TechRxiv*. 2023, 23612100. Prieiga per: doi: http://dx.doi.org/10.36227/techrxiv.23612100.

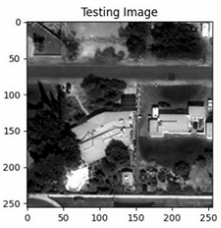
89. SUN, Fan, Zhiming LUO ir Shaozi LI. Boundary Difference over Union Loss for Medical Image Segmentation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2023, 14223 LNCS, 292–301. Prieiga per: doi: https://doi.org/10.48550/arXiv.2308.00220.
90. KINGMA, Diederik P ir Jimmy Lei BA. Adam: A Method for Stochastic Optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. 2014. Prieiga per: doi: https://doi.org/10.48550/arXiv.1412.6980.
91. BOCK, Sebastian, Josef GOPPOLD ir Martin WEISS. An improvement of the convergence proof of the ADAM-Optimizer. *arXiv*. 2018, 1804.10587. Prieiga per: doi: https://doi.org/10.48550/arXiv.1804.10587.
92. LOSHCHILOV, Ilya ir Frank HUTTER. Decoupled Weight Decay Regularization. *7th International Conference on Learning Representations, ICLR 2019*. 2017. Prieiga per: doi: https://doi.org/10.48550/arXiv.1711.05101.
93. ZHOU, Pan ir kt. Towards Understanding Convergence and Generalization of AdamW. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2024, 46(9), 6486–6493. Prieiga per: doi: https://doi.org/10.1109/TPAMI.2024.3382294.
94. D'ANGELO, Francesco ir kt. Why Do We Need Weight Decay in Modern Deep Learning? 2023. Prieiga per: https://doi.org/10.48550/arXiv.2310.04415.
95. HENDRYCKS, Dan ir Kevin GIMPEL. Gaussian Error Linear Units (GELUs). *arXiv*. 2016, 1606.08415. Prieiga per: doi: https://doi.org/10.48550/arXiv.1606.08415.
96. NIERADZIK, Lars ir kt. Effect of the output activation function on the probabilities and errors in medical image segmentation. *arXiv*. 2021, 2109.00903. Prieiga per: doi: https://doi.org/10.48550/arXiv.2109.00903.
97. SINGLA, Jai G ir Bakul VAGHELA. Semantic segmentation on multi-resolution optical and microwave data using deep learning. *arXiv*. 2024, 2411.07581. Prieiga per: doi: https://doi.org/10.48550/arXiv.2411.07581.
98. TOUZANI, Samir ir Jessica GRANDERSON. Open Data and Deep Semantic Segmentation for Automated Extraction of Building Footprints. *Remote Sensing*. 2021, 13(13), 2578. Prieiga per: doi: https://doi.org/10.3390/rs13132578.
99. DARANAGAMA, Samitha ir kt. Automatic Building Detection with Polygonizing and Attribute Extraction from High-Resolution Images. *ISPRS International Journal of Geo-Information*. 2021, 10(9), 606. Prieiga per: doi: https://doi.org/10.3390/ijgi10090606.
100. YU, Haojia ir kt. SuperpixelGraph: Semi-automatic generation of building footprint through semantic-sensitive superpixel and neural graph networks. *International Journal of Applied Earth Observation and Geoinformation*. 2023, 125, 103556. Prieiga per: doi: https://doi.org/10.1016/j.jag.2023.103556.
101. CHEN, Qi ir kt. An end-to-end shape modeling framework for vectorized building outline generation from aerial images. *ISPRS Journal of Photogrammetry and Remote Sensing*. 2020, 170, 114–126. Prieiga per: doi: https://doi.org/10.1016/j.isprsjprs.2020.10.008.
102. LIN, Tsung-Yi ir kt. Feature Pyramid Networks for Object Detection. 2016. Prieiga per: doi: https://doi.org/10.48550/arXiv.1612.03144.
103. CIRA, Calimanut Ionut ir kt. Insights into the Effects of Tile Size and Tile Overlap Levels on Semantic Segmentation Models Trained for Road Surface Area Extraction from Aerial Orthophotography. *Remote Sensing*. 2024, 16(16), 2954. Prieiga per: doi: https://doi.org/10.3390/rs16162954.

104. LU, Zhuheng ir kt. Fine-grained Metrics for Point Cloud Semantic Segmentation. *Pattern Recognition and Computer Vision, PRCV 2024, Lecture Notes in Computer Science*. 2025, 15041, 16. Prieiga per: doi: http://dx.doi.org/10.1007/978-981-97-8795-1_16.
105. HURTADO, Juana Valeria ir Abhinav VALADA. Semantic scene segmentation for robotics. *Deep Learning for Robot Perception and Cognition*. 2022, 279–311. Prieiga per: doi: https://doi.org/10.1016/B978-0-32-385787-1.00017-8.
106. Evidently AI. *Accuracy, precision, and recall in multi-class classification* [interaktyvus]. 2025 [žiūrėta 2025-05-14]. Prieiga per: https://www.evidentlyai.com/classification-metrics/multi-class-metrics.
107. ROBBINS, Herbert ir Sutton Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*. 1951, 22(3), 400–407. Prieiga per: doi: https://doi.org/10.1214/aoms/1177729586.
108. LAN, Qingfeng ir kt. Learning to Optimize for Reinforcement Learning. *arXiv*. 2023, 2302.01470. Prieiga per: doi: https://doi.org/10.48550/arXiv.2302.01470.
109. LIU, Liyuan ir kt. On the Variance of the Adaptive Learning Rate and Beyond. *8th International Conference on Learning Representations, ICLR 2020*. 2020. Prieiga per: doi: https://doi.org/10.48550/arXiv.1908.03265.
110. CHEN, Xiangning ir kt. Symbolic Discovery of Optimization Algorithms. *arXiv*. 2023, 2302.06675. Prieiga per: doi: https://arxiv.org/pdf/2302.06675.
111. WRIGHT, Less ir Nestor DEMEURE. Ranger21: a synergistic deep learning optimizer. *arXiv*. 2021, 2106.13731. Prieiga per: doi: https://arxiv.org/pdf/2106.13731.

Priedai

1 priedas. Pirmasis tyrimas, kuris buvo publikuotas DAMSS 2023

Classification of satellite images to create a map of the settlement area



ABSTRACT

This study aims to use advanced deep learning algorithms to automate the classification of satellite-taken land images, in order to facilitate the creation of detailed urban area maps. The study was conducted using object detection algorithms like AlexNet, VGGNet, InceptionNet, and ResNet. Classification of satellite images was done using the TensorFlow machine-learning framework [1]. Key performance metrics, such as Intersection Over Union (IOU), Dice Coefficient, and Pixel Accuracy [2], were employed to evaluate the accuracy of the classification.

PROJECT DESCRIPTION

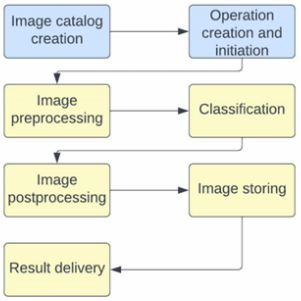
The aim of this study is to innovate and push forward in the field of urban mapping by blending state-of-the-art deep learning techniques with user-oriented software design. Users can upload satellite images to the system, which then utilizes cutting-edge algorithms for building detection, enhancing the precision of urban structure maps.

Central to this methodology is tackling the binary semantic segmentation challenge using specialized neural network models such as U-Net, optimized for high-precision image classification.

OBJECTIVES

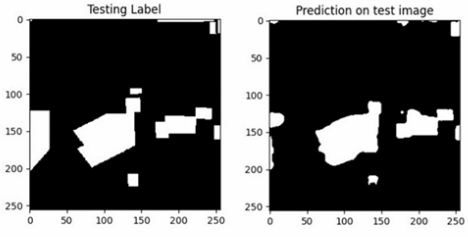
1. **Development of an automated classification platform.** A system allows users to submit satellite images effortlessly. Technologies: .NET 6 and C# with Blazor server framework.
2. **Data gathering.** This study emphasizes acquiring high-quality satellite images. It is crucial for the accuracy and effectiveness of the urban classification system.
3. **Data processing and model training.** The project focuses on refining data processing techniques and training robust deep learning models. Technologies: Python and TensorFlow.

FLOW DIAGRAM

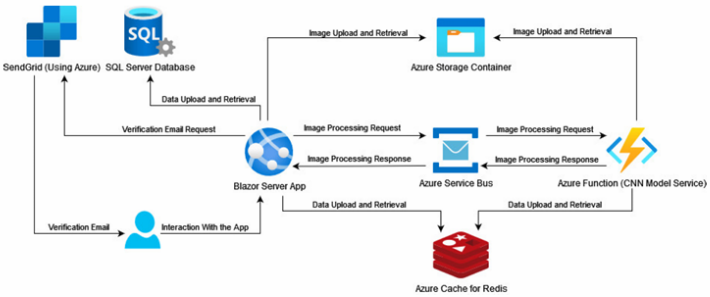


RESULTS

Convolutional Neural Network (CNN) models were used classification of satellite images in urban mapping. Metrics such as Intersection Over Union (IOU) and Pixel Accuracy were evaluated to ensure the accuracy of building segmentations in satellite images.

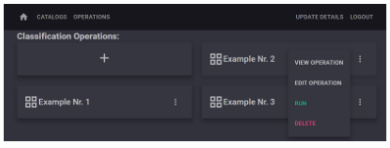


PLATFORM ARCHITECTURE



Azure Cloud: Serves as the foundational cloud environment, integrating and supporting all system components for efficient and scalable system workflow.
Azure Blob Storage Container: Stores large volumes of image data, offering scalability and easy access for processing satellite images.
Azure App Service: Hosts the system's main web application, ensuring scalable, secure user interactions and streamlined updates.
Azure SQL Server Database: Manages relational data, including user details, operation configurations, image metadata, and processing results, with robust capabilities.
Azure Function: Executes the CNN model, managing the intricate image analysis and calculation logic within a scalable, serverless environment.
Azure Service Bus: Facilitates asynchronous messaging between the main application and CNN function, enhancing communication and service decoupling.
Azure Cache for Redis: Provides high-speed caching to reduce latency and speed up data access in frequent operations.
SendGrid SaaS: Manages automated email communications such as verification emails and password resets, enhancing user engagement and system security.

PROTOTYPE OF THE SYSTEM



CONCLUSIONS

The proposed classification system demonstrates the potential to efficiently identify buildings from other objects in each area that could be used for more precise urban planning and infrastructure development strategies. The presented system is capable of generating accurate urban maps and identifying unauthorized constructions, thereby contributing to enhanced infrastructure planning.
Future research should include other one-stage and two-stage deep learning-based object detection algorithms/post-processing techniques. Additionally, we are planning to increase the diversity of the training dataset in order to improve the generalizability of the model.

REFERENCES

[1] N. Weir, D. Lindenbaum et.al. SpaceNet MVOI: a Multi-View Overhead Imagery Dataset. Computer Vision Foundation. 2019
[2] H. Choi, L. Hyun-Jik et.al. Comparative Analysis of Generalized Intersection over Union and Error Matrix for Vegetation Cover Classification Assessment. Sensors and Materials, vol. 31, no. 11 2019.

L. Buinauskas , A. Gadeikytė, E. Butkevičiūtė
Faculty of Informatics, KTU

2 priedas. Dalyvavimo DAMSS 2023 sertifikatas

DAMSS
DATA ANALYSIS METHODS FOR SOFTWARE SYSTEMS

CERTIFICATE OF PARTICIPATION

This is to certify that

Laurynas Buinauskas

Kaunas University of Technology

has attended the 14th conference

Data Analysis Methods for Software Systems

held in Druskininkai, Lithuania, November 30 - December 2, 2023
and has presented the paper

Classification of Satellite Images to Create a Map of the Settlement Area

Program Committee

Dr. Saulius Maskeliūnas

Prof. Gintautas Dzemyda

Druskininkai, 1 December 2023

LIETUVIAI ACADEMIA SCIENTIARUM

VILNIAUS UNIVERSITETAS · 1579 · UNIVERSITAS VILNENSIS

LIKS

3 priedas. Pagal *IoU* metriką surikiuotų testuotų bazinių modelių ir parametrų kombinacijos

Bazinis modelis	Parametrai	Laikas (s)	IoU	PA	Dice	Prec	Recall	ROC AUC	Hausd
convnextv2_huge	imagenet	616,301	0,772	0,92	0,868	0,87	0,88	0,95	39,461
convnextv2_large	imagenet	377,212	0,767	0,918	0,865	0,868	0,877	0,956	39,114
convnext_xxlarge	imagenet	617,641	0,767	0,918	0,864	0,873	0,869	0,948	38,635
convnextv2_large	fcmae_ft_in22k_in1k	444,374	0,766	0,918	0,864	0,871	0,872	0,957	39,276
convnextv2_large	fcmae	436,222	0,766	0,918	0,864	0,869	0,876	0,953	37,393
focalnet_huge_fl4	imagenet	766,007	0,765	0,916	0,863	0,854	0,891	0,96	41,349
focalnet_huge_fl3	imagenet	520,329	0,762	0,917	0,862	0,878	0,862	0,951	40,597
convnextv2_base	imagenet	282,941	0,762	0,916	0,862	0,863	0,877	0,953	41,542
focalnet_xlarge_fl4	imagenet	575,479	0,762	0,917	0,861	0,871	0,869	0,953	43,171
resnext101_32x16d	ssl	385,653	0,762	0,916	0,861	0,865	0,874	0,959	43,223
regnety_640	imagenet	405,392	0,759	0,916	0,86	0,872	0,866	0,954	39,004
resnext101_32x8d	instagram	256,869	0,759	0,914	0,859	0,857	0,879	0,95	39,966
resnext101_32x8d	swsl	275,214	0,758	0,914	0,858	0,856	0,88	0,955	43,107
maxxvitv2_nano_rw_256	imagenet	194,073	0,757	0,915	0,858	0,862	0,874	0,961	42,341
pvt_v2_b2_li	imagenet	203,732	0,757	0,914	0,858	0,862	0,873	0,962	41,398
convnext_tiny	in12k_ft_in1k	201,052	0,757	0,915	0,858	0,863	0,872	0,955	38,421
mit_b2	imagenet	214,053	0,757	0,912	0,858	0,844	0,892	0,962	41,296
convnext_tiny	fb_in22k	203,327	0,757	0,913	0,858	0,845	0,89	0,958	39,659
resnext101_32x16d	swsl	282,868	0,756	0,913	0,857	0,849	0,886	0,957	42,678
convnextv2_base	fcmae	257,471	0,756	0,914	0,858	0,863	0,87	0,948	40,87
convnextv2_base	fcmae_ft_in22k_in1k	253,405	0,756	0,914	0,857	0,855	0,879	0,951	41,53
focalnet_large_fl3	imagenet	460,635	0,756	0,914	0,858	0,866	0,865	0,95	43,898
pvt_v2_b4	imagenet	230,934	0,756	0,914	0,857	0,859	0,874	0,964	40,331
mit_b4	imagenet	242,446	0,755	0,911	0,857	0,835	0,901	0,957	39,983

resnest200e	imagenet	258,471	0,755	0,914	0,857	0,867	0,864	0,946	41,3
maxvit_rmlp_nano_rw_256	imagenet	195,807	0,755	0,914	0,857	0,866	0,866	0,957	41,651
convnext_base	fb_in22k_ft_in1k	241,749	0,755	0,912	0,856	0,844	0,889	0,955	40,809
focalnet_xlarge_fl3	imagenet	557,277	0,755	0,914	0,856	0,872	0,859	0,947	41,573
convnext_xlarge	fb_in22k_ft_in1k	253,18	0,755	0,914	0,857	0,865	0,865	0,951	41,7
convnext_tiny	in12k	207,495	0,754	0,913	0,857	0,864	0,864	0,951	39,681
focalnet_base_srf	imagenet	245,554	0,753	0,911	0,855	0,843	0,889	0,953	41,81
focalnet_tiny_srf	imagenet	210,884	0,753	0,912	0,855	0,851	0,879	0,955	40,296
regnety_160	imagenet	251,919	0,753	0,912	0,855	0,853	0,876	0,95	40,873
convnext_xlarge	imagenet	403,424	0,753	0,914	0,855	0,87	0,859	0,946	40,599
tf_efficientnetv2_l	imagenet	290,132	0,753	0,91	0,855	0,836	0,896	0,957	39,197
pvt_v2_b5	imagenet	259,466	0,752	0,91	0,855	0,834	0,899	0,96	40,34
swinv2_tiny_window16_256	imagenet	205,362	0,752	0,913	0,855	0,86	0,869	0,959	41,053
focalnet_large_fl4	imagenet	465,936	0,752	0,912	0,855	0,853	0,874	0,955	42,458
mit_b3	imagenet	223,064	0,752	0,913	0,855	0,861	0,87	0,955	42,37
swinv2_small_window16_256	imagenet	233,15	0,752	0,913	0,855	0,863	0,866	0,961	38,174
swinv2_base_window12to16_192to256	imagenet	264,157	0,752	0,913	0,855	0,861	0,868	0,962	39,73
mit_b1	imagenet	210,719	0,752	0,913	0,854	0,863	0,866	0,962	39,994
swinv2_small_window8_256	imagenet	207,724	0,751	0,911	0,854	0,846	0,882	0,95	40,995
convnext_large	fb_in22k	457,109	0,751	0,911	0,854	0,855	0,871	0,947	42,921
convnext_large	fb_in22k_ft_in1k	443,899	0,751	0,912	0,854	0,853	0,875	0,955	39,534
convnext_small	imagenet	240,025	0,751	0,912	0,854	0,857	0,87	0,947	42,207
convnextv2_tiny	imagenet	242,413	0,751	0,912	0,853	0,854	0,875	0,954	40,788
resnext101_32x8d	ssl	258,616	0,751	0,911	0,854	0,851	0,878	0,957	43,597
convnext_xlarge	fb_in22k	241,744	0,751	0,913	0,854	0,867	0,86	0,954	40,985
maxxvit_rmlp_small_rw_256	imagenet	214,092	0,75	0,912	0,854	0,857	0,873	0,966	40,445

regnety_120	imagenet	228,345	0,75	0,911	0,853	0,847	0,879	0,955	41,258
convnext_base	fb_in22k	240,434	0,75	0,911	0,853	0,854	0,872	0,95	41,297
convnext_base	clip_laion2b_augreg_ft_in1k	246,396	0,75	0,911	0,853	0,852	0,874	0,953	40,245
pvt_v2_b3	imagenet	218,935	0,749	0,911	0,853	0,859	0,865	0,959	40,758
swinv2_base_window16_256	imagenet	231,368	0,749	0,914	0,853	0,884	0,843	0,95	38,938
swinv2_tiny_window8_256	imagenet	196,136	0,749	0,911	0,853	0,847	0,881	0,958	40,467
resnest101e	imagenet	233,279	0,749	0,912	0,853	0,861	0,864	0,955	41,594
convnext_tiny	fb_in22k_ft_in1k	208,387	0,749	0,912	0,853	0,856	0,871	0,956	41,655
pvt_v2_b2	imagenet	205,53	0,749	0,911	0,852	0,854	0,871	0,957	42,003
swinv2_base_window8_256	imagenet	209,293	0,749	0,909	0,852	0,836	0,89	0,96	41,028
mit_b5	imagenet	271,43	0,749	0,91	0,852	0,851	0,872	0,952	43,426
focalnet_small_srf	imagenet	216,777	0,748	0,909	0,852	0,836	0,89	0,957	42,207
resnet50	swsl	210,632	0,748	0,909	0,852	0,843	0,881	0,951	41,993
maxvit_rmlp_tiny_rw_256	imagenet	208,429	0,748	0,912	0,852	0,859	0,866	0,96	40,677
convnext_tiny	imagenet	626,437	0,747	0,911	0,852	0,864	0,857	0,947	42,426
regnety_320	imagenet	321,705	0,747	0,911	0,852	0,864	0,858	0,946	42,145
convnext_large	imagenet	252,514	0,747	0,911	0,851	0,861	0,86	0,952	38,067
senet154	imagenet	240,116	0,746	0,911	0,851	0,863	0,858	0,949	43,885
resnest50d	imagenet	199,504	0,746	0,911	0,851	0,87	0,851	0,941	39,591
resnext101_32x16d	instagram	338,152	0,746	0,906	0,851	0,821	0,902	0,956	38,937
efficientnet-b7	advprop	269,828	0,746	0,91	0,851	0,86	0,859	0,953	40,02
convnext_base	imagenet	226,135	0,746	0,908	0,85	0,833	0,892	0,956	41,856
sebotnet33ts_256	imagenet	192,473	0,746	0,911	0,85	0,858	0,865	0,952	41,678
convnextv2_nano	fcmae	191,944	0,745	0,909	0,85	0,85	0,869	0,961	40,992
maxxvit_rmlp_nano_rw_256	imagenet	193,512	0,745	0,909	0,85	0,846	0,875	0,958	41,265
se_resnet152	imagenet	231,215	0,745	0,91	0,85	0,855	0,864	0,947	42,044

pvt_v2_b1	imagenet	196,902	0,744	0,907	0,85	0,824	0,903	0,96	41,684
convnext_base	clip_laion2b_augreg	383,79	0,744	0,91	0,85	0,863	0,855	0,949	41,456
regnetz_d32	imagenet	217,53	0,743	0,909	0,849	0,847	0,873	0,95	41,57
vgg19	imagenet	211,118	0,743	0,909	0,849	0,857	0,86	0,96	44,267
resnet50	ssl	207,373	0,743	0,908	0,848	0,849	0,868	0,948	42,875
regnetx_320	imagenet	282,603	0,743	0,91	0,849	0,862	0,856	0,949	47,585
halo2botnet50ts_256	imagenet	200,387	0,743	0,909	0,849	0,853	0,864	0,953	40,96
maxvit_nano_rw_256	imagenet	197,382	0,742	0,91	0,848	0,86	0,859	0,955	41,284
convnext_pico	imagenet	188,37	0,742	0,908	0,847	0,848	0,871	0,955	46,499
lamhalobotnet50ts_256	imagenet	198,058	0,742	0,91	0,848	0,868	0,847	0,95	42,885
dpn107	imagenet+5k	271,21	0,741	0,906	0,847	0,829	0,892	0,954	44,444
tf_efficientnetv2_m	imagenet	228,389	0,741	0,907	0,847	0,838	0,878	0,954	42,49
regnetz_e8	imagenet	239,84	0,741	0,909	0,848	0,86	0,855	0,949	42,376
convnext_large	mlp,clip_laion2b_augreg	361,058	0,741	0,91	0,848	0,871	0,843	0,942	42,024
resnext50_32x4d	ssl	212,732	0,741	0,911	0,847	0,876	0,841	0,953	42,092
dpn98	imagenet	224,958	0,741	0,905	0,847	0,829	0,887	0,952	44,213
resnext101_32x4d	swsl	240,016	0,741	0,91	0,847	0,87	0,847	0,945	43,277
convnextv2_nano	fcmae_ft_in22k_in1k	189,561	0,74	0,91	0,847	0,862	0,854	0,962	42,143
efficientnet-b6	advprop	244,462	0,74	0,909	0,848	0,853	0,864	0,948	41,294
resnext101_32x8d	imagenet	229,415	0,74	0,908	0,846	0,857	0,857	0,947	45,14
convnextv2_nano	imagenet	194,648	0,74	0,908	0,847	0,857	0,855	0,954	41,665
dpn131	imagenet	229,299	0,74	0,909	0,847	0,871	0,842	0,946	40,943
efficientnet-b7	imagenet	247,829	0,739	0,907	0,846	0,845	0,868	0,956	42,392
convnextv2_pico	imagenet	188,812	0,739	0,906	0,845	0,835	0,881	0,948	43,33
mit_b0	imagenet	207,559	0,739	0,906	0,846	0,843	0,869	0,956	43,017
se_resnet50	imagenet	214,235	0,739	0,906	0,846	0,84	0,872	0,946	43,867

regnetx_160	imagenet	232,319	0,738	0,907	0,846	0,856	0,854	0,944	44,541
convnext_nano	imagenet	673,282	0,737	0,909	0,845	0,866	0,846	0,937	41,685
regnety_040	imagenet	203,281	0,737	0,907	0,845	0,844	0,87	0,955	42,253
timm-efficientnet-b7	imagenet	239,397	0,737	0,906	0,845	0,844	0,866	0,952	43,102
regnetx_080	imagenet	230,077	0,736	0,904	0,844	0,831	0,879	0,948	45,15
timm-efficientnet-b8	advprop	753,475	0,736	0,905	0,844	0,833	0,878	0,954	41,96
efficientnet-b6	imagenet	235,123	0,736	0,907	0,845	0,853	0,857	0,953	43,565
efficientnet-b3	advprop	231,711	0,736	0,905	0,844	0,838	0,871	0,954	43,185
efficientnet-b5	advprop	237,55	0,736	0,907	0,844	0,853	0,856	0,957	42,664
regnetz_d8	imagenet	208,005	0,736	0,904	0,844	0,829	0,883	0,948	39,706
resnext101_32x4d	ssl	382,092	0,736	0,908	0,843	0,863	0,845	0,952	44,719
convnextv2_femto	imagenet	186,071	0,736	0,907	0,843	0,852	0,86	0,953	43,959
regnetz_b16	imagenet	195,914	0,736	0,906	0,843	0,846	0,864	0,956	42,433
regnety_080	imagenet	228,286	0,734	0,905	0,843	0,846	0,86	0,953	44,176
timm-efficientnet-b6	advprop	240,422	0,733	0,905	0,842	0,851	0,855	0,951	42,626
maxvit_rmlp_pico_rw_256	imagenet	196,067	0,733	0,905	0,842	0,86	0,842	0,949	41,028
pvt_v2_b0	imagenet	664,37	0,733	0,904	0,842	0,842	0,86	0,953	42,433
vgg19_bn	imagenet	213,437	0,733	0,906	0,842	0,861	0,841	0,952	41,82
se_resnet101	imagenet	228,081	0,733	0,907	0,841	0,868	0,839	0,945	42,094
se_resnext101_32x4d	imagenet	386,913	0,733	0,905	0,842	0,852	0,851	0,937	43,694
efficientnet-b4	imagenet	215,652	0,732	0,903	0,842	0,836	0,868	0,954	46,067
resnet152	imagenet	231,542	0,732	0,906	0,841	0,855	0,851	0,946	42,491
convnext_atto	imagenet	180,45	0,732	0,904	0,84	0,834	0,875	0,955	43,608
regnetz_040	imagenet	218,234	0,731	0,904	0,841	0,842	0,862	0,947	44,462
timm-efficientnet-b7	advprop	258,446	0,73	0,904	0,84	0,852	0,849	0,948	45,313
densenet161	imagenet	219,542	0,73	0,901	0,839	0,817	0,892	0,956	44,645

efficientnetv2_rw_s	imagenet	211,646	0,73	0,902	0,84	0,824	0,883	0,955	44,685
resnet18	swsl	192,256	0,73	0,903	0,839	0,835	0,868	0,946	42,502
efficientnet-b1	imagenet	210,083	0,729	0,904	0,839	0,843	0,859	0,953	41,901
se_resnext50_32x4d	imagenet	213,505	0,729	0,905	0,84	0,854	0,847	0,942	44,152
regnetz_c16	imagenet	198,896	0,729	0,903	0,839	0,838	0,863	0,945	41,925
efficientnet-b2	advprop	225,871	0,729	0,903	0,839	0,848	0,85	0,949	44,829
dpn92	imagenet+5k	226,486	0,729	0,907	0,839	0,873	0,831	0,95	46,173
lambda_resnet50ts	imagenet	199,814	0,728	0,905	0,839	0,865	0,832	0,948	45,817
botnet26t_256	imagenet	640,333	0,728	0,901	0,839	0,827	0,876	0,949	43,604
efficientnet-b3	imagenet	212,914	0,728	0,904	0,839	0,853	0,846	0,949	41,424
efficientnet-b4	advprop	231,299	0,728	0,904	0,839	0,849	0,853	0,954	43,868
timm-efficientnet-b5	advprop	239,137	0,728	0,903	0,839	0,851	0,846	0,949	43,757
efficientnet-b1	advprop	225,962	0,727	0,905	0,838	0,854	0,848	0,957	45,598
efficientnet-b5	imagenet	231,794	0,727	0,903	0,838	0,846	0,851	0,951	44,84
timm-efficientnet-b6	imagenet	228,398	0,727	0,902	0,838	0,838	0,859	0,951	43,698
resnet101	imagenet	223,021	0,727	0,904	0,838	0,855	0,843	0,945	44,062
convnext_femto	imagenet	184,78	0,726	0,903	0,837	0,847	0,851	0,945	43,56
timm-efficientnet-b2	advprop	225,312	0,725	0,904	0,837	0,86	0,836	0,951	46,773
efficientnet-b2	imagenet	212,911	0,725	0,902	0,837	0,838	0,86	0,954	44,816
efficientnetv2_rw_m	imagenet	216,73	0,725	0,903	0,836	0,852	0,844	0,94	44,698
densenet169	imagenet	213,976	0,724	0,904	0,836	0,853	0,844	0,945	44,733
resnext50_32x4d	imagenet	209,984	0,724	0,903	0,835	0,852	0,843	0,936	44,972
dpn68b	imagenet+5k	198,022	0,724	0,902	0,835	0,845	0,85	0,949	45,156
efficientnet-b0	advprop	223,42	0,724	0,901	0,835	0,84	0,854	0,951	45,505
timm-efficientnet-b5	imagenet	223,959	0,724	0,899	0,835	0,821	0,876	0,953	46,836
timm-efficientnet-b4	imagenet	219,027	0,724	0,901	0,835	0,831	0,866	0,951	45,925

timm-efficientnet-b4	advprop	233,045	0,723	0,9	0,835	0,832	0,863	0,955	45,885
timm-efficientnet-b0	advprop	224,526	0,723	0,901	0,836	0,836	0,86	0,951	47,27
timm-efficientnet-b1	advprop	226,085	0,723	0,901	0,835	0,839	0,857	0,948	46,95
efficientnetv2_rw_t	imagenet	202,324	0,723	0,9	0,835	0,841	0,848	0,942	44,205
timm-efficientnet-b3	noisy-student	201,522	0,723	0,902	0,835	0,849	0,844	0,951	45,29
timm-efficientnet-b2	imagenet	215,123	0,722	0,9	0,835	0,836	0,857	0,952	46,818
convnextv2_atto	imagenet	182,739	0,722	0,903	0,834	0,853	0,842	0,957	46,803
timm-tf_efficientnet_lite4	imagenet	211,415	0,722	0,901	0,835	0,842	0,85	0,951	47,428
resnext50_32x4d	swsl	206,449	0,722	0,903	0,834	0,86	0,833	0,943	46,963
resnet18	ssl	319,162	0,722	0,9	0,834	0,838	0,853	0,94	45,717
efficientnet-b0	imagenet	209,522	0,721	0,9	0,833	0,836	0,857	0,949	43,964
mobileone_s3	imagenet	211,158	0,721	0,899	0,834	0,83	0,86	0,949	45,678
dpn68	imagenet	210,195	0,721	0,9	0,833	0,835	0,858	0,951	46,056
timm-efficientnet-b2	noisy-student	197,664	0,721	0,901	0,833	0,841	0,851	0,953	44,265
densenet201	imagenet	234,138	0,72	0,902	0,833	0,858	0,832	0,942	45,499
lambda_resnet26t	imagenet	190,408	0,72	0,902	0,833	0,854	0,835	0,949	45,808
eca_botnext26ts_256	imagenet	189,722	0,719	0,9	0,833	0,86	0,823	0,94	44,736
lambda_resnet26rpt_256	imagenet	191,423	0,719	0,899	0,832	0,837	0,85	0,945	45,351
timm-efficientnet-b3	advprop	231,564	0,719	0,901	0,832	0,859	0,828	0,946	44,103
timm-efficientnet-b8	imagenet	259,419	0,719	0,901	0,832	0,848	0,842	0,948	43,419
regnetx_040	imagenet	210,598	0,719	0,901	0,832	0,85	0,837	0,942	43,952
timm-efficientnet-b7	noisy-student	268,451	0,718	0,9	0,831	0,852	0,834	0,949	46,043
timm-efficientnet-b3	imagenet	217,539	0,718	0,899	0,831	0,839	0,848	0,948	46,47
timm-efficientnet-b1	imagenet	212,155	0,718	0,899	0,832	0,839	0,846	0,952	47,011
inceptionv4	imagenet+background	229,794	0,718	0,9	0,832	0,848	0,838	0,95	44,496
resnet50	imagenet	213,377	0,718	0,9	0,831	0,854	0,831	0,942	43,542

mobileone_s4	imagenet	213,04	0,717	0,901	0,832	0,857	0,83	0,953	46,143
timm-efficientnet-b5	noisy-student	215,708	0,717	0,898	0,831	0,827	0,86	0,951	48,103
timm-efficientnet-b4	noisy-student	208,592	0,717	0,9	0,831	0,852	0,834	0,951	46,778
timm-efficientnet-b0	imagenet	214,396	0,717	0,899	0,831	0,842	0,844	0,951	46,335
densenet121	imagenet	212,303	0,716	0,899	0,83	0,848	0,834	0,943	45,352
inceptionv4	imagenet	230,794	0,714	0,898	0,829	0,843	0,837	0,948	46,128
timm-efficientnet-b6	noisy-student	244,894	0,714	0,898	0,829	0,848	0,831	0,945	47,308
timm-efficientnet-b1	noisy-student	197,985	0,711	0,899	0,826	0,853	0,826	0,949	45,317
resnet34	imagenet	209,179	0,71	0,897	0,826	0,84	0,838	0,952	46,91
timm-efficientnet-b0	noisy-student	198,206	0,708	0,898	0,824	0,85	0,826	0,948	46,562
inceptionresnetv2	imagenet	241,201	0,704	0,896	0,822	0,849	0,82	0,941	46,077
mobileone_s1	imagenet	214,637	0,704	0,892	0,821	0,813	0,86	0,948	59,521
mobileone_s2	imagenet	212,25	0,702	0,894	0,82	0,836	0,833	0,943	47,21
xception	imagenet	213,302	0,702	0,894	0,82	0,841	0,826	0,951	46,16
inceptionresnetv2	imagenet+background	220,541	0,701	0,895	0,819	0,85	0,815	0,942	49,312
mobileone_s0	imagenet	219,795	0,695	0,892	0,815	0,841	0,819	0,929	45,685
resnet18	imagenet	208,632	0,691	0,89	0,812	0,845	0,806	0,937	47,638
hiera_small_abswin_256	imagenet	223,237	0,656	0,875	0,784	0,844	0,749	0,928	53,514
convnextv2_huge	imagenet	616,301	0,772	0,92	0,868	0,87	0,88	0,95	39,461
convnextv2_large	imagenet	377,212	0,767	0,918	0,865	0,868	0,877	0,956	39,114
convnext_xxlarge	imagenet	617,641	0,767	0,918	0,864	0,873	0,869	0,948	38,635
convnextv2_large	fcmae_ft_in22k_in1k	444,374	0,766	0,918	0,864	0,871	0,872	0,957	39,276
convnextv2_large	fcmae	436,222	0,766	0,918	0,864	0,869	0,876	0,953	37,393
focalnet_huge_fl4	imagenet	766,007	0,765	0,916	0,863	0,854	0,891	0,96	41,349
focalnet_huge_fl3	imagenet	520,329	0,762	0,917	0,862	0,878	0,862	0,951	40,597
convnextv2_base	imagenet	282,941	0,762	0,916	0,862	0,863	0,877	0,953	41,542

focalnet_xlarge_fl4	imagenet	575,479	0,762	0,917	0,861	0,871	0,869	0,953	43,171
resnext101_32x16d	ssl	385,653	0,762	0,916	0,861	0,865	0,874	0,959	43,223
regnety_640	imagenet	405,392	0,759	0,916	0,86	0,872	0,866	0,954	39,004
resnext101_32x8d	instagram	256,869	0,759	0,914	0,859	0,857	0,879	0,95	39,966
resnext101_32x8d	swsl	275,214	0,758	0,914	0,858	0,856	0,88	0,955	43,107
maxxvitv2_nano_rw_256	imagenet	194,073	0,757	0,915	0,858	0,862	0,874	0,961	42,341
pvt_v2_b2_li	imagenet	203,732	0,757	0,914	0,858	0,862	0,873	0,962	41,398

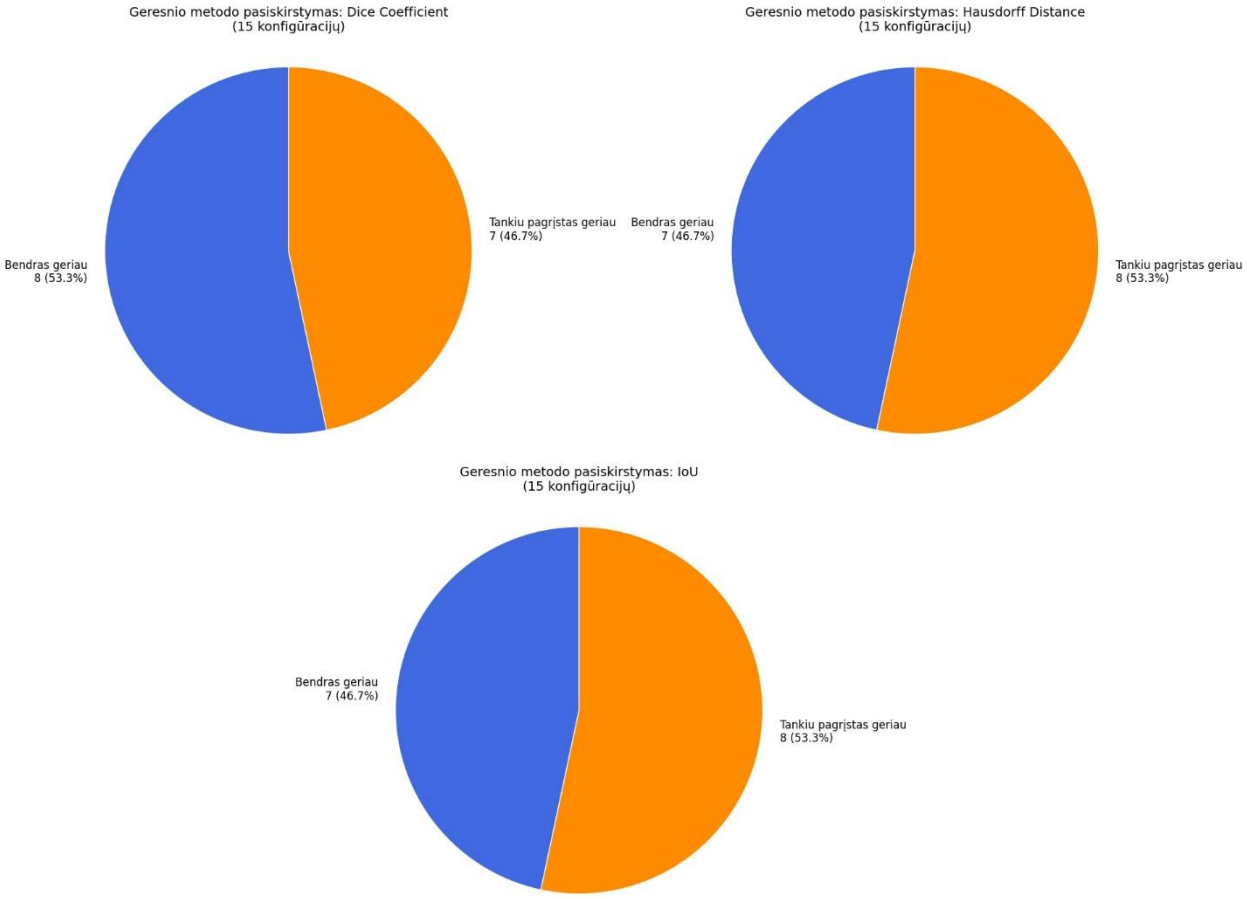
4 priedas. Pagal *IoU* surikiuotos metodų poros nefiltruotame Masačusetso rinkinyje

Modelių konfigūracijos	**Treniravimo konfigūracijos**	**Metodas**	**IoU**	**Dice**	**Hausd**	**PA**	**Prec**	**Recall**
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32	Bendras	0,724	0,83	27,34	0,942	0,861	0,827
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,10	Pagal tankį	0,704	0,813	30,65	0,939	0,85	0,825
DeepLabV3+, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32	Bendras	0,703	0,814	29,24	0,935	0,83	0,827
DeepLabV3+, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,10	Pagal tankį	0,705	0,814	29,64	0,939	0,846	0,826
DeepLabV3+, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 48	Bendras	0,708	0,818	26,6	0,937	0,837	0,831
DeepLabV3+, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 48, Slenkstis: 0,10	Pagal tankį	0,695	0,807	30,96	0,936	0,838	0,823
SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 32	Bendras	0,697	0,809	27,79	0,935	0,83	0,823
SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,20	Pagal tankį	0,684	0,801	29,59	0,931	0,815	0,819

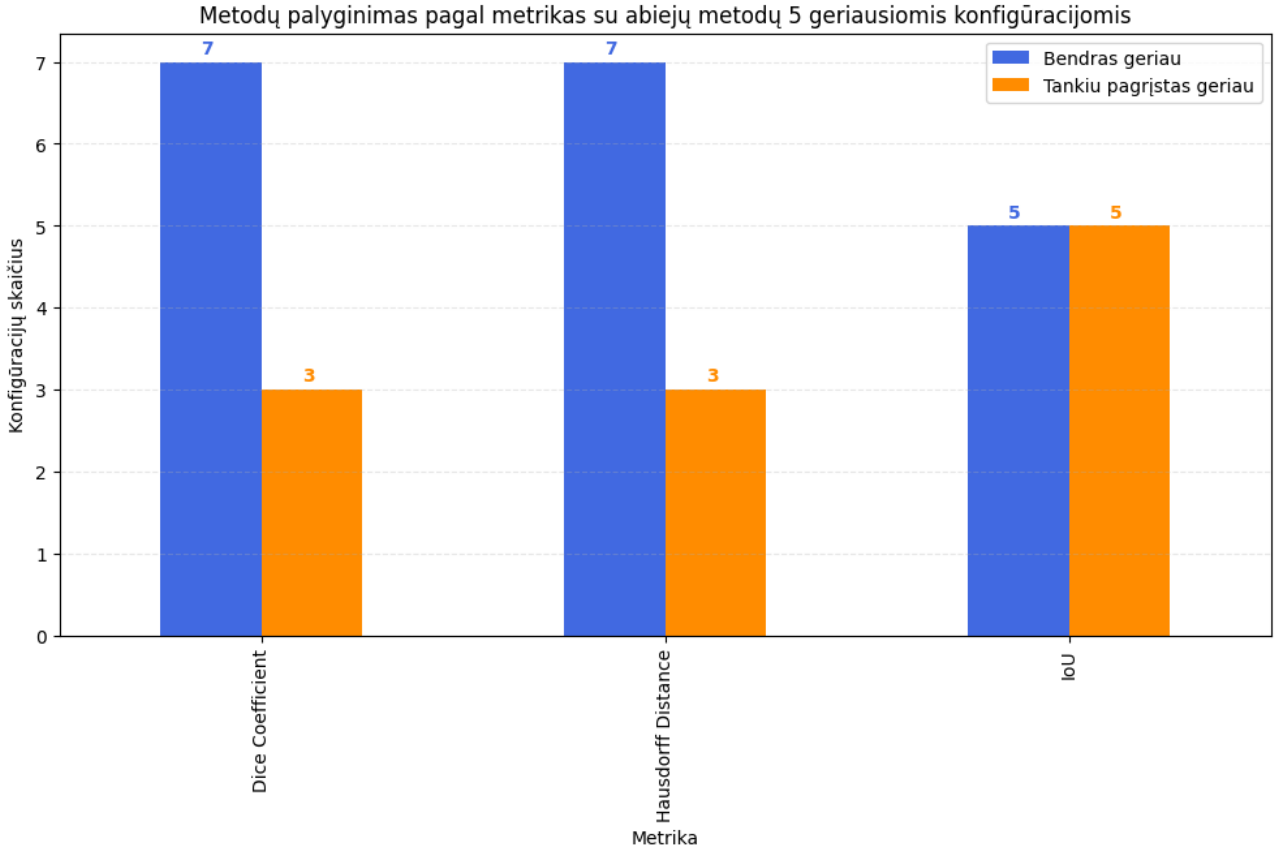
SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 48	Bendras	0,696	0,81	29,34	0,933	0,816	0,83
SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 48, Slenkstis: 0,20	Pagal tankį	0,682	0,799	29,55	0,93	0,816	0,812
SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32	Bendras	0,707	0,817	27,98	0,938	0,848	0,82
SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,20	Pagal tankį	0,669	0,783	30,81	0,932	0,83	0,804
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 32	Bendras	0,638	0,764	33,95	0,916	0,76	0,809
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,10	Pagal tankį	0,709	0,818	30,06	0,94	0,858	0,821
SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32	Bendras	0,701	0,812	30,91	0,936	0,839	0,819
SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,15	Pagal tankį	0,582	0,702	45,44	0,928	0,818	0,794
DeepLabV3+, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 48	Bendras	0,582	0,718	35,83	0,891	0,682	0,792
DeepLabV3+, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 48, Slenkstis: 0,10	Pagal tankį	0,696	0,808	30,68	0,936	0,827	0,833
DeepLabV3+, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 64	Bendras	0,582	0,718	36,51	0,89	0,68	0,785
DeepLabV3+, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 64, Slenkstis: 0,10	Pagal tankį	0,696	0,809	29,86	0,935	0,834	0,817
DeepLabV3+, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 64	Bendras	0,562	0,699	35,41	0,885	0,67	0,772

DeepLabV3+, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 64, Slenkstis: 0,10	Pagal tankį	0,699	0,813	29,12	0,935	0,829	0,827
SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 32	Bendras	0,569	0,704	39,99	0,896	0,709	0,761
SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,10	Pagal tankį	0,69	0,802	31,58	0,936	0,842	0,813
DeepLabV3+, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 32	Bendras	0,548	0,685	35,97	0,877	0,645	0,776
DeepLabV3+, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,10	Pagal tankį	0,704	0,813	30,39	0,938	0,838	0,834
DeepLabV3+, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 1e-05, E: 50, BS: 48	Bendras	0,568	0,706	38,41	0,892	0,702	0,744
DeepLabV3+, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 1e-05, E: 50, BS: 48, Slenkstis: 0,10	Pagal tankį	0,65	0,773	32,35	0,92	0,766	0,832
SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 64	Bendras	0,698	0,812	28,36	0,934	0,826	0,821
SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 64, Slenkstis: 0,18	Pagal tankį	0,334	0,39	15,66	0,836	0,76	0,191

5 priedas. Metodų pranašumo pasiskirstymas nefiltruotame Masačusetso rinkinyje



6 priedas. Abiejų metodų 5 geriausių konfigūracijų palyginimas nefiltruotame Masačusetso rinkinyje



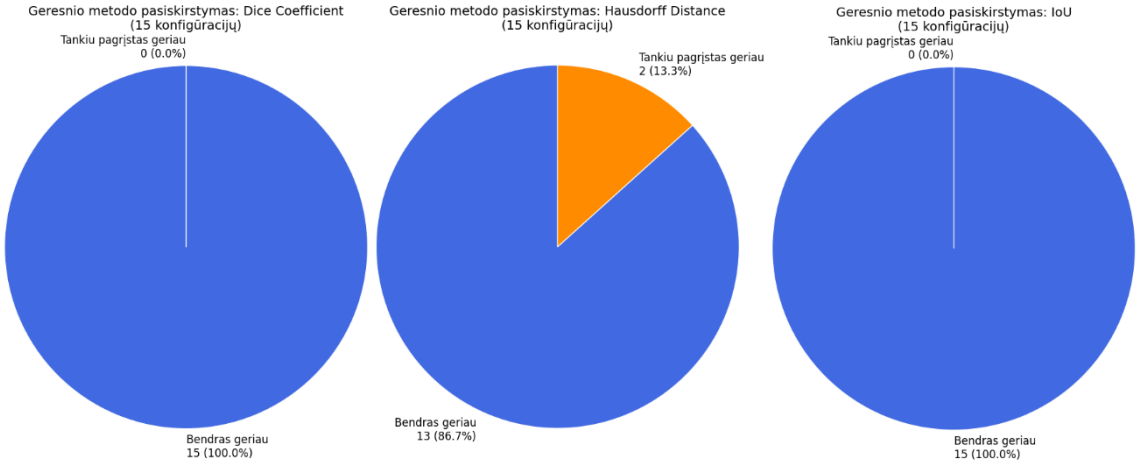
7 priedas. Pagal *IoU* surikiuotos metodų poros filtruotame Masačusetso rinkinyje

Modelio konfigūracijos	Treniravimo konfigūracijos	Metodas	IoU	Dice	Hausd	PA	Prec	Recall
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32	Bendras	0,723	0,829	28,28	0,942	0,855	0,832
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,15	Pagal tankį	0,711	0,822	29,88	0,939	0,851	0,821
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 48	Bendras	0,728	0,834	26,9	0,942	0,854	0,834
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 48, Slenkstis: 0,15	Pagal tankį	0,706	0,816	28,73	0,938	0,846	0,821
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 48	Bendras	0,721	0,828	27,35	0,941	0,853	0,829
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 48, Slenkstis: 0,15	Pagal tankį	0,705	0,816	29,35	0,938	0,85	0,817
Attention U-Net, convnext_tiny, in12k_ft_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32	Bendras	0,714	0,823	29,43	0,939	0,849	0,822
Attention U-Net, convnext_tiny, in12k_ft_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,20	Pagal tankį	0,699	0,812	31,34	0,935	0,839	0,812
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 64	Bendras	0,711	0,819	29,66	0,94	0,842	0,836
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 64, Slenkstis: 0,15	Pagal tankį	0,702	0,814	29,41	0,937	0,847	0,813
DeepLabV3+, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32	Bendras	0,709	0,819	28,49	0,938	0,839	0,832
DeepLabV3+, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,15	Pagal tankį	0,703	0,815	29,72	0,937	0,849	0,81

Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, binary_focal_jaccard_loss	LR: 0,0001, E: 50, BS: 32	Bendras	0,71	0,818	28,57	0,942	0,873	0,809
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, binary_focal_jaccard_loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,15	Pagal tankį	0,699	0,812	33,18	0,938	0,868	0,791
DeepLabV3+, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 48	Bendras	0,703	0,814	28,76	0,937	0,83	0,836
DeepLabV3+, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 48, Slenkstis: 0,15	Pagal tankį	0,693	0,808	29,21	0,933	0,821	0,825
SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32	Bendras	0,706	0,817	29,97	0,939	0,856	0,812
SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,15	Pagal tankį	0,689	0,805	29,59	0,934	0,841	0,802
SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 48	Bendras	0,692	0,805	29,4	0,934	0,832	0,817
SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 48, Slenkstis: 0,20	Pagal tankį	0,674	0,793	31,1	0,929	0,822	0,793
SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 32	Bendras	0,699	0,81	28,33	0,936	0,84	0,817
SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,20	Pagal tankį	0,664	0,782	31,81	0,929	0,812	0,81
SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 64	Bendras	0,683	0,798	29,28	0,933	0,829	0,815
SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 64, Slenkstis: 0,15	Pagal tankį	0,674	0,793	30,9	0,928	0,805	0,817
SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 48	Bendras	0,695	0,808	29,52	0,937	0,847	0,812

SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 48, Slenkstis: 0,15	Pagal tankį	0,656	0,772	31,15	0,932	0,825	0,814
SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32	Bendras	0,68	0,793	28,78	0,935	0,832	0,818
SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,20	Pagal tankį	0,671	0,787	30,78	0,932	0,824	0,813
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 1e-05, E: 50, BS: 64	Bendras	0,668	0,787	31,05	0,926	0,788	0,832
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 1e-05, E: 50, BS: 64, Slenkstis: 0,15	Pagal tankį	0,645	0,771	33,63	0,916	0,75	0,836

8 priedas. Metodų pranašumo pasiskirstymas filtruotame Masačusetso rinkinyje



9 priedas. Konfigūracijų poros surikiuotos pagal vidutinį *IoU* rezultatai Lietuvos rinkinyje

Modelio konfigūracija	Treniravimo konfigūracija	Treniravimo duomenys	Metodas	IoU	Dice	Hausdorff
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32	Lietuvos rinkinys (18000)	Bendras	0,528	0,623	95,3
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,30	Lietuvos rinkinys (18000)	Pagal tankumą	0,525	0,622	94,46
DeepLabV3+, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32	Masačusetso, Lietuvos rinkinys (9000)	Bendras	0,526	0,625	92,92

DeepLabV3+, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,25	Masačusetso, Lietuvos rinkinys (9000)	Pagal tankumą	0,52	0,62	95,28
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32	Lietuvos rinkinys (18000)	Bendras	0,527	0,622	94,78
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,20	Lietuvos rinkinys (18000)	Pagal tankumą	0,516	0,614	98,43
SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32	Lietuvos rinkinys (18000)	Bendras	0,52	0,618	95,72
SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,25	Lietuvos rinkinys (18000)	Pagal tankumą	0,52	0,617	96,36
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32	Lietuvos rinkinys (9000)	Bendras	0,515	0,613	96,93
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,25	Lietuvos rinkinys (9000)	Pagal tankumą	0,513	0,614	97,34
DeepLabV3+, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32	Lietuvos rinkinys (9000)	Bendras	0,516	0,615	95,29
DeepLabV3+, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,25	Lietuvos rinkinys (9000)	Pagal tankumą	0,511	0,612	95,03
SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 32	Lietuvos rinkinys (18000)	Bendras	0,51	0,606	97,28
SegFormer, convnextv2_large, fcmae_ft_in22k_in1k, 0,5 BCE Dice loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,20	Lietuvos rinkinys (18000)	Pagal tankumą	0,509	0,607	96,96
Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32	SN6 (9000), SN1 (filtruotas)	Bendras	0,327	0,437	106,2

Attention U-Net, convnextv2_large, fcmae_ft_in22k_in1k, 0,3 Focal 0,7 Jaccard loss	LR: 0,0001, E: 50, BS: 32, Slenkstis: 0,30	SN6 (9000), SN1 (filtruotas)	Pagal tankumą	0,333	0,444	106,98