

Kauno technologijos universitetas

Informatikos fakultetas

Palydoviniais vaizdais paremtų gyventojų tankio prognozavimo metodų tyrimas

Baigiamasis magistro studijų projektas

Gintautas Šedys

Projekto autorius

Doc. Andrius Kriščiūnas

Vadovas

Kaunas, 2025



Kauno technologijos universitetas

Informatikos fakultetas

Palydoviniais vaizdais paremtų gyventojų tankio prognozavimo metodų tyrimas

Baigiamasis magistro studijų projektas
Dirbtinio intelekto informatika (6211BX007)

Gintautas Šedys

Projekto autorius

Doc. Andrius Kriščiūnas

Vadovas

Dr. Eglė Butkevičiūtė

Recenzentė

Kaunas, 2025



Kauno technologijos universitetas

Informatikos fakultetas

Gintautas Šedys

Palydoviniais vaizdais paremtų gyventojų tankio prognozavimo metodų tyrimas

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 sąrašė;
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.

Gintautas Šedys

Patvirtinta elektroniniu būdu

Gintautas Šedys. Palydoviniais vaizdais paremtų gyventojų tankio prognozavimo metodų tyrimas. Baigiamasis magistro studijų projektas / vadovas doc. Andrius Kriščiūnas; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų kryptių grupė): Informatika.

Reikšminiai žodžiai: mašininis mokymas, konvoliuciniai neuroniniai tinklai, gyventojų tankumo žemėlapiai, dirbtinis intelektas.

Kaunas, 2025, 55p.

Santrauka

Gyventojų tankumo žemėlapiai yra būtinas įrankis resursams paskirstyti, teritorijoms planuoti ir pagalbai suteikti stichinių nelaimių atveju. Tradiciškai gyventojų tankumo žemėlapių sudarymo metodai dažnai tiesiog perskirto gyventojus apstatymo žemėlapyje bei naudoja daug išvestinių duomenų, kaip atstumai iki įvairių objektų. Dėl to šių modelių žemėlapiai gali būti gan netikslūs ir ne visada gali būti sudaromi dėl duomenų trūkumo. Šiame tyrime sprendžiama gyventojų tankumo žemėlapių sudarymo problema, pasiūlant 10 m raiškos gyventojų tankumo žemėlapio sudarymo modelį, paremtą CNN ir U-Net architektūromis. Modelis naudoja tik palydovinius vaizdus, todėl leidžia sudaryti šiuos žemėlapius visoms vietovėms ir visiems laikotarpiams, kai prieinami palydoviniai duomenys. Taip pat šiame darbe siūloma nauja gyventojų tankumo žemėlapių tikslumo vertinimo metrika – $\Delta Vid. \%$, kuri parodo procentinį skirtumą tarp surašymo duomenų ir modelio prognozuojamo gyventojų tankumo skirtinguose tankumo intervaluose.

Gintautas Šedys. Investigation of Population Density Prediction Methods Based on Satellite Imagery. Master's Final Degree Project/ supervisor Doc. Andrius Kriščiūnas; Faculty of Informatics, Kaunas University of Technology.

Study field and area (study field group): Informatics.

Keywords: machine learning, convolutional neural networks, population density maps, artificial intelligence.

Kaunas, 2025, 55p.

Summary

Gridded population maps are essential tools for resource allocation, urban planning, and disaster response. Traditionally, methods for creating gridded population maps often simply redistribute people based on settlement patterns and use a large amount of derived data, such as distances to various objects. As a result, maps produced by these models can be quite inaccurate and are not always available due to data limitations. This study addresses this problem by proposing a model, based on CNN and *U-Net* architectures, that can produce gridded population maps with 10 meter resolution, which relies solely on satellite imagery. This approach enables the generation of such maps for any location and time period where satellite data is available. The study also introduces a new metric for evaluating the accuracy of population density maps, $\Delta\text{Avg.}\%$, which shows the percentage difference between census based and model generated gridded population maps across different population density intervals.

Turinys

Lentelių sąrašas	8
Paveikslų sąrašas	9
Santrumpų ir terminų sąrašas	10
Įvadas	11
1. Gyventojų tankumo žemėlapių ir konvoliucinių neuroninių tinklų analizė	12
1.1. Problemos, kurias sprendžia GTŽ sudarymo modeliai	12
1.2. Gyventojų tankumo žemėlapių sudarymo modelių analizė	13
1.2.1. <i>GHS-Pop</i> – sprendimų priėmimui skirtas globalus GTŽ	13
1.2.2. <i>WorldPop</i> – GTŽ sudarytas naudojant atsitiktinio miško modelį	13
1.2.3. <i>LandScan</i> – GTŽ be mašininio mokymo	13
1.2.4. <i>MGWR</i> – masteliais paremtas modelis GTŽ sudaryti	14
1.2.5. <i>MNN</i> – GTŽ sudarytas be gyventojų perskirstymo	14
1.2.6. <i>POMELO</i> – aukštos raiškos GTŽ, sudarytas iš surašymo duomenų	14
1.2.7. <i>Popcorn</i> – spektrinių palydovų panaudojimas GTŽ sudaryti	14
1.2.8. <i>WPE</i> – GTŽ be mašininio mokymo	14
1.2.9. <i>GRUMPV1</i> – paprasta metodologija, be mašininio mokymo	15
1.2.10. <i>CNNReg</i> – CNN panaudojimas vietovės gyventojų skaičiui nustatyti	15
1.2.11. <i>RFBuild</i> – RF ir apstatymo žemėlapiu paremtas GTŽ	15
1.2.12. <i>MobileNetV2</i> – perkėlimo mokymosi panaudojimas GTŽ sudaryti	15
1.2.13. <i>SynthMod</i> – GTŽ kaimo ir miesto vietovių atskyrimas	15
1.2.14. Bendra modelių apžvalga	15
1.3. Konvoliucinių neuroninių tinklų analizė	17
1.3.1. CNN sprendžiami uždaviniai	17
1.3.2. CNN architektūra	19
1.3.3. <i>U-Net</i> architektūra	20
1.3.4. Nuostolių funkcijos naudojamos su CNN	20
1.4. „Google Earth Engine“ įrankio skirtu darbui su palydoviniais vaizdais analizė	21
1.5. Gyventojų tankumo žemėlapių ir konvoliucinių neuroninių tinklų analizės išvados	22
2. Gyventojų tankumo žemėlapių sudarymo metodologija	23
2.1. Sistemos aprašas ir reikalavimai	23
2.2. GTŽ prognozavimo modelio kūrimo metodologija	25
2.2.1. Modelio mokymui naudoti duomenys	25
2.2.2. Gyventojų surašymo duomenų agregavimas	26
2.2.3. Modelio mokymas ir modelio parametrų radimas	28
2.3. GTŽ modelio įvertinimo metodologija	28
2.3.1. Vizualinis įvertinimas	29
2.4. Gyventojų tankumo žemėlapių sudarymo metodologijos išvados	30
3. Gyventojų tankumo žemėlapių sistemos implementacija	31
4. Eksperimentai GTŽ modeliui sukurti	32
4.1. Eksperimentų atlikimo aplinka	32
4.2. Duomenų analizė	32
4.3. Duomenų paruošimas	34

4.4. CNN eksperimentai	35
4.4.1. Nuostolių funkcijos pasirinkimas	35
4.4.2. Sluoksnių skaičiaus pasirinkimas	39
4.4.3. Filtrų skaičiaus pasirinkimas	41
4.5. <i>U-Net</i> eksperimentai	43
4.5.1. Nuostolių funkcijos pasirinkimas	44
4.5.2. <i>U-Net</i> gylio pasirinkimas	46
4.5.3. Filtrų skaičiaus pasirinkimas	48
4.6. Eksperimentų rezultatai	49
Išvados	52
Literatūros sąrašas	53

Lentelių sąrašas

1 lentelė.	GTŽ modelių apžvalga	16
2 lentelė.	„Peržiūrėti pasirinktos teritorijos kanalus“ panaudos atvejo aprašas	24
3 lentelė.	„Prognozuoti GTŽ pasirinktai teritorijai“ panaudos atvejo aprašas	24
4 lentelė.	Modeliui mokytį skirti įvesties ir išvesties duomenys iš <i>Sentinel 1</i> , <i>Sentinel 2</i> , <i>VIIRS</i> palydovų bei surašymo GTŽ	26
5 lentelė.	Sistemos specifikacija	32
6 lentelė.	Modeliui mokytį naudotų kanalų statistiniai duomenys	33
7 lentelė.	Gyventojų tankumo verčių pasiskirstymas skirtinguose intervaluose	34
8 lentelė.	CNN modelių, mokytų su skirtingomis nuostolių funkcijomis, Δ Vid.% verčių lentelė skirtingiems gyventojų intervalams	38
9 lentelė.	CNN modelių, mokytų su skirtingais sluoksnių skaičiais, Δ Vid.% verčių lentelė skirtingiems gyventojų intervalams	40
10 lentelė.	CNN modelių, mokytų su skirtingais filtrų skaičiais, Δ Vid.% verčių lentelė skirtingiems gyventojų intervalams	42
11 lentelė.	<i>U-Net</i> modelių, mokytų su skirtingomis nuostolių funkcijomis, Δ Vid.% verčių lentelė skirtingiems gyventojų intervalams	46
12 lentelė.	<i>U-Net</i> modelių, mokytų su skirtingais gyliais, Δ Vid.% verčių lentelė skirtingiems gyventojų intervalams	47
13 lentelė.	<i>U-Net</i> modelių, mokytų su skirtingu filtru skaičiumi, Δ Vid.% verčių lentelė skirtingiems gyventojų intervalams	49
14 lentelė.	Įvairių miestų GTŽ, sudaryti su skirtingais modeliais, Δ Vid.% verčių lentelė skirtingiems gyventojų intervalams	51

Paveikslų sąrašas

1 pav.	Žemėlapis rodantis ar per pastaruosius 10 metų buvo darytas gyventojų surašymas . . .	12
2 pav.	<i>MNIST</i> klasifikavimo iliustracija	17
3 pav.	Gatvėje esančių objektų atpažinimo iliustracija	17
4 pav.	Gatvėje esančių objektų segmentavimo iliustracija	18
5 pav.	Veido savybių išgavimo iliustracija	18
6 pav.	GTŽ regresijos iliustracija	18
7 pav.	CNN architektūra	19
8 pav.	Konvoliucijos iliustracija (su Sobelio kraštų filtru)	19
9 pav.	Telkimo sluoksnis (maksimalios vertės pasirinkimas 2 x 2 pikselių dydžio lange) . . .	19
10 pav.	<i>U-Net</i> architektūra	20
11 pav.	GEE naudojimo pavyzdys su „Java Script“ kodu	22
12 pav.	Panaudos atvejų diagrama	23
13 pav.	„Prognozuoti GTŽ pasirinktai teritorijai“ veiklos diagrama	24
14 pav.	Metodologijos schema	25
15 pav.	Lietuvos gyventojų surašymo duomenys 500 m dydžio gardelėse	26
16 pav.	Kėdainių miesto GTŽ vizualinis vertinimas	30
17 pav.	GTŽ sistemos naudotojo sąsaja	31
18 pav.	GTŽ sistemos diegimo diagrama	31
19 pav.	Testavimo ir mokymo imčių atskirtis, pavaizduota Lietuvos žemėlapyje. Vienas imties elementas yra 5 x 5 km dydžio plotas	32
20 pav.	Standartizuotų, modeliui mokytui naudotų kanalų, histogramos	33
21 pav.	Įvesties ir išvesties duomenys Kėdainių miestui	34
22 pav.	CNN modelių, mokytų su skirtingomis nuostolių funkcijomis, Kėdainių miestui sudaryti GTŽ	37
23 pav.	CNN modelių, mokytų su skirtingomis nuostolių funkcijomis, nuostolių kreivės	39
24 pav.	CNN modelių, mokytų su skirtingais sluoksnių skaičiais, Kėdainių miestui sudaryti GTŽ	39
25 pav.	CNN modelių, mokytų su skirtingais filtrų skaičiais, Kėdainių miestui sudaryti GTŽ .	41
26 pav.	<i>U-Net</i> modelių, mokytų su skirtingomis nuostolių funkcijomis, Kėdainių miestui sudaryti GTŽ	45
27 pav.	<i>U-Net</i> modelių, mokytų su skirtingomis nuostolių funkcijomis, nuostolių kreivės . . .	45
28 pav.	<i>U-Net</i> modelių, mokytų su skirtingais gyliais, Kėdainių miestui sudaryti GTŽ	47
29 pav.	<i>U-Net</i> modelių, mokytų su skirtingu filtru skaičiumi, Kėdainių miestui sudaryti GTŽ .	48
30 pav.	Įvairių miestų GTŽ, sudaryti su skirtingais modeliais	50
31 pav.	Saugų kaimo GTŽ, sudaryti su „CNN SSIM 10 32“ ir „CNN SMAPE 10 10“ modeliais	51

Santrumpų ir terminų sąrašas

Santrumpos:

GTŽ – gyventojų tankumo žemėlapis;

ML – mašininis mokymas;

CNN – konvoliucinis neuroninis tinklas;

SML – simbolinis mašininis mokymas;

OSM – atvirasis kelių žemėlapis;

%TAA – bendras paskirstymo tikslumas;

RMSE – vidutinė šakninė kvadratinė paklaida;

RF – atsitiktinis miškas;

MAE – vidutinė absoliutinė paklaida;

MLP – daugiasluoksnis perceptronas;

MAPE – vidutinė absoliutinė procentinė paklaida;

MSE – vidutinė kvadratinė paklaida;

SMAPE – simetrinė vidutinė absoliuti procentinė paklaida;

SSIM – struktūrinio panašumo indeksas;

GEE – „Google Earth Engine“;

NIR – artimoji infraraudonoji spinduliuotė;

SWIR – trumpojo bangos ilgio infraraudonoji spinduliuotė.

Įvadas

Gyventojų tankumo žemėlapiai (GTŽ) yra būtinas įrankis resursų paskirstymui, teritorijų planavimui ir pagalbos suteikimui stichinių nelaimių atveju [1]. Tačiau tradiciniai GTŽ, paremti gyventojų surašymo duomenimis, yra limituoti žemos raiškos, kokybės ir prieinamumo [1, 2, 3, 4]. Pavyzdžiui, Lietuvoje GTŽ yra atnaujinami tik kartą per 10 metų, esant 100 m raiškai didesniuose miestuose ir 1 km raiškai likusioje valstybės teritorijoje [5]. O mažiau išsivysčiusiose pasaulio regionuose GTŽ duomenys gali būti prastos kokybės arba išvis neprieinami [1, 2, 3, 6].

Šią problemą sprendžia mašininio mokymo (ML) paremti GTŽ sudarymo modeliai. Didžioji dauguma šių modelių fokusuojasi ties aukštesne raiška ir kokybe bei turi sudėtingą metodologiją ir naudoja daug skirtingų duomenų šaltinių (palydoviniai vaizdai, atstumai iki mokyklų, surašymo GTŽ ir t. t.), dėl šių priežasčių šie modeliai geba sudaryti GTŽ tik tam tikrose teritorijose ir tik tam tikrais laiko periodais [7, 8, 9, 10].

Tiriamąo darbo tikslas

Šio tyrimo tikslas – ištirti GTŽ sudarymo modelius ir sukurti paprastą, mašininio mokymo (ML) paremtą modelį, kuris gebėtų sudaryti aukštos kokybės, 10 m raiškos GTŽ, tik iš palydovinių vaizdų įvesties, Lietuvos teritorijai.

Tiriamąo darbo uždaviniai

1. išanalizuoti esamus GTŽ sudarymo modelius, jų metodologijas, naudojamus duomenis bei šių modelių privalumus ir trūkumus;
2. apžvelgti konvoliucinius neuroninius tinklus (CNN) bei jų taikymą pikselių lygmens regresijos uždaviniams;
3. susirinkti ir pasiruošti surašymo GTŽ bei palydovinių vaizdų duomenis, reikalingus modeliui mokyti;
4. sukurti ML modelį, gebantį sudaryti GTŽ;
5. surasti tinkamą nuostolių funkciją bei geriausią architektūrą GTŽ modeliui;
6. surasti tinkamą metrika GTŽ modeliui įvertinti bei palyginti sukurtą GTŽ modelį su jau esamais GTŽ modeliais;
7. sukurti paprastą programą, leidžiančią matyti modelio sukurtą GTŽ.

1. Gyventojų tankumo žemėlapių ir konvoliucinių neuroninių tinklų analizė

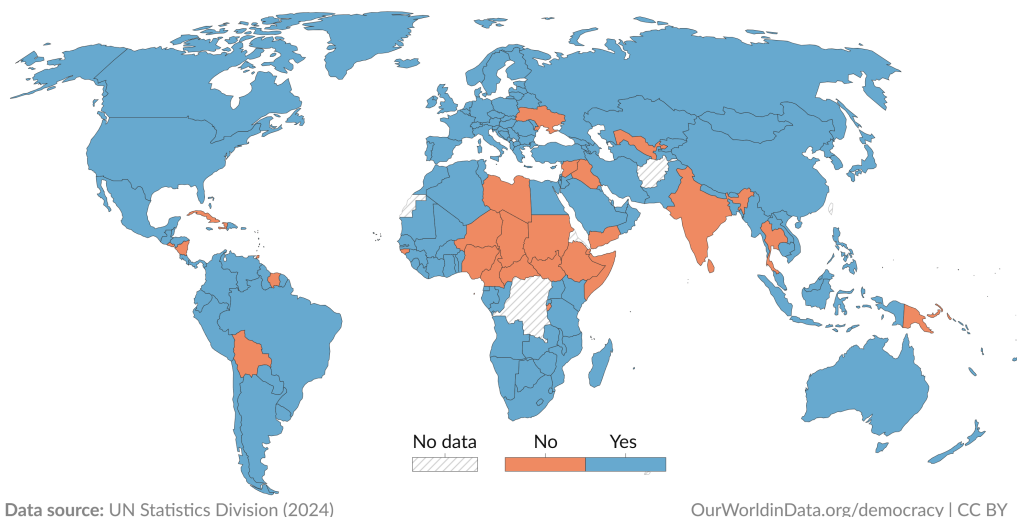
Šiame skyriuje susipažinsime su problemomis, kurias padeda išspręsti GTŽ modeliai, su jau egzistuojančiais GTŽ sudarymo modeliais ir jų ypatybėmis, apžvelgsime konvoliucinius neuroninius tinklus ir nuostolių funkcijas, su kuriomis šie tinklai yra mokomi bei „Google Earth Engine“ platformą, skirtą dirbti su geografiniais duomenimis.

1.1. Problemos, kurias sprendžia GTŽ sudarymo modeliai

GTŽ sudarymo modeliai atsirado kaip atsakas į esminius trūkumus, susijusius su tradiciniais gyventojų duomenų šaltiniais, tokiais kaip gyventojų surašymai žemų ir vidutinių pajamų šalyse. Nors surašymai teoriškai apima visą šalies populiaciją ir yra laikomi baziniu informacijos šaltiniu, praktiškai jie dažnai būna pasenę, neišsamūs arba visai nevykdomi, todėl negali patenkinti politikos planavimo poreikių [1].

Pagrindinės problemos, susijusios su surašymų duomenimis, yra šios:

1. **surašymų nebuvimas arba ribotas prieinamumas.** Daugelyje Afrikos ir kitų žemų pajamų šalių patikimų gyventojų surašymų nėra. Šių duomenų trūkumas reiškia, kad nėra būdo vertinti svarbius rodiklius, tokius kaip gyventojų skaičius, tankumas, mirtingumas ar gimstamumas. Gyventojų surašymų prieinamumas pateiktas paveikslėlyje 1 [4];
2. **žema erdvinė raiška.** tradiciniai surašymo duomenys paprastai pateikiami suagreguoti iki didelių administracinių vienetų (pvz., regionų ar provincijų), o tai riboja jų naudingumą. Tačiau sprendimų priėmimui vis dažniau reikia aukštos raiškos duomenų, pavyzdžiui, 1x1 km ar smulkesniu masteliu, kad būtų galima efektyviai planuoti infrastruktūrą [4];
3. **duomenų kokybės problemos.** dėl netikslių skaičiavimų, praleistų gyventojų grupių, pasikartojimų ar netinkamo geografinio priskyrimo, surašymų duomenys gali būti klaidingi. Taip pat, kai kurie plačiai naudojami duomenų rinkiniai, pavyzdžiui, *Africapolis*, remiasi surašymo informacija, kuri ne visada yra patikima ar teisinga [4, 11];
4. **netolygus duomenų padengimas.** Atokesnės ar sunkiau pasiekiamos vietovės dažnai patiria „nuotolio bausmę“ – duomenų kokybė ten blogesnė, dažniau pasitaiko netikslumų [4].



1 pav. Žemėlapis rodantis ar per pastaruosius 10 metų buvo darytas gyventojų surašymas [12]

GTŽ sudarymo modeliai siūlo sprendimą šioms problemoms, sudarydami GTŽ nepriklausomai nuo surašymo duomenų.

1.2. Gyventojų tankumo žemėlapių sudarymo modelių analizė

1.2.1. GHS-Pop – sprendimų priėmimui skirtas globalus GTŽ

GHS-Pop perskirsto žemos raiškos gyventojų surašymo duomenis apgyvendintų teritorijų žemėlapyje su linijiniu dazimetriniu kartografavimu, taip sukurdamas 100 m raiškos GTŽ, kuris prieinamas kas 5 metus nuo 1975-ųjų iki 2030-ųjų. Linijinis dazimetrinis kartografavimas, tai dideliame plote esamų gyventojų paskirstymas į mažesnio ploto tinklelio gardeles, *GHS-Pop* atveju šis paskirstymas yra proporcingas gardelės apstatymo lygiui. Apstatymo lygio žemėlapis yra sudarytas iš kitų dvejų modelių (*GHS-BUILT-S*, *GHS-BUILT-V*), kurie buvo sukurti pasinaudojant simboliniu mašininio mokymu (SML) ir taip pat yra sudėtingos *GHS-Pop* metodologijos dalis. Savo metodologijoje *GHS-Pop* naudoja gyventojų surašymo duomenis iš įvairių šaltinių: *Sentinel-2* ir *Landsat* palydovų vaizdus, skaitmeninius elevacijos modelius, vandens paviršiaus žemėlapius, vegetacijos indeksus ir duomenis iš atvirojo kelių žemėlapių (OSM). *GHS-Pop* tikslumui įvertinti naudojamos šios metrikos: R-tikslumą – 0,71; bendrą paskirstymo tikslumą (%TAA) – 83 %; vidutinę šakninę kvadratinę paklaidą (RMSE) – 45,4. *GHS-Pop* tiesiogiai nesudaro GTŽ, šis modelis iš pradžių sudaro apstatymo žemėlapi, o iš jo GTŽ [7].

1.2.2. WorldPop – GTŽ sudarytas naudojant atsitiktinio miško modelį

WorldPop, taip pat kaip ir *GHS-Pop*, perskirsto gyventojų surašymo duomenis į mažesnio ploto tinklelio gardeles, bet pasitelkdamas paprastesnę metodologiją. *WorldPop* naudoja atsitiktinio miško (RF) modelį GTŽ sudaryti ir gyventojams perskirstyti. Šio modelio GTŽ yra 100 m raiškos ir prieinamas kas 1 metus nuo 2000-ųjų iki 2020-ųjų. *WorldPop* savo metodologijoje naudoja skaitmeninius elevacijos modelius, vandens paviršiaus, topografinius, temperatūros, žemės dangos žemėlapius, OSM (atstumai iki mokyklų, ligoninių ir kitų objektų), *DMSP-OLS* ir *VIIRS* palydovų naktinių šviesų vaizdus, gyventojų surašymo duomenis ir *GHS-Pop* GTŽ. Metrikos tikslumui įvertinti nebuvo paminėtos. *WorldPop* tiesiogiai sudaro GTŽ kaip žymą, naudodamas administracinio vieneto gyventojų surašymo duomenis [8].

1.2.3. LandScan – GTŽ be mašininio mokymo

LandScan, kaip *GHS-Pop* ir *WorldPop* modeliai, taip pat perskirsto gyventojų surašymo duomenis, bet šis modelis, priešingai negu prieš tai minėti, nenaudoja ML, *LandScan* naudoja specifinius algoritmus gyventojų perskirstymo svoriams nustatyti. *LandScan* modelis priskiria specifinius svorius žemės dangos, topografiniams žemėlapiams, atstumams iki geležinkelių, oro uostų, miestų ir kitų objektų, su šiais svoriais modelis perskirsto gyventojų surašymo duomenis ir sudaro 1 km raiškos GTŽ, kuris prieinamas kas 1 metus nuo 1998 iki 2025. Metrikos tikslumui įvertinti nebuvo paminėtos. Kadangi *LandScan* nėra paremtas ML, jis nenaudoja jokios žymos modeliui mokytis. Taip pat verta paminėti, kad *LandScan* sudaro vidutinio gyventojų buvimo per 24 valandas GTŽ, o ne gyvenamosios vietos GTŽ [13].

1.2.4. MGWR – masteliais paremtas modelis GTŽ sudaryti

MGWR modelis išsiskiria atsižvelgimu į tai, kad skirtingi parametrai gali būti svarbūs skirtinguose masteliuose, pavyzdžiui, apstatymo lygis yra svarbus lokaliu mastu, o vietovės ekonominis išsivystymas – didesniu mastu. *MGWR*, kaip ir visi prieš tai minėti modeliai, sukuria GTŽ ir tada jį patikslina perskirstant gyventojų surašymo duomenis. *MGWR* sudaro 100 m raiškos GTŽ, kuris prieinamas tik 2018-ais metais. *MGWR* naudoja gyventojų surašymo, administracinių ribų, *GHS*, nakties šviesų, žemės panaudojimo tipo, atstumų iki įvairių objektų duomenis. *MGWR* tikslumui įvertinti naudoja šias metrikas: vidutinę absoliutinę paklaidą (MAE) – 20,698, RMSE – 37,914, R^2 – 0,64. Taip pat, kaip ir *WorldPop*, *MGWR* kaip žymą modeliui mokyti naudoja administracinio vieneto gyventojų surašymo duomenis [9].

1.2.5. MNN – GTŽ sudarytas be gyventojų perskirstymo

MNN tai CNN ir daugiasluoksnio perceptrono (MLP) kombinacija, skirta GTŽ sudaryti, nenaudojant gyventojų surašymo duomenų GTŽ patikslinimui. *MNN* naudoja CNN papildomiems parametrams iš duomenų išgauti ir tada su MLP sudaro GTŽ. *MNN* naudoja nakties šviesų, vegetacinių indeksų, žemės panaudojimo tipo, OSM ir atstumų iki įvairių objektų duomenis, *WorldPop* GTŽ bei gyventojų surašymo duomenis rezultatams įvertinti. *MNN* sudaro 100 m raiškos GTŽ, kuris prieinamas tik 2017 metais. Tikslumui įvertinti *MNN* naudoja procentinę RMSE (%RMSE) – 22,36 %, R^2 - 0,77. Kaip žymą modeliui mokyti *MNN* naudoja *WorldPop* GTŽ [10].

1.2.6. POMELO – aukštos raiškos GTŽ, sudarytas iš surašymo duomenų

POMELO sukuria apgyvendinimo žemėlapią ir tada perskirsto surašymo duomenis, taip sudarydamas 100 m raiškos GTŽ. Apgyvendinimo žemėlapiui sukurti *POMELO* naudoja MLP. MLP mokymui modelis naudoja OSM pastatų ir kelių informaciją, kelionės laiką iki miesto, kuris turi daugiau nei 50 000 gyventojų, nakties šviesas, topografinius, žemės panaudojimo tipo žemėlapius ir *GHS-BUILT*. Kaip žymą MLP naudoja administracinio vieneto gyventojų surašymo duomenis. *POMELO* tikslumas: R^2 – 0,85, MAE – 3100, vidutinė absoliutinė procentinė paklaida (MAPE) – 21,6 % [1].

1.2.7. Popcorn – spektrinių palydovų panaudojimas GTŽ sudaryti

Popcorn sudaro 10 m raiškos GTŽ, naudodamas tik *Snetinel-1*, *Sentinel-2* palydovų vaizdus ir gyventojų surašymo duomenis kaip žymą modeliui mokyti. Šis modelis neperskirsto surašymo duomenų savo metodologijoje. *Popcorn* naudoja *U-Net* architektūros CNN. Iš pradžių du *U-Net* modeliai išgauna savybes iš *Snetinel-1*, *Sentinel-2* vaizdų, tada sujungia šias savybes ir iš jų išgauna pastatų apgyvendinimo žemėlapią, kurį toliau transformuoja į GTŽ. *POMELO* tikslumas: R^2 – 0,6, MAE – 1,35, RMSE – 8,4 [14].

1.2.8. WPE – GTŽ be mašininio mokymo

WPE, taip pat kaip ir *LandScan*, nenaudoja mašininio mokymo, o priskiria taškus tam tikriems parametrams ir iš šių taškų sudaro apgyvendinimo tikėtinumą žemėlapią, kuris paverčiamas į 150 m raiškos GTŽ, perskirstant gyventojų surašymo duomenis šiame žemėlapyje. *WPE* prieinamas kiekvienais metais nuo 2013-ųjų metų. *WPE* apgyvendinimo tikėtinumo žemėlapiui sudaryti naudoja žemės paskirties tipo žemėlapius (aukšto pastatų tankumo vietovės – 200 taškų, vidutinio tankumo – 150, žemės ūkio vietovės – 25, vanduo, pelkės ir kitos negyvenamos vietovės – 0), OSM duomenis

(sankryžos 150 m spinduliu – 75, gyvenvietės 1 km spinduliu – 50) ir iš *Landsat* palydovo išgautus teritorijos tekstūros žemėlapius. *WPE* tikslumui nustatyti lygino savo modelio rezultatus su surašymo duomenimis, bet konkreti metrika nebuvo paminėta [15].

1.2.9. *GRUMPv1* – paprasta metodologija, be mašininio mokymo

GRUMPv1, kaip ir *WPE* su *LandScan*, nenaudoja mašininio mokymo. *GRUMPv1* perskirsto didelių administracinių vienetų (valstybių, apskričių) gyventojų surašymo duomenis miesto ir kaimo vietovėms 1 km raiškos gardelėje. Perskirstymui *GRUMPv1* naudoja nakties šviesas ir gyvenamų vietovių administracines ribas (mažesnėms gyvenvietėms, kur nakties šviesų neužtenka apgyvendinimui nustatyti). *GRUMPv1* prieinamas kas 5 metus nuo 1990-ųjų iki 2000-ųjų metų. Metrikos tikslumui nustatyti paminėtos nebuvo [16].

1.2.10. *CNNReg* – CNN panaudojimas vietovės gyventojų skaičiui nustatyti

CNNReg sudaro 1 km raiškos GTŽ, naudodamas CNN regresijos uždaviniui spręsti – 1 x 1 km dydžio plotui nustatyti gyventojų skaičių. *CNNReg* naudoja tik *Landsat* palydovinius vaizdus ir JAV 2000-ųjų, 2010-ųjų metų gyventojų surašymo duomenis 1 x 1 km dydžio gardelėse kaip žymą modeliui mokytį. *CNNReg* tikslumas: MAE – 23, R^2 – 0,9, MAPE – 73,78 % [17].

1.2.11. *RFBuild* – RF ir apstatymo žemėlapiu parentas GTŽ

RFBuild GTŽ yra sudarytas iš dviejų modelių. Vienas iš šių modelių, naudodamasis apstatymo žemėlapiu (*GHS-BUILT* ir kiti apstatymo žemėlapiai) patikslina administracines ribas, o kitas su RF modelių paskirsto surašymo duomenis šiose ribose. RF paskirsto surašymo duomenis naudodamas žemės panaudojimo tipo, naktinių šviesų, temperatūros, elevacijos, OSM, atstumų iki įvairių objektų duomenis, taip sudarydamas 100 m raiškos GTŽ. *RFBuild* tikslumas: RMSE – 8,1, MAE – 1,6 [18].

1.2.12. *MobileNetV2* – perkėlimo mokymosi panaudojimas GTŽ sudaryti

Kaip ir *CNNReg*, *MobileNetV2* sprendžia regresijos uždavinį – surasti gyventojų skaičių 36 x 36 m dydžio plotui. Nuo *CNNReg*, *MobileNetV2* skiriasi tuo, kad *MobileNetV2* naudojo iš bendruomenių surinktus gyventojų duomenis, taip išgaudamas aukštos raiškos GTŽ, skirtą modeliui mokytį. Dėl duomenų trūkumo *MobileNetV2* naudojo perkėlimo mokymą modeliui sukurti. *MobileNetV2* naudojo tik labai aukštos raiškos vaizdus (2 cm raiškos) GTŽ sudaryti. *MobileNetV2* tikslumo metrika yra 63 % (klasifikuojant GTŽ į apgyvendintas ir neapgyvendintas teritorijas) [19].

1.2.13. *SynthMod* – GTŽ kaimo ir miesto vietovių atskyrimas

SynthMod sudaro 500 m raiškos GTŽ perskirstydamas surašymo duomenis atskirai kaimo ir miesto vietovėms. *SynthMod* naudojo Kinijos Henano provincijos gyventojų surašymo duomenis, kurie buvo atskirti tarp miesto ir kaimo vietovių. Kaimo vietovių nustatymui *SynthMod* naudojo nakties šviesas ir žemės panaudojimo tipo žemėlapi. Nustačius kaimo vietoves, jose gyventojai buvo paskirstyti tolygiai. Miesto vietovių gyventojų perskirstymui buvo sukurtas regresijos modelis, kuris naudojo nakties šviesas gyventojams perskirstyti. *SynthMod* R^2 tikslumas buvo 0,75 [20].

1.2.14. Bendra modelių apžvalga

Bendra visų išanalizuotų modelių apžvalga yra pateikta lentelėje 1. Iš šios lentelės pastebime, kad didžioji dalis GTŽ sudarymo modelių perskirsto surašymo duomenis pagal apgyvendinimo tikėtumą,

o tikėtinumo žemėlapis gali būti sudarytas pasinaudojant skirtingomis metodologijomis, tokiomis kaip RF, SML ar svorių priskyrimu, nenaudojant ML. Tik *MNN*, *WPE*, *MobileNetV2* ir *GRUMPV1* neperskirstė surašymo duomenų, o sukūrė GTŽ tiesiogiai. ML modeliai mokymo metu kaip žymą dažniausiai naudoja gyventojų skaičių, esantį administraciniame vienete (apskirtyje, rajone, mieste), *MNN* kaip žymą naudojo *WorldPop* modelio GTŽ (100 m raiška), *CNNReg* kaip žymą naudojo JAV gyventojų surašymo duomenis (1 km gardelės), o *MobileNetV2* kaip žymą naudojo iš bendruomenių surinktus gyventojų duomenis. Modelių raiškos buvo tarp 10 m ir 1 km. Dauguma modelių naudojo išvestinius vaizdus, tokius kaip žemės paskirtis, apstatymo lygis, vandens kaukės, kelių ir atstumų iki įvairių objektų duomenis iš tokių šaltinių kaip OSM ir nakties šviesos vaizdus. Tik *Popcorn*, *WPE*, *MobileNetV2* ir *CNNReg* modeliai naudoja spektrinių palydovų vaizdus tiesiogiai.

Buvo pastebėta, kad GTŽ sudarymo modeliai neturi standartinės ir žmogui suprantamos metrikos jų tikslumui nustatyti. Dažniausiai naudojamos metrikos yra MAE, R^2 , MAPE, RMSE. Tačiau šios metrikos nėra lengvai suprantamos žmogui (R^2 , MAPE) arba priklauso nuo modelio mokymo duomenų (MAE, MAPE). MAE tinka lyginti modelius tik tame pačiame administraciniame vienete, o MAPE skirsis priklausomai nuo įvertinamo regiono. Tam tikri modeliai metrikų įvertinimui išvis nenaudojo.

1 lentelė. GTŽ modelių apžvalga. Šioje lentelėje pateikiama: modelio pavadinimas; prieinamumas laike kas n metų; raiška metrais; mokymo žymos raiška metrais (adm. vnt. - administracinis vienetas (skirtingos vietovės turi skirtingas raiškas)); ar modelis perskirsto surašymo duomenis; ar modelis naudoja spektrinius palydovų vaizdus; ar modelis naudoja nakties šviesas; ar modelis naudoja išvestinius vaizdus (pvz.: skaitmeniniai elevacijos modeliai, vandens paviršiaus žemėlapiai); ar modelis naudoja OSM duomenis (pvz.: atstumus iki mokyklų, ligoninių); ar modelis naudoja kitus modelius

Modelis	Metodologija	Prieinamumas (metais)	Raiška (metrais)	Mokymo žymos raiška (metrais)	Perskirsto surašymo duomenis	Naudoja spektrinių palydovų vaizdus	Naudoja nakties šviesas	Naudoja išvestinius vaizdus	Naudoja OSM duomenis	Naudoja kitus modelius
<i>GHS-Pop</i>	SML	5	100	-	Taip	-	-	Taip	Taip	<i>GHS-BUILT</i>
<i>WorldPop</i>	RF	1	100	Adm. vnt.*	Taip	-	Taip	Taip	Taip	<i>GHS-BUILT</i>
<i>LandScan</i>	Svoriai	1	1000	-	Taip	-	-	Taip	NaN	-
<i>MGRW</i>	<i>MGRW</i>	-	100	Adm. vnt.*	Taip	-	Taip	Taip	Taip	<i>GHS-BUILT</i>
<i>MNN</i>	<i>MNN</i>	-	100	100	-	-	Taip	Taip	Taip	World Pop
<i>POMELO</i>	MLP	-	100	Adm. vnt.*	Taip	-	Taip	Taip	Taip	<i>GHS-BUILT</i>
<i>Popcorn</i>	<i>U-Net</i>	-	10	Adm. vnt.*	-	Taip	-	-	-	-
<i>WPI</i>	Svoriai	1	150	-	Taip	Taip	-	Taip	Taip	-
<i>GRUMPV1</i>	Svoriai	5	1000	-	Taip	-	Taip	-	Taip	-
<i>CNNReg</i>	CNN	1	1000	1000	-	Taip	-	-	-	-
<i>RFBUILD</i>	RF	-	100	Adm. vnt.*	Taip	-	Taip	Taip	Taip	<i>GHS-BUILT</i>
<i>MobileNetV2</i>	CNN	-	36	36	-	Taip	-	-	-	-
<i>SynthMod</i>	Reg.	-	500	Adm. vnt.*	Taip	-	Taip	Taip	-	-

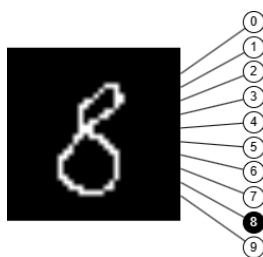
1.3. Konvoliucinių neuroninių tinklų analizė

Šiame skyrelyje iš pradžių aptarsime dažniausiai sprendžiamus uždavinius su CNN, CNN ir *U-Net* architektūras bei nuostolių funkcijas skirtas CNN ir *U-Net* modelių mokymui.

1.3.1. CNN sprendžiami uždaviniai

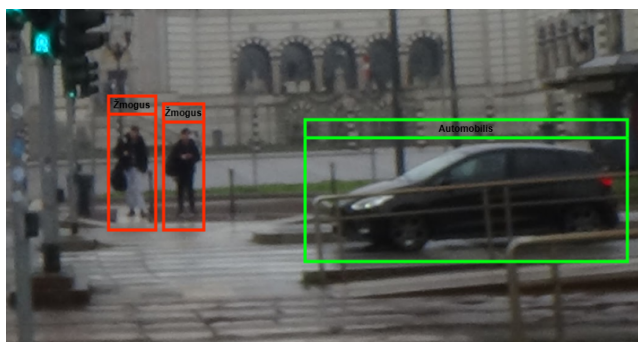
Konvoliuciniai neuroniniai tinklai (CNN) yra naudojami įvairaus pobūdžio problemoms spręsti, dalis šių problemų yra pateiktos žemiau.

Vaizdų klasifikavimas - paprasčiausias CNN uždavinys, kurio tikslas yra vaizdui priskirti kategoriją. Vienas iš labiausiai žinomų klasifikavimo uždavinių yra *MNIST* ranka rašytų skaičių atpažinimas, kur įvestis yra nespalvotas ranka rašyto skaičiaus paveikslukas, o išvestis viena iš dešimties kategorijų, nusakančių koksai tai skaičius (žr. 2 pav.) [21].



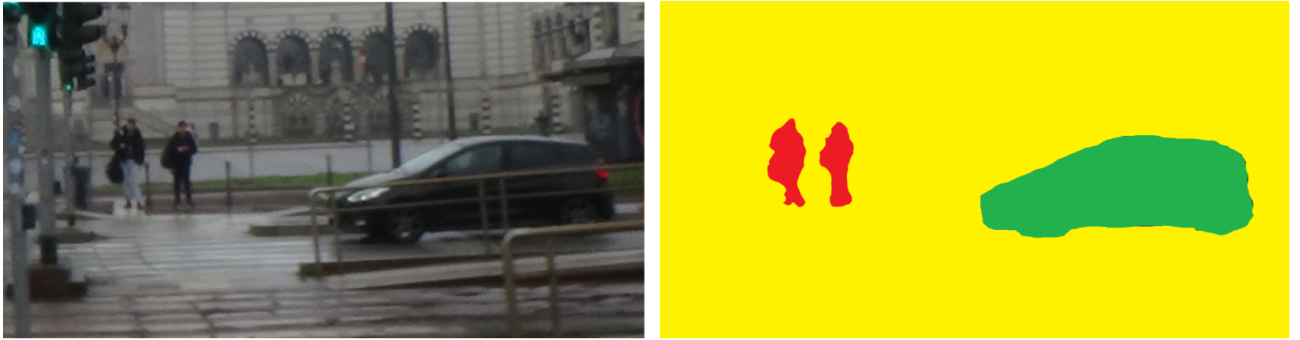
2 pav. *MNIST* klasifikavimo iliustracija

Objektų atpažinimas - tai išplėstas klasifikavimo uždavinys, kurio tikslas ne tik priskirti kategoriją, bet ir nustatyti kur tam tikros kategorijos objektas yra paveikslėlyje. Šio uždavinio pavyzdys būtų gatvėje esančių objektų atpažinimas, šios užduoties įvestis būtų gatvės paveikslukas, o išvestis būtų objektų (gyventojų, mašinų, stulpų ir t.t.) kategorijų ir aprėpties langelių sąrašas (žr. 3 pav.) [21].



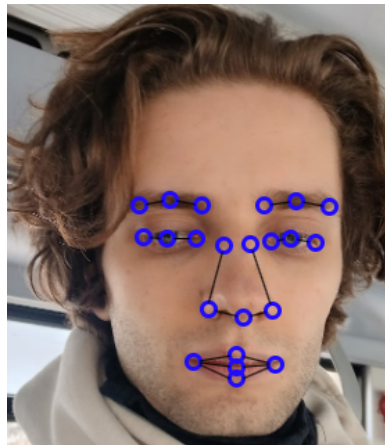
3 pav. Gatvėje esančių objektų atpažinimo iliustracija

Semantinis segmentavimas - iš vaizdo į vaizdą uždavinys. Šio uždavinio tikslas panašus kaip ir objektų atpažinimo, bet vietoje aprėpties langelių sąrašo, semantinio segmentavimo išvestis yra kategorijų vaizdas. Gatvėje esančių objektų pavyzdžio išvestis būtų vaizdas, kurio pikselių vertės būtų kategorijos (1 - žmogus, 2 - mašina ir t.t.) (žr. 4 pav.) [21].



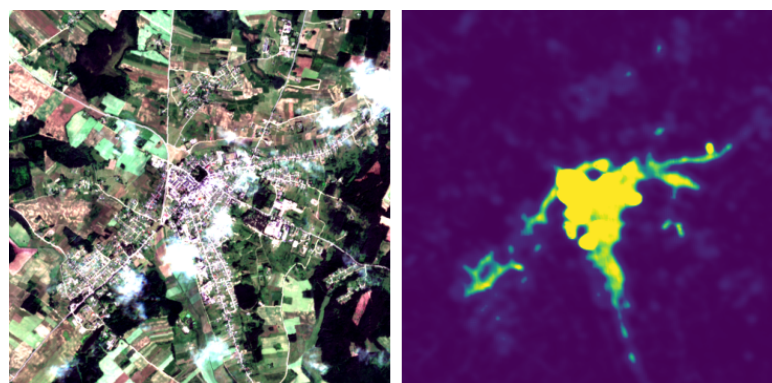
4 pav. Gatvėje esančių objektų segmentavimo iliustracija

Savybių išgavimas - šio uždavinio tikslas iš vaizdo išgauti tam tikras savybes. Vienas labiausiai žinomų šio uždavinio pavyzdžių yra veidų atpažinimas, kurio tikslas - identifikuoti skirtingų žmonių veidus. Veidų atpažinimo metu CNN išgauna tam tikrą savybių sąrašą, pagal kurį galima identifikuoti žmogaus veidą (žr. 5 pav.) [21].



5 pav. Veido savybių išgavimo iliustracija

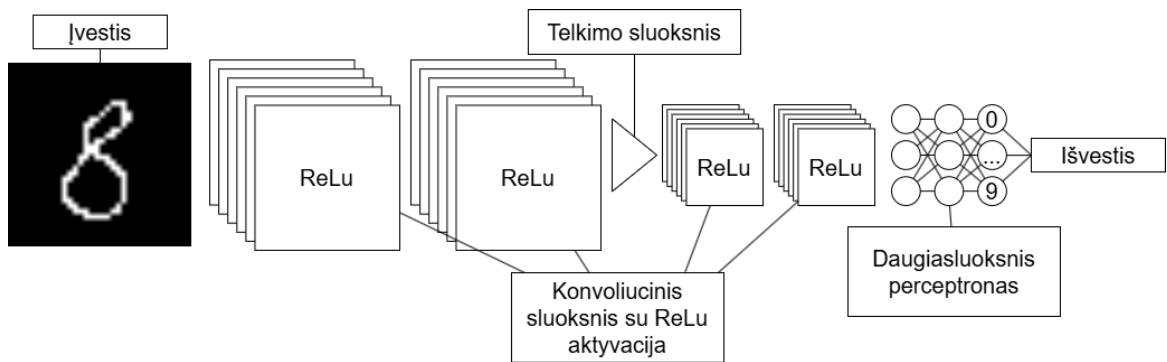
Pikselių lygmens regresija - šis uždavinys, kaip ir semantinis segmentavimas, yra iš vaizdo į vaizdą uždavinys. Šio uždavinio tikslas yra iš vaizdo sudaryti šilumos žemėlapi, nusakantį tam tikrą vaizdo savybę. Keli šio uždavinio pavyzdžiai būtų: vaizdų raiškos gerinimas, augalų maistinių medžiagų trūkumo nustatymas, GTŽ sudarymas (žr. 6 pav.) [22, 23].



6 pav. GTŽ regresijos iliustracija

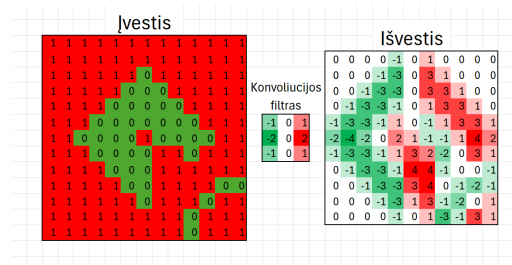
1.3.2. CNN architektūra

CNN (žr. 7 pav.) - tai specifinis neuroninis tinklas, kurio įvestis yra vaizdai. CNN pagrinde susideda iš trijų sluoksnių: konvoliucijos, aktyvavimo ir telkimo [24].



7 pav. CNN architektūra

Konvoliucijos sluoksnis (žr. 8 pav.) - pagrindinė CNN dalis. Šis sluoksnis su mokomu konvoliucijos filtru apdoroja vaizdą, išryškindamas ir išgaudamas tam tikras jo savybes [24].

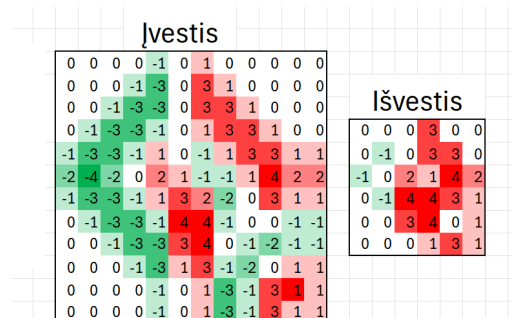


8 pav. Konvoliucijos iliustracija (su Sobelio kraštų filtru)

Aktyvavimo sluoksnis - kaip ir paprastose neuroniniuose tinkuose, šis sluoksnis yra naudojamas stabilizuoti prieš tai buvusio sluoksnio vertėms. Vienas iš šio sluoksnio pavyzdžių būtų aktyvavimo sluoksnis su sigmoidine aktyvacijos funkcija (žr. 1), kuri visas vertes įspaudžia tarp 0 ir 1 [24].

$$S(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

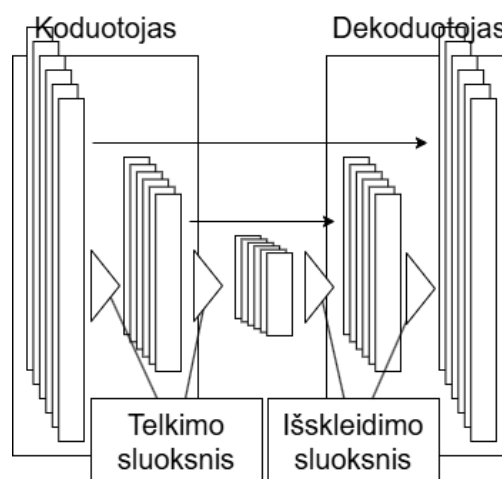
Telkimo sluoksnis (žr. 9 pav.) - šis sluoksnis yra skirtas svarbiausioms savybėms išgauti ir vaizdai sumažinti. Dažniausiai naudojamas telkimo sluoksnis yra maksimalios vertės pasirinkimas 2 x 2 pikselių dydžio lange [24].



9 pav. Telkimo sluoksnis (maksimalios vertės pasirinkimas 2 x 2 pikselių dydžio lange)

1.3.3. U-Net architektūra

U-Net (žr. 10 pav.) - tai CNN modifikacija, skirta smulkesnių vaizdo detalių išsaugojimui. Įprastiniai CNN, telkimo sluoksnyje praranda dalį detalių, šią problemą sprendžia *U-Net*, kuris atstato pradinį vaizdą su išskleidimo sluoksniu ir prieš tai buvusiais sluoksniais. *U-Net* architektūra turi smėlio laikrodžio formą. Iš pradžių koduotojas (įprasta CNN architektūra) užkoduoja vaizdus į mažesnio dydžio vaizdus, tada dekoduoja (išskleidimo sluoksniai su konvoliucijos sluoksniais) atstato pradinio dydžio vaizdą su informacija iš ankstesnių sluoksnių [25].



10 pav. *U-Net* architektūra

1.3.4. Nuostolių funkcijos naudojamos su CNN

CNN modeliams mokytis naudojamos skirtingos nuostolių funkcijos, priklausomai nuo specifinio uždavinio. Žemiau pateiktos dažniausiai naudojamos nuostolių funkcijos, kur Y – tikros vertės, X – modelio prognozuotos vertės.

Klasifikacijos uždavinių nuostolių funkcijos

Kryžminė entropija (žr. 2) – tai dažniausiai klasifikacijos uždaviniams naudojama nuostolių funkcija. Kryžminė entropija parodo skirtumą tarp modelio prognozuotų verčių tikimybių pasiskirstymo (X_k) ir tikrų verčių pasiskirstymo (Y_k). Ši nuostolių funkcija yra jautresnė mažoms X_k tikimybėms (pvz. 0,1, 0,2). Čia k - yra klasė kuriai skaičiuojama kryžminė entropija [21].

$$Loss = - \sum_{k=1}^K Y_k \cdot \log X_k \quad (2)$$

Regresijos uždavinių nuostolių funkcijos

Vidutinė absoliutinė paklaida (MAE) (žr. 3) ir vidutinė kvadratinė paklaida (MSE) (žr. 4) – tai dažniausiai regresijos uždaviniams naudojamos nuostolių funkcijos. MAE yra labiau atsparesnė kraštutinėms vertėms lyginant su MSE, nes MSE apskaičiuoja šių reikšmių kvadratą [21].

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y_i - X_i| \quad (3)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - X_i)^2 \quad (4)$$

Simetrinė vidutinė absoliuti procentinė paklaida (SMAPE) (žr. 5) – nuostolių funkcija, kuri turi daug pranašumų lyginant su MAE ir MSE nuostolių funkcijomis [26].

1. SMAPE reikšmės turi ribas tarp 0 ir 2 (MSE ir MAE vertės gali augti iki begalybės);
2. SMAPE vertės yra reliatyvios lyginamoms reikšmėms, jeigu tikra ir prognozuota reikšmės yra didelės (pvz. 1000 ir 1500), tai SMAPE vertė bus gan maža (0,5);
3. SMAPE vertės yra simetrinės – SMAPE prognozių pervertinimą ir nuvertinimą traktuoja vienodai;
4. SMAPE yra stabili su mažomis vertėmis;
5. SMAPE yra labiau tinkama subjektyvioms užduotims (pvz. vaizdų atkūrimui), dėl lyginamų reikšmių reliatyvumo;
6. SMAPE toleruoja kraštutines vertes, nes SMAPE vertės visada bus tarp 0 ir 2.

$$SMAPE = \frac{1}{n} \sum_{i=1}^n \frac{|Y_i - X_i|}{(|Y_i| + |X_i|)/2} \quad (5)$$

Struktūrinio panašumo indeksas (SSIM) (žr. 6, 7, 8 ir 9) – tai žmogaus regos sistema paremta ir plačiai naudojama metrika panašumui tarp dviejų vaizdų nustatyti. SSIM metrikos pranašumas lyginant su kitomis metrikomis yra tas, kad SSIM lygina ne individualius pikselius, o vaizdų regionus. SSIM įvertina vaizdų panašumus tarp ryškumų (žr. 7), kontrastų (žr. 8) ir struktūrų (žr. 7). SSIM ir yra šių panašumų pasvertų su svoriais α , β , γ sandauga. Kur μ – vaizdo pikselių imties vidurkis, σ^2 – imties variacija, σ_{YX} – Y ir X kovariacija, c_1 , c_2 , c_3 – konstantos skirtos mažų reikšmių stabilizavimui [27].

$$SSIM = l(Y, X)^\alpha \cdot c(Y, X)^\beta \cdot s(Y, X)^\gamma \quad (6)$$

$$l = \frac{2\mu_Y\mu_X + c_1}{\mu_Y^2 + \mu_X^2 + c_1} \quad (7)$$

$$c = \frac{2\sigma_Y\sigma_X + c_2}{\sigma_Y^2 + \sigma_X^2 + c_2} \quad (8)$$

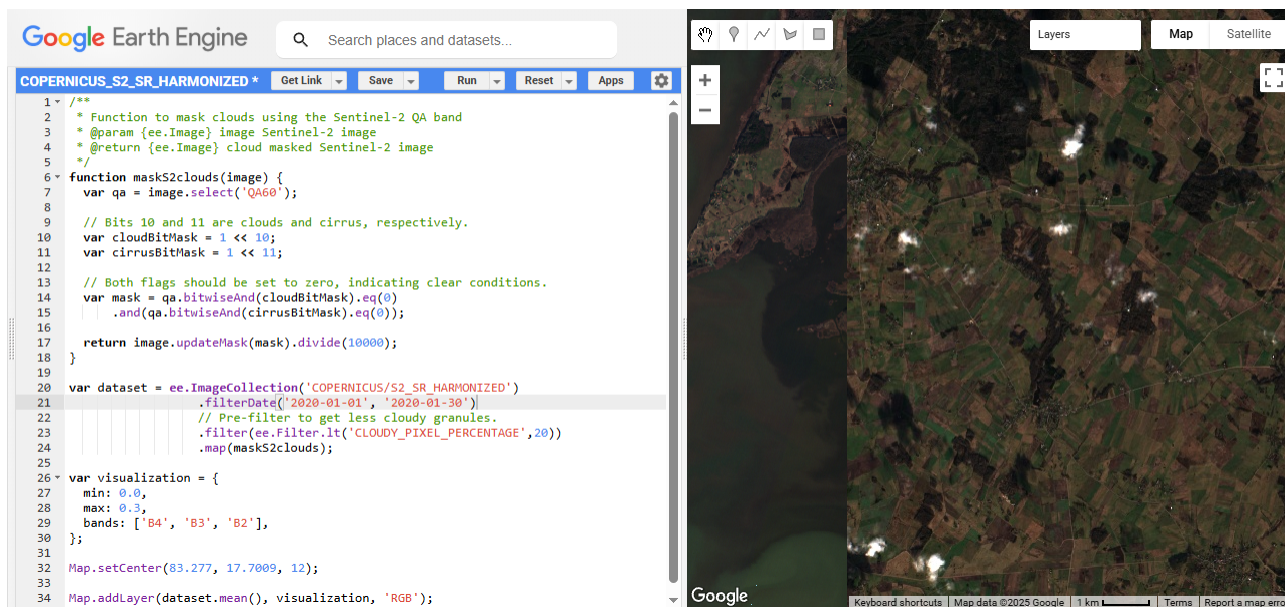
$$s = \frac{\sigma_{YX} + c_3}{\sigma_Y\sigma_X + c_3} \quad (9)$$

1.4. „Google Earth Engine“ įrankio skirtas darbu su palydoviniais vaizdais analizė

„Google Earth Engine“ (GEE) - tai debesijos platforma skirta didelio masto palydovinių vaizdų analizei. Ši platforma turi prieigą prie „Google“ kompiuterijos pajėgumų bei daugelio skirtingų geografinių duomenų, pavyzdžiui, kaip palydoviniai vaizdai ar žemės paskirties žemėlapiai. GEE

suteikia galimybę manipuluoti, apdoroti ir vizualizuoti geografinius duomenis su paprasta vartotojo sąsaja arba su gan paprastu „Java Script“ (11 pav.) arba „Python“ kodu [28].

GEE taip pat leidžia įsikelti savo geografinius duomenis įvairiais formatais, tokiais kaip „GEO TIFF“ ar CSV, kur stulpeliai nurodo poligono formą, kuris aprašo geografinę teritoriją, bei šio poligono savybes [28].



11 pav. GEE naudojimo pavyzdys su „Java Script“ kodu

1.5. Gyventojų tankumo žemėlapių ir konvoliucinių neuroninių tinklų analizės išvados

Išanalizuoti 13 GTŽ sudarymo modelių. Didžioji šių modelių dalis tiesiog perskirsto gyventojus, pasinaudodami gyventojų tikėtinumo žemėlapiiais, kurie buvo sudaryti taikant įvairias metodologijas, tokio žemėlapiio pavyzdys būtų *GHS-BUILT*. Daugelis išanalizuotų modelių naudoja išvestinius duomenis, tokius kaip atstumai iki mokyklų arba žemės paskirties žemėlapius, GTŽ sudarymui. Tik *MNN*, *WPE*, *MobileNetV2* ir *GRUMPV1* modeliai prognozuoja GTŽ tiesiogiai. Taip pat iš šios modelių analizės pastebėta, kad nėra paprastos ir plačiai naudojamos metrikos, skirtos GTŽ tikslumui įvertinti.

Buvo atlikta CNN ir *U-Net* architektūrų analizė. Šios analizės metu pastebėta, kad abi architektūros yra tinkamos pikselių lygmens regresijos uždaviniams spręsti. Taip pat buvo išanalizuotos nuostolių funkcijos, tinkamos šioms architektūroms mokytis. Dažniausiai naudojamos nuostolių funkcijos yra MAE, MSE, SMAPE ir SSIM. Kur SMAPE ir SSIM turi ribotas reikšmes: SMAPE – tarp 0 ir 2, SSIM – tarp 0 ir 1.

Galiausiai buvo apžvelgta GEE platforma, kuri suteikia prieigą prie didelio kiekio geografinių duomenų bei palengvina darbą su jais.

2. Gyventojų tankumo žemėlapių sudarymo metodologija

Šiame skyriuje pateikiama metodologijos, skirtos GTŽ prognozavimo modeliui sukurti ir jo tikslumui įvertinti bei sistemos, su kuria šis modelis būtų naudojamas, aprašas.

2.1. Sistemos aprašas ir reikalavimai

Kuriama GTŽ sistema, leis naudotojui pasirinkti 5 x 5 km dydžio teritoriją ir šiai teritorijai sukurti GTŽ.

GTŽ sudarymo sistemos bei GTŽ sudarymo modelio funkciniai ir nefunkciniai reikalavimai pateikti žemiau.

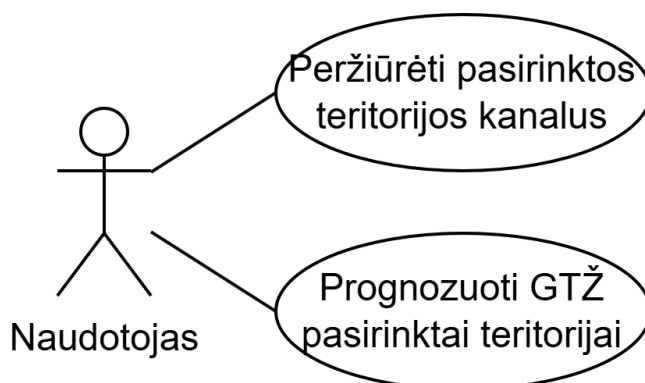
Funkciniai reikalavimai

- modelis turi prognozuoti GTŽ;
- modelio įvestis turi būti 500 x 500 pikselių dydžio žemėlapis, su kanalais tik iš palydovinių vaizdų, su 10 m raiška vienam pikseliui;
- modelio išvestis turi būti 500 x 500 pikselių dydžio GTŽ, su 10 m raiška vienam pikseliui;
- sistema turi leisti pasirinkti 5 x 5 km dydžio teritoriją, kuriai būtų prognozuojamas GTŽ;
- sistema turi leisti matyti pasirinktos teritorijos kanalus, kurie bus naudojami GTŽ prognozavimui.

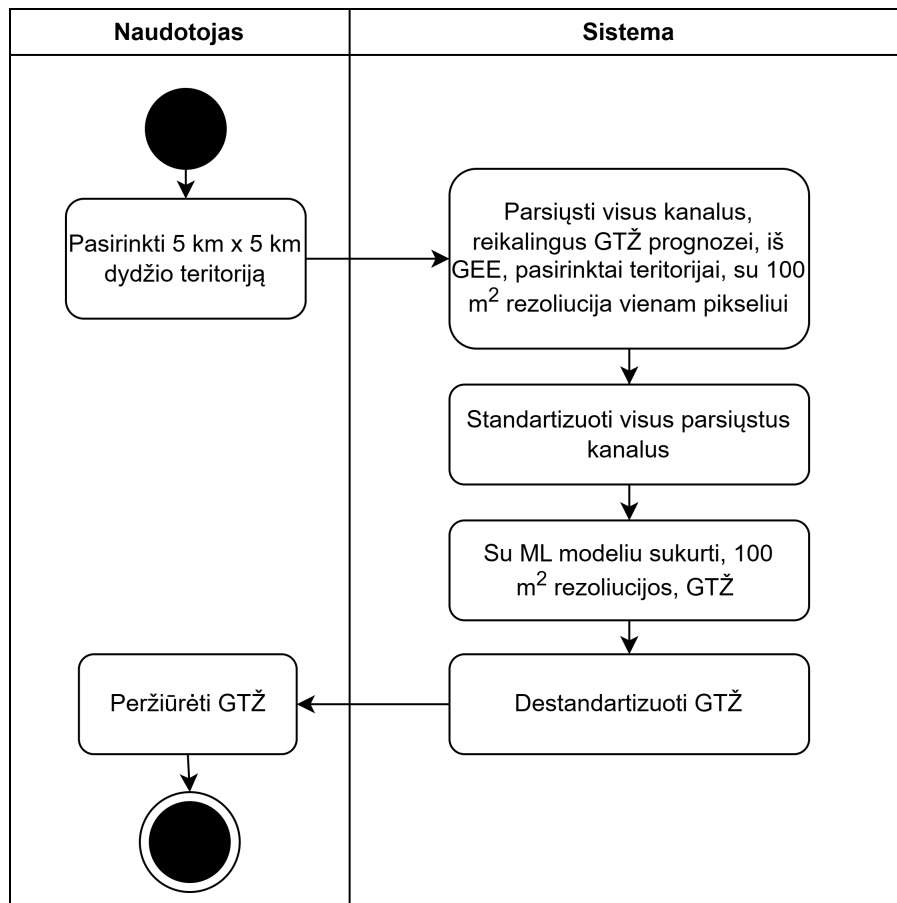
Nefunkciniai reikalavimai

- modelis turi sugebėti identifikuoti negyvenimas teritorijas;
- modelis turi sugebėti identifikuoti įvairaus gyventojų tankumo teritorijas;
- modelis turi sugebėti identifikuoti industrines teritorijas;
- modelio tikslumas turi būti didesnis už *GHS-Pop* tikslumą.

Visi sistemos funkciniai reikalavimai taip pat yra pateikiami panaudos atvejų (12 pav.) bei veiklos diagramų pavidalu (13 pav.). Panaudos atvejai detalizuoti 2 lentelėje ir 3 lentelėje.



12 pav. Panaudos atvejų diagrama



13 pav. „Prognozuoti GTŽ pasirinktai teritorijai“ veiklos diagrama

2 lentelė. „Peržiūrėti pasirinktos teritorijos kanalus“ panaudos atvejo aprašas

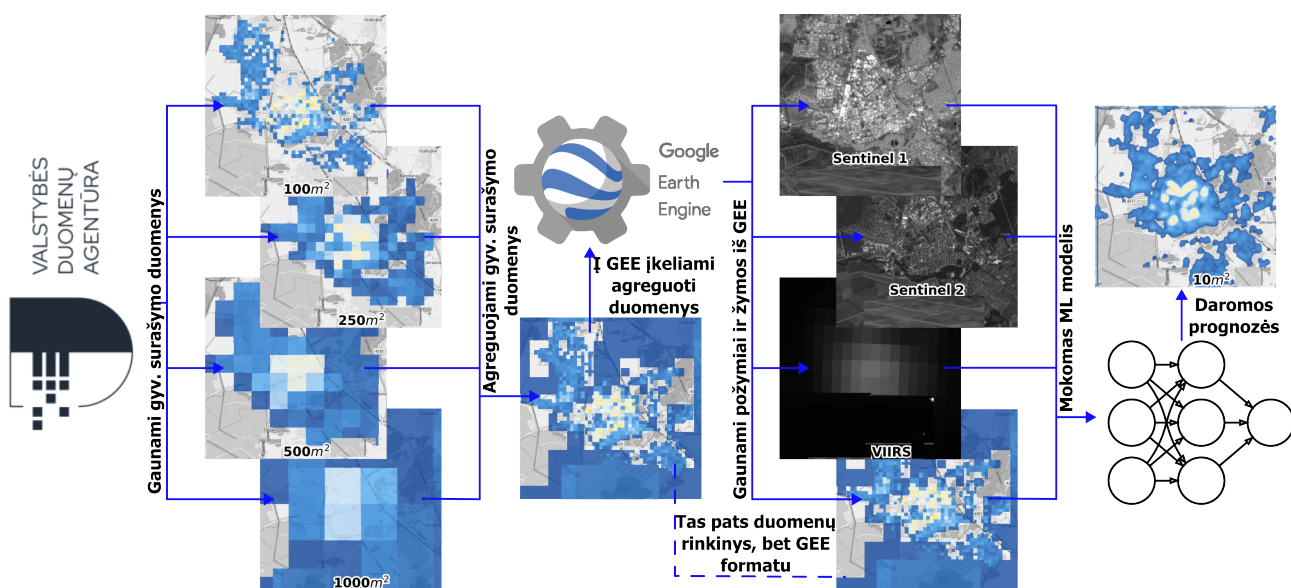
Pavadinimas	Peržiūrėti pasirinktos teritorijos kanalus
Tikslas	Galimybė peržiūrėti pasirinktos teritorijos kanalus
Aktoriai	Naudotojas
Pradinės sąlygos	Turima prieiga prie GEE ir pasirinktas kanalas, kurį norima peržiūrėti
Scenarijus	Naudotojas pasirenka kanalą, gaunami kanalo duomenys iš GEE, pateikiamas pasirinktas kanalas
Galutinės sąlygos	Pateiktas pasirinktas kanalas

3 lentelė. „Prognozuoti GTŽ pasirinktai teritorijai“ panaudos atvejo aprašas

Pavadinimas	Prognozuoti GTŽ pasirinktai teritorijai
Tikslas	Galimybė sukurti GTŽ pasirinktai teritorijai
Aktoriai	Naudotojas
Pradinės sąlygos	Turima prieiga prie GEE ir pasirinkta 5 x 5 km dydžio teritorija, kuriai bus atliekamos GTŽ prognozės
Scenarijus	Naudotojas pasirenka 5 x 5 km dydžio teritoriją, šiai teritorijai gaunami modeliui reikalingų kanalų duomenys iš GEE, kanalai standartizuojami ir pateikiami modeliui, modelis atlieka GTŽ prognozes, GTŽ destandartizuojamas, galutinis GTŽ pateikimas naudotojui
Galutinės sąlygos	Pateiktas 5 x 5 km GTŽ, su 10 m raiška

2.2. GTŽ prognozavimo modelio kūrimo metodologija

14 paveikslėlyje pavaizduota GTŽ prognozavimo modelio kūrimo metodologija. Pirmiausia Lietuvos gyventojų surašymo duomenys, pateikti 100 m, 250 m, 500 m ir 1 km dydžio gardelėse, yra gaunami iš *Valstybės duomenų agentūros* tinklalapio, tada šie gyventojų surašymo duomenys yra sujungiami į vieną bendrą GTŽ ir įkeliami į GEE aplinką. Su GEE atliekamas palydovų kanalų apdorojimas, kuris susideda iš koordinacių sistemos suvienodinimo bei debesuotų pikselių pašalinimo. Po kanalų apdorojimo visi duomenys (*Sentinel 1*, *Sentinel 2*, *VIIRS* ir surašymo GTŽ) yra parsiončiami kaip 5 x 5 km dydžio teritorijų vaizdai su 10 m raiška. Parsiųsti duomenys yra standartizuojami. Su standartizuotais duomenimis yra mokomas ML modelis, kurio įvestis yra palydoviniai kanalai, o išvestis yra GTŽ kanalas.



14 pav. Metodologijos schema

2.2.1. Modelio mokymui naudoti duomenys

Modeliui mokytis skirti įvesties ir išvesties duomenys pateikti 4 lentelėje. Pasinaudojant GEE, *Sentinel 2* ir *Sentinel 1* palydovų kanalai buvo suvidurkinti 2021 metų vasaros periode, prieš tai iš *Sentinel 2* kanalų pašalinus debesuotus pikselius. Iš *Sentinel 1* debesuotų pikselių pašalinti neprireikė, nes *Sentinel 1* yra radaro palydovas, kurio vaizdams oro sąlygos reikšmės neturi. *VIIRS* nakties šviesų intensyvumo kanalas buvo suvidurkintas visame 2021 metų periode, prieš tai iš šio kanalo taip pat pašalinus debesuotus pikselius.

Debesuoti pikseliai buvo identifikuoti pasinaudojant *Sentinel 2 QA60* bei *VIIRS CF_CVG* kanalais, kurie parodo pikselių debesuotumą.

Po kanalų suvidurkinimo visi kanalai su GEE buvo transformuoti į *EPSG:3346* koordinacių sistemą, skirtą naudoti Lietuvoje, kad kanalai nebūtų iškraipyti Lietuvos teritorijoje ir būtų užtikrintas jų tarpusavio suderinamumas.

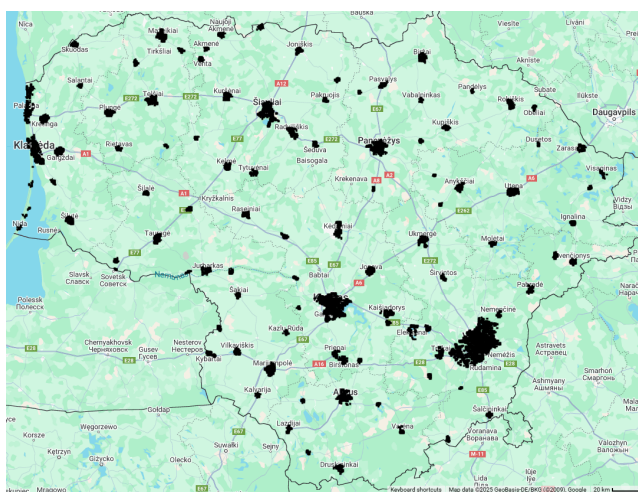
Po domenų paruošimo su GEE, visi duomenys buvo parsiųsti kaip 10 m raiškos vaizdai pasirinktoms teritorijoms.

4 lentelė. Modeliui mokyti skirti įvesties ir išvesties duomenys iš *Sentinel 1*, *Sentinel 2*, *VIIRS* palydovų bei surašymo GTŽ [29]

Palydovas	Kanalas	Raiška	Vertė
<i>Sentinel 2</i>	B1	60 m	443,9 nm - 442,3 nm (Aerozolių) atspindžio koeficientas
<i>Sentinel 2</i>	B2	10 m	496,6 nm - 492,1 nm (Mėlynos) atspindžio koeficientas
<i>Sentinel 2</i>	B3	10 m	560 nm - 559 nm (Žalios) atspindžio koeficientas
<i>Sentinel 2</i>	B4	10 m	664,5 nm - 665 nm (Raudonos) atspindžio koeficientas
<i>Sentinel 2</i>	B5	20 m	703,9 nm - 703,8 nm (Raudono krašto 1) atspindžio koeficientas
<i>Sentinel 2</i>	B6	20 m	740,2 nm - 739,1 nm (Raudono krašto 2) atspindžio koeficientas
<i>Sentinel 2</i>	B7	20 m	782,5 nm - 779,7 nm (Raudono krašto 3) atspindžio koeficientas
<i>Sentinel 2</i>	B8	10 m	835,1 nm - 833 nm (Artimojo infraraudonojo (NIR)) atspindžio koeficientas
<i>Sentinel 2</i>	B8A	20 m	864,8 nm - 864 nm (Raudono krašto 4) atspindžio koeficientas
<i>Sentinel 2</i>	B9	60 m	945 nm - 943,2 nm (Vandens garų) atspindžio koeficientas
<i>Sentinel 2</i>	B11	20 m	1613,7 nm - 1610,4 nm (trumpų bangų infraraudonojo (SWIR) 1) atspindžio koeficientas
<i>Sentinel 2</i>	B12	20 m	2202,4 nm - 2185,7 nm (trumpų bangų infraraudonojo (SWIR) 2) atspindžio koeficientas
<i>Sentinel 1</i>	VV	10 m	Vertikalios poliarizacijos atspindys
<i>Sentinel 1</i>	VH	20 m	Kryžminės (vertikalios/horizontalios) poliarizacijos atspindys
<i>VIIRS</i>	Šv.	500 m	Nakties šviesos intensyvumas
Surašymas	GTŽ	100 m	Gyventojų tankumas 100 x 100 m dydžio teritorijoje

2.2.2. Gyventojų surašymo duomenų agregavimas

Surašymo GTŽ buvo sudarytas agreguojant keturių skirtingų raiškų (100 m, 250 m, 500 m, 1 km) gyventojų surašymo duomenis gautus iš *Valstybės duomenų agentūros*, šie duomenys buvo sudaryti iš geografinių teritorijų (poligonų) ir joms priskirto gyventojų skaičiaus. Agregacijos tikslas buvo sudaryti kiek įmanoma aukštesnės raiškos GTŽ visai Lietuvos teritorijai. 1 km raiškos duomenys buvo pateikti visai Lietuvos teritorijai, o visų likusių raiškų (100 m, 250 m, 500 m) duomenys tik didesniems Lietuvos miestams ir miesteliams (žr. 15 pav.).



15 pav. Lietuvos gyventojų surašymo duomenys 500 m dydžio gardelėse

Pagrindinė agregacijos algoritmo mintis pateikta žemiau:

1. iš žemesnės raiškos (250 m) poligonų atimame aukštesnės raiškos (100 m) poligonus bei jų gyventojus, taip gaudami gyventojų poligonus kurie užpildo duomenų trūkumą 100 m raiškoje;
2. kartojame 1 žingsnį su 500 m ir 1 km raiškomis, taip gaudami trūkumo poligonus 250 m ir 500 m raiškoms;
3. visus duomenų trūkumo ir 100 m raiškos gyventojų poligonus konvertuojame į gyventojų tankumo poligonus, padalindami gyventojų skaičių iš poligono ploto;
4. visus duomenų trūkumo ir 100 m raiškos gyventojų tankumo poligonus sujungiame į bendrą duomenų rinkinį.

Pilna agregacijos algoritmo implementacija, kartu su algoritmu skirtu trūkumo poligonams rasti, yra pateikta žemiau, „Python“ kodo pavidalu.

```
def gdf_diff_hellper(group):
    # Randame kiek gyventoju gyvena large_geo ir small_geo polygonu sankirotje
    group['intersect'] = group['polygon_small'].intersection(group['polygon_large'])
    group['area_intersect'] = group['intersect'].area
    group['area_perc'] = group['area_intersect'] / group['area_small']
    group['pop_small'] = group['pop_small'] * group['area_perc']
    intersection_pop = group['pop_small'].sum()

    # Randame poligona kuris yra large_geo be small_geo poligonu
    small_geo = unary_union(group['polygon_small'])
    big_geo = group['polygon_large'].to_numpy()[0]
    big_pop = group['POP_large'].to_numpy()[0]

    # Apskaiciuojame sankirtos poligona bei jo gyventoju skaičiu
    new_geo = big_geo.difference(small_geo)
    new_pop = big_pop - intersection_pop
    new_pop = 0 if new_pop < 0 else new_pop
    res = {
        'POP': new_pop, 'polygon': new_geo,
        'area': new_geo.area, 'id': group['id_b'].to_numpy()[0]
    }
    return res

def gdf_diff(large_geo, small_geo):
    # Rasti kurie large_geo poligonai kertasi su small_geo poligonais
    merged = gpd.sjoin(
        large_geo, small_geo, how='left', predicate='intersects',
        lsuffix='large', rsuffix='small'
    )

    # Apskaicioti kiekvieno large_geo skirtuma su small_geo, kurie su jo kertasi
    diff = merged.groupby('id_large').apply(lambda group: gdf_diff_hellper(group)).tolist()

    diff = pd.DataFrame(diff)
    diff = gpd.GeoDataFrame(diff, geometry=diff['polygon'])
    diff = diff[diff['area'] >= 10**2].reset_index(drop=True) # Pasalinam mazus plotus
    return diff

def scale_pop_to_100(df):
```

```

# Konvertuojame gyventoju skaičiu i tankuma - žmones per 100 m^2
df = df.copy()
df['POP'] = df['POP'] / (df['area'] / (100**2))
return df

# Skaiciuojame gyventoju "likucius" tarp kiekvienos raiskos poros
diff_0250_0100 = gdf_diff(gyv_0250, gyv_0100)
diff_0500_0250 = gdf_diff(gyv_0500, gyv_0250)
diff_1000_0500 = gdf_diff(gyv_1000, gyv_0500)

# Sujungiame viska i viena duomenu rinkini su 100x100 m raiska
full_df = pd.concat([
    scale_pop_to_100(gyv_0100),
    scale_pop_to_100(diff_0250_0100),
    scale_pop_to_100(diff_0500_0250),
    scale_pop_to_100(diff_1000_0500)
], ignore_index=True).reset_index(drop=True)

```

Turint gyventojų tankumo poligonus, juos įkeliamė į GEE ir konvertuojame į rastrinį vaizdą, taip sudarydami surašymo 100 m raiškos GTŽ.

2.2.3. Modelio mokymas ir modelio parametrų radimas

Modelių mokymui buvo naudojami 5 x 5 km dydžio, 10 m raiškos vaizdai gauti iš įvairių Lietuvos miestų. Visi duomenys buvo standartizuoti.

Modelių testavimui buvo atidėti miestų duomenys, kurie pakliūna į tam tikrą geografinę teritoriją.

U-Net ir CNN modelių parametrms rasti eksperimentai buvo atlikti nuosekliai, kur kiekvienas eksperimentas daro įtaką tolimesniam eksperimentui.

Iš pradžių buvo mokomi modeliai su skirtingomis nuostolių funkcijomis. Pagal šių modelių rezultatus tolimesniems eksperimentams buvo pasirinkta nuostolių funkcija, davusi geriausius rezultatus.

Toliau buvo mokomi modeliai su skirtingais sluoksnių skaičiais (*U-Net* – skirtingais gyliais). Pagal šių modelių rezultatus buvo pasirinktas modelio sluoksnių skaičius / gylis tolimesniems eksperimentams.

Galiausiai buvo mokomi modeliai su skirtingais filtrų skaičiais ir pasirinktas filtrų skaičius, davęs geriausius rezultatus.

2.3. GTŽ modelio įvertinimo metodologija

Dėl nusistovėjusių ir dažnai naudojamų metrikų, skirtų GTŽ sudarymo modelių įvertinimui, nebuvimo, buvo sukurta $\Delta Vid.\%$ procentinio skirtumo metrika (žr. 10 ir 11) tarp surašymo GTŽ (X) ir ML modelio sudaryto GTŽ (Y), gyventojų tankumo intervale $[a, b]$, kur:

- a – gyventojų tankumo intervalo pradžia;
- b – gyventojų tankumo intervalo pabaiga;

- X – surašymo GTŽ;
- Y – ML modelio sudarytas GTŽ.

$$sum(Y, X, a, b) = \sum_{a \leq X_i \leq b} Y_i \quad (10)$$

$$\Delta Vid.\%(Y, X, a, b) = \left(1 - \frac{\min(sum(Y, X, a, b), sum(X, X, a, b))}{\max(sum(Y, X, a, b), sum(X, X, a, b))} \right) \cdot 100 \quad (11)$$

Šios metrikos idėja yra pasirinkus gyventojų tankumo intervalą (pvz. 2 ir 4), apskaičiuoti santykį tarp į šį intervalą patenkančių surašymo GTŽ reikšmių sumos ir atitinkamų surašymo GTŽ reikšmėms ML modelio GTŽ reikšmių sumoms.

Tokia metrika buvo sukurta dėl trijų priežasčių:

1. paprasta metrikos implementacija – metrikai tereikia filtravimo, sumos bei minimalios ir maksimalios vertės funkcijų;
2. metrikos suprantamumas žmogui – metrika pasako kiek procentų, tam tikrame gyventojų tankumo intervale, skiriasi modelio tankumo vidurkis nuo surašymo tankumo vidurkio. Kai metrika lygi 0 %, tai tankumo reikšmės tarp surašymo ir ML modelio sudaryto GTŽ sutampa, kai 50 % – reikšmės skiriasi dvigubai, kai 100 % – reikšmės skiriasi kelis ar net kelias dešimtis kartų;
3. metrikos pritaikomumas skirtingiems gyventojų tankumo intervalams – dėl eksponentinio gyventojų pasiskirstymo [30] reikalinga metrika kuri gali įvertinti tikslumą skirtinguose tankumo intervaluose.

Pagrindinis šios metrikos trūkumas yra nestabilumas su mažomis gyventojų tankumo reikšmėmis. Pavyzdžiui, jeigu surašymo GTŽ vidurkis būtų 0,1, o ML modelio sudaryto GTŽ vidurkis būtų 0,4, tai $\Delta Vid.\%$ reikšmė būtų 75 % procentai, nors realybėje abu GTŽ parodo, kad teritorija praktiškai negyvenama.

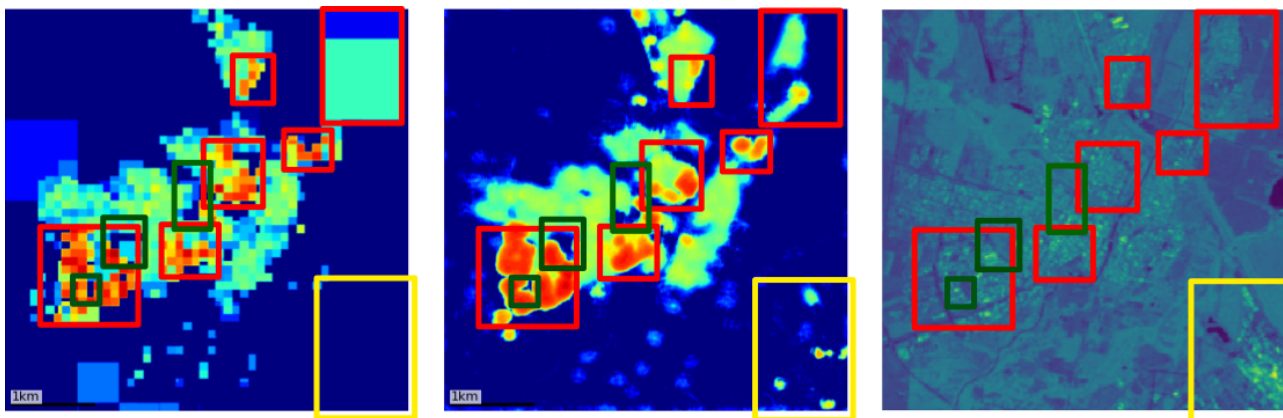
2.3.1. Vizualinis įvertinimas

Be $\Delta Vid.\%$ metrikos pildomai bus atliekamas GTŽ vizualinis vertinimas, kurio metu bus tikrinama, ar GTŽ modelis sugeba atskirti gyvenamas teritorijas nuo negyvenamų teritorijų, ar modelis sugeba atskirti pramonines teritorijas ir ar modelis sugeba atskirti aukšto ir žemo gyventojų tankumo teritorijas.

Detalus vizualinės analizės pavyzdys Kėdainių miestui pateiktas 16 paveikslėlyje. Šiame paveikslėlyje matome, kad:

1. surašymo ir ML modelio sudaryti GTŽ turi panašias negyvenamas teritorijas (mėlynos spalvos plotas);
2. aukšto gyventojų tankumo teritorijos, kurios apibrauktos raudonai, sutampa abejuose GTŽ;

3. ML modelio GTŽ sugeba identifikuoti negyvenamas teritorijas (parkus ir kitas rekreacines erdves) šalia aukšto tankumo teritorijų (apibrauktos žaliai);
4. ML modelis sugeba identifikuoti industrines teritorijas, šio atvėju tai yra geltonai apibraukta trąšų gamybos bendrovės teritorija.



16 pav. Kėdainių miesto GTŽ vizualinis vertinimas. Pirmas vaizdas – surašymo GTŽ; Antras vaizdas – ML modelio sudarytas GTŽ; Trečias vaizdas – radaro palydovo *Sentinel 1* vaizdas

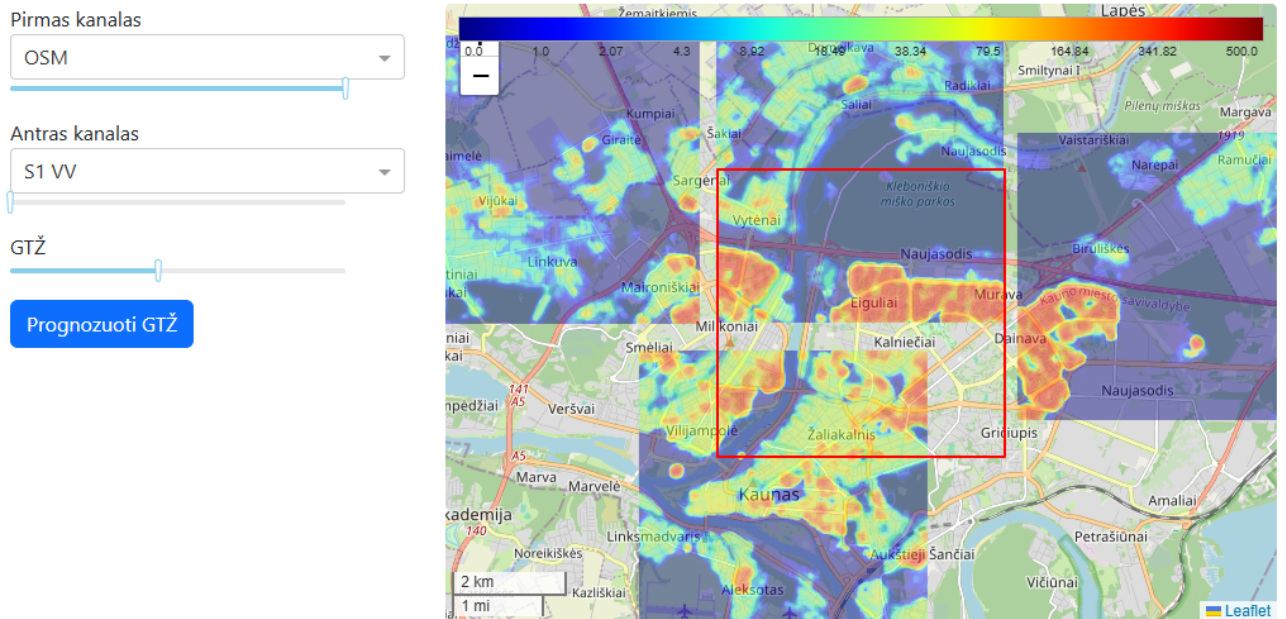
2.4. Gyventojų tankumo žemėlapių sudarymo metodologijos išvados

Pateikta metodologija apima visus esminius žingsnius, reikalingus aukštos raiškos GTŽ prognozavimui naudojant palydovinius duomenis ir ML metodus. Pagrindė metodologija susideda iš:

1. surašymo GTŽ sudarymo iš Lietuvos gyventojų surašymo duomenų;
2. surašymo GTŽ duomenų ir *Sentinel-1*, *Sentinel-2*, *VIIRS* palydovų duomenų parsisiuntimo iš GEE bei jų standartizavimo;
3. CNN ir *U-Net* modelių mokymo su skirtingais parametrais, kad būtų galima rasti tinkamą modelio architektūrą bei nuostolių funkciją modeliui mokyti;
4. modelio įvertinimo su $\Delta Vid. \%$ metrika.

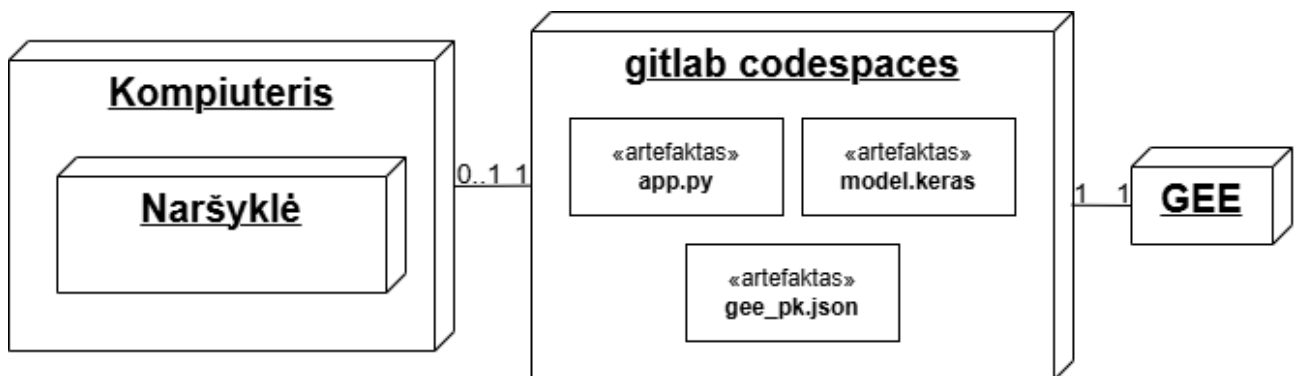
3. Gyventojų tankumo žemėlapių sistemos implementacija

Šiame skyriuje pateikiama gyventojų tankumo žemėlapių sistemos implementacija (17 pav.).



17 pav. GTŽ sistemos naudotojo sąsaja

18 paveikslėlyje matome sistemos statinį vaizdą, kur sistema susideda iš trijų pagrindinių failų: *app.py* – pagrindinis *Python* programos failas, *model.keras* – GTŽ sudarymo modelio *keras* failas ir *gee_pk.json* konfigūracijos failas, skirtas prisijungimui prie GEE.



18 pav. GTŽ sistemos diegimo diagrama

4. Eksperimentai GTŽ modeliui sukurti

Šiame skyriuje yra pateikiama duomenų, naudojamų GTŽ modeliams sukurti, analizė bei ML modelių kūrimo eksperimentai, skirti GTŽ sudarymui. Eksperimentai fokusuojasi į CNN ir *U-Net* modelių architektūras, skirtas aukštos raiškos GTŽ sudarymui, jų nuostolių funkcijas ir modelių optimizavimą, kas ir yra šio tiriamojo darbo tikslas. Skyriaus gale yra pateikiamas geriausio GTŽ modelio palyginimas su *GHS-POP* modeliu bei geriausio GTŽ modelio rezultatai skirtingoms vietovėms.

4.1. Eksperimentų atlikimo aplinka

Visi eksperimentai aprašyti šiame skyrelyje buvo atlikti „Google Colab“ platformoje su „Python 3.10“ programavimo kalba ir „tensorflow“ biblioteka skirta ML. Sistemos specifiška pateikta 5 lentelėje.

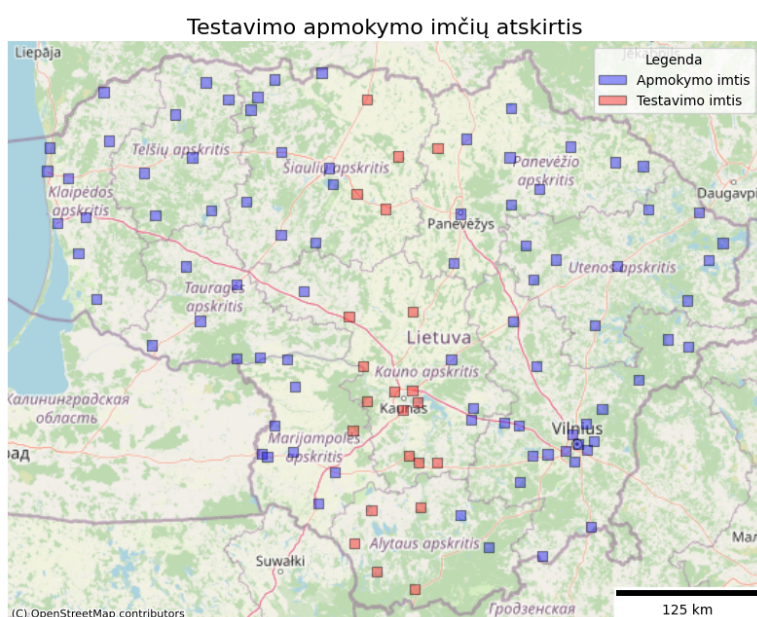
5 lentelė. Sistemos specifikacija

CPU	x86_64
CPU Branduoliai	6
CPU Gijos	12
CPU RAM	83.48 GB
GPU	NVIDIA A100-SXM4-40GB
GPU RAM	39.56 GB
OS	posix

4.2. Duomenų analizė

Eksperimentai buvo atlikti su 108 miestų, 10 m raiškos, 5 x 5 km plotų vaizdais. 86 miestai buvo naudojami modeliams mokytis, o 22 miestai buvo naudojami modeliams testuoti. Visi toliau pateikti modelių įvertinimai buvo apskaičiuoti su testavimo imtimi. Visi testavimui ir mokymui panaudoti miestai pateikti 19 paveikslėlyje.

Testavimo ir mokymo imtys buvo sudarytos atidedant miestus esančius tarp platumų 23,4664 ir 24,2794 testavimui. Į testavimo imtį patenka antras pagal dydį miestas Lietuvoje – Kaunas.



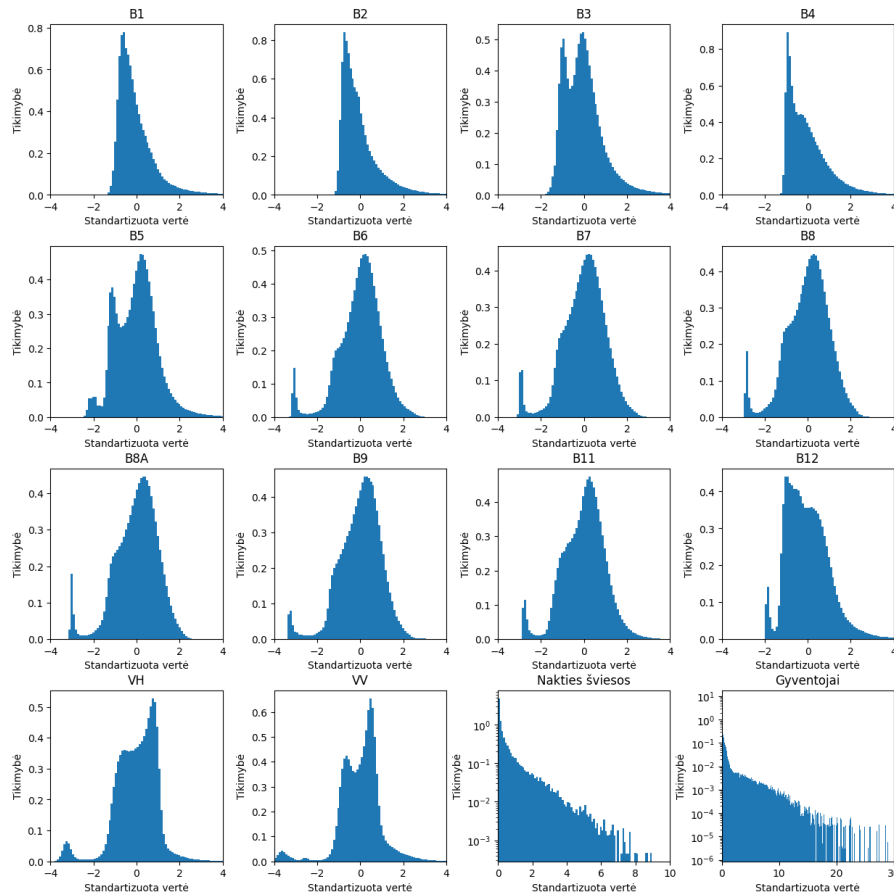
19 pav. Testavimo ir mokymo imčių atskirtis, pavaizduota Lietuvos žemėlapyje. Vienas imties elementas yra 5 x 5 km dydžio plotas

Duomenų statistinė analizė pateikta 6 lentelėje, o 20 paveikslėlyje pateiktos palydovų spektrinių kanalų bei gyventojų ir nakties šviesų duomenų histogramos. Histogramų sudarymui visi duomenys buvo standartizuoti. Prieš apskaičiuojant standartinį nuokrypį gyventojų ir nakties šviesų duomenims, šie duomenys buvo simetriškai atspindėti (žr. 12), dėl pastarųjų duomenų eksponentinio pasiskirstymo.

$$A = B \cup (-B), \text{ kai } \{B \in \mathbb{R} \mid B \geq 0\} \quad (12)$$

6 lentelė. Modeliui mokyti naudotų kanalų statistiniai duomenys: vidurkis (Vid.), standartinis nuokrypis (SD), minimali vertė (Min.), pirmas kvartilis (Q1), antras kvartilis (Q2), trečiasis kvartilis (Q3), maksimali vertė (Max.)

	B1	B2	B3	B4	B5	B6	B7	B8	B8A	B9	B11	B12	VH	VV	Nak.	Gyv.
Vid.	416	489	737	620	1125	2571	3095	3198	3326	3385	1917	1174	-17	-10	7	5
SD	241	305	333	415	432	777	990	1068	1053	995	650	581	3	3	12	29
Min.	43	1	1	1	60	1	1	33	0	29	35	21	-29	-25	0	0
Q1	266	292	496	302	801	2154	2507	2545	2701	2801	1504	739	-19	-12	0	0
Q2	353	406	697	520	1140	2648	3179	3303	3444	3485	1972	1109	-16	-10	1	0
Q3	493	578	882	810	1378	3064	3758	3924	4050	4042	2334	1526	-15	-9	7	0
Max.	5560	17392	16880	16256	12323	12602	12041	15544	12273	11828	13300	14891	15	22	136	1036



20 pav. Standartizuotų, modeliui mokyti naudotų kanalų, histogramos

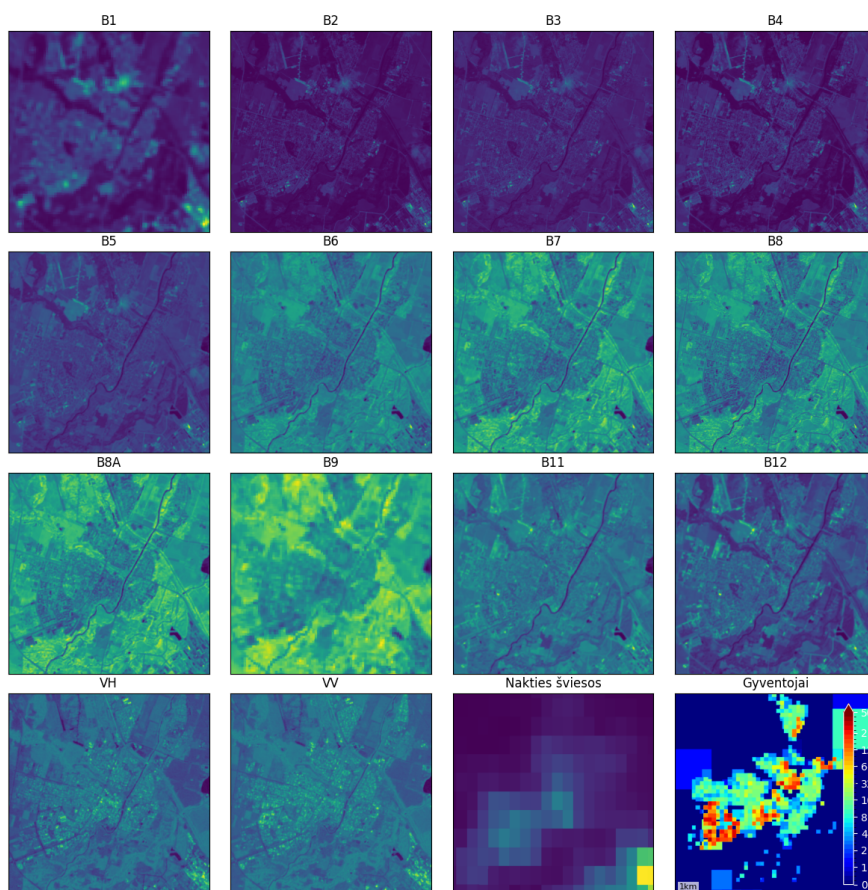
Iš statistinės analizės matome, kad *Sentinel-1* ir *Sentinel-2* palydovų visų spektrinių kanalų duomenys turi normalųjį pasiskirstymą. Nakties šviesų ir gyventojų duomenys turi eksponentinį pasiskirstymą, kur didžioji dalis verčių yra lygios nuliui.

7 lentelėje pateiktas gyventojų tankumo verčių pasiskirstymas skirtinguose intervaluose. Kaip matyti iš lentelės 7, net 72 % pikselių neturi gyventojų, o likusiųjų gyventojų tankumo pasiskirstymas mažėja eksponentiškai — kuo didesnis gyventojų tankumas, tuo pikselių su tokia verte mažiau.

7 lentelė. Gyventojų tankumo verčių pasiskirstymas skirtinguose intervaluose

Nuo	Iki	Proporcija
0	0,5	72,17 %
0,5	1	4,86 %
1	2	3,89 %
2	4	4,89 %
4	8	3,86 %
8	16	4,03 %
16	32	3,00 %
32	64	1,07 %
64	128	0,99 %
128	256	0,92 %
256	∞	0,34 %

Visų spektrinių kanalų, nakties šviesų ir GTŽ vizualinė reprezentacija yra pateikta 21 paveikslėlyje.



21 pav. Įvesties ir išvesties duomenys Kėdainių miestui

4.3. Duomenų paruošimas

Visi duomenys naudojami tolimesniuose eksperimentuose buvo standartizuoti. Nakties šviesų duomenims, prieš standartizavimą, papildomai buvo pritaikyta logaritminė transformacija.

Galutinė modelio įvestis buvo 500 x 500 x 16 dydžio matrica, kur 500 x 500 – teritorijos dydis, 16 – skirtingi kanalai (12 – *Sentinel-2* kanalų, 2 – *Sentinel-1* kanalų, vienas nakties šviesų kanalas ir vienas logaritmiškai transformuotas nakties šviesų kanalas).

4.4. CNN eksperimentai

Šiame skyrelyje pateikti eksperimentai skirti surasti nuostolių funkciją, su kuria mokytas CNN modelis turėtų mažiausią $\Delta\text{Vid.}\%$ (žr. 11) vertę visuose gyventojų intervaluose, bei nustatyti CNN modelio sluoksnių ir filtrų skaičiaus poveikį $\Delta\text{Vid.}\%$ vertėms.

CNN modelio architektūros šablonas, pagal kurį sukurti visi CNN modeliai aptarti šiame skyrelyje, pateiktas žemiau „Python“ kodo pavidalu. Kiekvienas modelis turi 500 x 500 x 16 dydžio įvestį, nustatytą sluoksnių skaičių, kur kiekvienas sluoksnis turi nustatytą skaičių filtrų, 3 x 3 dydžio konvoliucijos kaukę bei „relu“ aktyvacijos funkciją, papildomą sluoksnį su vienu filtru, 3 x 3 dydžio konvoliucijos kaukę bei „linear“ aktyvacijos funkcija. Šis papildomas sluoksnis yra skirtas modelio išvesčiai. Visi modeliai buvo mokyti su „adam“ optimizatoriumi.

```
from tensorflow.keras.layers import Input, Conv2D
from tensorflow.keras import Sequential

def create_cnn_model(n_layers, n_filters, loss):
    model = Sequential()
    model.add(Input(shape=(500, 500, 16)))
    for i in range(n_layers):
        model.add(Conv2D(n_filters, (3, 3), activation='relu', padding='same'))
    model.add(Conv2D(1, (3, 3), activation='linear', padding='same'))
    model.compile(optimizer="adam", loss=loss)
    return model

create_cnn_model(5, 5, 'mse').summary()
```

4.4.1. Nuostolių funkcijos pasirinkimas

Nuostolių funkcijai atrasti, buvo mokyta 11 CNN modelių, su 10 sluoksnių ir 10 filtrų per sluoksnį, iš jų 3 modeliai buvo mokyti su MSE (žr. 4), SMAPE (žr. 5) ir SSIM (žr. 6) nuostolių funkcijomis, o likę 8 modeliai buvo mokyti su pastarųjų nuostolių funkcijų kombinacijomis ir modifikacijomis.

Visos modifikacijos buvo taikomos SMAPE nuostolių funkcijai, dėl jos žinomos minimalios (0) ir maksimalios (2) vertės, su tikslu subalansuoti gyventojų eksponentinio pasiskirstymo (7 lentelė) poveikį skirtingiems gyventojų rėžiams. Pavyzdžiui, jeigu 90 % GTŽ pikselių verčių yra 1, tai modelio, kuris sudaro GTŽ kurio visos pikselių vertės yra 1, SMAPE vertė būtų 0,2. Paprasčiausia SMAPE modifikacija buvo B_SMAPE (žr. 13). B_SMAPE, tai SMAPE verčių, apskaičiuotų gyventojų intervalams, kur intervalai buvo eksponentiškai didėjantys, vidurkis. Prieš tai pateiktam pavyzdžiui B_SMAPE vertė būtų 1,77. W_SMAPE (žr. 14) – nuostolių funkcija panaši į B_SMAPE, kur kiekvieno gyventojų intervalo SMAPE vertė buvo padauginta iš intervalo dydžio proporcijos visam intervalui. W_SMAPE vertė, kai $b = 8$, prieš tai pateiktam pavyzdžiui būtų 1,99. Tiek B_SMAPE, tiek W_SMAPE gyventojų intervalų sudarymui naudoja Y (surašymo GTŽ vertes).

BB_SMAPE (žr. 15) ir BW_SMAPE (žr. 16), tai B_SMAPE ir W_SMAPE atmainos, kur

atitinkama pradinė nuostolių funkcija buvo apskaičiuota du kartus ir susumuota, gyventojų intervalų sudarymui naudojant tiek Y (surašymo GTŽ vertes), tiek X (modelio GTŽ vertes).

B_SMAPE_S (žr. 17), W_SMAPE_S (žr. 18), BB_SMAPE_S (žr. 19) ir BW_SMAPE_S (žr. 20) nuostolių funkcijos buvo sudarytos prie B_SMAPE , W_SMAPE , BB_SMAPE , BW_SMAPE pridėdant SSIM nuostolių funkciją.

$$B_SMAPE = \frac{1}{b} \sum_{i=0}^b SMAPE(Y_{2^i:2^{i+1}}, X_{2^i:2^{i+1}}) \quad (13)$$

$$W_SMAPE = \sum_{i=0}^b SMAPE(Y_{2^i:2^{i+1}}, X_{2^i:2^{i+1}}) \cdot \frac{2^{i+1} - 2^i}{2^i} \quad (14)$$

$$BB_SMAPE = B_SMAPE(Y, X) + B_SMAPE(X, Y) \quad (15)$$

$$BW_SMAPE = W_SMAPE(y, X) + W_SMAPE(X, y) \quad (16)$$

$$B_SMAPE_S = B_SMAPE + SSIM \quad (17)$$

$$W_SMAPE_S = W_SMAPE + SSIM \quad (18)$$

$$BB_SMAPE_S = BB_SMAPE + SSIM \quad (19)$$

$$BW_SMAPE_S = BW_SMAPE + SSIM \quad (20)$$

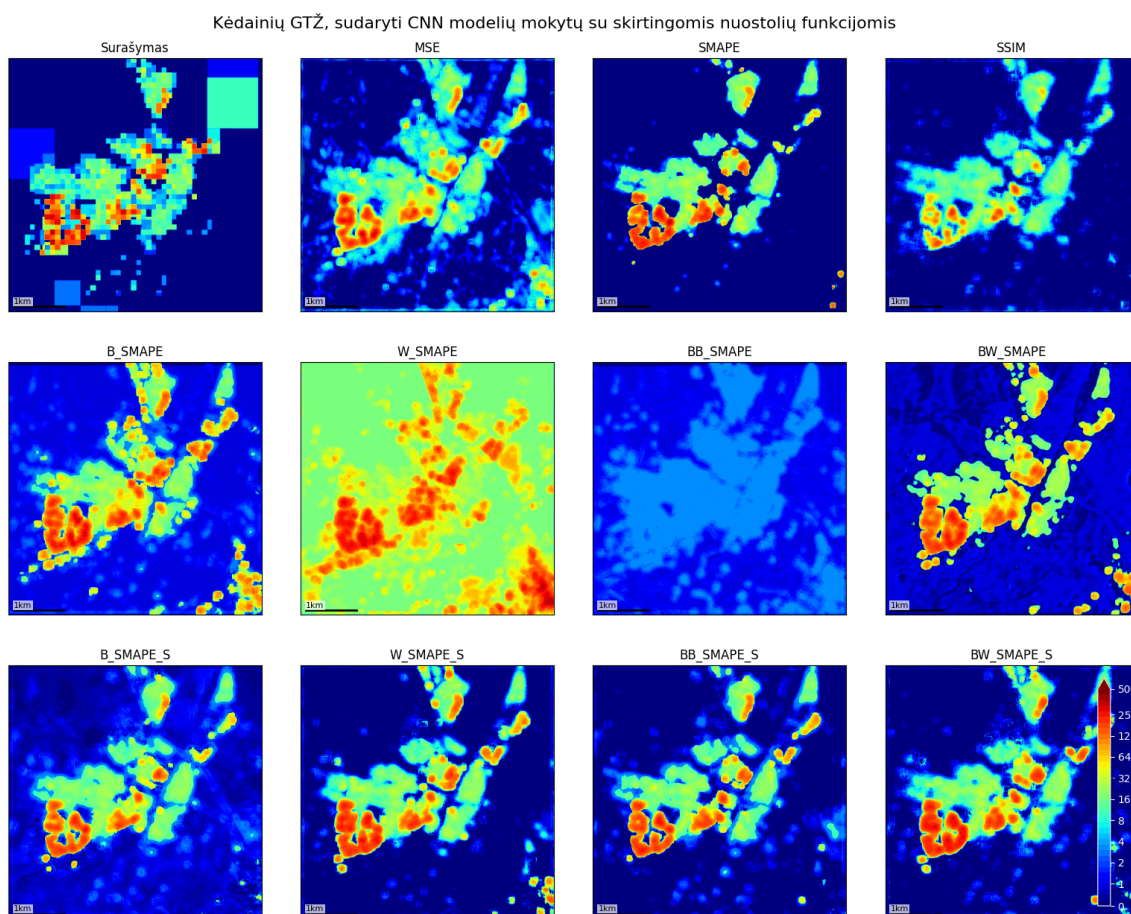
- Y – Surašymo GTŽ.
- X – Modelio sudarytas GTŽ.
- 2^b – Didžiausia gyventojų intervalo vertė.

8 lentelėje ir 22 paveikslėlyje pateikti CNN modelių, mokytų su prieš tai minėtomis nuostolių funkcijomis, rezultatai. Iš 8 lentelės matome, kad SMAPE duoda geriausias rezultatus – visuose gyventojų intervaluose ši nuostolių funkcija duoda mažas Δ Vid.% vertes (23 %), o 22 paveikslėlyje matome, kad ši nuostolių funkcija aiškiai išskiria gyvenamas vietas ir industrines teritorijas. SSIM nuostolių funkcija žemesniuose gyventojų intervaluose duoda dar geresnius rezultatus (6 % - 12 %) negu SMAPE, tačiau rezultatai pradeda blogėti, kai intervalas pasiekia 16 gyventojų ir daugiau. Iš 22 paveikslėlio matome, kad SSIM dar geriau išskiria industrines teritorijas. MSE nuostolių funkcija buvo linkusi pervertinti gyventojų skaičių per ± 46 % žemesniuose gyventojų intervaluose. Geresnius rezultatus (8 % - 30 %) MSE pradėjo duoti, kai intervalas pasiekia 8 gyventojus. Iš 22 paveikslėlio matome, kad MSE nesugeba išskirti industrinių teritorijų.

B_SMAPE ir W_SMAPE – SMAPE verčių, apskaičiuotų gyventojų intervalams, vidurkis nedavė geresnių rezultatų negu SMAPE nuostolių funkcija. Tiek iš 8 lentelės, tiek iš 22 paveikslėlio matome, kad B_SMAPE ir W_SMAPE stipriai pervertina gyventojų skaičių mažuose gyventojų intervaluose (B_SMAPE – iki 16 gyventojų (per 60 %), W_SMAPE – iki 256 gyventojų (per 90 %)). B_SMAPE ir W_SMAPE nesugeba išskirti industrinių teritorijų.

Intervalų sudarymas naudojant Y (surašymo GTŽ vertes) ir X (modelio GTŽ vertes) pablogino B_SMAPE rezultatus ir pagerino W_SMAPE rezultatus. Modelis su BB_SMAPE sugebėjo diferencijuoti tik tarp apstatytų ir neapstatytų teritorijų, visuose apstatytuose teritorijose gyventojų tankumo vertę prilygindamas 3 (kai gyventojų intervalas didesnis už 4, vidurkis visada būna 3, o standartinis nuokrypis 0). BW_SMAPE, lyginant su W_SMAPE, pagerino rezultatus žemesniuose gyventojų intervaluose.

SSIM nuostolių funkcijos pridėjimas prie SMAPE modifikacijų pagerino visus rezultatus visuose gyventojų intervaluose, taip pat visos SMAPE modifikacijos su SSIM nuostolių funkcija sugebėjo identifikuoti industrines teritorijas.

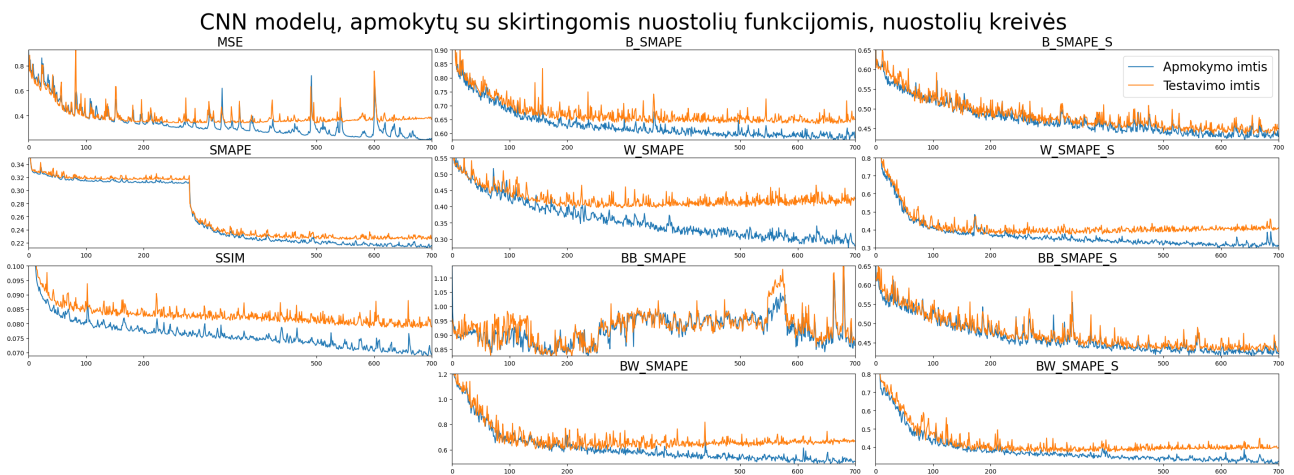


22 pav. CNN modelių, mokyti su skirtingomis nuostolių funkcijomis, Kėdainių miestui sudaryti GTŽ. Žemėlapis apatiniame dešiniajame kampe yra industrinė teritorija

8 lentelė. CNN modelių, mokytų su skirtingomis nuostolių funkcijomis, Δ Vid.% verčių lentelė skirtingiems gyventojų intervalams. Eilučių reikšmės: Gyventojai – gyventojų intervalas (Nuo, Iki), Surašymas – Surašymo GTŽ vidurkis (Vid.) ir standartinis nuokrypis (SN), toliau visų mokytų modelių vidurkis (Vid.), standartinis nuokrypis (SN) ir GTŽ vidurkio procentinis skirtumas lyginant su surašymo GTŽ vidurkiu (Δ Vid.%). Paskutiniame stulpelyje pateiktas modelio visų intervalų Δ Vid.% verčių vidurkis

Gyventojai	Nuo	0,0	0,5	1,0	2,0	4	8	16	32	64	128	256	Vid.
	Iki	0,5	1,0	2,0	4,0	8	16	32	64	128	256	500	
Surašymas	Vid.	0,1	0,7	1,4	3,0	6	11	22	44	93	182	319	
	SN	0,1	0,2	0,3	0,6	1	2	4	9	18	35	54	
MSE	Vid.	1,4	1,3	2,6	5,2	9	13	20	35	81	133	181	
	SN	6,5	4,2	7,6	9,9	14	17	18	38	58	67	70	
	ΔVid. %	92 %	46 %	46 %	42 %	30 %	15 %	8 %	20 %	12 %	26 %	43 %	
SMAPE	Vid.	0,5	0,5	1,3	3,8	8	14	21	35	96	156	212	
	SN	7,3	4,0	8,2	12,1	16	19	21	44	76	83	79	
	ΔVid. %	80 %	28 %	7 %	21 %	23 %	17 %	5 %	18 %	2 %	14 %	33 %	
SSIM	Vid.	0,5	0,8	1,6	3,6	6	10	16	21	49	92	142	
	SN	3,8	2,5	4,6	5,9	9	11	12	25	51	71	81	
	ΔVid. %	80 %	12 %	12 %	16 %	6 %	10 %	28 %	50 %	46 %	49 %	55 %	
B_SMAPE	Vid.	3,6	3,1	5,4	9,3	15	21	30	53	118	170	219	
	SN	14,1	10,1	16,3	19,2	23	26	28	46	65	69	68	
	ΔVid. %	97 %	77 %	74 %	67 %	59 %	46 %	27 %	17 %	20 %	6 %	31 %	
W_SMAPE	Vid.	25,1	22,3	28,2	33,8	40	45	55	90	170	226	273	
	SN	28,9	22,2	28,1	33,5	38	40	44	65	74	69	54	
	ΔVid. %	99 %	96 %	95 %	91 %	85 %	74 %	59 %	50 %	45 %	19 %	14 %	
BB_SMAPE	Vid.	1,4	1,7	2,0	2,5	3	3	3	3	3	3	3	
	SN	0,7	0,8	0,8	0,8	0	0	0	0	0	0	0	
	ΔVid. %	92 %	58 %	30 %	16 %	53 %	74 %	86 %	93 %	96 %	98 %	99 %	
BW_SMAPE	Vid.	2,3	1,7	3,7	7,2	13	20	29	50	111	158	199	
	SN	12,3	8,7	14,2	18,9	24	26	28	45	59	59	54	
	ΔVid. %	95 %	58 %	62 %	58 %	52 %	41 %	24 %	12 %	16 %	13 %	37 %	
B_SMAPE_S	Vid.	1,2	1,3	2,2	4,9	8	14	20	32	82	135	185	
	SN	6,6	3,9	7,2	11,0	14	17	18	37	65	71	72	
	ΔVid. %	91 %	46 %	36 %	38 %	27 %	18 %	6 %	26 %	11 %	25 %	41 %	
W_SMAPE_S	Vid.	1,1	1,0	2,5	5,6	10	16	23	43	109	164	209	
	SN	9,3	5,8	10,9	14,9	18	22	24	47	68	65	52	
	ΔVid. %	90 %	30 %	44 %	46 %	37 %	26 %	4 %	2 %	14 %	9 %	34 %	
BB_SMAPE_S	Vid.	0,8	1,0	1,9	4,6	8	14	21	35	87	140	187	
	SN	6,6	4,9	7,4	11,2	15	18	20	39	65	70	68	
	ΔVid. %	87 %	30 %	26 %	34 %	27 %	20 %	4 %	20 %	6 %	23 %	41 %	
BW_SMAPE_S	Vid.	1,1	1,0	2,4	5,3	10	16	24	45	115	176	224	
	SN	9,3	5,1	9,6	14,1	22	25	26	51	75	73	55	
	ΔVid. %	90 %	30 %	41 %	43 %	41 %	30 %	10 %	3 %	19 %	3 %	29 %	

Nors geriausi rezultatai buvo modelio su SMAPE nuostolių funkcija, tolimesniems eksperimentams atlikti naudosime SSIM nuostolių funkciją, dėl pastarosios funkcijos gebėjimo išskirti industrines teritorijas bei stabilaus mokymosi, nes kaip matome iš 23 paveikslėlio, modelis su SSIM nuostolių funkcija mokosi tolygiai per epochas, kol modelis su SMAPE nuostolių funkcija ilgą laiką visiškai nesimoko, ir tada SMAPE vertė staigiai nukrenta. W_SMAPE_S, BB_SMAPE_S ir BW_SMAPE_S nuostolių funkcijos nebuvo pasirinktos dėl to, kad jos sumažina Δ Vid.% tik per ± 3 % lyginant su SSIM, bet prailgina modelio mokymąsi kelis kartus.

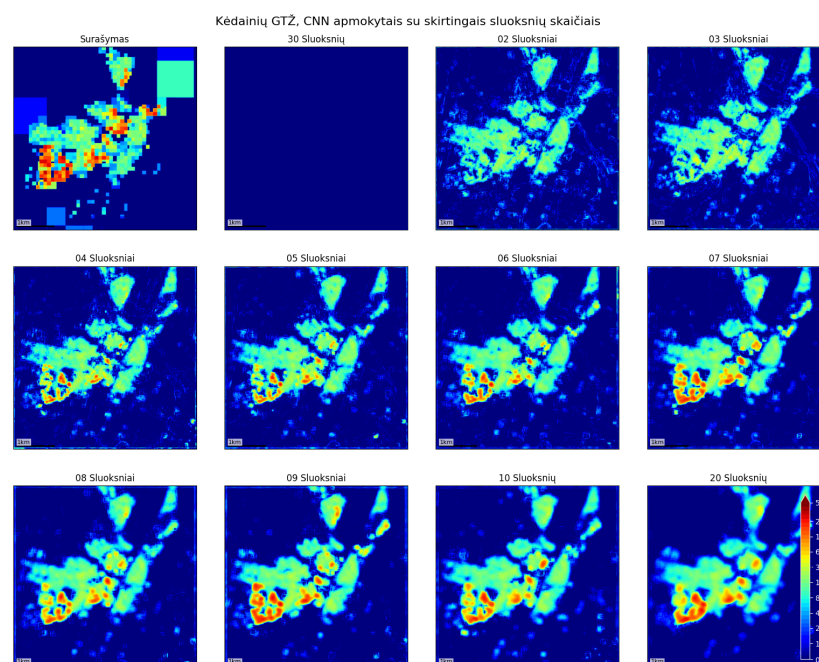


23 pav. CNN modelių, mokyty su skirtingomis nuostolių funkcijomis, nuostolių kreivės

4.4.2. Sluoksnių skaičiaus pasirinkimas

Sluoksnių skaičiui atrasti buvo mokyta 12 CNN modelių, su SSIM nuostolių funkcija, 10 filtrų per sluoksnį ir su skirtingais sluoksnių skaičiais. Šių modelių rezultatai pateikti 9 lentelėje ir 24 paveikslėlyje. Šie rezultatai parodo, kad:

1. nepriklausomai nuo sluoksnių skaičiaus visi modeliai su SSIM nuostolių funkcija sugeba išskirti industrines teritorijas;
2. modeliai su 3 sluoksniais ir mažiau nesugeba išskirti aukšto gyventojų tankumo (64 gyv. ir daugiau) teritorijų;
3. modeliai kurie turėjo nuo 4 iki 6 sluoksnių, sunkiai atskyrė aukšto gyventojų tankumo teritorijas;
4. modeliai su 5 ir daugiau sluoksnių, atskyrė aukšto gyventojų tankumo teritorijas;
5. modelis su 30 sluoksniais nesugebėjo mokytis, tai tikriausiai įvyko dėl didelio sluoksnių skaičiaus;
6. modelis su 9 sluoksniais davė geriausius rezultatus – 28 Δ Vid.%.



24 pav. CNN modelių, mokyty su skirtingais sluoksnių skaičiais, Kėdainių miestui sudaryti GTŽ

9 lentelė. CNN modelių, mokytų su skirtingais sluoksnių skaičiais, Δ Vid.% verčių lentelė skirtingiems gyventojų intervalams

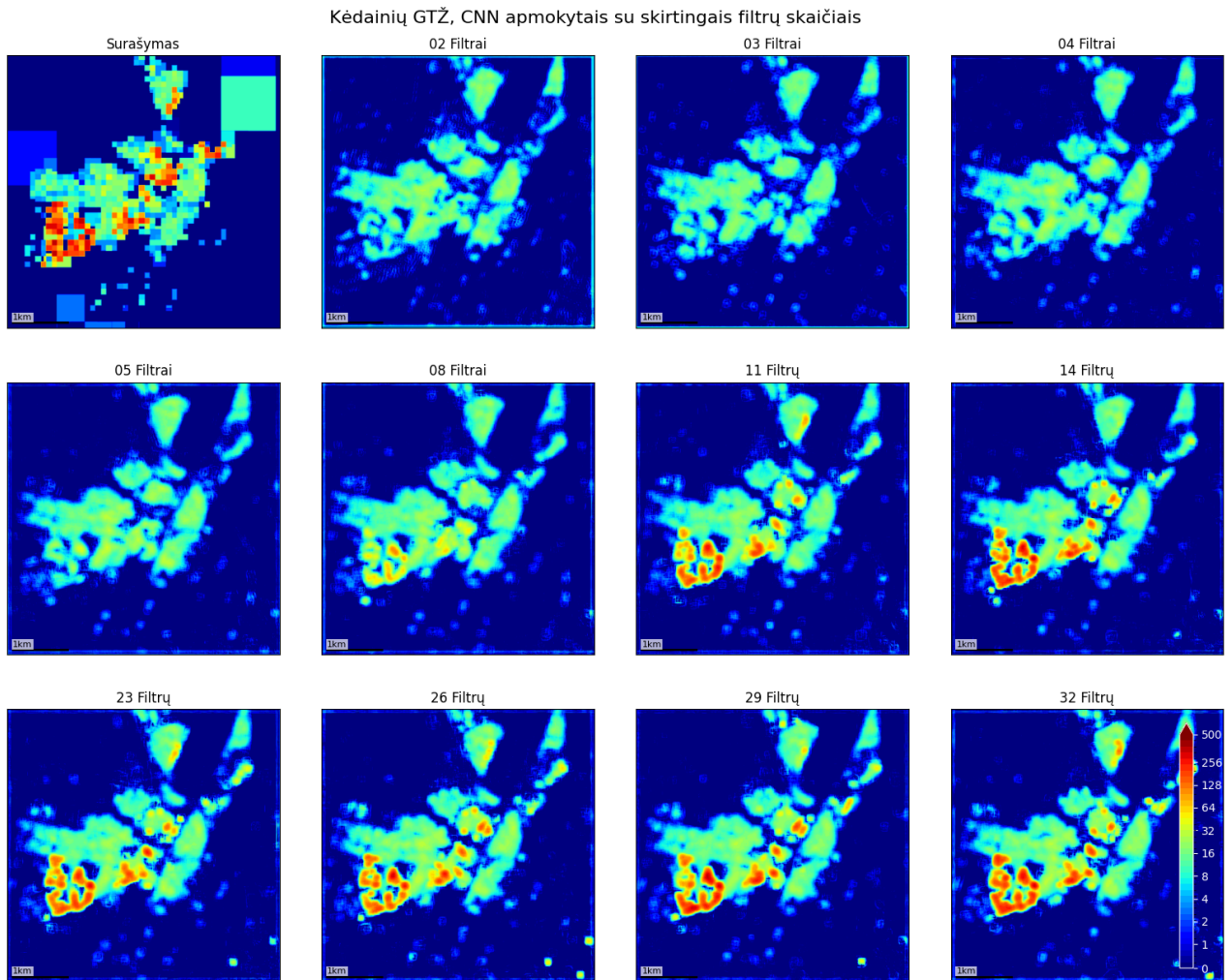
Gyventojai	Nuo Iki	0,0	0,5	1,0	2,0	4	8	16	32	64	128	256	Vid.
		0,5	1,0	2,0	4,0	8	16	32	64	128	256	500	
Surašymas	Vid.	0,1	0,7	1,4	3,0	6	11	22	44	93	182	319	
	SN	0,1	0,2	0,3	0,6	1	2	4	9	18	35	54	
1 Sluoksniis	Vid.	0,5	0,8	1,4	2,8	4	6	10	8	6	6	7	
	SN	1,1	1,4	2,1	3,0	4	4	5	5	4	4	3	
	ΔVid. %	80 %	12 %	0 %	6 %	25 %	41 %	55 %	80 %	92 %	96 %	97 %	53 %
2 Sluoksniai	Vid.	0,5	0,9	1,4	2,9	5	8	13	12	10	13	15	
	SN	1,2	1,6	2,2	3,6	5	6	7	8	8	8	8	
	ΔVid. %	80 %	22 %	0 %	3 %	16 %	29 %	41 %	72 %	88 %	92 %	95 %	49 %
3 Sluoksniai	Vid.	0,4	0,8	1,4	3,2	6	9	14	13	13	18	22	
	SN	1,2	1,5	2,3	3,9	5	6	7	9	10	11	11	
	ΔVid. %	75 %	12 %	0 %	6 %	8 %	22 %	36 %	68 %	85 %	90 %	92 %	45 %
4 Sluoksniai	Vid.	0,4	0,8	1,3	3,0	5	9	14	15	28	52	84	
	SN	2,1	1,9	2,8	4,3	6	7	9	14	34	50	60	
	ΔVid. %	75 %	12 %	7 %	0 %	9 %	21 %	35 %	64 %	69 %	71 %	73 %	40 %
5 Sluoksniai	Vid.	0,5	0,8	1,3	3,0	5	9	14	16	30	58	92	
	SN	2,3	2,0	2,8	4,3	6	7	9	15	35	52	61	
	ΔVid. %	80 %	12 %	7 %	0 %	9 %	22 %	35 %	63 %	66 %	67 %	71 %	39 %
6 Sluoksniai	Vid.	0,4	0,8	1,4	3,1	5	9	14	16	36	69	111	
	SN	2,6	2,0	3,3	4,7	6	8	9	17	41	59	70	
	ΔVid. %	75 %	12 %	0 %	3 %	9 %	21 %	35 %	62 %	61 %	62 %	65 %	37 %
7 Sluoksniai	Vid.	0,5	0,8	1,4	3,2	6	9	15	19	50	96	148	
	SN	3,5	2,5	3,5	5,6	8	10	11	22	53	71	80	
	ΔVid. %	80 %	12 %	0 %	6 %	4 %	17 %	32 %	55 %	45 %	47 %	53 %	32 %
8 Sluoksniai	Vid.	0,5	0,8	1,4	3,2	6	9	15	19	44	82	128	
	SN	3,2	1,9	3,4	4,9	7	8	10	19	47	65	81	
	ΔVid. %	80 %	12 %	0 %	6 %	1 %	15 %	29 %	55 %	51 %	54 %	59 %	33 %
9 Sluoksniai	Vid.	0,5	0,8	1,5	3,5	6	11	17	23	59	114	176	
	SN	4,1	2,4	4,1	6,3	9	11	13	26	61	82	94	
	ΔVid. %	80 %	12 %	6 %	14 %	7 %	4 %	22 %	46 %	36 %	37 %	44 %	28 %
10 Sloksnių	Vid.	0,4	0,9	1,5	3,5	6	9	15	19	45	86	128	
	SN	3,1	2,0	3,7	5,0	7	9	10	19	48	64	74	
	ΔVid. %	75 %	22 %	6 %	14 %	1 %	15 %	32 %	56 %	51 %	52 %	59 %	35 %
20 Sloksnių	Vid.	0,5	0,8	1,4	3,4	6	10	15	21	53	101	154	
	SN	3,7	1,9	3,4	4,9	8	10	11	23	57	75	90	
	ΔVid. %	80 %	12 %	0 %	11 %	3 %	13 %	29 %	51 %	42 %	44 %	51 %	30 %
30 Sloksnių	Vid.	0,6	0,6	0,6	0,5	0	0	0	0	0	0	0	
	SN	0,1	0,1	0,1	0,1	0	0	0	0	0	0	0	
	ΔVid. %	83 %	14 %	57 %	83 %	91 %	95 %	97 %	98 %	99 %	99 %	99 %	83 %

Tolimesniems eksperimentams CNN modelio filtrų skaičiui rasti naudosime SSIM nuostolių funkciją ir 9 sluoksnių modelio architektūrą.

4.4.3. Filtrų skaičiaus pasirinkimas

Filtrų skaičiui atrasti buvo mokyta 13 CNN modelių, su SSIM nuostolių funkcija, 9 sluoksniais ir skirtingais filtrų skaičiais. Šių modelių rezultatai pateikti 10 lentelėje ir 25 paveikslėlyje. Šie rezultatai parodo, kad:

1. nepriklausomai nuo filtrų skaičiaus, modelis su SSIM nuostolių funkcija bei 9 sluoksniais visada sugeba išskirti industrines teritorijas;
2. modeliai, kurie turi mažiau negu 8 filtrus, visiškai nesugeba išskirti aukšto gyventojų tankumo teritorijų (32 gyv. ir daugiau);
3. modelis su 8 filtrais sunkiai išskiria aukšto gyventojų tankumo teritorijas;
4. vis modeliai, kurie turi 11 filtrų ir daugiau, duoda gerus rezultatus visuose gyventojų tankumo intervaluose;
5. didėjant filtrų skaičiui, didėja tikslumas aukšto gyventojų tankumo teritorijose;
6. geriausias modelis turėjo 32 filtrus, jo $\Delta Vid. \%$ vertė buvo 28 %.



25 pav. CNN modelių, mokyty su skirtingais filtrų skaičiais, Kėdainių miestui sudaryti GTŽ

10 lentelė. CNN modelių, mokytų su skirtingais filtrų skaičiais, Δ Vid.% verčių lentelė skirtingiems gyventojų intervalams

Gyventojai	Nuo	0,0	0,5	1,0	2,0	4	8	16	32	64	128	256	Vid.
	Iki	0,5	1,0	2,0	4,0	8	16	32	64	128	256	500	
Surašymas	Vid.	0,1	0,7	1,4	3,0	6	12	22	44	93	182	319	
	SN	0,1	0,2	0,3	0,6	1	2	4	9	18	36	54	
2 Filtrai	Vid.	0,6	1,0	1,7	3,1	5	9	13	12	10	10	11	
	SN	1,4	1,8	2,6	3,6	5	6	7	8	6	6	5	
	ΔVid. %	83 %	30 %	17 %	3 %	11 %	25 %	40 %	73 %	89 %	94 %	96 %	
3 Filtrai	Vid.	0,5	0,8	1,4	2,9	5	8	12	11	9	9	10	
	SN	1,5	1,6	2,4	3,4	4	6	6	7	6	5	5	
	ΔVid. %	80 %	12 %	0 %	3 %	15 %	28 %	44 %	75 %	90 %	95 %	97 %	
4 Filtrai	Vid.	0,4	0,8	1,3	3,0	5	9	14	13	10	11	12	
	SN	1,0	1,4	2,1	3,4	4	6	8	9	7	6	6	
	ΔVid. %	75 %	12 %	7 %	0 %	9 %	23 %	35 %	70 %	88 %	93 %	96 %	
5 Filtrai	Vid.	0,4	0,8	1,4	3,2	6	10	15	14	12	14	16	
	SN	1,3	1,4	2,3	3,8	5	7	8	11	9	9	8	
	ΔVid. %	75 %	12 %	0 %	6 %	4 %	18 %	31 %	67 %	86 %	92 %	95 %	
8 Filtrai	Vid.	0,4	0,8	1,4	3,4	6	10	15	17	20	27	36	
	SN	1,4	1,6	2,3	4,2	6	7	8	12	15	17	19	
	ΔVid. %	75 %	12 %	0 %	11 %	0 %	17 %	33 %	62 %	78 %	85 %	88 %	
11 Filtrų	Vid.	0,5	0,9	1,4	3,3	6	10	15	20	45	85	135	
	SN	3,0	2,1	3,1	5,0	8	9	10	20	46	65	76	
	ΔVid. %	80 %	22 %	0 %	9 %	1 %	15 %	32 %	55 %	52 %	53 %	57 %	
14 Filtrų	Vid.	0,5	0,8	1,5	3,5	6	10	16	22	54	104	165	
	SN	3,9	2,4	4,1	6,5	8	11	12	26	57	77	88	
	ΔVid. %	80 %	12 %	6 %	14 %	3 %	12 %	30 %	50 %	42 %	43 %	48 %	
17 Filtrų	Vid.	0,5	0,9	1,5	3,4	6	10	15	20	49	93	144	
	SN	3,5	2,3	3,8	5,3	7	10	11	22	51	71	79	
	ΔVid. %	80 %	22 %	6 %	11 %	0 %	13 %	31 %	54 %	47 %	49 %	54 %	
20 Filtrų	Vid.	0,5	0,8	1,5	3,3	6	10	15	19	42	76	122	
	SN	3,0	1,7	3,1	4,9	7	9	10	19	45	62	77	
	ΔVid. %	80 %	12 %	6 %	9 %	3 %	17 %	32 %	57 %	55 %	58 %	61 %	
23 Filtrų	Vid.	0,5	0,9	1,6	3,6	6	11	16	23	55	102	156	
	SN	3,8	2,5	4,3	6,2	8	11	12	25	55	72	79	
	ΔVid. %	80 %	22 %	12 %	16 %	4 %	7 %	26 %	47 %	41 %	43 %	51 %	
26 Filtrų	Vid.	0,5	0,9	1,5	3,5	6	11	17	21	50	97	152	
	SN	3,4	2,3	3,9	5,7	8	10	12	23	53	75	86	
	ΔVid. %	80 %	22 %	6 %	14 %	3 %	7 %	24 %	52 %	45 %	46 %	52 %	
29 Filtrų	Vid.	0,6	1,0	1,6	3,6	6	11	17	22	56	106	166	
	SN	3,9	2,9	4,3	6,3	9	11	13	25	58	80	92	
	ΔVid. %	83 %	30 %	12 %	16 %	7 %	8 %	25 %	49 %	40 %	41 %	47 %	
32 Filtrų	Vid.	0,5	0,8	1,5	3,5	6	11	17	24	59	114	176	
	SN	4,1	2,4	4,1	6,3	9	12	13	26	61	82	95	
	ΔVid. %	80 %	12 %	6 %	14 %	7 %	4 %	22 %	46 %	36 %	37 %	44 %	

Geriausias CNN modelis buvo mokytas su SSIM nuostolių funkcija, turėjo 10 sluoksnių ir 32 filtrus kiekviename sluoksnyje.

4.5. *U-Net* eksperimentai

Šiame skyrelyje pateikti eksperimentai, skirti surasti nuostolių funkciją, su kuria mokytas *U-Net* modelis turėtų mažiausią $\Delta\text{Vid.}\%$ vertę visuose gyventojų intervaluose bei nustatyti *U-Net* modelio gylio ir filtrų skaičiaus poveikį $\Delta\text{Vid.}\%$ vertėms.

U-Net modelio architektūros šablonas, pagal kurį sukurti visi *U-Net* modeliai aptarti šiame skyrelyje, pateiktas žemiau „Python“ kodo pavidalu. Kiekvienas modelis turi 500 x 500 x 16 dydžio įvestį, kraštų papildymą iki 512 x 512 dydžio, kad būtų užtikrintas modelio veikimas suspaudimo sluoksniuose ($512 = 2^9$), kodavimo ir dekodavimo blokus, kurių skaičius priklauso nuo modelio gylio, papildomą sluoksnį su vienu filtru, 3 x 3 dydžio konvoliucijos kaukę bei „linear“ aktyvacijos funkcija. Šis papildomas sluoksnis yra skirtas modelio išvesčiai, nuimti kraštus iki 500 x 500 dydžio. Kiekvienas kodavimo blokas turi po tris konvoliucijos sluoksnius su nustatytu skaičiumi filtrų, 3 x 3 dydžio konvoliucijos kaukę bei „relu“ aktyvacijos funkcija, vieną suspaudimo sluoksnį, kuris sumažina įvestį per pusę, išrinkdamas iš 4 gretimų pikselių didžiausią vertę. Kiekvienas dekodavimo blokas turi išplėtimo sluoksnį, kuris padidina įvestį dvigubai, apjungimo sluoksnį, kuris prie išplėtimo sluoksnio prideda atitinkamą suspaudimo sluoksnį, tris konvoliucijos sluoksnius. Visi modeliai buvo mokyti su „adam“ optimizatoriumi.

```
from tensorflow.keras.layers import Input, Conv2D, MaxPooling2D, UpSampling2D, Cropping2D
from tensorflow.keras.layers import concatenate, Conv2DTranspose, ZeroPadding2D
from tensorflow.keras.models import Model

def Conv2Dx3(layer, n_filters):
    layer = Conv2D(n_filters, (3, 3), activation='relu', padding='same')(layer)
    layer = Conv2D(n_filters, (3, 3), activation='relu', padding='same')(layer)
    layer = Conv2D(n_filters, (3, 3), activation='relu', padding='same')(layer)
    return layer

def create_encoder_block(layer, n_filters):
    layer = Conv2Dx3(layer, n_filters)
    pool = MaxPooling2D(pool_size=(2, 2))(layer)
    return pool, layer

def create_decoder_block(layer, skip_layer, n_filters):
    layer = Conv2DTranspose(n_filters, (2, 2), strides=(2, 2), padding='same')(layer)
    layer = concatenate([layer, skip_layer], axis=3)
    layer = Conv2Dx3(layer, n_filters)
    return layer

def create_unet_model(depth, n_filters, loss):
    inputs = Input(shape=(500, 500, 16))
    # Padding for having shape of 512 x 512 (512 = 2^9)
    layer = ZeroPadding2D(padding=((6, 6), (6, 6)))(inputs)
    skip_layers = []
    for i in range(depth - 1): # Encoder
        layer, skip = create_encoder_block(layer, n_filters)
        skip_layers.append(skip)
    layer = Conv2Dx3(layer, n_filters) # Middle
    for i in range(depth - 1): # Decoder
        layer = create_decoder_block(layer, skip_layers.pop(), n_filters)
```

```
layer = Conv2D(1, (1, 1), activation='linear', padding='same')(layer)
# Remove padding
layer = Cropping2D(cropping=((6, 6), (6, 6)))(layer)
model = Model(inputs=inputs, outputs=layer)
model.compile(optimizer="adam", loss=loss)
return model
```

4.5.1. Nuostolių funkcijos pasirinkimas

Taip pat kaip ir CNN modeliai, U-Net modeliai buvo mokyti su skirtingomis nuostolių funkcijomis bei jų modifikacijomis. Iš viso buvo mokomi 11 *U-Net* modelių su MSE, SMAPE, SSIM nuostolių funkcijomis bei jų modifikacijomis. Mokomi modeliai turėjo po 3 kodavimo ir dekodavimo blokus bei kiekvienas modelio sluoksnis turėjo po 8 filtrus.

Eksperimentų rezultatai pateikti 26 paveikslėlyje ir 11 lentelėje. Gauti rezultatai buvo panašūs į CNN modelių rezultatus. Modelis mokytas su SMAPE nuostolių funkcija davė geriausius rezultatus (29 Δ Vid.%). SSIM nuostolių funkcija, kaip ir su CNN architektūra, davė geresnius rezultatus (4 - 13 Δ Vid.%) mažose gyventojų tankumo intervaluose (iki 16 gyv.). MSE nuostolių funkcija pervertino tankumą (50 - 65 Δ Vid.%) mažose gyventojų tankumo intervaluose (iki 8 gyv.). SSIM ir SMAPE atpažino industrines teritorijas, MSE neatpažino.

B_SMAPE ir W_SMAPE, taip pat kaip ir su CNN architektūra, stipriai pervertino gyventojų tankumą mažose tankumo intervaluose (B_SMAPE intervale iki 16 gyventojų, W_SMAPE intervale iki 128 gyventojų) bei nesugebėjo atpažinti industrinių teritorijų.

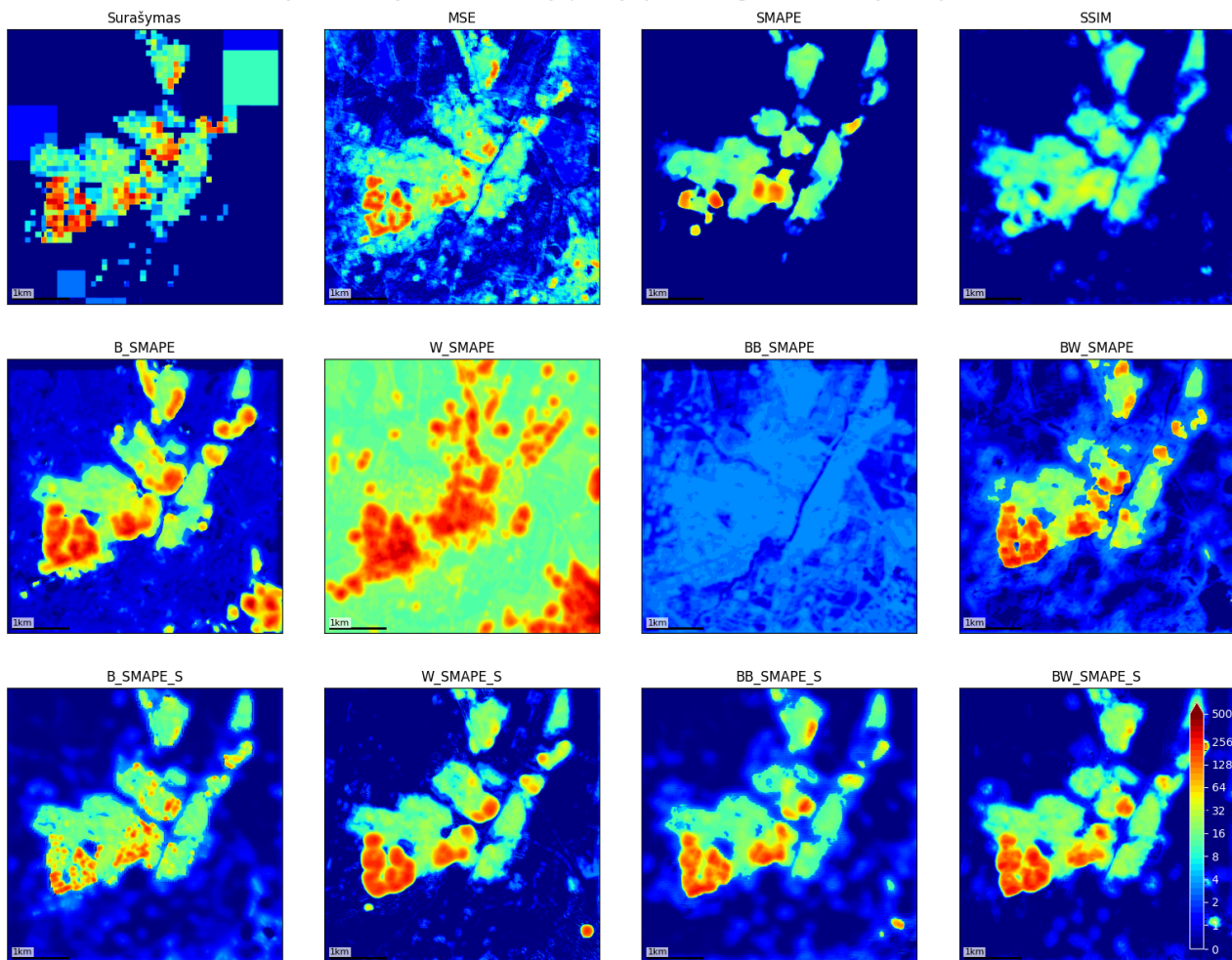
BB_SMAPE visiškai nesugebėjo atpažinti gyvenamų teritorijų, o BW_SMAPE pagerino W_SMAPE rezultatus.

Kaip ir su CNN, visų nuostolių funkcijų ir SSIM funkcijos kombinacijos pagerino rezultatus bei sugebėjo atpažinti industrines teritorijas. Geriausius rezultatus davė BW_SMAPE_S nuostolių funkcija (31 Δ Vid.%).

27 paveikslėlyje pateiktos visų nuostolių funkcijų nuostolių kreivės. Iš šių kreivų matome, kad su visomis nuostolių funkcijomis, išskyrus BB_SMAPE, B_SMAPE_S ir BB_SMAPE_S, modelis mokėsi stabiliai. SMAPE kaip ir su CNN architektūra, po 200 epochų turėjo staigų nukritimą.

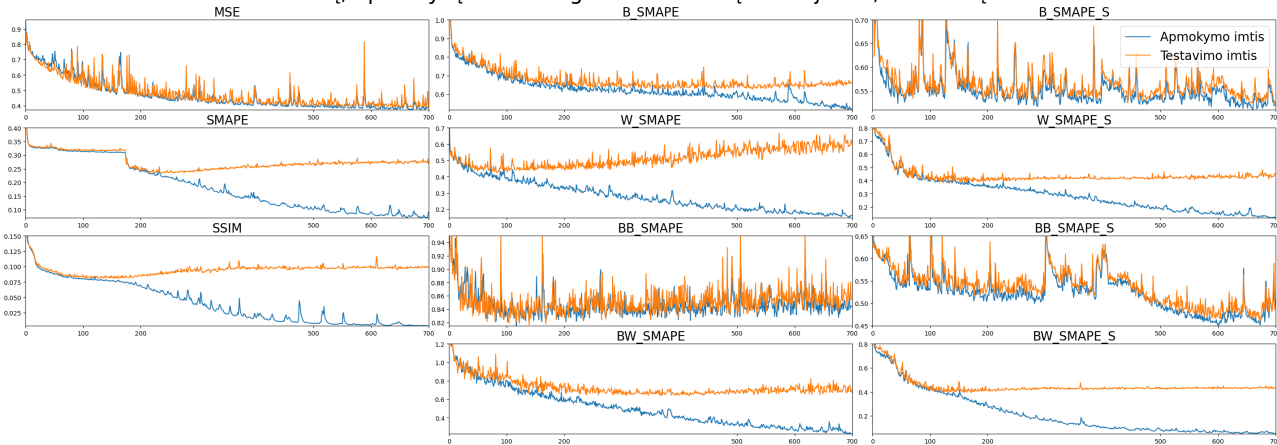
Tolimesniems *U-Net* eksperimentams buvo pasirinkta BW_SMAPE_S nuostolių funkcija, dėl jos duodamų gerų rezultatų bei stabilaus modelio mokymo. Taip pat verta paminėti, kad BW_SMAPE_S nuostolių funkcijos rezultatai testavimo imčiai neblogėjo, kai modelis pradėjo persimokyti.

Kėdainių GTŽ, sudaryti U-Net modelių apmokytų su skirtingomis nuostolių funkcijomis



26 pav. U-Net modelių, mokyty su skirtingomis nuostolių funkcijomis, Kėdainių miestui sudaryti GTŽ

U-Net modelių, apmokytų su skirtingomis nuostolių funkcijomis, nuostolių kreivės



27 pav. U-Net modelių, mokyty su skirtingomis nuostolių funkcijomis, nuostolių kreivės

11 lentelė. *U-Net* modelių, mokytų su skirtingomis nuostolių funkcijomis, Δ Vid.% verčių lentelė skirtingiems gyventojų intervalams

Gyventojai	Nuo Iki	0,0	0,5	1,0	2,0	4	8	16	32	64	128	256	Vid.
		0,5	1,0	2,0	4,0	8	16	32	64	128	256	500	
Surašymas	Vid.	0,1	0,7	1,4	3,0	6	12	22	44	93	182	319	
	SN	0,1	0,2	0,3	0,6	1	2	4	9	18	36	54	
	ΔVid. %	95 %	65 %	58 %	50 %	36 %	17 %	9 %	28 %	19 %	31 %	45 %	
MSE	Vid.	2,1	2,0	3,4	6,0	10	14	20	32	75	124	173	
	SN	7,1	5,1	8,2	10,7	14	17	18	33	62	72	74	
	ΔVid. %	95 %	65 %	58 %	50 %	36 %	17 %	9 %	28 %	19 %	31 %	45 %	
SMAPE	Vid.	0,4	0,6	1,2	3,7	7	12	19	30	63	101	134	
	SN	5,9	2,9	5,9	10,3	15	18	21	42	68	80	77	
	ΔVid. %	75 %	14 %	14 %	18 %	15 %	6 %	14 %	33 %	32 %	44 %	58 %	
SSIM	Vid.	0,4	0,8	1,5	3,6	6	10	16	16	14	16	18	
	SN	1,4	1,7	2,7	4,4	6	7	8	10	10	9	8	
	ΔVid. %	75 %	12 %	6 %	16 %	4 %	13 %	30 %	64 %	84 %	90 %	94 %	
B_SMAPE	Vid.	3,6	2,9	4,6	8,8	14	22	31	55	126	186	240	
	SN	15,4	11,5	14,4	19,0	24	28	28	50	74	79	74	
	ΔVid. %	97 %	75 %	69 %	65 %	58 %	47 %	28 %	20 %	25 %	1 %	24 %	
W_SMAPE	Vid.	28,3	23,0	29,9	39,3	47	53	67	114	186	239	272	
	SN	42,9	32,9	38,6	47,0	51	53	58	76	77	73	61	
	ΔVid. %	99 %	96 %	95 %	92 %	87 %	77 %	66 %	61 %	49 %	23 %	14 %	
BB_SMAPE	Vid.	1,4	1,7	2,1	2,6	3	3	3	3	3	3	3	
	SN	0,8	0,9	0,9	0,7	0	0	0	0	0	0	0	
	ΔVid. %	92 %	58 %	33 %	13 %	51 %	74 %	86 %	93 %	96 %	98 %	99 %	
BW_SMAPE	Vid.	2,0	1,8	3,6	6,7	12	19	28	49	109	152	188	
	SN	10,9	7,0	12,5	16,7	21	26	30	49	69	69	61	
	ΔVid. %	95 %	61 %	61 %	55 %	48 %	37 %	19 %	8 %	14 %	16 %	41 %	
B_SMAPE_S	Vid.	1,7	1,6	3,3	6,5	10	16	22	34	66	98	126	
	SN	10,5	7,9	15,0	16,5	21	22	26	41	62	69	71	
	ΔVid. %	94 %	56 %	57 %	53 %	42 %	25 %	0 %	24 %	29 %	46 %	60 %	
W_SMAPE_S	Vid.	1,2	1,1	2,5	5,4	9	15	23	42	102	156	199	
	SN	9,1	6,5	10,5	13,8	17	21	24	47	68	66	49	
	ΔVid. %	91 %	36 %	44 %	44 %	36 %	23 %	2 %	4 %	8 %	14 %	37 %	
BB_SMAPE_S	Vid.	1,3	1,4	2,5	4,9	8	13	19	33	84	133	170	
	SN	7,5	5,1	8,4	11,7	15	19	22	42	68	72	63	
	ΔVid. %	92 %	50 %	44 %	38 %	28 %	10 %	13 %	25 %	9 %	27 %	46 %	
BW_SMAPE_S	Vid.	1,2	1,2	2,4	5,3	10	15	23	41	103	161	203	
	SN	8,9	5,5	8,5	13,5	19	22	26	48	74	76	64	
	ΔVid. %	91 %	41 %	41 %	43 %	37 %	22 %	3 %	7 %	9 %	11 %	36 %	

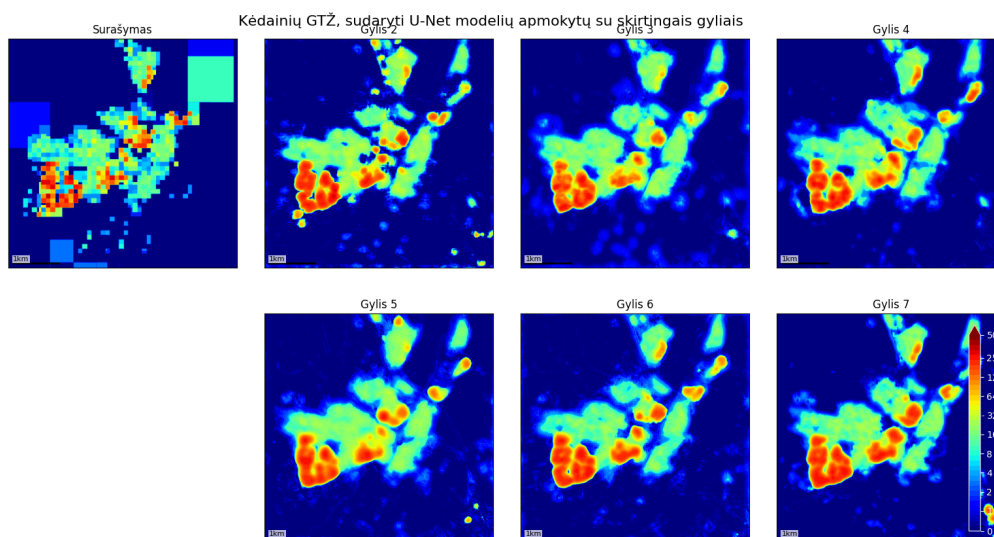
4.5.2. *U-Net* gylio pasirinkimas

U-Net modelio gyliui atrasti buvo mokytą 6 *U-Net* modeliai, su BW_SMAPE_S nuostolių funkcija, skirtingais gyliais ir 8 filtrais kiekviename sluoksnyje. *U-Net* modelio sluoksnių skaičius gali būti išreikštas su 21 formule.

$$\text{sluoksniai}(\text{gylis}) = \text{gylis} \cdot 3 + 3 \quad (21)$$

Modelių, su skirtingais gyliais, rezultatai pateikti 12 lentelėje ir 28 paveikslėlyje. Šie rezultatai parodo, kad:

1. *U-Net* modelis, nepriklausomai nuo gylio, davė gerus rezultatus. Tokius rezultatus galima paaikškinti apskaičiuojant, kiek sluoksnių turi modelis su mažiausiu gyliu. Modelis su mažiausiu gyliu 2 turi 9 sluoksnius, o iš CNN rezultatų jau žinome, kad visi modeliai, kurie turėjo 7 ir daugiau sluoksnių, davė gerus rezultatus;
2. iš 28 paveikslėlio matome kad didėjant modelio gyliui, modelio triukšmas (atsitiktinai prognozuojamos žemo gyventojų tankumo teritorijos) mažėja;
3. geriausio modelio gylis buvo 6, su 28 Δ Vid.‰ verte.



28 pav. *U-Net* modelių, mokyty su skirtingais gyliais, Kėdainių miestui sudaryti GTŽ

12 lentelė. *U-Net* modelių, mokyty su skirtingais gyliais, Δ Vid.‰ verčių lentelė skirtingiems gyventojų intervalams

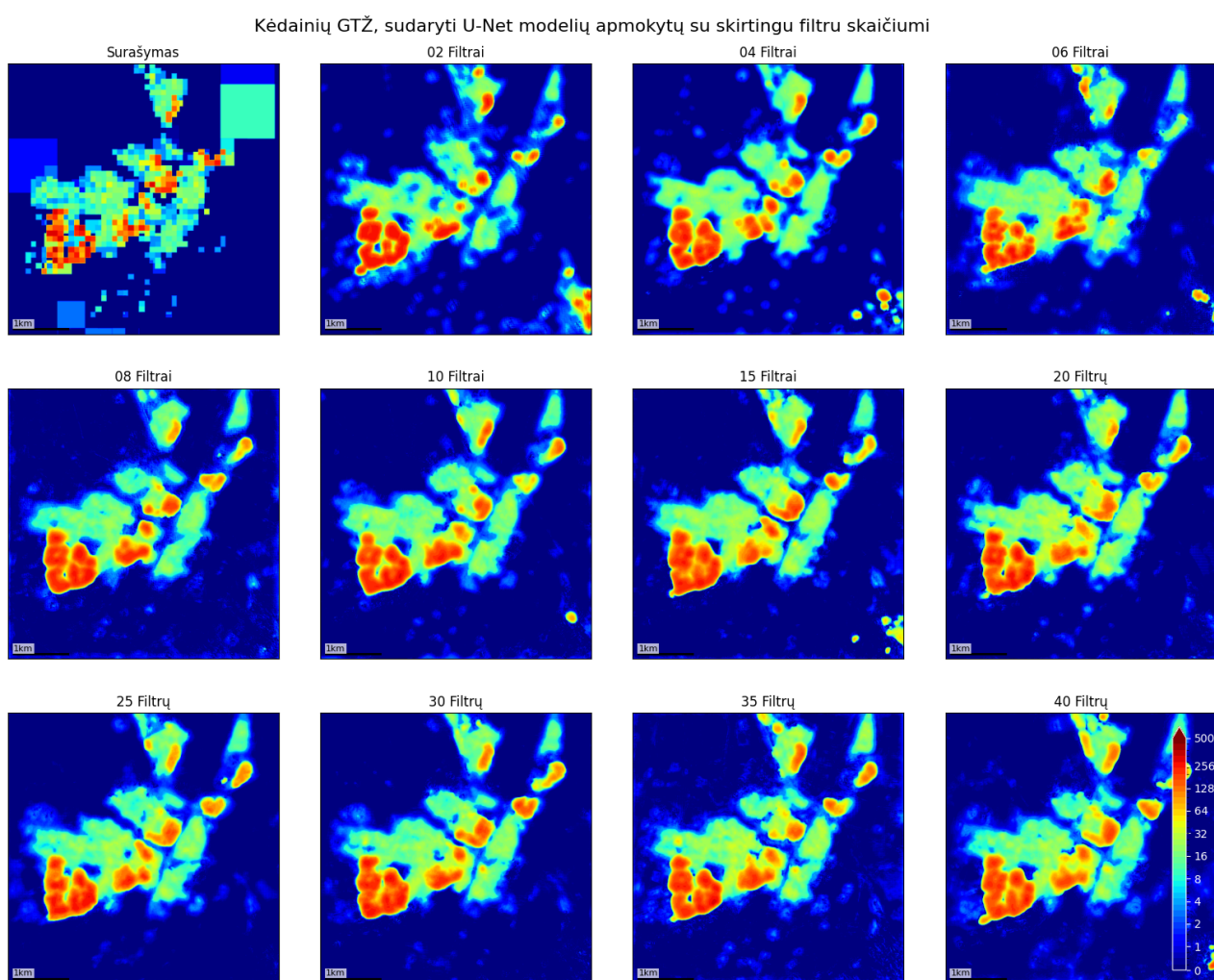
Gyventojai	Nuo	0,0	0,5	1,0	2,0	4	8	16	32	64	128	256	Vid.
	Iki	0,5	1,0	2,0	4,0	8	16	32	64	128	256	500	
Surašymas	Vid.	0,1	0,7	1,4	3,0	6	12	22	44	93	182	319	
	SN	0,1	0,2	0,3	0,6	1	2	4	9	18	36	54	
Gylis 2	Vid.	1,0	1,0	2,4	5,1	10	16	25	46	110	167	213	
	SN	9,0	5,8	10,9	13,7	20	24	27	52	74	72	56	
	Δ Vid. ‰	90 %	30 %	41 %	41 %	39 %	29 %	9 %	3 %	14 %	8 %	33 %	30 %
Gylis 3	Vid.	1,2	1,2	2,4	5,3	10	15	23	41	103	161	203	
	SN	8,9	5,5	8,5	13,5	19	22	26	48	74	76	64	
	Δ Vid. ‰	91 %	41 %	41 %	43 %	37 %	22 %	3 %	7 %	9 %	11 %	36 %	31 %
Gylis 4	Vid.	1,2	1,4	3,0	6,3	10	16	23	43	105	165	211	
	SN	9,6	6,4	11,9	15,9	19	23	26	49	72	77	68	
	Δ Vid. ‰	91 %	50 %	53 %	52 %	40 %	25 %	4 %	2 %	11 %	9 %	33 %	33 %
Gylis 5	Vid.	1,4	1,3	2,8	6,0	10	17	24	42	104	158	203	
	SN	10,0	5,5	11,0	15,4	19	24	26	49	73	71	58	
	Δ Vid. ‰	92 %	46 %	50 %	50 %	41 %	29 %	5 %	6 %	10 %	13 %	36 %	34 %
Gylis 6	Vid.	1,1	1,0	2,2	5,2	9	14	21	42	111	163	205	
	SN	8,7	4,8	9,7	14,6	19	22	25	50	70	68	50	
	Δ Vid. ‰	90 %	30 %	36 %	42 %	34 %	15 %	5 %	5 %	15 %	10 %	35 %	28 %
Gylis 7	Vid.	1,3	1,2	3,0	6,1	10	16	24	47	124	181	221	
	SN	11,4	7,8	14,6	18,1	21	27	31	56	77	68	48	
	Δ Vid. ‰	92 %	41 %	53 %	50 %	39 %	28 %	8 %	5 %	24 %	0 %	30 %	33 %

Tolimesniems eksperimentams *U-Net* modelio filtrų skaičiui rasti naudosime *BW_SMAPE_S* nuostolių funkciją ir architektūrą, kurios gylis 6.

4.5.3. Filtrų skaičiaus pasirinkimas

Filtrų skaičiui atrasti buvo mokyta 11 *U-Net* modelių, kurių gylis 6, su *BW_SMAPE_S* nuostolių funkcija ir skirtingais filtrų skaičiais. Šių modelių rezultatai pateikti 13 lentelėje ir 29 paveikslėlyje. Šie rezultatai parodo, kad:

1. nepriklausomai nuo filtrų skaičiaus modelis sugeba išskirti aukšto tankumo gyventojų teritorijas;
2. modeliai su mažiau negu 8 filtrais nesugeba išskirti industrinių teritorijų;
3. neskaitant industrinių teritorijų, visi modeliai davė panašius rezultatus (30 Δ Vid.%);
4. geriausias modelis turėjo 25 filtras su 29 Δ Vid.% verte.



29 pav. *U-Net* modelių, mokyty su skirtingu filtru skaičiumi, Kėdainių miestui sudaryti GTŽ

13 lentelė. *U-Net* modelių, mokytų su skirtingu filtru skaičiumi, Δ Vid.% verčių lentelė skirtingiems gyventojų intervalams

Gyventojai	Nuo Iki	0,0	0,5	1,0	2,0	4	8	16	32	64	128	256	Vid.
		0,5	1,0	2,0	4,0	8	16	32	64	128	256	500	
Surašymas	Vid.	0,1	0,7	1,4	3,0	6	12	22	44	93	182	319	
	SN	0,1	0,2	0,3	0,6	1	2	4	9	18	36	54	
	Δ Vid. %	92 %	50 %	33 %	31 %	31 %	20 %	0 %	3 %	20 %	2 %	33 %	28 %
2 Filtrai	Vid.	1,4	1,4	2,1	4,4	9	15	22	43	117	178	213	
	SN	10,9	7,8	9,8	15,0	23	29	33	58	89	81	65	
	Δ Vid. %	92 %	50 %	33 %	31 %	31 %	20 %	0 %	3 %	20 %	2 %	33 %	28 %
4 Filtrai	Vid.	1,0	1,1	2,6	5,6	10	16	23	43	111	164	206	
	SN	9,0	6,0	11,7	14,3	18	22	26	48	72	71	55	
	Δ Vid. %	90 %	36 %	46 %	46 %	40 %	26 %	4 %	2 %	15 %	10 %	35 %	31 %
6 Filtrai	Vid.	1,3	1,5	2,8	6,2	11	17	24	46	117	178	226	
	SN	10,0	8,6	9,7	15,5	21	27	28	55	82	80	58	
	Δ Vid. %	92 %	53 %	50 %	51 %	44 %	31 %	8 %	4 %	20 %	2 %	29 %	34 %
8 Filtrai	Vid.	1,1	1,0	2,2	5,2	9	14	21	42	111	163	205	
	SN	8,7	4,8	9,7	14,6	19	22	25	50	70	68	50	
	Δ Vid. %	90 %	30 %	36 %	42 %	34 %	15 %	5 %	5 %	15 %	10 %	35 %	28 %
10 Filtrų	Vid.	1,1	1,1	2,5	5,5	10	16	24	46	118	173	217	
	SN	9,6	6,8	11,4	14,9	20	24	28	50	71	66	50	
	Δ Vid. %	90 %	36 %	44 %	45 %	40 %	27 %	7 %	3 %	20 %	5 %	31 %	31 %
15 Filtrų	Vid.	1,1	1,1	2,4	5,8	10	17	25	44	108	161	206	
	SN	9,4	5,8	9,7	14,1	17	22	25	45	67	67	56	
	Δ Vid. %	90 %	36 %	41 %	48 %	41 %	30 %	11 %	1 %	13 %	11 %	35 %	32 %
20 Filtrų	Vid.	1,2	1,3	2,9	6,4	12	19	28	46	108	163	205	
	SN	9,0	5,5	10,1	13,9	19	22	26	45	67	65	53	
	Δ Vid. %	91 %	46 %	51 %	53 %	47 %	37 %	20 %	4 %	13 %	10 %	35 %	37 %
25 Filtrų	Vid.	1,0	1,2	2,2	5,4	9	14	22	41	102	160	210	
	SN	8,4	4,9	7,9	12,8	16	20	22	44	68	70	56	
	Δ Vid. %	90 %	41 %	36 %	44 %	34 %	18 %	0 %	6 %	8 %	12 %	34 %	29 %
30 Filtrų	Vid.	1,1	1,2	2,4	5,4	10	16	24	44	112	165	207	
	SN	9,4	5,7	9,7	14,1	19	23	26	50	69	66	54	
	Δ Vid. %	90 %	41 %	41 %	44 %	36 %	25 %	6 %	1 %	16 %	9 %	35 %	31 %
35 Filtrų	Vid.	1,2	1,3	2,6	6,1	11	18	28	48	111	166	207	
	SN	8,5	5,5	9,3	14,0	19	23	26	47	68	64	54	
	Δ Vid. %	91 %	46 %	46 %	50 %	44 %	35 %	18 %	7 %	16 %	8 %	35 %	36 %
40 Filtrų	Vid.	1,5	1,4	3,3	6,7	11	17	26	49	115	166	207	
	SN	10,5	6,2	12,2	16,5	21	24	27	50	67	61	50	
	Δ Vid. %	93 %	50 %	57 %	55 %	46 %	31 %	13 %	10 %	18 %	8 %	35 %	37 %

Geriausias *U-Net* modelis buvo mokytas su BW_SMAPE_S nuostolių funkcija, turėjo 25 filtrus kiekviename sluoksnyje ir šio modelio gylys buvo 6.

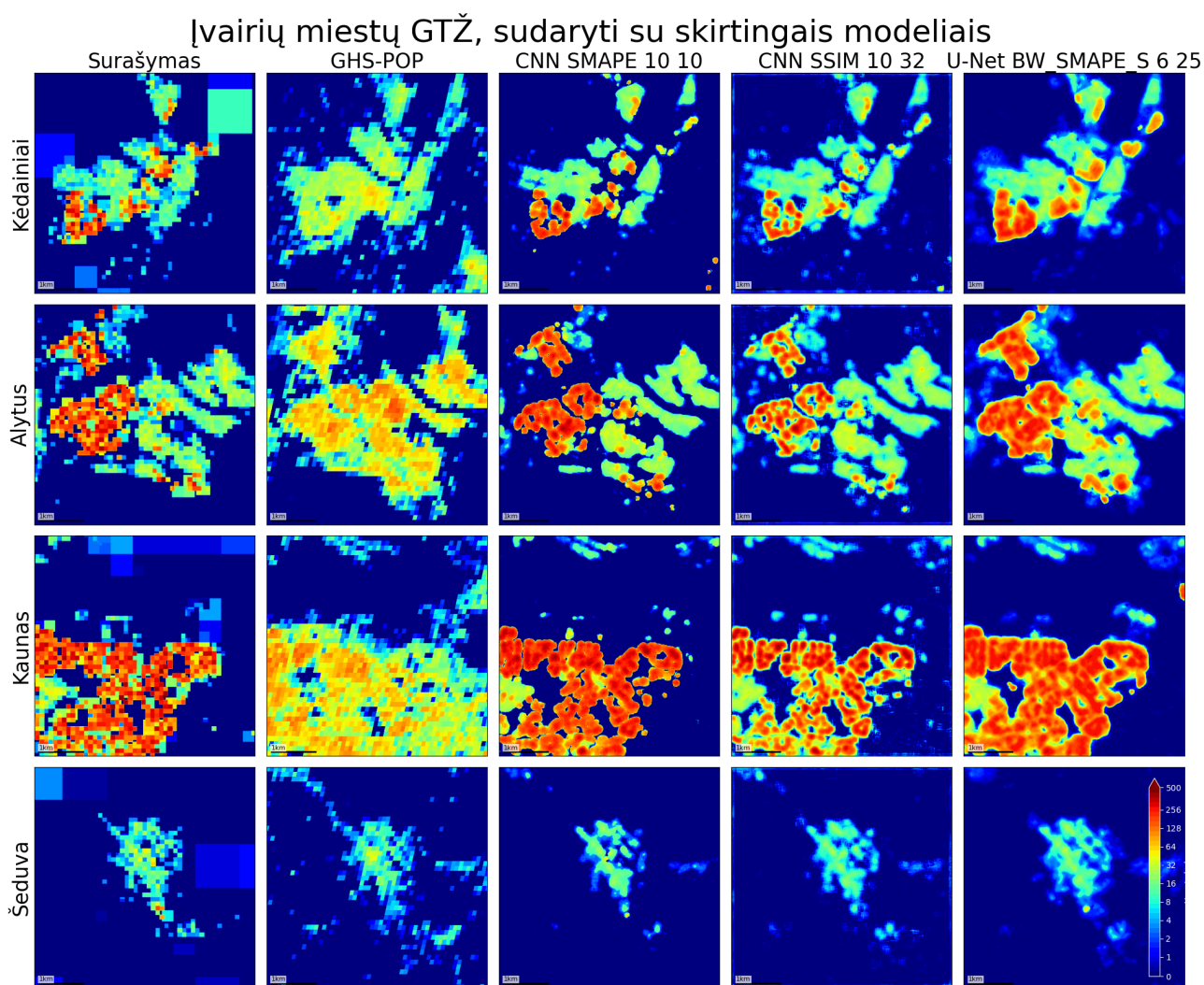
4.6. Eksperimentų rezultatai

30 paveikslėlyje ir 14 lentelėje pateikti geriausių modelių rezultatai. Modeliai pavadinti pagal nuostolių funkciją, modelio gylį arba sluoksnių skaičių, filtrų skaičių. Iš 30 paveikslėlio, 14 lentelės bei prieš tai buvusių skyrelių galime pasakyti, kad:

1. geriausias modelis buvo „CNN SMAPE 10 10“, šio modelio 22 Δ Vid.% vertė buvo mažiausia;
2. dėl stabilaus modelio mokymo, geriausia nuostolių funkcija modeliu mokyti buvo SSIM ir jos

kombinacijos su modifikuota SMAPE;

- geram didelio gyventojų tankumo teritorijų atpažinimui, modelis turi turėti bent 7 sluoksnius;
- geram didelio gyventojų tankumo teritorijų bei industrinių teritorijų atpažinimui, modelis turi turėti bent 10 filtrų;
- 30 paveikslėlyje matome, kad „CNN SMAPE 10 10“ modelis labai aiškiai išskyrė gyvenamas teritorijas, aukšto tankumo teritorijas bei industrines teritorijas, o „CNN SSIM 10 32“ ir „U-Net BW SMAPE_S 6 25“ modeliai turėjo labiau gradientinį perėjimą tarp skirtingų teritorijų;
- lyginant su *GHS-POP* modeliu, visi mūsų modeliai duoda geresnius rezultatus. 30 paveikslėlyje matome kad *GHS-POP* neatiskira industrinių teritorijų, o tiesiog paskirsto gyventojus tolygiai apstatytoms teritorijoms.

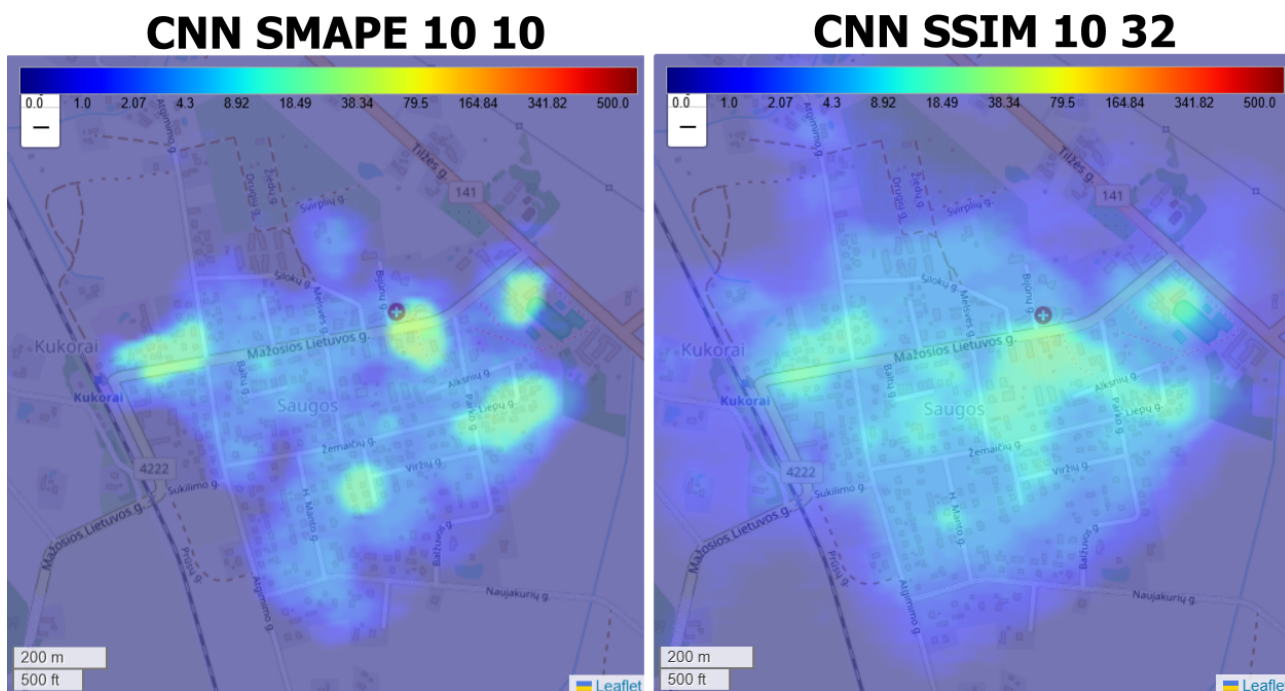


30 pav. Įvairių miestų GTŽ, sudaryti su skirtingais modeliais

14 lentelė. Įvairių miestų GTŽ, sudaryti su skirtingais modeliais, Δ Vid.% verčių lentelė skirtingiems gyventojų intervalams

Gyventojai	Nuo Iki	0,0	0,5	1,0	2,0	4	8	16	32	64	128	256	Vid.
Surašymas	Vid. SN	0,1	0,7	1,4	3,0	6	12	22	44	93	182	319	
GHS-POP	Vid. SN	1,2	1,4	2,3	5,3	9	16	28	32	38	49	61	
	ΔVid. %	91 %	50 %	39 %	43 %	36 %	25 %	19 %	28 %	59 %	73 %	80 %	49 %
CNN SMAPE 10 10	Vid. SN	0,5	0,5	1,3	3,8	8	14	21	36	96	156	212	
	ΔVid. %	80 %	28 %	7 %	21 %	23 %	17 %	5 %	18 %	2 %	14 %	33 %	22 %
CNN SSIM 10 32	Vid. SN	0,5	0,8	1,5	3,5	6	11	17	24	59	114	176	
	ΔVid. %	80 %	12 %	6 %	14 %	7 %	4 %	22 %	46 %	36 %	37 %	44 %	28 %
U-Net BW_SMAPE_S 6 25	Vid. SN	1,0	1,2	2,2	5,4	9	14	22	41	102	160	210	
	ΔVid. %	90 %	41 %	36 %	44 %	34 %	18 %	0 %	6 %	8 %	12 %	34 %	29 %

31 paveikslėlyje matome „CNN SMAPE 10 10“ modelio trūkumą, šis modelis nesugeba atpažinti labai mažo tankumo teritorijų bei vienkiemių, „CNN SSIM 10 32“ šias teritorijas atpažįsta.



31 pav. Saugų kaimo GTŽ, sudaryti su „CNN SSIM 10 32“ ir „CNN SMAPE 10 10“ modeliais

Išvados

1. Buvo išanalizuoti 13 GTŽ sudarymo modelių. Pastebėta kad didžioji jų dalis nesudaro GTŽ tiesiogiai, o tiesiog perskirsto gyventojų surašymo duomenis ant žemėlapių, kuris nusako žmonių tikėtinumą (pvz.: teritorijų apstatymo žemėlapis kaip *GHS-BUILT*). Daug GTŽ sudarymo modelių naudoja išvestinius duomenis, kaip atstumai iki svarbių objektų, maža modelių dalis naudoja spektrinių palydovų vaizdus tiesiogiai. GTŽ sudarymo modelių metodologijos stipriai skiriasi viena nuo kitos, nuo paprasto perskirstymo iki RF ar CNN modelių.
2. Buvo išanalizuotos CNN ir *U-Net* architektūros. Šios architektūros yra tinkamos pikselių lygmens regresijos uždaviniams. Tokiems uždaviniams spręsti dažniausiai naudojamos MAE, MSE, SMAPE ir SSIM nuostolių funkcijos.
3. Buvo gauti Lietuvos gyventojų surašymo duomenys, su skirtingomis raiškomis, iš *Valstybės duomenų agentūros*. Jie buvo sujungti į vieną bendrą rastrinį duomenų rinkinį su 100 m raiška. Buvo gauti *Sentinel 1*, *Sentinel 2*, *VIIRS* palydovų vaizdai, su įvairiomis raiškomis. Visi duomenys buvo parsųsti iš GEE kaip 10 m raiškos vaizdai ir standartizuoti. Duomenų rinkinį sudarė 108 Lietuvos miestų, 5 x 5 km plotų, vaizdai.
4. Buvo sukurtas GTŽ sudarymo modelis „CNN SSIM 10 32“, kuris sugebėjo sudaryti tikslų 10 m raiškos GTŽ. Šis GTŽ sugeba atskirti industrines teritorijas, žemo bei aukšto gyventojų tankumo teritorijas ir negyvenamas teritorijas.
5. „CNN SSIM 10 32“ modelis buvo mokytas su SSIM nuostolių funkcija, šio modelio architektūra buvo CNN modelis su 10 sluoksniais, kur kiekvienas sluoksnis turi 32 filtras.
6. Buvo sukurta $\Delta\text{Vid.}\%$ metrika, skirta GTŽ tikslumui įvertinti skirtinguose gyventojų tankumo intervaluose. Ši metrika palygina kiek procentų skiriasi surašymo GTŽ vertės nuo modelio GTŽ verčių skirtinguose gyventojų tankumo intervaluose, sudarytuose pagal surašymo GTŽ vertes. Modelio „CNN SSIM 10 32“ $\Delta\text{Vid.}\%$ verčių vidurkis visuose intervaluose buvo 28 %. Modelio *GHS-POP* $\Delta\text{Vid.}\%$ verčių vidurkis visuose intervaluose buvo 49 %. Iš vizualinės analizės buvo pastebėta, kad *GHS-POP* modelis paskirsto gyventojus apstatymo teritorijomis, į kurias įtraukia ir industrines teritorijas.
7. Buvo sukurta paprasta programa, skirta peržiūrėti modelio sukurtą GTŽ.

Literatūros sąrašas

1. Nando Metzger, John E Vargas-Muñoz, Rodrigo C Daudt, Benjamin Kellenberger, Thao Ton-That Whelan, Ferda Ofli, Muhammad Imran, Konrad Schindler, and Devis Tuia. Fine-grained population mapping from coarse census counts and open geodata. *Scientific Reports*, 12 (1):20085, 2022. doi: 10.1038/s41598-022-24495-w. URL <https://doi.org/10.1038/s41598-022-24495-w>. Published online 2022 Nov 22.
2. Tracy A. Kugler, Kathryn Grace, David J. Wrathall, Alex de Sherbinin, David Van Riper, Christoph Aubrecht, Douglas Comer, Susana B. Adamo, Guido Cervone, Ryan Engstrom, Carolynne Hultquist, Andrea E. Gaughan, Catherine Linard, Emilio Moran, Forrest Stevens, Andrew J. Tatem, Beth Tellman, and Jamon Van Den Hoek. People and pixels 20 years later: the current data landscape and research trends blending population and environmental data. *Population and Environment*, 41(2):209–234, December 2019. ISSN 0199-0039. doi: 10.1007/s11111-019-00326-5. Epub 2019 Nov 22.
3. Tobias G. Tiecke, Xianming Liu, Amy Zhang, Andreas Gros, Nan Li, Gregory Yetman, Talip Kilic, Siobhan Murray, Brian Blankespoor, Espen B. Prydz, and Hai-Anh H. Dang. Mapping the world population one building at a time, 2017. URL <https://arxiv.org/abs/1712.05839>.
4. Valentin Seidler, Edson C. Utazi, Amelia B. Finaret, Sebastian Luckeneder, Gregor Zens, Maksym Bodarenko, Abigail W. Smith, Sarah E. K. Bradley, Andrew J. Tatem, and Patrick Webb. Subnational variations in the quality of household survey data in sub-saharan africa. *Nature Communications*, 16(1):3771, 2025. ISSN 2041-1723. doi: 10.1038/s41467-025-58776-5. URL <https://doi.org/10.1038/s41467-025-58776-5>.
5. Aurelija Šturaitė. Lietuvos gyventojų tankumo kartografovimo metodų lyginamoji analizė / Comparative analysis of methods of cartographic representation of population density in Lithuania. Magistro darbas, Vilniaus Universitetas, Gamtos Mokslų Fakultetas, Kartografijos Centras, Vilnius, Lithuania, 2016.
6. K. B. Stevens. From top to bottom: gridded human population estimates in data-poor situations. *Revue scientifique et technique (International Office of Epizootics)*, 42:111–119, May 2023. ISSN 0253-1933. doi: 10.20506/rst.42.3354.
7. Martino Pesaresi, Marcello Schiavina, Panagiotis Politis, Sergio Freire, Katarzyna Krasnodębska, Johannes H. Uhl, Alessandra Carioli, Christina Corbane, Lewis Dijkstra, Pietro Florio, Hannah K. Friedrich, Jing Gao, Stefan Leyk, Linlin Lu, Luca Maffenini, Ines Mari-Rivero, Michele Melchiorri, Vasileios Syrris, Jamon Van Den Hoek, and Thomas Kemper and. Advances on the global human settlement layer by joint assessment of earth observation and population survey data. *International Journal of Digital Earth*, 17(1):2390454, 2024. doi: 10.1080/17538947.2024.2390454. URL <https://doi.org/10.1080/17538947.2024.2390454>.
8. Christopher T. Lloyd, Heather Chamberlain, David Kerr, Greg Yetman, Linda Pistolesi, Forrest R. Stevens, Andrea E. Gaughan, Jeremiah J. Nieves, Graeme Hornby, Kytt MacManus, Parmanand Sinha, Maksym Bondarenko, Alessandro Sorichetta, and Andrew J. Tatem and. Global spatio-temporally harmonised datasets for producing high-resolution gridded population

- distribution datasets. *Big Earth Data*, 3(2):108–139, 2019. doi: 10.1080/20964471.2019.1625151. URL <https://doi.org/10.1080/20964471.2019.1625151>. PMID: 31565697.
9. Zhen Lei, Shulei Zhou, Penggen Cheng, and Yijie Xie. Improved population mapping for china using the 3d building, nighttime light, points-of-interest, and land use/cover data within a multiscale geographically weighted regression model. *ISPRS International Journal of Geo-Information*, 13(9), 2024. ISSN 2220-9964. doi: 10.3390/ijgi13090335. URL <https://www.mdpi.com/2220-9964/13/9/335>.
 10. Luxiao Cheng, Lizhe Wang, Ruyi Feng, and Jining Yan. Remote sensing and social sensing data fusion for fine-resolution population mapping with a multimodel neural network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:5973–5987, 2021. doi: 10.1109/JSTARS.2021.3086139.
 11. Osama Naz, Muhammad Ibrahim, A. F. Mohiuddin, Asad A. Khan, and Zafar Samad. Public health data quality and evidence use in developing countries: a call to action. *Frontiers in Public Health*, 11:1194499, 2023. doi: 10.3389/fpubh.2023.1194499. URL <https://doi.org/10.3389/fpubh.2023.1194499>.
 12. Bastian Herre and Pablo Arriagada. “Data Page: Population census recently completed”, part of the following publication: Bastian Herre and Pablo Arriagada (2023) - “State Capacity”. Data adapted from UN Statistics Division. <https://ourworldindata.org/grapher/population-census-world-bank>, 2023. [online resource].
 13. Amy N. Rose and Eddie A. Bright. The landscan global population distribution project: Current state of the art and prospective innovation, 2014. URL <https://api.semanticscholar.org/CorpusID:131581529>.
 14. Nando Metzger, Rodrigo Caye Daudt, Devis Tuia, and Konrad Schindler. High-resolution population maps derived from sentinel-1 and sentinel-2. *Remote Sensing of Environment*, 314:114383, 2024. ISSN 0034-4257. doi: <https://doi.org/10.1016/j.rse.2024.114383>. URL <https://www.sciencedirect.com/science/article/pii/S0034425724004097>.
 15. Charlie Frye. Esri’s 2016 world population estimate methodology. Esri white paper, Esri, Redlands, CA, USA, 2017. URL <https://www.esri.com>.
 16. Columbia University Center for International Earth Science Information Network (CIESIN). Documentation for the global rural-urban mapping project, version 1 (grumpv1): Urban extent polygons, revision 01, 2017. URL <https://doi.org/10.7927/H4C53HSR>.
 17. Caleb Robinson, Fred Hohman, and Bistra Dilkina. A deep learning approach for population estimation from satellite imagery. In *Proceedings of the 1st ACM SIGSPATIAL Workshop on Geospatial Humanities*, GeoHumanities ’17, page 47–54, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450354967. doi: 10.1145/3149858.3149863. URL <https://doi.org/10.1145/3149858.3149863>.
 18. F. J. Reed, A. E. Gaughan, F. R. Stevens, G. Yetman, A. Sorichetta, and A. J. Tatem. Gridded population maps informed by different built settlement products. *Data (Basel)*, 3:33, September 4 2018. doi: 10.3390/data3030033.

19. Konstantin Klemmer, Godwin Yeboah, João Porto de Albuquerque, and Stephen A Jarvis. Population mapping in informal settlements with high-resolution satellite imagery and equitable ground-truth, 2020. URL <https://arxiv.org/abs/2009.08410>.
20. Xianfeng Liu, Xiufang Zhu, Yaozhong Pan, Yuqi Ma, Tianqi Li, and Shuchen Chen. Mapping population distribution by integrating night-time light satellite imagery and land-cover data. In *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 2186–2189, 2015. doi: 10.1109/IGARSS.2015.7326238.
21. Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 33(12):6999–7019, 2022. doi: 10.1109/TNNLS.2021.3084827.
22. Huajun Liu, Fuqiang Liu, Xinyi Fan, and Dong Huang. Polarized self-attention: Towards high-quality pixel-wise regression, 2021. URL <https://arxiv.org/abs/2107.00782>.
23. Wei Yao, Zhigang Zeng, Cheng Lian, and Huiming Tang. Pixel-wise regression using u-net and its application on pansharpening. *Neurocomputing*, 312:364–371, 2018. ISSN 0925-2312. doi: <https://doi.org/10.1016/j.neucom.2018.05.103>. URL <https://www.sciencedirect.com/science/article/pii/S0925231218307008>.
24. Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J. Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1), 2021. ISSN 2196-1115. doi: 10.1186/s40537-021-00444-8. URL <https://doi.org/10.1186/s40537-021-00444-8>.
25. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. URL <https://arxiv.org/abs/1505.04597>.
26. Qihui Han, Jie Liu, and Cheolkon Jung. Lightweight generative network for image inpainting using feature contrast enhancement. *IEEE Access*, 10:86458–86469, 2022. doi: 10.1109/ACCESS.2022.3198974.
27. Jim Nilsson and Tomas Akenine-Möller. Understanding ssim, 2020. URL <https://arxiv.org/abs/2006.13846>.
28. Noel Gorelick, Matt Hancher, Mike Dixon, Simon Ilyushchenko, David Thau, and Rebecca Moore. Google earth engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*, 2017. doi: 10.1016/j.rse.2017.06.031. URL <https://doi.org/10.1016/j.rse.2017.06.031>.
29. Sentinel Hub. Sentinel-2 bands. <https://custom-scripts.sentinel-hub.com/custom-scripts/sentinel-2/bands/>, n.d. Accessed: 2025-05-05.
30. Luca Bergamasco and Pietro Asinari. Scalable methodology for the photovoltaic solar energy potential assessment based on available roof surface area: Application to piedmont region (italy). *Solar Energy*, 85:1041–1055, 05 2011. doi: 10.1016/j.solener.2011.02.022.