



Kauno technologijos universitetas
Matematikos ir gamtos mokslų fakultetas

Giliojo mokymosi modelių sujungimas vaizdams segmentuoti
Baigiamasis magistro studijų projektas

Kazimieras Buškus
Projekto autorius

Prof. dr. Evaldas Vaičiukynas
Vadovas

Doc. dr. Mantas Lukoševičius
Vadovas

Kaunas, 2023



Kauno technologijos universitetas
Matematikos ir gamtos mokslų fakultetas

Giliojo mokymosi modelių sujungimas vaizdams segmentuoti

Baigiamasis magistro studijų projektas
Didžiųjų verslo duomenų analitika (6213AX001)

Kazimieras Buškus

Projekto autorius

Prof. dr. Evaldas Vaičiukynas

Vadovas

Doc. dr. Mantas Lukoševičius

Vadovas

Doc. dr. Paulius Palevičius

Recenzentas

Doc. dr. Asta Daunorienė

Recenzentė

Kaunas, 2023



Kauno technologijos universitetas

Matematikos ir gamtos mokslų fakultetas

Kazimieras Buškus

Giliojo mokymosi modelių sujungimas vaizdams segmentuoti

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.

Kazimieras Buškus

Patvirtinta elektroniniu būdu

Buškus, Kazimieras. Giliojo mokymosi modelių sujungimas vaizdams segmentuoti. Magistro studijų baigiamasis projektas / vadovai prof. dr. Evaldas Vaičiukynas ir doc. dr. Mantas Lukoševičius; Kauno technologijos universitetas, Matematikos ir gamtos mokslų fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Taikomoji matematika (Matematikos mokslai).

Reikšminiai žodžiai: semantinis segmentavimas, gilusis mokymasis, modelių ansambliai, žinių distiliavimas.

Kaunas, 2023. 112 p.

Santrauka

Dirbtiniai neuroniniai tinklai padeda, skatina, o tam tikrais panaudojimo atvejais iš esmės įgalina efektyvų užduočių sprendimą daugelyje industrijų, nuo medicinos iki autonominio vairavimo ar žemės ūkio. Vieni produktyviausių vaizdų analizei, taip pat ir semantiniam segmentavimui, naudojamų sprendimų yra paremti giliaisiais konvoliuciniais neuroniniais tinklais. Siekiant dar labiau efektyvinti algoritmus sukurtos įvairios metodikos, viena tokių – kelių modelių jungimas sudarant kolektyvus. Šios technikos, taikomos ir taip resursams reikliems algoritmams, dar labiau didina reikiamų skaičiavimų kiekius. Tampa svarbu rasti balansą tarp pasirenkamo modelio ir jo našumo tam, jog vykdoma ekonominė veikla liktų efektyvi, o šiam tikslui gali būti panaudojama modelių suspaudimo (distiliavimo) metodika.

Magistrinio darbo projekte tiriamas atskirų modelių, homogeninių ir heterogeninių ansamblių bei distiliuotų modelių semantinio segmentavimo efektyvumas vienam specifiniam (jūros dugno vaizdų) bei trimis standartiniams vaizdinių duomenų rinkiniams pasitelkiant penkias giliųjų konvoliucinių tinklų architektūras. Pagal projekte atliktų eksperimentų rezultatus daroma išvada, jog homogeniniai ansambliai pasiteisina didesniems duomenų rinkiniams bei labiau išreikštomis segmentavimo klasėms, o distiliavimo metodikos efektyvesnis naudojimas pastebimas mažesniems duomenų rinkiniams.

Projektą sudaro keturios pagrindinės dalys: pirmiausia atliekama susijusių metodikų literatūros analizė, aprašoma darbo metodika, atliekami semantinio segmentavimo eksperimentai įvairioms modelių konfigūracijoms ir aprašomi gauti empiriniai rezultatai, pateikiamos darbo išvados.

Buškus, Kazimieras. Deep learning model ensembling for semantic segmentation. Master's Final Degree Project / supervisors prof. Evaldas Vaičiukynas and assoc. prof. Mantas Lukoševičius; Faculty of Mathematics and Natural Sciences, Kaunas University of Technology.

Study field and area (study field group): Applied Mathematics (Mathematical Sciences).

Keywords: semantic segmentation, deep learning, ensemble modeling, knowledge distillation.

Kaunas, 2023. 112.

Summary

Artificial neural networks help, prompt, and for specific use cases essentially enable efficient task solving in various industries, from medicine to autonomous driving or agriculture. In image analysis and semantic segmentation, some of the most productive algorithms are based on deep convolutional neural networks. Various techniques are implemented to make the algorithms' performance more efficient – one of them being model ensembling. These solutions, applied to already resource-intensive models, further increase the required computation amounts. It becomes essential to find a balance between the chosen model and its performance for the economic activity to remain effective. For this purpose, one can employ model compression (distillation) methodology.

This project investigates semantic segmentation performance using individual models, compositions of homogeneous and heterogeneous ensembles and distilled models for one specific (seabed transect) and three standard semantic segmentation datasets employing five deep convolutional architectures. According to the results of the experiments carried out in the project, it is concluded that homogeneous ensembles work for larger datasets and more distinguishable segmentation classes. In comparison, the distillation methodology is more effective for smaller datasets.

The project consists of four main parts: first, a literature analysis of related methodologies is performed, the work methodology is described, semantic segmentation experiments are performed for various model configurations, empirical results are described, and the conclusions of the work are presented.

Turinys

Lentelių sąrašas	8
Paveikslų sąrašas	10
Santrumpų ir terminų sąrašas	12
Įvadas.....	13
1. Literatūros apžvalga	14
1.1. Semantinis segmentavimas.....	14
1.2. Mokymosi modelių ansambliai	18
1.3. Ansamblių efektyvinimas	22
1.4. Perkeliamasis mokymasis.....	32
1.5. Mokymosi modelių pateikimas	36
1.6. Apibendrinimas	37
2. Tyrimo objektas ir metodai	39
2.1. Konvoliuciniai neuroniniai tinklai.....	39
2.2. Modelių mokymas	43
2.3. Modelių konfigūracija	44
2.3.1. Modelių architektūros.....	45
2.4. Modelių kolektyvai.....	49
2.4.1. Momentinis ansambliavimas	50
2.4.2. Vertinimo rodikliai	51
2.5. Modelių kolektyvo distiliavimas	53
3. Tyrimų rezultatai.....	55
3.1. Uždavinys, programinė bei techninė įranga	55
3.2. Duomenų rinkiniai.....	56
3.3. Empiriniai rezultatai	59
3.3.1. Arkties duomenų rinkinys	60
3.3.2. Cityscapes duomenų rinkinys.....	69
3.3.3. Floodnet duomenų rinkinys.....	72
3.3.4. Camvid duomenų rinkinys	75
3.4. Ilgesnis modelių mokymas	78
3.5. Diskusija	80
3.6. Ekonominės rekomendacijos.....	81
Išvados	83
Literatūros sąrašas	84
Priedai.....	99
1 priedas. Ansamblio sudarymo schemas	99
2 priedas. <i>Kather-CRC-2016</i> testavimo rezultatai.....	99
3 priedas. Adaptyvaus ansamblio struktūra.....	100
4 priedas. <i>Boosted CNN</i> architektūra	100
5 priedas. <i>BCL</i> sandara	100
6 priedas. Modelių-specialistų testavimo rezultatai	101
7 priedas. CNN ansamblio architektūra objektų atpažinimui.....	101
8 priedas. Deeplab modelio struktūra (<i>Cityscapes</i> segmentavimo pavyzdys)	102
9 priedas. PAN modelio struktūra (<i>Cityscapes</i> segmentavimo pavyzdys).....	104
10 priedas. FPN modelio struktūra (<i>Cityscapes</i> segmentavimo pavyzdys)	106

11	priedas. LinkNet modelio struktūra (Cityscapes segmentavimo pavyzdys)	108
12	priedas. PSPNet modelio struktūra (Cityscapes segmentavimo pavyzdys)	110
13	priedas. Standartinių duomenų rinkinių semantinės klasės	112

Lentelių sąrašas

1 lentelė. Standartiniai semantinio segmentavimo duomenų rinkiniai [13].....	15
2 lentelė. Modelių greitinimo, suspaudimo metodikos [84]	23
3 lentelė. Testavimo rezultatai <i>MNIST</i> rinkiniui [5]	25
4 lentelė. Testavimo rezultatai kalbos atpažinimo užduočiai [5].....	26
5 lentelė. Testavimo rinkinio klasifikavimo paklaidos [128] (<i>EnD</i> – tradicinis, <i>EnD</i> ² – siūlomas distiliavimas)	28
6 lentelė. Testavimo rinkinio klasifikavimo paklaidos vaizdiniais duomenimis [128] (<i>PRR</i> – spėjimo tikslumas (angl. <i>prediction rate</i>), <i>ERR</i> – klaidų lygis (angl. <i>error rate</i>), <i>AUX</i> – agreguoti duomenys)	28
7 lentelė. <i>PASCAL VOC</i> segmentavimo rezultatai atskiriems modeliams, ansambliui bei distiliuotam modeliui [137]	31
8 lentelė. <i>SIFT Flow</i> segmentavimo rezultatai atskiriems modeliams, ansambliui bei distiliuotam modeliui [137]	31
9 lentelė. Rezultatai naudojant <i>AFDS</i> [147] (<i>MACS</i> (angl. <i>multiply-accumulate</i>) nurodo operacijų sumažėjimą mažinant modelį).....	36
10 lentelė. Arkties vaizdų rinkinio informacija (skliausteliuose nurodomas atskirų klasių žymėjimų skaičius).....	56
11 lentelė. Arkties duomenų mokymo parametrai vienai biologinei klasei.....	60
12 lentelė. Arkties rinkinio testavimo rezultatai vienai biologinei klasei.....	60
13 lentelė. Arkties rinkinio segmentavimo ansambliu rezultatai vienai biologinei klasei.....	61
14 lentelė. Momentinio ansamblio testavimo rezultatai biologinei klasei.....	62
15 lentelė. Arkties rinkinio keturių biologinių klasių mokymo parametrai	62
16 lentelė. Modelių bei momentinių ansamblių segmentavimo rezultatai (IOU) <i>Ophiuroidea</i> klasei (x5 – homogeninis kolektyvas).....	63
17 lentelė. Modelių bei momentinių ansamblių segmentavimo rezultatai (IOU) <i>Polychaeta</i> klasei (x5 – homogeninis kolektyvas).....	64
18 lentelė. Modelių bei momentinių ansamblių segmentavimo rezultatai (IOU) <i>Ceriantharia</i> klasei (x5 – homogeninis kolektyvas.....	65
19 lentelė. Modelių bei momentinių ansamblių segmentavimo rezultatai (IOU) <i>Laminaria</i> klasei (x5 – homogeninis kolektyvas).....	66
20 lentelė. Distiliavimo rezultatai (IOU) <i>Laminaria</i> klasei (x5 – homogeninis kolektyvas, dist. – distiliuotas modelis).....	67
21 lentelė. Cityscapes duomenų mokymo parametrai.....	69
22 lentelė. Segmentavimo rezultatai Cityscapes duomenų rinkiniui (x5 – homogeninis ansamblis).....	69
23 lentelė. Distiliavimo testavimo standartiniams rinkiniams konfigūracija.....	71
24 lentelė. Distiliuotų modelių segmentavimo rezultatai Cityscapes duomenų rinkiniui (distiliuotiems modeliams nurodoma mokymosi etapo α reikšmė, x5 – homogeninis kolektyvas, pilkame fone pateikti pradiniai (žr. 22 lentelę) rezultatai)	71
25 lentelė. Floodnet duomenų parametrai	72
26 lentelė. Segmentavimo rezultatai Floodnet duomenų rinkiniui (x5 – homogeninis kolektyvas).	72
27 lentelė. Distiliuotų modelių segmentavimo rezultatai Floodnet duomenų rinkiniui (distiliuotiems modeliams nurodoma mokymosi etapo α reikšmė, x5 – homogeninis kolektyvas, pilkame fone pateikti pradiniai (žr. 26 lentelę) rezultatai)	74
28 lentelė. Camvid duomenų mokymo parametrai	75

29 lentelė. Segmentavimo rezultatai Camvid duomenų rinkiniui (x5 – homogeninis ansamblis) ...	75
30 lentelė. Distiliuotų modelių segmentavimo rezultatai Camvid duomenų rinkiniui (distiliuotiems modeliams nurodoma mokymosi etapo α reikšmė, x5 – homogeninis kolektyvas, pilkame fone pateikti pradiniai (žr. 29 lentelę) rezultatai)	77
31 lentelė. Atskirų momentinio modelių ansamblio narių rezultatai (mokant 1000 epochų).....	78
32 lentelė. Distiliavimo rezultatai standartiniams rinkiniams (pilkame fone pateikti pradiniai (žr. 22, 26, 29 lenteles) rezultatai, modelį-mokytoją ruošiant 200 epochų)	79
33 lentelė. Laiko testavimo konfigūracija	82

Paveikslų sąrašas

1 pav. Kompanijų <i>Arterys (a)</i> , <i>Ares (b)</i> , <i>Hexagon (c)</i> , <i>MVTec (d)</i> marketingo medžiaga	15
2 pav. Standartinių semantinio segmentavimo rinkinių pavyzdžiai	17
3 pav. Kaskadinis modelių mokymas [79].....	22
4 pav. Modeliams-specialistams priskirtų klasių pavyzdžiai [5]	26
5 pav. Modelių ansamblio spėjimo savybės [128].....	27
6 pav. Klasifikavimui naudojamas 3 klasių duomenų rinkinys [128].....	28
7 pav. Bendroji modelių mokymo schema [137]	30
8 pav. Perkeliamojo mokymosi iliustracija [144]	33
9 pav. Perkeliamojo mokymosi klasifikacija [145]	34
10 pav. Induktyviojo mokymosi iliustracija [144].....	34
11 pav. <i>AFDS</i> koncepcija [147]	35
12 pav. <i>ReLU (a)</i> bei <i>sigmoid (b)</i> aktyvacijos funkcijų pavydžiai [154].....	40
13 pav. Konvoliucijos operacija su užpildymu (brūkšniuotų linijų nurodomas plotas) [155].....	40
14 pav. Pavyzdinis išvesties skaičiavimas trijų kanalų matricai su dviem konvoliuciniais filtrais (3x3x3), kai užpildymas = 0, žingsnis = (1, 3).....	41
15 pav. Maksimalaus telkimo (angl. <i>max pooling</i>) operacijos iliustracija.....	41
16 pav. Paketinės normalizacijos operacijos taikymo konteksto iliustracija	42
17 pav. <i>Softmax</i> veikimo iliustracija [160]	43
18 pav. Abstrakti užkoduotojo-dekoduotojo architektūros schema	45
19 pav. ResNet modelio bloko algoritmo principo iliustracija	46
20 pav. PSPNet architektūra [163].....	46
21 pav. LinkNet architektūra [164].....	47
22 pav. FPN architektūros principo iliustracija [165].....	48
23 pav. PAN architektūra: modelis (a), <i>GAU</i> blokas (b), <i>FPA</i> modulis (c) [166].....	48
24 pav. <i>ASPP</i> blokas [167].....	49
25 pav. Modelių jungimo algoritmo iliustracija.....	50
26 pav. Mokymosi koeficiento grafikas pritaikius kosinusinį atvėsimą (iš viso 100 epochų kas 20 epochų išsaugant momentinį modelį)	51
27 pav. Sumaišymo matricos pavyzdys kelių klasių spėjimams	52
28 pav. Pavyzdiniai sužymėtos transektos keturioms biologinės klasėms	57
29 pav. Cityscapes duomenų rinkinio pavyzdžiai: mokymo (pirma eilutė) ir testavimo (antra eilutė) vaizdai (a) ir segmentavimo kaukės (b)	58
30 pav. Floodnet duomenų rinkinio pavyzdžiai: mokymo (pirma eilutė) ir testavimo (antra eilutė) vaizdai (a) ir segmentavimo kaukės (b)	58
31 pav. Camvid duomenų rinkinio pavyzdžiai: mokymo (pirma eilutė) ir testavimo (antra eilutė) vaizdai (a) ir segmentavimo kaukės (b)	59
32 pav. „B1_0332“ transektos Ophiuroidea klasės segmentavimas LinkNet modeliu bei homogeniniu ansambliu (x5) (raudonai vaizduojamas ekspertinio žymens apvadas, žaliai – neteisingai žymėta klasė, mėlynai – neteisingai nepažymėta klasė)	63
33 pav. „G3_0100“ transektos Polychaeta klasės segmentavimas PAN modeliu bei homogeniniu ansambliu (x5) (raudonai vaizduojamas ekspertinio žymens apvadas, žaliai – neteisingai žymėta klasė, mėlynai – neteisingai nepažymėta klasė)	64

34 pav. „B7_0237“ transektos Ceriantharia klasės segmentavimas LinkNet modeliu bei homogeniniu ansambliu (x5) (raudonai vaizduojamas ekspertinio žymens apvadas, žaliai – neteisingai žymėta klasė, mėlynai – neteisingai nepažymėta klasė)	65
35 pav. „D1(1)_0822“ transektos Laminaria klasės segmentavimas Deeplab modeliu bei homogeniniu ansambliu (x5) (raudonai vaizduojamas ekspertinio žymens apvadas, žaliai – neteisingai žymėta klasė, mėlynai – neteisingai nepažymėta klasė)	66
36 pav. Sėkmingų biologinių klasių segmentavimo Deeplab homogeniniu ansambliu modeliu pavyzdžiai (raudonai vaizduojamas ekspertinio žymens apvadas, žaliai – neteisingai žymėta klasė, mėlynai – neteisingai nepažymėta klasė)	67
37 pav. „D1(1)_0822_30s“ transektos <i>Laminaria</i> klasės segmentavimas Deeplab modeliu, Deeplab homogeniniu ansambliu, LinkNet bei distiliuotu LinkNet modeliais (raudonai vaizduojamas ekspertinio žymens apvadas, žaliai – neteisingai žymėta klasė, mėlynai – neteisingai nepažymėta klasė).....	68
38 pav. Cityscapes testavimo rinkinio segmentavimai Deeplab modeliu ir homogeniniu ansambliu (x5)	70
39 pav. Deeplab homogeninio ansamblio narių efektyvumai (Cityscapes).....	71
40 pav. Cityscapes testavimo rinkinio segmentavimai homogeniniais ansambliais (x5) ir distiliuotais modeliais PAN ir FPN architektūroms	72
41 pav. Floodnet testavimo rinkinio segmentavimai Deeplab modeliu ir homogeniniu ansambliu (x5)	73
42 pav. Deeplab homogeninio ansamblio narių efektyvumai (Floodnet).....	74
43 pav. Floodnet testavimo rinkinio segmentavimai homogeniniais ansambliais (x5) ir distiliuotais PAN ir FPN modeliais.....	75
44 pav. Camvid testavimo rinkinio segmentavimai FPN modeliu ir homogeniniu ansambliu (x5). 76	
45 pav. Deeplab homogeninio ansamblio narių efektyvumai (Camvid).....	77
46 pav. Camvid testavimo rinkinio segmentavimai homogeniniais ansambliais (x5) ir distiliuotais PAN ir FPN modeliais.....	78
47 pav. Modelių segmentavimo trukmė bei efektyvumas Floodnet rinkiniui (* – homogeninis kolektyvas)	82

Santrumpų ir terminų sąrašas

Camvid – standartinis miesto gatvių vaizdų semantinio segmentavimo duomenų rinkinys;

Cityscapes – standartinis miesto gatvių vaizdų semantinio segmentavimo duomenų rinkinys;

CNN – konvoliuciniai (sąsūkų) neuroniniai tinklai (angl. *Convolutional Neural Networks*);

Deeplab – „DeepLabv3+“ konvoliucinio neuroninio tinklo architektūra;

Floodnet – standartinis aeronuotraukų semantinio segmentavimo duomenų rinkinys;

FPN – požymių piramidės konvoliucinio neuroninio tinklo architektūra (angl. *Feature Pyramid Network*);

ImageNet – standartinis bendrinio tipo vaizdų semantinio segmentavimo duomenų rinkinys;

IOU – semantinio segmentavimo tikslumo metrika (angl. *Intersection-Over-Union*);

mIOU – semantinio segmentavimo klasių IOU vidurkio metrika (angl. *mean Intersection-Over-Union*);

PAN – piramidinė dėmesio konvoliucinio neuroninio tinklo architektūra (angl. *Pyramid Attention Network*);

PSPNet – piramidinė scenos analizės konvoliucinio neuroninio tinklo architektūra (angl. *Pyramid Scene Parsing Network*);

ResNet – liekamųjų ryšių konvoliucinio neuroninio tinklo architektūra (angl. *Residual Network*);

SGD – stochastinio gradientinio nusileidimo algoritmas;

SOTA – geriausi iki šios dienos pasiekti rezultatai (angl. *state of the art*).

Įvadas

Mašininio mokymosi kontekste jau kurį laiką pastebima ši tendencija: individualioms, specifinėms dalykinėms sritims ne visada kuriami nauji mašininio mokymosi modeliai, o naudojama jau egzistuojanti informacija, architektūros, duomenų rinkiniai bei modelių pateikimo produkcinei aplinkai sprendimai pritaikant tik specifines duomenų apdorojimo grandinės dalis ir algoritmus [1, 2, 3]. Visiškai naujų (angl. *from scratch*) modelių architektūrų kūrimas kiekvienai konkrečiai užduočiai yra netinkamas aukštos pridėtinės vertės kūrimui, o modelių ar duomenų pakartotinis panaudojimas gerina pačių algoritmų efektyvumą. Produkcinei aplinkai skirtų modelių kūrimo eiga gali būti nusakoma kaip akademiniam kontekste sukurtų efektyviausių (SOTA) modelių pastovus perdirbimas juos pritaikant iš(si)keltam tikslui [1]. Skirtinguose etapuose gali būti sudaromi algoritmų ansambliai [4] bei atliekamas ansamblių koncentravimas (distiliavimas, angl. *distillation*) [5, 6] gaunant kompaktiškesnius, tačiau efektyvius modelius, pasižyminčius mažesniu poreikiu skaičiavimo resursams. Taip pat taikomas perkeliamasis mokymasis (angl. *transfer learning*) [7], kurio metu žinios, išmoktos bendrinėje naudojimo srityje (t.y. su tam tikro tipo duomenimis), panaudojamos kitu tikslu skirtingos tikslinės srities rėmuose. Norint gauti modelius, pasiekiančius geriausius rezultatus būtent specifinės dalykinės srities problematikai spręsti, naudojamos jau sukurtomis architektūromis: modeliai gali būti transformuojami juos pritaikant turimiems duomenims bei siekiamoms išvestims, rezultatams gerinti modeliai gali būti jungiami į ansamblius, kurių nariai ruošiami naudojant perkeliamąjį mokymąsi. Šiuo atveju pagrindinė optimizacija turi būti atliekama būtent korektiškai sudarant modelių ansamblius bei juos paprastinant ribotų resursų diegimo (angl. *model deployment*) ir pateikimo (angl. *model serving*) aplinkose. Kompiuterinės regos užduotims spręsti būtent gilieji konvoliuciniai neuroniniai tinklai (CNN) [8] teikia geriausius rezultatus bei efektyviausius spėjimus, o bene plačiausiai šie algoritmai naudojami vaizdų segmentavimo uždaviniuose [9]. Tačiau dėl šių modelių taikymui reikalingų resursų (skaičiavimo, darbinės, grafinės atminties ir kt.) kiekio, metodų naudojimas gali tapti sudėtingas. Geriausiems rezultatams pasiekti reikalingas efektyvus duomenų, naudojamų mokymosi bei spėjimo (angl. *inference*) etapų metu, valdymo mechanizmas, lengviau įgyvendinamas diegiant kompaktišką (pritaikius distiliavimą) CNN modelį.

Darbo tikslas – ištirti įvairių giliojo mokymosi architektūrų vaizdų segmentavimo efektyvumą pritaikant ansambliavimo bei žinių distiliavimo technikas.

Darbo tikslo įgyvendinimui iškeliami šie uždaviniai:

1. apžvelgti metodus, skirtus giliojo mokymosi modeliams sujungti ir distiliuoti;
2. pasirinkti ir paruošti vaizdų rinkinius giliam mokymuisi;
3. pritaikyti pasirinktus semantinio segmentavimo modelius;
4. pritaikyti modelių jungimą kolektyvams sudaryti;
5. išbandyti distiliavimo technikas spėjimų greitaveikai gerinti;
6. apibendrinti eksperimentų rezultatus ir pateikti išvagas.

1. Literatūros apžvalga

1.1. Semantinis segmentavimas

Semantinis segmentavimas (angl. *semantic segmentation*), arba taško lygmens klasifikavimas, yra kompiuterinės regos (angl. *computer vision*) užduočių pogrupis, kuomet siekiama grupuoti tai pačiai klasei priklausančias atskiras paveikslėlio dalis [10]. Semantinio segmentavimo naudojimas padeda nepriklausomiems agentams aiškiau suprasti ir, jei reikalinga, sąveikauti su išoriniu pasauliu. Šio tipo algoritmai taikomi daugelyje industrijų nuo gamybos pramonės (kokybės kontrolė [11], darbo sauga [12]), medicinos (vaizdų analizė [13], gydymo planavimas [14]), mažmeninės prekybos (prekių išdėstymas ir valdymas [15]), automobilių pramonės (autonominis vairavimas [16, 17, 18, 19]) iki agrokultūros [20, 21, 22, 23]. Toliau pateikiami keli semantinio segmentavimo, taikomo kartu su kitomis vaizdo analizės ir apdorojimo technologijomis ir teikiančio ekonominę naudą, pavyzdžiai.

Mažmeninėje drabužių prekyboje virtualaus matavimosi sektoriaus vertė 2021 m. siekė 2,79 mlrd. USD, o iki 2028 m. sieks 8,5 mlrd. USD¹. Ši technologija leidžia pirkėjams peržiūrėti įvairias aprangos kombinacijas, pritaikyti drabužių dydį bei juos stilizuoti neliečiant pačių rūbų ar net nesilankant parduotuvėje naudojantis aplikacijomis, pritaikytomis išmaniesiems telefonams, papildytosios realybės (angl. *augmented reality*) veidrodžiams ar virtualios realybės (angl. *virtual reality*) akiniams. Kompanijų² teigimu, naudojantis šia technologija galima pritraukti ir išlaikyti klientus teikiant suasmenintą vartojimo patirtį, motyvuoti pirkėją bei sutaupyti iš karto neįsigyjant ir nesandėliuojant kuo didesnei vartotojų auditorijai pritaikyto asortimento, taip pat sumažinti netinkančių prekių grąžinimo kaštus.

Vaizdų analizė taip pat svarbi ir tradicinėse rinkose, pavyzdžiui medicinos sektoriuje. Jau nuo praeito šimtmečio vidurio tyrėjai siūlė ir diegė paprastesnes ar sudėtingesnes sistemas, skirtas padėti priimti klinikinius sprendimus, o skaitmenizacija, augantis asmeninės medicinos poreikis bei proveržiai dirbtinio intelekto srityje šią tendenciją tik išryškino [24, 25]. Dirbtinio intelekto taikymo medicinoje sektorius 2022 m. buvo vertinamas virš 15 mlrd. USD, o tikėtinas augimas iki 2030 m. siekia 187 mlrd. USD³. Vaizdų analizei pasitelkiant gilųjį mokymąsi dažniausiai naudojami rentgenologinio, kompiuterinės tomografijos, magnetinio rezonanso ar ultragarsinės diagnostikos tyrimų metu gaunami vaizdai [26]. Dirbtiniu intelektu, taip pat ir semantiniu segmentavimu, paremtus sprendimus siūlančios kompanijos⁴ teigia, jog naudojant jų produktus trumpėja tyrimų laikas (spartėja brangios technikos atsiperkamumas), mažinami specialistų atliekamos analizės bei tyrimų rezultatų aprašymo kaštai, be to, teigiama, jog kai kurios sistemos klinikinius duomenis vertina efektyviau už gydytojus-specialistus [27]. Medicininės įrangos kompanijos, pavyzdžiui *Philips*, diegia vaizdų analizės (pvz., organų ar jų dalių segmentavimo) sprendimus į parduodamų prietaisų programinę įrangą⁵.

¹ <https://reports.valuates.com/market-reports/360I-Auto-9Q319/virtual-fitting-room>

² <https://ares.snap.com>, <https://3dlook.me>

³ <https://www.precedenceresearch.com/artificial-intelligence-in-healthcare-market>

⁴ <https://www.arterys.com>, <https://www.aidoc.com>

⁵ <https://www.philips.com/a-w/imalytics/specialized-modules/model-based-segmentation.html>



1 pav. Kompanijų *Arterys* (a), *Ares* (b), *Hexagon* (c), *MVTEC* (d) marketingo medžiaga

Statybos sektoriuje⁶ šias technologijas pritaikant aeronuotraukoms galima efektyviau atlikti miestų planavimo, geodezinių matavimų, pastatų projektų dokumentacijos etapų atskiras veiklas; žemės ūkyje⁷ sprendimai taikomi trąšų bei insekticidų skirstymo apsaugant augalus procesams įgyvendinti, šiltnamių automatizavimui ir pan. Taigi, matoma, jog vaizdų analizės technologijų panaudojimo atvejų (žr. 1 pav.) galima rasti įvairiose srityse, o sprendimus diegiančios kompanijos teigia šių technologijų taikymo ekonominėse veiklose naudą.

Semantinio segmentavimo modelių ruošimo, diegimo bei pateikimo strategija ir naudojama infrastruktūra yra individualiai taikomos kiekvienai aplikacijai. Šis pasirinkimas priklauso nuo vartotojų lūkesčių, panaudojimo atvejo reikalavimų, galimų naudoti technologijų. Reikia mokytis ir pačius modelius, o šis procesas reikalauja didelio duomenų kiekio (semantinio segmentavimo atveju – ekspertų (ar sistemų) žymėtos vaizdinės informacijos). Dalis industrijoje naudojamų vaizdų rinkinių pateikiama 1 lentelėje toliau:

1 lentelė. Standartiniai semantinio segmentavimo duomenų rinkiniai [13]

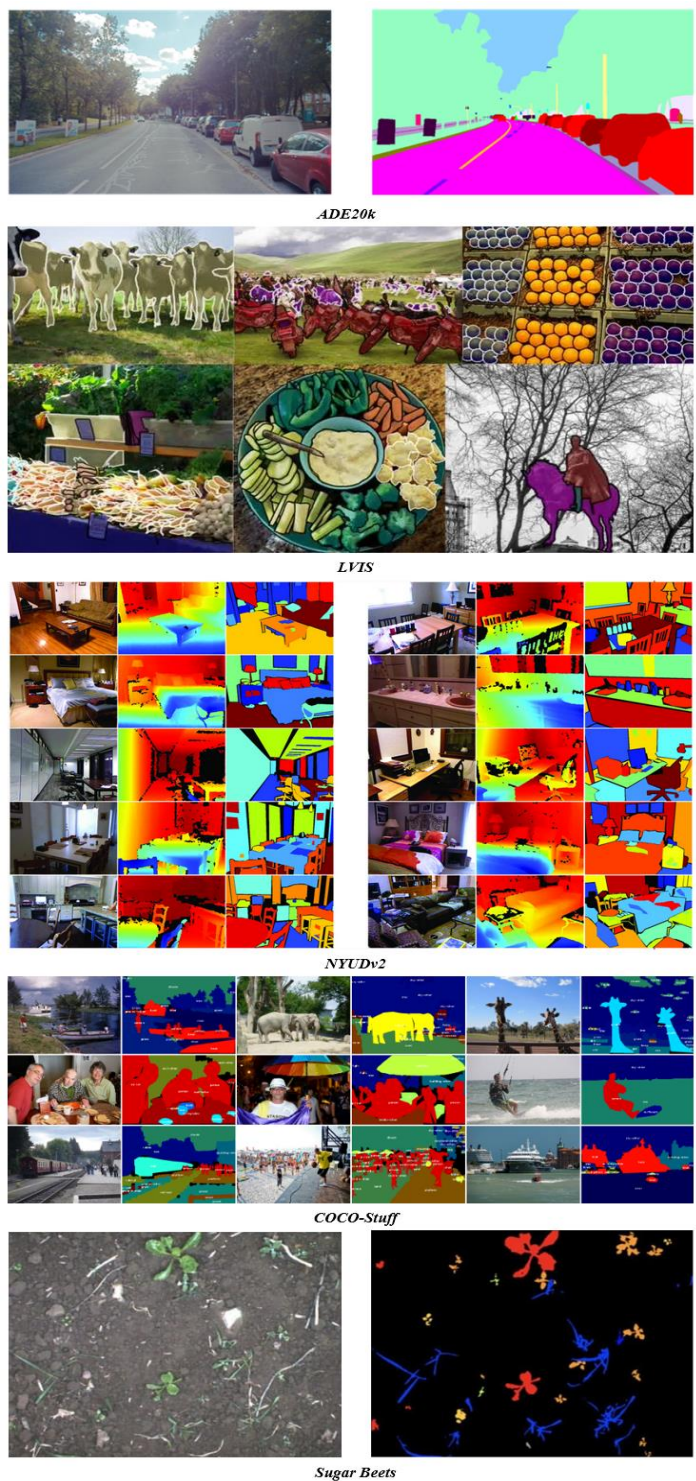
Duomenų rinkinys	Aplinka/naudojimas	Metai	Klasių skaičius	Rezoliucija	Vaizdų kiekis
<i>PASCAL VOC</i> [28]	Bendrini	2012	21	Skirtinga	2913
<i>PASCAL CONTEXT</i> [29]	Bendrini	2014	540	Skirtinga	10650

⁶ <https://hexagon.com/products/product-groups/geosystems>

⁷ <https://www.mvtec.com/application-areas/agriculture>

Duomenų rinkinys	Aplinka/naudojimas	Metai	Klasių skaičius	Rezoliucija	Vaizdų kiekis
<i>SBD</i> [30]	Bendrinis	2011	21	Skirtinga	11355
<i>Microsoft COCO</i> [31]	Bendrinis	2014	80	Skirtinga	328124
<i>Cityscapes</i> [32]	Miestas	2016	30	2048x1024	5000
<i>Camvid</i> [33]	Miestas	2009	32	960x720	701
<i>Stanford background</i> [34]	Laukas	2009	8	320x420	715
<i>Sift Flow</i> [35]	Laukas	2011	33	256x256	2688
Youtube-Objects-Jain [36]	Objektai	2014	10	480x360	10167
Adobe's Portrait Segmentation [37]	Portretai	2016	2	600x800	1800
MINC [38]	Medžiagos	2015	23	Skirtinga	14561
NYUDv2 [39]	Vidus	2012	40	480x640	1449
<i>ADE20k</i> [40]	Bendrinis	2017	150	Skirtinga	22210
<i>Coco - stuff</i> [41]	Daiktai	2018	172	Skirtinga	164000
<i>PASCAL-PART</i> [42]	Bendrinis (žymėtos atskiros dalys)	2014	80	Skirtinga	19740
<i>iSAID</i> [43]	Paviršiaus nuotraukos iš oro	2019	15	Skirtinga	2806
<i>Open Images Dataset V6</i> [44]	Bendrinis	2020	350	Skirtinga	9 mln.
<i>MinneApple</i> [45]	Obuoliai	2019	1	720x1280	1671
<i>LVIS</i> [46]	Bendrinis	2019	1203	Skirtinga	164000
<i>UNIMIB 2016</i> [47]	Padėklai su maistu	2017	73	3264x2448	1027
<i>A2D2</i> [48]	Miestas	2020	38	920x1208	41280
<i>Sugar Beets</i> [49]	Augalai	2016	3	1296x966	12340

Pavyzdiniai rinkinių vaizdai pateikiami 2 pav. toliau:



2 pav. Standartinių semantinio segmentavimo rinkinių pavyzdžiai

Taigi, semantiniu segmentavimu paremta vaizdų analizė taikoma daugelyje sričių ir industriju, egzistuoja įvairiems taikymams skirtų duomenų rinkinių. Paruošus modelius juos siekiama toliau efektyvinti – tam skirtos ansambliavimo bei distiliavimo metodikos aptariamos tolesniuose poskyriuose.

1.2. Mokymosi modelių ansambliai

Vienas veiksmingiausių ir plačiausiai naudojamų būdų mašininio mokymosi algoritmų vykdomos užduoties rodikliams gerinti – mokymo imties (padalintos išskiriant svarbesnius stebėjimus ir pan.) naudojimas keliems skirtingiems modeliams (identišku ar skirtingu architektūru) parengti ir šių modelių agreguotų išvesčių naudojimas galutinei prognozei gauti – vadinamieji modelių ansambliai (angl. *ensemble methods*) [50]. Šie rinkiniai pagrįsti hipoteze, jog derinant kelis modelius kartu galima gauti „galingesnį“ modelį, nes bendruoju atveju skirtingi algoritmai yra linkę nedaryti tų pačių klaidų testavimo etapo metu. Tariant, jog atskiri modeliai išgauna skirtingas žinias iš duomenų ir galimai atsižvelgia į skirtingas pavyzdžių savybes, būtų logiška modelius jungti į ansamblį tam, kad būtų galima panaudoti kiekviename modelyje esančią mokymo metu gautą informaciją. Taip sudaroma efektyvesnė sistema, nes pavyzdžių žymų (angl. *labels, targets*) prognozės atveju, kai vienas modelis klysta, tikėtina, jog kiti modeliai prognozuos teisingai.

Pagrindinė ansamblių idėja yra pakankamai paprasta: keli „silpni“ modeliai (angl. *weak learners*) mokomi spręsti tą pačią užduotį ir vėliau sujungiami geresniems rezultatams gauti [51]. Ruošiant bet kurį modelį tikimasi minimizuoti poslinkį (angl. *bias*) ir dispersiją (angl. *variance*), tačiau šie rodikliai neigiamai koreliuoja. Norint pasiekti optimalų konkrečios užduoties sprendimą reikia spręsti būtent šią dalinio nesuderinamumo problemą: modelis turi turėti pakankamai laisvės laipsnių užduoties duomenų rinkiniui apibendrinti, tačiau neturi būti pernelyg prisitaikyta prie konkrečios duomenų imties. Jeigu būtų galima tiesiogiai išspręsti šią poslinkio-dispersijos kompromiso problemą, daugelis ansamblių tipų neturėtų prasmės. Konstruojant ansamblius trivialūs („silpni“) modeliai ir sudaro sudėtingesnio, efektyvesnio galutinio modelio (ansamblio) pagrindą. Atskiri algoritmai dažnu atveju arba yra pernelyg šališki (stebimas žymus poslinkis), arba per daug jautrūs (stebima didelė dispersija), jog būtų tinkami naudoti po vieną. Ansamblių sudarymas remiasi tuo, kad silpnesni, poslinkio arba dispersijos kompromisinio lygmens netenkinantys modeliai sujungiami taip sukuriant efektyvesnį ir tvaresnį (t.y. įvairiose situacijose, pavyzdžiui nebūdingo pavyzdžio spėjime, patenkinamus rezultatus gaunanti) modelį.

Ansamblių kūrimo metodikos gali būti skirstomos į 3 pagrindines grupes [52]: stochastiniu atrinkimu (angl. *bootstrap aggregation, bagging*), hipotezių stiprinimu (angl. *boosting*) grįstas metodikas bei daugybinių struktūrų (angl. *stacking*) (bendra tokių algoritmų grupė – bazinių modelių jungimas metamodeliais (angl. *metamodels*)) sudarymą. Pačios metodikos pasirinkimas priklauso ir nuo to, kokie – ar identišku, ar skirtingu struktūrų ir net veikimo principų – modeliai bus naudojami ansamblyje. Dažniausiai (*bagging, boosting*) taikomas vieno tipo mašininio mokymosi algoritmas ir kuriami skirtingai mokyti baziniai modeliai, iš kurių sudaromas homogeninis (vienalypis) ansamblis. Kitų metodų atveju, taip pat naudojant daugybines struktūras, baziniai mokymosi algoritmai gali skirtis, o modeliai jungiami į heterogenišką (daugialypį) ansamblį. Kuriant šias modelių kombinacijas svarbu tai, jog bazinių modelių pasirinkimas bei jų agregavimas į ansamblį derėtų. Jei pasirenkami baziniai modeliai pasižymi mažu poslinkiu, ir, neišvengiamai, didesne dispersija, į tai turėtų būti atsižvelgiama sudarant galutinius rinkinius, o metodika turi gebėti mažinti būtent dispersiją. Galima teigti, jog naudojant *bagging* gaunamas modelių ansamblis, kurio dispersija yra mažesnė nei jo komponentų, o *boosting* metodika ir daugybinių struktūrų sudarymu kuriami ansambliai, kurių bendras poslinkis yra mažesnis už atskirų komponentų.

Toliau, remiantis [51, 52, 53, 54], trumpai aptariamos *boosting, bagging* ir daugybinių struktūrų metodikos.

Bagging technika paremta paprastųjų atsitiktinių grąžintinių imčių pakartojimais (pakartotinės imties metodas, angl. *bootstrap*), kuomet iš pradinio stebėjimų rinkinio sudaromas pasirinktas kiekis vienodo dydžio imčių. Norint, jog šios imtys būtų reprezentatyvios prognozuojamos populiacijos atžvilgiu, pradinio duomenų rinkinio dydis turėtų būti pakankamai didelis, jog būtų galima fiksuoti didžiąją dalį informacijos sklaidos kompleksiskumo (tuomet ir atrinktosios imtys gerai aproksimuos populiaciją). Taip pat svarbu, jog pradinės imties dydis būtų didesnis už atrinktųjų imčių, kad būtų galima išvengti priklausomumo. Dažnai nėra įmanoma įvertinti net pradinės duomenų imties reprezentatyvumo bei nepriklausomumo, tačiau praktiškai taikant pakartotines imtis gaunami apytiksliai įverčiai yra naudingi efektyvių prognozių sudarymui. Naudojant *bagging* paruošiami keli nepriklausomi modeliai, o galutinis (ansamblio) modelis gaunamas vidurkinant (ar kitaip agreguojant) jų išvestis. Bet kurioms daugiau mokymo duomenų, plačios įvesčių / išvesčių aibės reikalaujančioms užduotims nėra galimybės paruošti visiškai nepriklausomus modelius, nes tai reikalautų labai didelio duomenų kiekio. Tačiau apytikslės pakartotinių imčių reprezentatyvumo bei nepriklausomumo savybės lemia, jog tokiomis imtimis mokyti modeliai yra, iš praktinės pusės, nepriklausomi. Svarbu paminėti, jog *bagging* atveju modeliai gali būti mokomi paraleliai. Vidurkinant ar kitaip agreguojant mokytų bazinių modelių išvestis, konkrečiam pavyzdžiui tikėtina prognozė nesikeičia, tačiau dispersiją galima sumažinti.

Boosting – nuosekli modelių ansamblių sudarymo metodika. *Boosting* metu modeliai ruošiami iteratyviai: modelio mokymasis tam tikro žingsnio metu priklauso nuo mokymo ankstesniajame. Kaip jau minėta, bendruoju atveju gaunamas ansamblis, kurio poslinkis yra mažesnis už atskirų modelių. *Boosting* metu kiekviena sekos mokymosi imtis paruošiama taip, jog modelis būtų priverstas labiau atsižvelgti į anksčiau sekoje mokytų bazinių modelių klaidingai klasifikuotus (ar regresuotus) pavyzdžius. Taip pasiekama, jog kiekvieno naujai ruošiamo modelio pagrindiniai resursai būtų skiriami patiems sudėtingiausiems imties pavyzdžiams, o iteratyvaus proceso pabaigoje būtų gautas modelis su mažesniu poslinkiu. Būtent dėl poslinkio mažinimo aspekto *boosting* modeliais dažniausiai pasirenkami didesnį poslinkį linkę formuoti algoritmai (pvz., sprendimų medžių atveju bus renkamosi naudoti mažesnę kiekį išsiskojimų turinčius medžius). Kitas svarbus aspektas – modelių sudėtingumas. Dėl to, jog skaičiavimai *boosting* atveju negali būti lygiagretinami, sudėtingų modelių naudojimas gali tiesiog per ilgai trukti. Pasirinkus bazinius modelius taikomos skirtingos technikos nuosekliai mokymuisi, t.y. koku būdu bus atsižvelgiama į ankstesnes modelių klaidas. Dažniausiai naudojami keli metaalgoritmai: *Adaboost* [55] ir *Gradient boosting* [56]. *Adaboost* atnaujina mokymo imties pavyzdžių svorius kiekviename etape, o *Gradient boosting* atveju atnaujinami patys stebėjimai, t.y. skiriasi šių algoritmų optimizavimo uždavinio sprendimas. *Adaboost* atveju optimizuojama iteratyviai: ieškoma lokaliai optimali svorio-klasifikatoriaus (regresoriaus) pora, o *Gradient boosting* atveju bazinis modelis kiekvienoje iteracijoje mokomas priešinga esamo ansamblio (sudaryto modelio) nuostolių funkcijos (angl. *cost function*) gradiento kryptimi.

Daugybinių struktūrų metodika siekia mažinti ansamblio komponentų poslinkį, o atskiri modeliai dažnai skiriasi struktūriškai ar net veikia skirtingais optimizavimo principais. Dažniausiai atskirų modelių išvestys galutinę ansamblio prognozę formuoja ne tiesiogiai, o panaudojant papildomą metamodelį. Gali būti mokomas tam tikras atskirų modelių kiekis n bei papildomas metamodelis, skirtas optimizuotam išvesčių vertimui į ansamblio prognozę. Dėl to, jog atskirų modelių paruošimo (mokymosi) metu prognozės nėra aktualios (modelių išvestys yra įvestys metamodeliui), susiduriama su duomenų skirstymo problema: rinkinys turi būti dalinamas mažiausiai į dvi dalis (viena – bazinių

modelių, kita – metamodelio rengimui). Dėl skirtingų atskirų modelių ruošimo resursų reikalavimų ir jų mokymosi trukmės atsirandanti duomenų skirstymo problema iš dalies sprendžiama k skirsnių (angl. *k-fold*) mokymusi (vadinamasis *Super Learner* metodas [57]), kuomet kiekvienam stebėjimui (ar stebėjimų grupei) spėjimai (įvestys metamodeliui) gaunami naudojant ant $k - 1$ skirsnių mokytus atskirus modelius. Paties metamodelio pasirinkimas yra platus: priklausomai nuo užduoties galima naudoti paprastus regresinius modelius, atsitiktinius miškus ar apdoroti išvestis neuroniniais tinklais, *SVM* (angl. *support vector machine*) modeliais ar kitais algoritmais [52, 54] (CNN – [58, 59, 60, 61]).

Neuroninių tinklų ansambliai gali būti konstruojami pasitelkiant neigiamos koreliacijos mokymosi metodiką [62]. Šio būdo idėja – skatinti atskirus modelius ansamblyje išmokti skirtingus mokymosi imties aspektus (įvairias jo dalis) tam, jog visas ansamblis geriau apibendrintų pilną duomenų rinkinį. Naudojantis šiuo algoritmu visi ansamblio nariai mokomi paraleliai pritaikant papildomą koreliacijos baudos narį nuostolių funkcijose. To paties proceso metu atskiri modeliai yra ir mokomi, ir kombinuojami į ansamblį.

Be jau aptarto modelių heterogeniškumo / homogeniškumo egzistuoja dar vienas svarbus skirtumas modelių ansamblių kūrime: numanomi ir išreikšti ansambliai (angl. *implicit / explicit ensembles*). Atskiras mašininio mokymosi modelis taip pat gali elgtis kaip homogeniškų modelių ansamblis: tokios situacijos gali susidaryti naudojant išmetimo (angl. *dropout*) [63], stochastinio gylio (angl. *stochastic depth*) [64] algoritmus, jų sintezę (*Swapout*) [65] ar kitas panašias metodikas (pvz., *Disturblabel* [66], kuomet mokymosi metu kiekvieno paketo (angl. *batch*) žymenys pakeičiami su tam tikra tikimybe).

Konstruojant modelių ansamblius bene svarbiausia bet kurio algoritmo dalis yra bendrojo sprendimo, sudaryto iš atskirų narių rezultatų, sudarymas (angl. *decision fusion*) (pats sprendimo jungimas gali būti ir ansamblių sudarymo metodikos dalis). Egzistuoja keli pagrindiniai būdai modelių išvesčių jungimui: klasifikavimo atveju dažniausiai naudojami pagal vidutinę (angl. *average voting*), maksimalią (angl. *majority voting*) skirstinio (gaunamo iš *softmax* [67] sluoksnio ir pan.) reikšmę gauti rezultatai, taip pat pasitelkiamos sprendimo šablonų (angl. *decision template*) [68], Bajeso [69], *Dempster-Shafe* [70] ar kitos technikos. Jei žinoma konkreti informacija apie bendrinės srities ar konkrečios problemos tikslinę populiaciją, galimas ir svertinis balsavimas. *Ju et al.* [71] iškėlė hipotezę, jog naivus sprendimų jungimas (pvz., vidurkinimas) nėra pakankamai adaptyvus duomenims ir gali lemti prastesnius rezultatus, nes per daug atsižvelgiama į didelio poslinkio modelių prognozavimo išvestis.

Literatūroje siūlomi ir nauji, inovatyvesni sprendimai modifikuojant klasikinius sprendimų jungimo algoritmus. *Paladini et al.* [72] pristato dvi technikas būtent CNN ansamblių sudarymui: *Mean-Ensemble-CNN* bei *NN-Ensemble-CNN* vėžinių darinių klasifikavimo kontekste jungia keturių populiarių neuroninių tinklų architektūrų išvestis. Naudojant *Mean-Ensemble-CNN*, skirtingai nuo klasikinių sprendimų jungimo technikų, ansamblio prognozuojama klasė nustatoma pagal kiekvieno modelio *softmax* išvesčių kiekvienai atskirai klasei vidurkį, o galutinis rezultatas skaičiuojamas pagal klasių tikimybių skirstinių vidurkių maksimumą (žr. 1 Priedą). *NN-Ensemble-CNN* metodikoje paskutinio neuroninio tinklo sluoksnio išvestys jungiamos į bendrą vektorį, kuris toliau naudojamas nedideliame, keturių sluoksnių neuroniniam tinklui mokyti. Eksperimentų metu abu metodai pasirodė efektyviau nei atskirai klasifikuojantys baziniai modeliai (žr. 2 Priedą).

Pastebima, jog literatūroje kolektyvai dažniausiai sudaromi aprašytais klasikiniiais metodais naudojant skirtingas modelių kombinacijas ar tik minimaliai keičiant mokymosi modelių struktūras. Tačiau randama ir bandymų modifikuoti pačių ansamblių sudarymo metodikų bazinius žingsnius. Keletas šių patobulintų tradicinių ansamblių sudarymo technikų aptariama toliau.

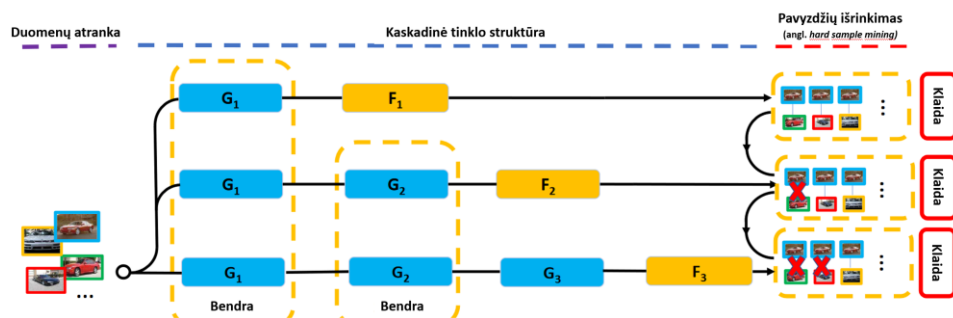
Jameel'as et al. [73] pasiūlė adaptyvaus CNN ansamblių karkaso kūrimo techniką multispektrinių vaizdų klasifikacijos kontekste naudojant interaktyvų mokymąsi (angl. *online learning*). Autorių sukurtas karkasas susideda iš penkių modulių, tarp kurių – dinaminio ansamblio klasifikatorius (angl. *Dynamic Ensemble Classifier, DEC*), gebantis dinamiškai koreguoti naujo pavyzdžio klasifikavimo svorius atskiriems modeliams (agregavimui naudojamas daugumos balsavimas) (žr. 3 Priedą). *DEC* iš esmės yra prie duomenų prisitaikantis modelių kolektyvas, kurio kiekvienas narys yra atsakingas už atskirų spektro juostų (atskirų įvesties kanalų) klasifikavimą. Jei tam tikra ansamblio klasifikuojama duomenų rinkinio dalis *DEC* nepasiekia tam tikros ribinės vertės (klasifikavimo tikslumas < 50 %), tuomet šie pavyzdžiai agreguojami *K*-vidurkių metodu (angl. *K-means*) ir priskiriami naujai hipotetinei klasei *X* – galimai naujai spektro juostai. Atrinktais pavyzdžiais mokomas naujas CNN modelis perduodamas į *DEC* kolektyvą. Taip sudaromas įvairialypis ansamblis, gebantis efektyviai klasifikuoti skirtingus duomenis.

Dėl tam tikrų ribojimų *boosting* naudojimui CNN architektūroms (pvz., dėl mokymosi paketais galimas persimokymas, nes *boosting* CNN naudojamas kaip tam tikrų neuronų pasirinkimas ir aktyvacija kiekvienoje iteracijoje), *Han et al.* [74] pasiūlė inkrementinio *boosting* idėją. Autoriai realizavo metodiką, kuomet, naudojant *AdaBoost*, informacija apie mokymo duomenis yra kaupiama per kelis paketus ir tik tuomet perduodama CNN: kiekvienoje iteracijoje pasirenkamos tik tam tikros CNN išvestys (paskutinio paslėpto sluoksnio (angl. *hidden layer*) savybės (angl. *features*)) kombinuojamos su anksčiau pasirinktomis išvestimis, o parametrai atnaujinami taip, kad dažniausiai pasirenkamos išvestys turėtų didžiausius svorius.

Moghimi et al. [75] *boosting* inkorporavo į giliojo CNN mokymosi architektūros nuostolių funkciją. Pagal autorių pasiūlytą karkasą galima ruošti ir heterogeniškus modelius, o rezultatai parodė, jog tokie ansambliai pasiekia SOTA efektyvumus populiariuose vaizdų klasifikacijos uždaviniuose.

Brahimi et al. [76] taip pat pasiūlė inkorporuotą *boosting* struktūrą mokant giliuosius CNN. Autoriai rekurentinės (angl. *recurrent*) architektūros CNN sustiprina pridėdami papildomus blokus (angl. *boosted blocks*), sudarytus iš skirtingo skaičiaus konvoliucinių sluoksnių (angl. *boosted Convolutional Layer, BCL*) (žr. 4, 5 Priedus). Šie blokai pagerina viso tinklo prognozės išraišką naudojant netiesinę kelių poslinkių aktyvacijos funkciją (angl. *multi-bias nonlinear activation function*) [77] bei sujungtus *ReLU* (angl. *rectified linear unit*) blokus svarbiausios informacijos (po konvoliucinių sluoksnių apdorojimo) išsaugojimui.

BIER (angl. *boosting independent embeddings robustly*) [78] metodikoje naudojami didelio neuroninio tinklo išmoktų savybių erdvės atvaizdžių vektoriai (angl. *embeddings*) atskiriems modeliams parengti pasitelkus *boosting*. *HDC* (angl. *hard-aware cascaded embedding*) [79] ansambliai sudaromi tarsi kaskada (žr. 3 pav.) iš skirtingo sudėtingumo modelių, o kiekvienas mokymo imties pavyzdys naudojamas atnaujinti tik tų modelių, kurie pavyzdį laiko sudėtingu, svorius.



3 pav. Kaskadinis modelių mokymas [79]

[80] pristatoma CNN ansamblių parengimo metodika baigtiniam (angl. *end-to-end*) mokymuisi įgyvendinti. Autoriai prijungia nedideles neuroninių tinklų architektūras prie trečiojo sluoksnio neuroninio tinklo bloko (blokas papildomai skaidomas pagal matricos kanalus, gaunami baziniai modeliai nuo 1 iki L) ir prie kiekvienos ketvirtojo bloko išvesties (gaunami baziniai modeliai $L+1$ iki $2L$). Taip pasiekiami ne tik geresni segmentavimo rezultatai, bet ir sutaupoma skaičiavimo resursų, nes net 98 % slankiojo kablelio operacijų (angl. *floating-point operations*) atliekama būtent blokuose, kuriais baziniai modeliai dalinasi. Galutinė ansamblio prognozė sudaroma sujungus visų bazinių modelių savybių reprezentacijos sluoksnius.

1.3. Ansamblių efektyvinimas

Modernius mašininio mokymosi algoritmus, ypač neuroninius tinklus, sudėtingus tiek architektūra, tiek optimizavimo ypatybėmis (pvz., *GPT-3* [81] kalbos modelis turi 175 milijardus parametru, *Google AI Language Transformer* [82] – vieną trilijoną) ir ypač jų ansamblius gali būti sudėtinga pritaikyti daug spėjimų reikalaujantiems panaudojimo atvejams. Jei rezultatus reikia (o dažniausiai taip ir yra) gauti per kelias sekundes ar dar greičiau, šis procesas tampa dar labiau komplikotas, o norint pasiekti optimalius rezultatus naudojant ansamblius, tokių architektūrų produkcinis pateikimas dažnu atveju nėra galimas. Išmaniųjų įrenginių amžiuje būdai, kuriais modelių ansambliai sumažinami (suspaudžiami) neprarandant (ar prarandant tik nedidelę dalį) prognozavimo efektyvumo, yra svarbūs tiek tyrėjams, tiek verslui. Tokių technikų pavyzdžiai [83] – genėjimas (angl. *pruning*), kuomet modelių svoriai, nedarantys įtakos spėjimo galiai, yra panaikinami (skirtingai nuo išmetimo technikos, kai naudojant ne visus parametrus siekiama išvengti persimokymo (angl. *overfitting*)), kvantavimas (remiamasi svorių tam tikruose sluoksniuose pasiskirstymu, artimu normaliajam, kai svoriai arba išdalinami į vienodo dydžio intervalus arba tiesiog pakeičiamas skaičiaus tikslumas), faktorizacija (taikomi tradiciniai ir specifiniai matricų faktorizavimo algoritmai sluoksnių parametru mažinimui), taip pat neuroninės architektūros paieška (angl. *Neural Architecture Search*) ar ansamblio žinių distiliavimas (angl. *knowledge distillation*) (žr. 2 lentelę). Toliau šiame poskyryje bus aptariama būtent žinių distiliavimo technika.

2 lentelė. Modelių greitinimo, suspaudimo metodikos [84]

Metodika	Aprašymas	Taikymai
Parametrų atmetimas ir dalinimasis (angl. <i>pruning and sharing</i>)	Sumažinama perteklinių parametrų, nedarančių įtakos prognozių efektyvumui	Konvoliuciniai ir visiškai sujungti sluoksniai
Šaltiniai		
Comparing Biases for Minimal Network Construction with Back-Propagation [85] Second Order Derivatives for Network Pruning: Optimal Brain Surgeon [86] Compressing Deep Convolutional Networks using Vector Quantization [87] Compressing Neural Networks with the Hashing Trick [88] Learning both Weights and Connections for Efficient Neural Network [89] Data-free Parameter Pruning for Deep Neural Networks [90] BinaryConnect: Training Deep Neural Networks with binary weights during propagations [91] Deep Learning with Limited Numerical Precision [92] Learning Structured Sparsity in DeepNeural Networks [93] Less is More: Towards Compact CNNs [94]	Fast ConvNets Using Group-Wise Brain Damage [95] Neural Networks with Few Multiplications Deep neural networks are robust to weight binarization and other non-linear distortions [96] XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks [97] Quantized Convolutional Neural Networks for Mobile Devices [98] Pruning Filters For Efficient ConvNets [99] Soft Weight-Sharing For Neural Network Compression [100] Loss-aware Binarization of Deep Networks [101] Trained Ternary Quantization [102] Towards the Limit of Network Quantization [103]	
Metodika	Aprašymas	Taikymai
Faktorizacija	Naudojamos matricų dekompozicijos metodikos informatyvių parametrų gavimui	Konvoliuciniai ir visiškai sujungti sluoksniai
Šaltiniai		
Learning Separable Filters [104] Predicting Parameters in Deep Learning [105] Low-rank matrix factorization for Deep Neural Network training with high-dimensional output targets [106] Exploiting Linear Structure Within Convolutional Networks for Efficient Evaluation [107]	Speeding up Convolutional Neural Networks with Low Rank Expansions [108] Speeding-up Convolutional Neural Networks Using Fine-tuned CP-Decomposition [109] Convolutional neural networks with low-rank regularization [110]	
Metodika	Aprašymas	Taikymai
Perkelti / suspausti konvoliuciniai filtrai	Parametrų išsaugojimui kuriami atskirai sudaromų struktūrų konvoliuciniai filtrai	Konvoliuciniai sluoksniai

Šaltiniai		
Group Equivariant Convolutional Networks [111]	Exploiting Cyclic Symmetry in Convolutional Neural Networks [114]	Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning [115]
Doubly Convolutional Neural Networks [112]		
Understanding and Improving Convolutional Neural Networks via Concatenated Rectified Linear Units [113]		
Multi-Bias Non-linear Activation in Deep Neural Networks [77]		
SqueezeDet: Unified, Small, Low Power Fully Convolutional Neural Networks for Real-Time Object Detection for Autonomous Driving [116]		
Metodika	Aprašymas	Taikymai
Žinių distilavimas (angl. <i>knowledge distillation</i>)	Kompaktiško neuroninio tinklo mokymas distilijuojant žinias iš modelio-mokytojo	Konvoliuciniai ir visiškai sujungti sluoksniai
Šaltiniai		
Model compression [6]	Face model compression by distilling knowledge from neurons [120]	Net2net: Accelerating learning via knowledge transfer [121]
Do deep nets really need to be deep? [117]		
Distilling the knowledge in a neural network [5]		
Fitnets: Hints for thin deep nets [118]		
Bayesian dark knowledge [119]		
	Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer [122]	

Žinių distiliavimas – tai kompaktiškesnio algoritmo (mažesnis modelis, modelis-studentas) mokymas naudojant distiliuotas žinias, perimtas iš didelio modelio ar modelių ansamblio (sudėtingo modelio, modelio-mokytojo) [123]. Naudojantis šiomis jau išmoktomis žiniomis galima efektyviai paruošti nedidelį modelį reikšmingai nepakenkiant baziniam prognozavimo efektyvumui.

Svarbu paminėti, jog prieš distilijuojant, t.y. bandant modelių-mokytojų išgautą informaciją perkelti modeliams-mokiniam, patys didieji modeliai (ansambliai) turėtų pasiekti aukštą, idealiu atveju – SOTA – našumą. Kompleksiški ansambliai gerus prognozavimo rezultatus dažniausiai ir pasiekia dėl to, jog yra persotinami informacija iš mokymosi rinkinio. Todėl prieš distilijuojant modelio (ansamblio) žinias reiktų įsitikinti, jog jis geba pakankamai gerai apibendrinti informaciją ir efektyviai prognozuoja dar nematytų pavyzdžių rinkinius.

Klasifikavimo uždavinių sprendimui naudojamo sudėtingo modelio (pvz., gilus neuroninio tinklo) pagrindinis tikslas – vidutinės logaritminės tikimybės teisingai prognozuoti klasę maksimizavimas [67]. Tačiau modelis spėjimo procese pateikia ir daugiau išmoktos informacijos. Klasifikuojant lengvojo automobilio pavyzdį neuroninis tinklas, skirtas atpažinti transporto priemones, paskutiniame sluoksnyje (*softmax* klasifikatorius) galėtų teigti 80 % tikimybę, jog teikiamas pavyzdys yra automobilis, 19,8 % – jog sunkvežimis, 0,2 % – jog laivas. Šia sprendimo entropija ir galima remtis distilijuojant išmoktas žinias [5].

Kaip jau minėta, vienas paprasčiausių būdų mašininio mokymosi algoritmų efektyvumui gerinti – skirtingų modelių mokymas ir jų išvesčių agregavimas. Tačiau ši strategija brangiai kainuoja tiek resursų, tiek laiko prasme ir dažnai nėra tinkama pateikimui produkcinėje aplinkoje. *Caruana et al.*

[6] pademonstravo, jog bendrai ansamblyje įgytos žinios gali būti tarsi suspaustos į mažesnę modelį. Pagrindinė autorių idėja buvo greito ir kompaktiško modelio (neuroninio tinklo su paslėptu sluoksniu) naudojimas efektyvesnio, tačiau lėtesnio ir didesnio algoritmo (modelių kolektyvo) išmoktai funkcijai aproksimuoti. Remiamasi gerai žinoma dirbtinių neuroninių tinklų savybe – universalia aproksimacija, t.y. tinklas su pakankamai dideliu paslėptu sluoksniu ir pakankamai duomenų gali aproksimuoti bet kokią funkciją iki pasirenkamo tikslumo. Autoriai suspaustą modelį ruošė ne ant pradinės duomenų imties (kuri paprastai yra nedidelė), tačiau ant pseudoduomenų, generuotų pagal marginalųjį skirtinį naivaus Bajeso (angl. *Naive Bayes Estimation*) [69] algoritmu, bei sukurtą procedūrą *MUNGE*, generuojančią pavyzdžius pagal neparametrinę skirstinio aproksimaciją. Pasiękti rezultatai įrodė modelių kompresijos našumą: aštuoniuose dvejetainės klasifikacijos užduotyse modelis pasirodė vidutiniškai geriau už kolektyvinius sprendimo algoritmus, o vidutinis laikas klasifikacijai (10 tūkst. stebėjimų) bei modelių užimamos atminties kiekis sumažėjo atitinkamai virš 2400 ir virš 6000 kartų.

Modelių distiliavimo idėją išplėtojo *Hinton et al.* [5]. Naudojant suspaudimo metodiką autoriai pasiekė gerų rezultatų mokant neuroninius tinklus *MNIST* [124] (žr. 3 lentelę) duomenų rinkiniu, ir pademonstravo, jog galima reikšmingai pagerinti komercinėje aplinkoje (automatiniam kalbos atpažinimui) taikyto modelio efektyvumą distiliuojant kolektyvo išmoktas žinias į vieną, kompaktiškesnę modelį. Pagal autorių pateiktą metodiką, pasinaudojant modelio-mokytojo žiniomis klasifikavimo probleminės užduoties atveju modeliui-studentui paruošiamas perkeliama rinkinys (angl. *transfer set*). Paprasčiausios algoritmo formos atveju, modelio-mokytojo žinios į distiliuotą modelį perkeliama pavyzdžių klasifikacijos žymenims priskiriant minkštųjų žymenų skirstinį (angl. *soft target distribution*). Straipsnyje *Hinton et al.* taip pat pristato problemos, kuomet ansambliai, net ir kuriami panaudojant įvairias lygiagretinimo technikas, gali reikalauti per daug resursų dėl pačių modelių dydžio arba labai didelių duomenų imčių, sprendimo idėją.

3 lentelė. Testavimo rezultatai *MNIST* rinkiniui [5]

Modelis	Architektūra	Testavimo klaidų skaičius	Temperatūros parametras
Modelis-mokytojas (fiksotos tikslo įvestys (angl. <i>hard targets</i>))	Du visiškai sujungti sluoksniai su 1200 neuronų	67	1
Modelis-studentas (fiksotos tikslo įvestys (angl. <i>hard targets</i>))	Du visiškai sujungti sluoksniai su 800 neuronų	146	1
Distiliuotas modelis (fiksotos ir tikimybinės tikslo įvestys (angl. <i>hard and soft targets</i>))	Du visiškai sujungti sluoksniai su 800 neuronų	74	20

Autoriai parodo, jog turint dideles duomenų imtis galima naudoti modelius-specialistus, kuriais koncentruojamasi į dažnai sumaišomų (pvz., klystama tarp žymenų porų) klasių poaibius. Tai gali padėti optimizuoti resursus, reikalingus galutinio ansamblio parengimui. Pagrindinis modelių specialistų trūkumas – per didelis prisitaikymas, persimokymas – sprendžiamas naudojant švelniųjų žymų reprezentacijas. Ansamblis sudaromas iš modelio-apibendrintojo (mokomo ant visų duomenų ir žymų) ir modelių-specialistų, mokytų tik ant dažnai maišomų klasių poaibio (kitos klasės priskiriamos „šiukšlių“ išvesčiai (angl. *dustbin, void class*)). Tam, jog būtų galima išvengti persimokymo, modeliams-specialistams priskiriami modelio-apibendrintojo svoriai, tuomet specialistai mokomi naudojant pusę modeliui-specialistui priskirtų žymų duomenų rinkinio ir pusę

atsitiktinai parinktų pavyzdžių iš pilno rinkinio. Po mokymo nenaudojamos klasės logitas (angl. *logit*) didinamas logaritmine perteklinai modeliui-specialistui priskirtų klasių proporcija. Pačios klasės specialistams priskiriamos klasterizuojant modelio-apibendrintojo prognozių kovariacijų matricą. Tokio klasių atrinkimo pavyzdys pateikiamas 4 pav. toliau:

JFT 1: Tea party; Easter; Bridal shower; Baby shower; Easter Bunny; ...
JFT 2: Bridge; Cable-stayed bridge; Suspension bridge; Viaduct; Chimney; ...
JFT 3: Toyota Corolla E100; Opel Signum; Opel Astra; Mazda Familia; ...

4 pav. Modeliams-specialistams priskirtų klasių pavyzdžiai [5]

Vienas šios strategijos privalumų yra daug greitesnis modelių-specialistų mokymas lyginant su modelių-apibendrintoju, taip pat modelių-specialistų mokymosi procesą galima lygiagretinti. Eksperimentų metu buvo mokomas 61 modelis-specialistas, klasifikuojama 301 klasė: pastebimas 1,1 % absoliutus (4,4 % santykinis) testavimo rinkinio klasifikavimo tikslumo pagerėjimas. Taip pat svarbu pabrėžti, jog kuo daugiau specialistų klasifikuoja priskirtą klasę, tuo didesnis tikslumo didėjimas pastebimas (žr. 6 Priedą). Dėl to, jog modelių-specialistų mokymas nepriklauso nuo kitų specialistų ruošimo, galimai didesnio modelių kiekio parengimas tam tikrų uždavinių kontekstuose leistų reikšmingai pagerinti prognozių rezultatus. Tačiau naudojant šią strategiją spėjimo procesas tampa ne toks akivaizdus ir pavyzdžio priskyrimui į tam tikrą klasę reikalingas atskiras algoritmas. Pirmiausia naudojantis modeliu-apibendrintoju randamos n labiausiai tikėtinos klasės k , ir tuomet, žinant aktyvių modelių-specialistų poaibį (kur k kertasi su specialistui priskirtų klasių aibe), minimizuojamas pilnasis tikimybių skirstinys visoms klasėms pagal santykinę entropiją (angl. *Kullback-Leibler divergence*) [67].

Kalbos atpažinimo užduoties kontekste [5] autoriai parodė, jog naudojant distiliavimo strategiją ansamblio, suspausto į vieną modelį, efektyvumas reikšmingai lenkia tokio paties dydžio tiesiogiai mokyto modelio efektyvumą. Eksperimentų metu naudotas gilus neuroninis tinklas Markovo modelio (angl. *Hidden Markov Model*) būsenų tikimybiniam skirstiniui iš trumpų akustinių pavyzdžių gauti. Tinklo architektūra – 8 paslėpti sluoksniai po 2560 *ReLU* blokus, *softmax* išvestis su 14000 žymų, iš viso apie 85 mln. parametrų. Diversifikuotam modelių ansamblio kūrimui naudota dešimt identiškų architektūrų tinklų su skirtingais (atsitiktinai priskirtais) pradiniais svorių variantais, agregavimui naudojamas tikimybių vidurkis. 4 lentelėje toliau galima matyti, jog distiliuotas modelis geba išsaugoti daugiau nei 80 % klasifikacijos pagerėjimo lyginant pradinį modelį ir modelių ansamblį.

4 lentelė. Testavimo rezultatai kalbos atpažinimo užduočiai [5]

Sistema	Testavimo tikslumas (angl. <i>test frame accuracy</i>)	Klaidingų žodžių dažnis (angl. <i>word error rate, WER</i>)
Bazinis lygmuo (angl. <i>baseline</i>)	58,9 %	10,9 %
Ansamblis (<i>10xEnsemble</i>)	61,1 %	10,7 %
Distiliuotas modelis	60,8 %	10,7 %

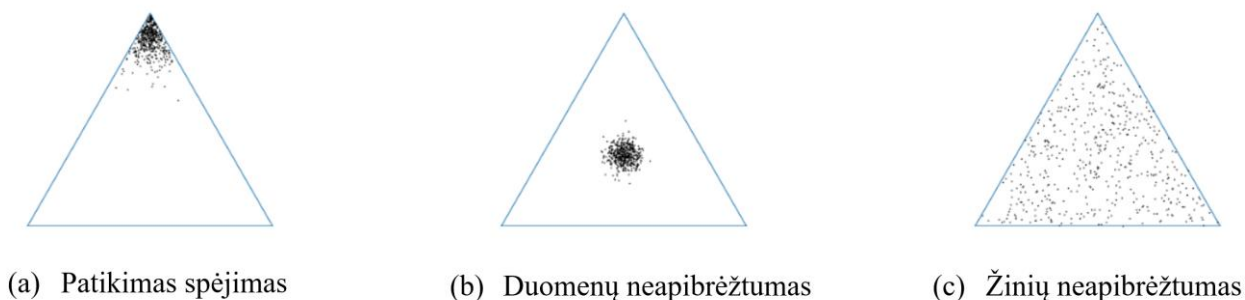
Panašias idėjas mokslininkai pritaikė ir kitų tipų neuroniniams tinklams. *Wang'as et al.* [125] apibūdina *a posteriori* prognozių skirstinio Monte Karlo aproksimacijos (Bajeso neuroninių tinklų [69]) distiliavimą į kompaktiškesnę formą – gilųjį neuroninį tinklą. Autoriai moko modelį-studentą modelio-mokytojo (Bajeso neuroninių tinklų ansamblio) išmokto skirstinio aproksimacijai. Idėja panaši į [5]: vienos iteracijos metu mokomas modelis-mokytojas (naudojant *SGLD* (angl. *stochastic*

gradient Langevin dynamics) [126] algoritmą), o modelio-mokinio svoriai atnaujinami naudojant MAP (angl. *maximum a posteriori estimation*) išvestis su tam tikra normaliaja paklaida.

Wong'as et al. [127] automatinio kalbos atpažinimo problematikos kontekste pasiūlė mokytojo-studento tipo mokymo metodiką, tinkamą realaus laiko aplikacijoms, panaudojant įvairias hipotezių kombinavimo technikas, pavyzdžiui MBR (angl. *Minimum Bayes Risk*). Modeliai-studentai mokomi pagal tiesiogines ansamblio išvestis, o atgalinė sklaida (angl. *backpropagation*) vykdoma visam kolektyvui.

Malinin'as et al [128] teigia, jog [5, 125, 127] siūlomi metodai, kurie iš esmės apibūdina vieno taškinio sąlyginio klasių skirstinio įverčio gavimą naudojant sąlyginių skirstinių ansamblį, neišsaugo informacijos apie ansamblio įvairovę (angl. *diversity*), taigi nelieka ir prielaidų žinių neapibrėžtumui (pvz., bendrajai atsitiktinių dydžių informacijai (angl. *mutual information*) [129]) nusakyti. Savo darbe autoriai tiria visų ansamblio prognozių skirstinio, o ne tik prognozių vidurkių, modeliavimą. Teigiama, jog toks ansamblio skirstinių distiliavimas išsaugo ne tik žinias, padedančias ansambliai geriau klasifikuoti dirbant išvien, tačiau ir pačią skirstinių informaciją.

Pagal autorių metodiką, pradinis sukurtas modelių ansamblis turėtų pasižymėti elgesiu, nurodytu 5 pav. toliau.



5 pav. Modelių ansamblio spėjimo savybės [128]

Konkrečiau, kiekvienam pradines klasifikacijos klases atitinkančiam testavimo imties pavyzdžiui ansamblis turėtų pateikti a arba b tipo rezultatus. Jei pateikiamas visiškai kitoks pavyzdys, t.y. ansamblis neturi žinių, jog pavyzdžio įvestys reprezentuoja būtent nurodomą klasę, ansamblio prognozės turėtų skirtis tarp atskirų modelių. Kaip tik šias žinių spragas ir galima aproksimuoti bendrosios informacijos matu (1).

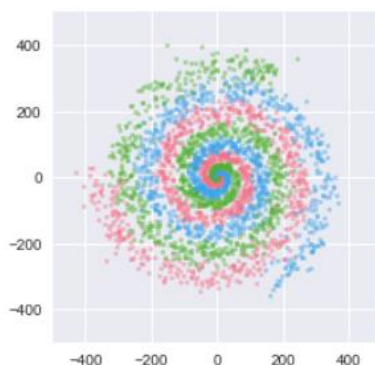
$$\underbrace{\mathcal{I}[y, \theta | \mathbf{x}^*, \mathcal{D}]}_{\text{Žinių neapibrėžtumas}} = \underbrace{\mathcal{H}[\mathbb{E}_{\mathbf{p}(\theta|\mathcal{D})}[P(y|\mathbf{x}^*, \theta)]]}_{\text{Visas neapibrėžtumas}} - \underbrace{\mathbb{E}_{\mathbf{p}(\theta|\mathcal{D})}[\mathcal{H}[P(y|\mathbf{x}^*, \theta)]]}_{\text{Tikėtinas duomenų neapibrėžtumas}} \quad (1)$$

čia pirmasis narys apibūdina ansamblio (narių skirstinių vidurkių), antrasis – vidutinę visų narių skirstinių entropiją.

Viso ansamblio skirstinio entropija turėtų būti didelė, kai ansamblis nėra tikras savo sprendimu (galimai dėl klasių panašumo arba nematyto tipo pavyzdžių). *A posteriori* skirstinių vidurkio entropijos ir *a posteriori* skirstinių entropijos vidurkio skirtumas nebus lygus nuliui tik tuo atveju, kai modelių prognozės skirsis. Jei šis skirtumas atsiranda dėl klasių persidengimo, kiekvieno ansamblio nario prognozės skirstinio entropija bus aukšta (žr. 5 pav. b), ir tokiu atveju, pagal (1), bendroji informacija bus nedidelė. Jei pats pavyzdys bus nematytas (modeliai negeba tiksliai apibūdinti

pavyzdžio pagal mokymo imties duomenis), ansamblio narių prognozių skirstiniai vienas nuo kito žymiai skirsis, o pats entropijų vidurkis galimai bus mažesnis ir tokiu atveju bus gaunama aukšta žinių neapibrėžtumo reikšmė.

Eksperimentų metu buvo naudojamas rinkinys, pavaizduotas 6 pav. toliau:



6 pav. Klasifikavimui naudojamas 3 klasių duomenų rinkinys [128]

Šis trijų klasių (tūkstantis duomenų pavyzdžių kiekvienai) rinkinys naudojamas šimto neuroninių tinklų (su atsitiktiniu svorių inicializavimu) ansamblio mokymui, taip pat paruošiamas ir mažesnis dešimties tinklų ansamblis. Testavimo imties klasifikavimo tikslumas nurodytas 5 lentelėje toliau:

5 lentelė. Testavimo rinkinio klasifikavimo paklaidos [128] (EnD – tradicinis, EnD^2 – siūlomas distiliavimas)

Modelių skaičius	Atskiras modelis	Ansamblis	EnD	EnD^2
10	13,21	12,63	12,57	12,52
100		12,37	12,27	12,47

Pagal rezultatus matomas ansamblių efektyvumo pagerėjimas lyginant su vienu neuroniniu tinklu, o šimto ir dešimties narių kolektyvų rezultatai skiriasi nežymiai. Naudojant tradicinę distiliavimo strategiją [5] išlaikoma didžioji dalis ansamblio informacijos, o siūlomu algoritmu gaunami tik šiek tiek prastesni rezultatai, tačiau galima atskirti žinių spragas pagal (1) formulę.

Distiliavimas taip pat išbandytas ant standartinių *CIFAR-10*, *CIFAR-100* [130] bei *TinyImageNet* [131] paveikslėlių duomenų rinkinių naudojant *VGG-16* [132] neuroninio tinklo architektūrą. Rezultatai šiems rinkiniams pateikiami 6 lentelėje toliau:

6 lentelė. Testavimo rinkinio klasifikavimo paklaidos vaizdiniais duomenims [128] (PRR – spėjimo tikslumas (angl. *prediction rate*), ERR – klaidų lygis (angl. *error rate*), AUX – agreguoti duomenys)

Duomenų rinkinys	Kriterijus	Individualus modelis	Ansamblis	EnD	EnD^2	EnD_{+AUX}	EnD^2_{+AUX}
<i>C10</i>	ERR	$8,0 \pm 0,4$	$6,2 \pm NA$	$6,7 \pm 0,3$	$7,3 \pm 0,2$	$6,7 \pm 0,2$	$6,9 \pm 0,2$
	PRR	$84,6 \pm 1,2$	$86,8 \pm NA$	$84,8 \pm 0,8$	$85,3 \pm 1,1$	$85,1 \pm 0,1$	$85,7 \pm 0,3$
<i>C100</i>	ERR	$30,4 \pm 0,3$	$26,3 \pm NA$	$28,0 \pm 0,4$	$27,9 \pm 0,3$	$28,2 \pm 0,3$	$28,0 \pm 0,5$
	PRR	$72,5 \pm 1,0$	$75,0 \pm NA$	$73,1 \pm 0,5$	$73,7 \pm 0,7$	$74,0 \pm 0,3$	$74,0 \pm 0,2$
<i>TIM</i>	ERR	$41,8 \pm 0,6$	$36,6 \pm NA$	$38,3 \pm 0,2$	$37,6 \pm 0,2$	$38,5 \pm 0,3$	$37,3 \pm 0,5$
	PRR	$70,8 \pm 1,1$	$73,8 \pm NA$	$72,2 \pm 0,2$	$73,1 \pm 0,1$	$72,6 \pm 1,3$	$72,7 \pm 1,1$

Pagal gautus rezultatus matoma, jog siūloma ansamblių distiliavimo strategija geriau pasirodo *CIFAR-100* bei *TinyImageNet* rinkinių atveju, o teisingo klasifikavimo rodikliu arba sutinka, arba pasirodo geriau už tradicinį distiliavimą. Autoriai teigia, jog tai galimai reiškia, jog siūlomas distiliavimas išlaiko daugiau informacijos apie ansamblio žinias bei jo įvairovę.

Ba et al. [117] suspaudžia SOTA gilaus mokymosi modelį (taip pat ir modelių ansamblių) mokant mažesnę modelį atkartoti modelio-mokytojo išvestis. Autoriai modelius ruošia ant *TIMIT* [133] bei *CIFAR-10* [130] duomenų imčių, o modelio-studento mokymo rinkiniui sudaryti naudoja modelio-mokytojo logitų išvestis. Tam, jog modelis-studentas greičiau išmoktų modelio-mokytojo vidinį informacijos modeliavimą, papildomai modeliui-studentui pridedamas ribojantis tiesinis sluoksnis su k tiesinės funkcijos blokų po įvesties sluoksniu. Rezultatai rodo, jog su dešimt kartų parametų kiekiu mažesni modeliai-studentai pasiekia tik apie 1,4-5,1 % mažesnę tikslumą su *TIMIT*, 3,2-3,4 % su *CIFAR-10* duomenų rinkiniais.

Romero et al. [118] modelio-studento (gilesnio už modelį-mokytoją, tačiau su mažiau parametų vidiniuose sluoksniuose) mokymo bei galutinio prognozavimo gerinimui panaudojo ne tik galutines, bet ir tarpines modelio-mokytojo išvestis. Autoriai sudarė papildomus parametrus (angl. *intermediate-level hints*) tam, jog modelio-studento paslėptas sluoksnis gebėtų mokytis iš modelio-mokytojo paslėpto sluoksniu išvesčių. Rezultatai parodė, jog daugiau nei tris kartus mažesnis modelis klasifikavimo tikslumu pasirodo ~1,5 % geriau nei didesnis SOTA modelis-mokytojas. Vienas šio metodo trūkumų – toks žinių perkėlimas iš esmės galimas tik homogeniškiems modeliams.

Luo et al. [120] pristatė žinių perkėlimą naudojant kompaktiškesnes nei žymenų prognozavimo skirstinio sluoksniu (pvz., *softmax*), tačiau tiek pat informacijos išsaugančias neuronų išvestis paslėptuose tinklo sluoksniuose. Autoriai naudojo „informatyvių“ neuronų pasirinkimą paskutiniame paslėptame sluoksnyje interpretuojant neuronus tarsi grafą ir atsirenkant juos pagal energijos funkciją, randančią įvairius bei nekoreliuotus neuronus. Rezultatai parodė, jog veidų atpažinimo užduoties kontekste modelis-studentas pasiekia geresnius rezultatus su beveik 52 kartais mažiau parametų ir 90 kartų greitėjančiu spėjimo procesu.

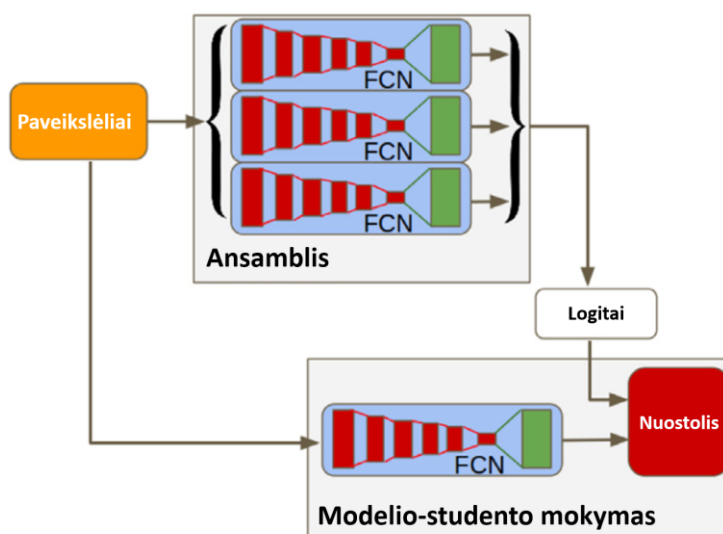
[121] autoriai pristatė algoritmą, kaip greitai perduoti žinias iš vieno neuroninio tinklo kitam. Naudojantis funkcinės priklausomybes išsaugančių (angl. *function-preserving transformations*) neuroninių tinklų architektūrų keitimo koncepcija autoriai inicializuoja modelius-studentus taip, jog jie reprezentuotų kuo panašesnę funkciją į modelį-mokytoją, o tolesnis mokymas vykdomas įprastu būdu.

Aloysius et al. [134] CNN mokymo laiko mažinimui pristatė ansamblinę architektūrą, kuomet baziniai ansamblio modeliai yra mokomi atpažinti atskiras klases, suskirstytas pagal panašumus hierarchiniu klasterizavimo algoritmu (žr. 7 Priedą). Taip pat autoriai pristato naują tikimybinį ansamblio klasifikatorių ansamblio išvesčių vertimui į klasės žymenį. Metodika žymiai sumažinami reikalingi skaičiavimo resursų kiekiai net didelėms mokymo imtims.

Taikant *ABE* [135] (angl. *attention-based ensemble*) modeliai-studentai tiesiogiai nejungiami prie modelio-mokytojo išvesčių, o naudojami tam tikri dėmesio (angl. *attention*) moduliai, užtikrinantys diversifikuotų atvaizdžių funkcijų mokymą (sudaromos dėmesio kaukės skirtingoms savybių erdvės vietoms).

Panašiai ir *Zagoruyko et al.* [122] CNN modelius-studentus moko atkartoti gerą efektyvumą pasiekiančių modelių-mokytojų dėmesio atvaizdžius (angl. *attention maps*). Mokant modelius-studentus nuostolių funkcija keičiama taip, jog būtų atsižvelgiama ne tik į teisingus klasifikavimo rezultatus, bet ir į teisingą dėmesio atvaizdžių (pagal modelį-mokytoją) sudarymą. Atvaizdžiai skaičiuojami priklausomai nuo architektūros: CNN – priešpaskutiniame mažinimo (angl. *downsample*) bloke, o autorių naudojamame modelyje ResNet (angl. *Residual Network*) [136] – kiekvienoje sluoksnių grupėje. Eksperimentai parodo, jog taip mokyti modeliai pasiekia geresnius rezultatus nei atskiri modeliai-studentai, o pritaikant žinių distiliavimą [5] pasiekiamas modeliui-mokytojui artimas tikslumas.

Holliday et al. [137] pasiūlė metodiką, kuomet semantinio segmentavimo CNN modelio suspaudimas yra pakankamas naudojimui atsako realiu laiku (angl. *real-time*) reikalaujantiems panaudojimo atvejams. Šiam suspaudimui atlikti autoriai naudoja [5] modelių distiliavimo strategiją ją pritaikant heterogeninių ansamblių suspaudimui. Ansamblis sudaromas iš kelių skirtingų gylių (50, 101, 152) ResNet [136], taip pat *AlexNet* [138], *VGG-16* [132], *LeNet* [139] modelių. Keičiama kai kurių modelių struktūra, pavyzdžiui ResNet dekonvoliuciniam sluoksniui priskiriama dvinarės interpoliacijos funkcija, o svoriai iššaldomi (mokymosi koeficientas nustatomas į 0). Pastebima, jog atskirai mokomas modelis-studentas pasirodo žymiai prasčiau nei modelis-mokytojas ar distiliuotas modelis. Tiek distiliuoto, tiek pilno (modelio-mokytojo) modelių rezultatai yra pakankamai artimi, o tai galimai reiškia, jog iš modelio-mokytojo buvo perduota pagrindinė išmoktų žinių dalis.



7 pav. Bendroji modelių mokymo schema [137]

Autoriai pirmiausia moko segmentavimo modelį ant visos duomenų imties, o mokymas tęsiamas su perkeliama imtimi (kaip žymenys naudojami ansamblio logitų vektoriai) nuostolių funkcijai taikant Euklido atstumą (žr. 7 pav.) Eksperimentinėje aplinkoje taip paruošti modeliai buvo išbandyti su *SBD* [30], išplėstu *PASCAL VOC* [28] bei *SIFT Flow* [35] duomenų rinkiniais, ansamblio išvesčių agregavimui naudotas aritmetinis skirstinių vidurkis, o patys ansambliai sudaryti iš kelių modelių (*FCN* [8], ResNet, *LeNet*), darančių vienodą įtaką galutiniam sprendimui. Pradiniai modelių svoriai nustatyti iš prieinamų duomenų rinkinių mokymo rezultatų: ResNet bei *LeNet* naudojami ImageNet rinkinio [140] svoriai.

Rezultatai *PASCAL VOC* rinkiniu pateikiami 7 lentelėje toliau:

7 lentelė. PASCAL VOC segmentavimo rezultatai atskiriems modeliams, ansambliui bei distiliuotam modeliui [137]

Modeliai	IOU vidurkis (%)	Taškų tikslumas (angl. <i>pixel accuracy</i> , %)	Laikas (sek.)
VGG-16 (32s)	63,6	90,5	0,1
VGG-16 (16s)	65,0	91,0	0,1
VGG-16 (8s)	65,5	91,2	0,1
GoogLeNet	54,7	88,4	0,06
ResNet-50	60,6	90,4	0,06
ResNet-101	64,0	91,2	0,11
ResNet-152	65,0	91,5	0,11
Ansamblis	67,7	92,1	1,29
Suspaustas Resnet-152	66,1	91,7	0,11

Stebima, jog nors distiliuotas modelis ir nepasiekia ansamblio segmentavimo tikslumo, tačiau gauti rezultatai lenkia visus atskirų modelių rodiklius, o spėjimo metu sugaištama dešimt kartų mažiau laiko.

SIFT Flow rinkiniui taip pat pastebimas geras ansamblio apibendrinimas, suspaustais modeliais pasiekiami geresni rezultatai, spėjimo laikas taip pat žymiai sutrumpėja (žr. 8 lentelę).

8 lentelė. SIFT Flow segmentavimo rezultatai atskiriems modeliams, ansambliui bei distiliuotam modeliui [137]

Modeliai	IOU vidurkis (%)	Taškų tikslumas (angl. <i>pixel accuracy</i> , %)	Geometrinis tikslumas (angl. <i>geometric pixel accuracy</i> , %)	Laikas (sek.)
VGG-16 (32s)	33,9	84,3	93,6	0,14
VGG-16 (16s)	38,0	84,9	93,8	0,14
VGG-16 (8s)	37,5	85,9	94,6	0,14
ResNet-50	34,9	83,9	93,8	0,07
ResNet-101	35,5	84,5	94,0	0,09
ResNet-152	36,7	84,9	94,3	0,12
ResNet-50-skip (ne ansamblio dalis)	38,6	85,4	94,4	0,07
ResNet-152-skip (ne ansamblio dalis)	38,8	86,3	95,0	0,12
Ansamblis	39,1	86,9	95,1	1,25
Distiliuoti modeliai				
VGG-16 (8s)	38,4	86,3	94,6	0,14
FCN-ResNet-152	37,0	85,7	94,5	0,12
FCN-ResNet-152-skip	40,0	86,5	94,8	0,12
FCN-ResNet-50	35,4	84,9	94,1	0,07
FCN-ResNet-50-skip	37,9	85,2	94,2	0,07

[5, 125, 127] strategijomis siekiama išmokyti vieną modelį, kuris gerai aproksimuotų vidutinę ansamblio prognozę ir taip pasiekti geresnius rodiklius bei sutaupyti skaičiavimo resursų. Tačiau

naudojant šias metodikas nėra įmanoma įvertinti pačios prognozės neapibrėžtumo, nes prarandama ansamblio prognozių įvairovė, todėl nėra galimybės atskirti (1) lygties narių. Savo darbe *Malinin 'as et al.* [141] siūlo patį ansamblį, kuris galimai (jei tinkamai paruoštas) yra nežinomo tikrojo žinių skirstinio imčių grupė (t.y. savaime tam tikri skirstiniai), aproksimuoti *Prior Network* modeliu [141], suteikiančiu galimybę parametrizuoti sąlyginį skirstinį pagal turimus skirstinius (šiuo atveju – išvesčių). Autoriai sudaro perkeliamąjį duomenų rinkinį iš pradinio mokymo rinkinio įvesčių ir žymenų, gaunamų pagal ansamblio prognozių skirstinius kiekvienam pavyzdžiui. Tuomet *Prior Network* modelis mokomas minimizuojant (neigiamą) logaritmą tikėtino funkciją kiekvienam atskiros klasės skirstiniui. Metodikos kontekste skirstinio paieškos konvergavimo pagreitinimui naudojamas ir temperatūros koeficientas T [5]: didinant šį koeficientą siekiama suartinti empirinės mokymo imties klasių bei paties modelių išvesčių skirstinius. Šia metodika ir pasirinktu prognozavimo modeliu gaunamos prognozės geba pateikti papildomos informacijos: atsiranda galimybė nusakyti, ar modeliui trūksta žinių apie klasių atskiriamumą, ar apie pačią klasę.

Logiška būtų teigti, jog jei ansamblis ir būna distiliuojamas, o galutinės prognozės sudarymui paruošiamas modelis nėra masyvus ir reikalaujantis daug resursų, tačiau vis tiek pakankamai galingas svarbiausioms savybėms iš pradinės duomenų imties išgauti, galimai naudingas turėtų būti ir tolesnis rezultatų apdorojimas. *Teichmann 'as et al.* [142] pateikia metodiką visiškai sujungtų (angl. *fully-connected*) sąlyginių atsitiktinių laukų (angl. *conditional random fields, CRF*) naudojimui kartu su CNN segmentavimu. Teigdami sąlyginę nepriklausomybę *FullCRFs* karkasui [143], autoriai performuoja didelę dalį *CRF* atliekamo spėjimo proceso identiškais konvoliucijų operacijomis, o tai leidžia efektyviai panaudoti lygiagretinamus resursus.

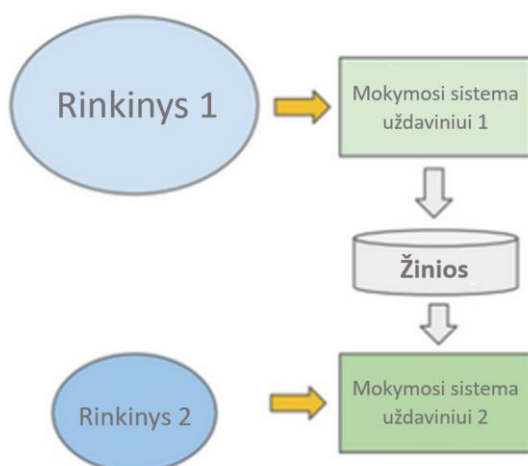
1.4. Perkeliamasis mokymasis

Kaip žinių perkėlimo terpę tikslinga naudoti susijusių dalykinių sričių problemų sprendimus, t.y. panaudoti žinias, gautas sprendžiant vieno tipo uždavinį susijusios srities uždaviniams, kuomet nereikia modelių kurti „nuo nulio“ [144]. Būtent tokio tipo žinių perdavimą apibrėžia perkeliamasis mokymasis (angl. *transfer learning*) [145]. Šio mokymosi tipo tikslas – neizoliuoti mokymosi paradigmos ir panaudoti žinias, įgytas vienos užduoties sprendimo metu, kitų, susijusių problemų sprendimui (žr. 8 pav.) Gilaus mokymosi kontekste ši motyvacija yra ypač stipri, nes sudėtingų problemų sprendimui adekvatus giliųjų neuroninių tinklų parengimas reikalauja labai didelio duomenų kiekio. Būtent negebėjimas parengti didesnį duomenų rinkinį specializuotai sričiai ar konkrečiai užduočiai spręsti ir yra pagrindinis perkeliamojo mokymosi veiksnys. Net turint technines galimybes naudoti SOTA modelius, jų efektyvumas siaurai apibrėžtai (mokymo imties prasme) užduočiai bus nedidelis palyginti su sritimis, kuriose egzistuoja dideli sužymėtų duomenų kiekiai. Taigi, pats perkeliamasis mokymasis peržengia konkrečių užduočių ribas: bandoma panaudoti iš anksto paruoštus (bendrinėse, daug žymėtų duomenų turinčiose srityse) (angl. *pre-trained*) modelius (ar jų dalis) naujoms užduotims spręsti. Vienas geriausiai žinomų pavyzdžių – ImageNet [140] duomenų rinkinys: populiariausių SOTA modelių svoriai šiam rinkiniui yra prieinami daugelyje mašininio mokymosi karkasų ir bibliotekų, juos galima parsisiųsti ir naudoti savo modeliams ruošti.

Perkeliamasis mokymasis (angl. *transfer learning*)

Paruošimas naujoms užduotims paremtas jau išmoktais uždaviniais:

- Mokymosi procesas gali būti greitesnis, tikslesnis ir / arba reikalauti mažiau duomenų.



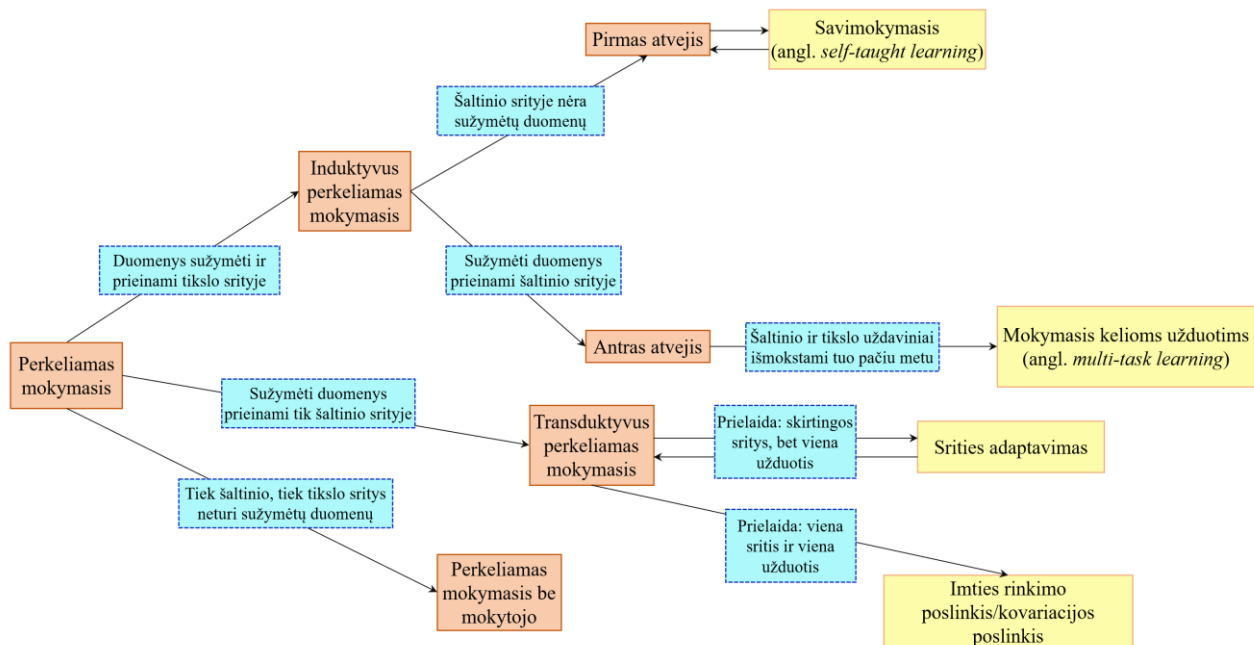
8 pav. Perkeliamojo mokymosi iliustracija [144]

Formalus perkeliamojo mokymosi apibrėžimas yra toks [145]: turint bendrinę (pirminę⁸) sritį D_s ir jai atitinkamą užduotį (tikslinę⁹ sritį) T_s bei paieškos sritį D_t ir atitinkamą užduotį (reikšmių sritį) T_t , perkeliamojo mokymosi tikslas – sąlyginio tikimybių skirstinio $P(Y_t|X_t)$ D_t kontekste radimas panaudojant informaciją gautą iš D_s ir T_s , kai $D_s \neq D_t$ arba $T_s \neq T_t$. Daugeliu atvejų bendrinės srities duomenų rinkinio dydis yra eksponentiškai didesnis už tikslinės srities duomenų rinkinio dydį. Svarbu pabrėžti, jog gali skirtis tiek marginalieji (įvesčių) skirstiniai skirtingose bendrinėje ir tikslinėje srityse, tiek ir sąlyginiai skirstiniai (žymenų erdvė) skirtingoms užduotims.

Egzistuoja ne viena perkeliamojo mokymosi strategija ir dar daugiau šių strategijų įgyvendinimo technikų. Jų pasirinkimas priklauso nuo užduoties, kurios kontekste naudojamas vienas ar kitas mašininio mokymosi algoritmas ar architektūra, taip pat nuo galimų įvesčių tipo bei turimų duomenų kiekio (žr. 9 pav.)

⁸ Verčiama iš anglų kalbos – pirminė ir tikslinė sritis atitinka *source and target domains*

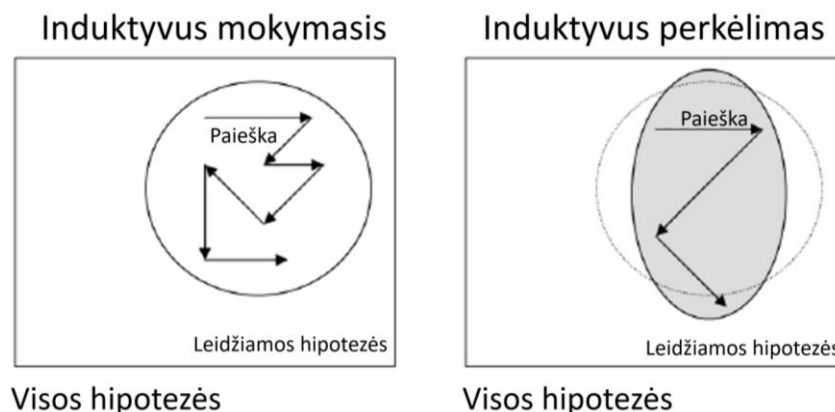
⁹ Verčiama iš anglų kalbos – pirminė ir tikslinė užduotys atitinka *source and target tasks*



9 pav. Perkeliamojo mokymosi klasifikacija [145]

Pagal skirstymą 9 pav. skiriamos trys pagrindinės perkeliamojo mokymosi grupės: induktyvus perkeliamasis mokymasis, kuomet pirminė ir tikslinė sritys sutampa (bent jau iš dalies), tačiau skiriasi reikšmių sritys (užduotys); perkeliamasis mokymasis be mokytojo, kuomet modeliai tikslinei sričiai mokomi be žymenų (angl. *unsupervised learning*); transduktyvus perkeliamasis mokymasis, kuomet tarp pirminės ir tikslinės užduočių egzistuoja panašumų, tačiau skiriasi jų apibrėžimo sritys.

Giliojo mokymosi kontekste naudojamą perkeliamą mokymąsi atitinka induktyvusis variantas. Šią strategiją įgyvendinančių algoritmų tikslas – nustatyti atvaizdžius pagal mokymo duomenų rinkinį. Tam, jog būtų gaunamas gerai duomenis (ypač tuos, kurie labai skiriasi nuo mokymosi rinkinio) apibendrinantis algoritmas, pačiam mokymo rinkiniui iškeliamos tam tikros prielaidos (angl. *inductive bias*). Šios prielaidos gali būti apibrėžiamos kaip prognozavimo hipotezių erdvės ribojimas ar šių hipotezių paieškos algoritmas (žr. 10 pav.) Nustatant tam tikrą apibrėžimo sritį ar konkrečią užduotį svarbu žinoti, jog prielaidos daro įtaką tam, kokias žinias modelis galės perimti. Induktyviojo mokymosi metu algoritmai išnaudoja tam tikras pirminės užduoties prielaidas tikslinės užduoties sprendimui, pavyzdžiui sumažinama galimų suformuoti hipotezių erdvė (pvz., naudojant *L2-SP* [146] reguliarizaciją).



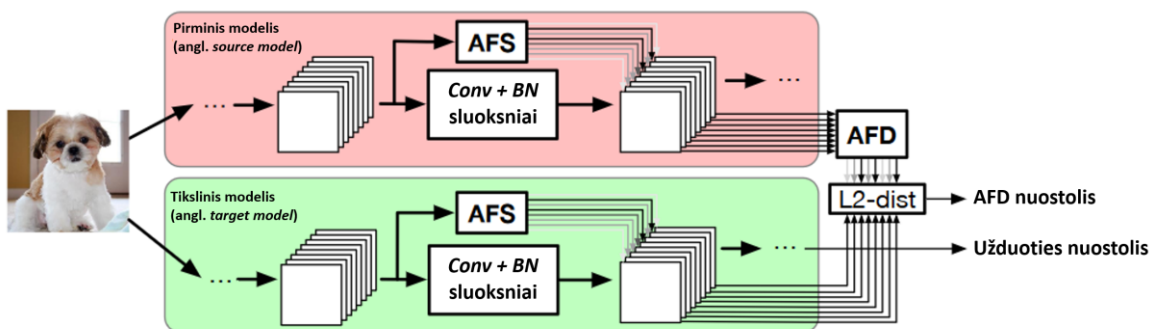
10 pav. Induktyviojo mokymosi iliustracija [144]

Giliųjų neuroninių tinklų kontekste galima atskirti dvi pagrindines perkeliamojo mokymosi metodikas – iš anksto mokytų modelių naudojimą savybėms išskirti bei paruoštų modelių derinimą (angl. *fine-tuning*).

Pirmuoju atveju naudojamos SOTA architektūros, mokytos ant didelių tam tikros probleminės srities (ji gali būti ir labai plati, pavyzdžiui ImageNet atveju galima teigti, jog probleminė sritis – bendrinis paveikslėlių klasifikavimas) duomenų rinkinių. Paprastai specifinei užduočiai spręsti panaudojamas modelis be paskutinio (ar be kelių paskutinių) sluoksnio. Tikimasi, jog pasinaudojant iš anksto paruošto modelio galimybėmis išreikšti pagrindines tam tikros tikslinės apibrėžimo srities pavyzdžių savybes, ruošiant kitą pasirinktą algoritmą šie atvaizdžiai savyje turės pakankamai daug informacijos ir bus lengvai atskiriami tarp tikslinės srities pavyzdžių.

Jei pirmuoju būdu pasirinkto algoritmo svoriai nėra atnaujinami, tai paruoštų modelių derinimo kontekste šis procesas jau vykdomas. Šios metodikos darbo eiga iš esmės nelabai skiriasi nuo visiškai naujai ruošiamo modelio mokymo, o skirtumas tik tas, jog naudojamas atskiras pirminis duomenų užkodavimo (angl. *backbone*) modelis. Jis sudaromas tikslinės užduoties apibrėžimo bei reikšmių sritims pritaikant pasirinkto paruošto modelio architektūrą arba vidinių sluoksnių svorius įkomponuojant į savo kuriamą modelį. Taip pat yra galimybė neatnaujinti kai kurių užkodavimo modelio struktūroje esančių sluoksnių (juos išaldyti, angl. *freeze*), jei žinoma tam tikra informacija (pvz., tam tikruose sluoksniuose pastebima tinkama entropija tarp skirtingų klasių pavyzdžių išvesčių).

Literatūroje randama ir perkeliamojo mokymosi bei modelių distiliavimo sintezės pavyzdžių. Wang'as et al. [147] pasiūlė AFDS (angl. *attentive feature distillation and selection*) algoritmą, kuriuo perkeliamojo mokymosi metu ne tik derinamos reguliarizacijos (praleidžiant neuronų jungtis (angl. *neuron skipping*)) reikšmės, tačiau ir nustatomos perkėlimui reikšmingos savybės.



11 pav. AFDS koncepcija [147]

Autoriai naudoja savybių atvaizdžių (kiekvieno konvoliucinio sluoksnio išvesčių) atkartojimą, pavadintą AFD (angl. *attentive feature distillation*) (žr. 11 pav.) AFD išmoksta (naudojamas nedidelis neuroninis tinklas) skirti kiekvieno atskiro kanalo svarbą, o ši svarba panaudojama kaip svoriai modelio-studento atvaizdžių vektorių reguliarizacijai naudojant Euklido atstumą. Taip perduodama informacija kurias modelio-mokytojo kanalų išvestis naudoti. Modelio-mokinio pagreitinimui siūloma AFS (angl. *attentive feature selection*) metodika dinaminiam modelio mažinimui (angl. *pruning*). Ši dalis išmoksta pasirinkti svarbiausius konvoliucijų išvesties kanalus ir praleisti ne tokius svarbius priklausomai nuo to, kokia buvo konvoliucijos įvestis. Tokiu būdu nustatomi retai naudojami neuronai, kurie gali būti šalinami taip sumažinant resursų poreikį. AFD ir AFS kartu veikia tarsi

informacijos tėkmės sklendės: *AFD* pritaiko tinkamą reguliarizaciją derinant žinių, perkeliamų iš modelio-mokytojo, kiekį; *AFS* stabdo nesvarbią ir praleidžia tik pagrindinę informaciją tolesniems sluoksniams.

Autorių metodika išbandoma šešiams duomenų rinkiniams naudojant ResNet-101 su papildomais pagalbiniais tinklais *AFS* ir *AFD* įgyvendinimui. Pagal rezultatus matoma, jog *AFDS* pasirodo geriau daugumos duomenų rinkinių testavimo atvejais palyginus su *network slimming (NS)* [148] bei *soft filter pruning (SFP)* [149] metodikomis (žr. 9 lentelę).

9 lentelė. Rezultatai naudojant *AFDS* [147] (*MACS* (angl. *multiply-accumulate*) nurodo operacijų sumažėjimą mažinant modelį)

Mažinimas <i>MACs</i> būdu		<i>NS</i>	<i>SFP</i>	<i>AFDS</i>
<i>MIT Indoors 67</i>	2x	81,83 ± 0,35	79,43 ± 0,50	82,05 ± 0,43
	5x	69,38 ± 0,27	60,43 ± 0,31	69,93 ± 0,52
	10x	1,50 ± 0,30	58,49 ± 0,34	66,72 ± 0,53
<i>Standford Dogs 120</i>	2x	87,21 ± 0,58	81,74 ± 0,26	87,41 ± 0,56
	5x	73,44 ± 0,27	61,20 ± 0,31	75,14 ± 0,52
	10x	1,33 ± 0,50	59,63 ± 0,23	70,70 ± 0,33
<i>Caltech-256-30</i>	2x	85,87 ± 0,38	77,26 ± 0,28	85,15 ± 0,75
	5x	66,57 ± 0,23	64,27 ± 0,31	66,64 ± 0,32
	10x	0,39 ± 0,04	57,11 ± 0,54	61,45 ± 0,43
<i>Caltech-256-60</i>	2x	88,02 ± 0,45	84,59 ± 0,28	87,15 ± 0,75
	5x	73,95 ± 0,27	68,38 ± 0,59	74,46 ± 0,52
	10x	5,05 ± 0,11	61,27 ± 0,49	70,16 ± 0,53
<i>CUB-200-2011</i>	2x	78,88 ± 0,65	75,65 ± 0,26	78,03 ± 0,45
	5x	73,44 ± 0,27	61,50 ± 0,31	73,35 ± 0,52
	10x	0,52 ± 0,50	57,88 ± 0,23	69,07 ± 0,43
<i>Food-101</i>	2x	83,78 ± 0,61	75,65 ± 0,26	84,21 ± 0,65
	5x	73,36 ± 0,45	17,10 ± 0,17	79,12 ± 0,52
	10x	0,99 ± 0,04	3,85 ± 0,09	76,95 ± 0,49

1.5. Mokymosi modelių pateikimas

Atlikus visus pagrindinius modelio paruošimo grandinės etapus, panaudojus perkeliama mokymąsi, o geriausius gautus modelius sukombinavus į ansamblį ir jį distiliavus, viena paskutinių probleminės užduoties sprendimo stadijų – modelio pateikimo (angl. *model serving*) diegimas (angl. *deployment*). Mašininio mokymosi algoritmų spėjimo proceso naudojimas gali būti suskirstytas į 3 pagrindines kategorijas [150, 151, 152]:

- Spėjimai pagal užklausą: šiuo atveju pateikiamas modelis realizuojamas serveryje kaip galinis užklauso metodas (angl. *endpoint*). Aplikacija pasirinktu protokolu santykinai trumpo laiko intervalo ribose pateikia ir gauna informaciją apie siųstą užklausą su rezultatais. Tokie modeliai dažniausiai įgyvendinami kaip mikroservisai (angl. *microservices*) arba kaip nutolusios funkcijos (angl. *serverless functions*). Platformos, skirtos būtent tokių aplikacijų diegimui (pvz.: *Kubernetes*, *Docker*), palengvina modelių spėjimo procedūrų įgyvendinimą ir

- plėtrą. Pačios platformos rūpinasi naujų servisų (arba jų dalių) paleidimų (pvz., sulaukiant daugiau užklausų), automatiniu atnaujinimu bei suteikia atsparumą trikdžiams;
- Spėjimas paketais (angl. *batch processing*): modelis spėjimo procesui įkraunamas tam tikrais laiko intervalais arba tuomet, kai susikaupia nustatytas kiekis apdorojimui skirtos informacijos. Ši pateikimo strategija dar vadinama atjungtuoju pateikimu (angl. *offline serving*). Dėl šiandien naudojamų duomenų bazių efektyvumo (pvz.: *redis*, *Aerospike*), duomenų apdorojimas paketais yra pakankamai paprastai įgyvendinamas ir iš esmės reikia rūpintis tik teisingu duomenų pateikimu modeliui. Tačiau šio tipo apdorojimas gal tapti sudėtingas sukaupus didesnius duomenų kiekius daugeliui atskirų vartotojų. Tokiems ribojimams spręsti taikomas horizontalus plėtimas (angl. *horizontal scaling*), procesų lygiagretinimas;
 - Modeliai realizuojami pačiuose jų pateikiamą informaciją naudojančiuose prietaisuose (angl. *edge devices*). Šiuo atveju nustatymam duomenų teikėjui (pvz., serveriui) užklausos nėra siunčiamos, o aplikacija duomenis apdoroja pačiame prietaise. Tačiau toks modelių diegimas gali būti sudėtingas dėl (potencialiai) didžiulio kraštinių elementų kiekio bei jų technologinės įvairovės. Daugelio įrenginių diegimams (angl. *multi-device deployment*) taip pat sukurtos platformos, padedančios užtikrinti efektyvų modelių pateikimą (pvz., *Fritz AI*). Dažnai tokie dieginiai naudojami kartu su anksčiau minėtomis kategorijomis: pavyzdžiui, jei prietaisas paprastai kreipiasi į serverį modelio rezultatui gauti, nelikus interneto ryšio lokaliai įkraunamas (pvz., naršyklėje) paprastesnis neuroninio tinklo modelis ir spėjimas duomenims atliekamas būtent juo.

Sprendžiant, kurį modelio pateikimo variantą rinktis, svarbu įvertinti kelis aspektus. Pirmiausia svarbu susidaryti konkretaus naudojimo atvejo specifikaciją aptariant svarbiausius nefunkcinius reikalavimus. Vieni svarbiausių yra atsakymo vėlinimo (angl. *latency*) reikalavimai, duomenų saugumas, tinklo pasiekiamumo galimybės bei aplikacijai skirti resursai tiek kainos (pvz., debesijos paslaugoms), tiek skaičiavimo pajėgumų prasme. Priklausomai nuo to, kokie vėlinimo reikalavimai nustatomi konkrečiai aplikacijai, gali būti pasirinkta modelį diegti tam tikroje realaus laiko sprendimus užtikrinančioje architektūroje. Jei nėra reikalaujama greitų modelio spėjimų rezultatų, reikėtų svarstyti apie paketinio apdorojimo architektūras.

1.6. Apibendrinimas

Apžvelgus literatūrą randama įvairių modelių ansambliavimo bei distiliavimo technikų. Algoritimų ansambliavimas taikant neuroninius tinklus dažniausiai atliekamas daugybinių struktūrų metodu, o distiliavimas – *Hinton et al.* [5] pristatyta modelio-mokytojo ir modelio-studento technika bei jos modifikacijomis, taip pat taikymuose su CNN pastebima įvairių algoritimų sintezė. Literatūroje pateikiami rezultatai tekstinės ir vaizdinės informacijos klasifikavimo užduotims rodo modelių ansambliavimo ir distiliavimo efektyvumą.

Modelių spėjimai segmentuojant vaizdinius duomenis giliaisiais konvoliuciniais neuroniniais tinklais yra ypač svarbūs, nes naudodamos šią informaciją įvairios komercinės sistemos interpretuoja savo aplinką ir priima sprendimus. Algoritams ruošti egzistuoja daug įvairių vaizdų rinkinių, skirtų aibei aplikacijų, tačiau modelių parengimas ir taikymas (ypač naudojant didelius rinkinius) užtrunka ilgai ir reikalauja didelių techninių resursų. Žinant, jog beveik 90 % visų šio tipo algoritmams skirtų skaičiavimų panaudojama būtent spėjimo proceso įgyvendinimui [153], labai svarbu tampa rasti

efektyvias metodikas, leidžiančias sistemoms sutaupyti skaičiavimo resursų stipriai nepaveikiant segmentavimo efektyvumo.

Šiame darbe nuspręsta tirti modelių ansambliavimo bei distiliavimo metodikas semantinio segmentavimo kontekste. Projekte išsikeltas tikslas – ištirti įvairių giliojo mokymosi architektūrų vaizdų segmentavimo efektyvumą pritaikant ansambliavimo bei žinių distiliavimo technikas. Šiam tikslui pasiekti iškeliami tokie uždaviniai:

1. apžvelgti metodus, skirtus giliojo mokymosi modeliams sujungti ir distiliuoti;
2. pasirinkti ir paruošti vaizdų rinkinius giliajam mokymuisi;
3. pritaikyti pasirinktus semantinio segmentavimo modelius;
4. pritaikyti modelių jungimą kolektyvams sudaryti;
5. išbandyti distiliavimo technikas spėjimų greitimeikai gerinti;
6. apibendrinti eksperimentų rezultatus ir pateikti išvagas.

2. Tyrimo objektas ir metodai

Tyrimo metu buvo mokomi ir testuojami giliojo mokymosi modeliai bei rengiamos įvairios konfigūracijos vaizdinių duomenų rinkinių semantinio segmentavimo uždaviniams spręsti. Eksperimentams atlikti ruošiami pasirinktų giliųjų CNN architektūrų modeliai, segmentavimo rezultatų efektyvumui gerinti modeliai jungiami į heterogeninius kolektyvus, homogeniniams ansambliams formuoti panaudojama kosinusinio atvėsimo technika, taip pat pritaikomas žinių distiliavimas. Tolesniuose poskyriuose glaustai aptariami pagrindiniai naudojamų metodikų aspektai.

2.1. Konvoliuciniai neuroniniai tinklai

Konvoliuciniai neuroniniai tinklai (CNN) – giliojo mokymosi modelių klasė, daugiausia naudojama kompiuterinės regos uždaviniuose. Šie tinklai iš dalies sukurti bandant atkartoti vizualinės informacijos atpažinimo procesus žmogaus smegenyse pasitelkiant tarpusavyje susietų neuronų sluoksnius. Svarbiausia CNN inovacija ir skiriamoji savybė – konvoliucinių sluoksnių naudojimas. Naudojant šias struktūras CNN išmoksta atpažinti skirtingas paveikslėlio savybes: kampus, kraštus, tekstūras ir pan. Konvoliuciniai sluoksniai pritaiko nedidelių matricių su mokomais svoriais aibę (vadinamuosius konvoliucinius filtrus): požymių žemėlapiai (angl. *feature maps*) gaunami dauginant įvesties matricos narius tam tikroje srityje ir susumuojant gautus rezultatus (2).

$$Y_{ij}^l = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} \omega_{ab} Y_{(i+a)(j+b)}^{l-1} \quad (2)$$

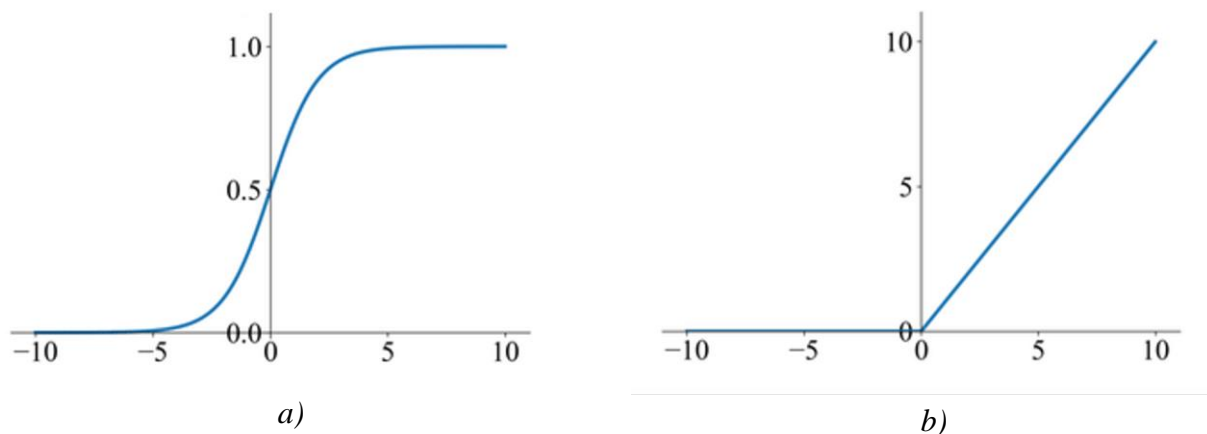
čia Y_{ij}^l – l ojo sluoksnio ij matricos nario išvestis, m – $m \times m$ konvoliucinio filtro dydis.

Konvoliucijos operacija giliojo mokymosi kontekste reiškia matricos elementų sandaugą ir sudėtį (sudedami sudauginti filtro elementai). Išvesčiai (formulėje (2) pateikiamas vieno kanalo skaičiavimas) taip pat gali būti pritaikoma aktyvacijos funkcija (angl. *activation function*), jog būtų užtikrinamas tinklo netiesiškumas (angl. *non-linearity*), taip pat pridedamas poslinkio (angl. *bias*) parametras (3).

$$A_{ij}^l = f(Y_{ij}^l) + b \quad (3)$$

čia Y_{ij}^l – l – ojo sluoksnio ij matricos nario išvestis, f – aktyvacijos funkcija, b – poslinkio parametras.

Konvoliuciniuose tinkluose po kitų sluoksnio operacijų dažnai taikoma glodintos tiesinės funkcijos (angl. *rectified linear unit, ReLU*) (4), taip pat sigmoidės (angl. *sigmoid*) (5) aktyvacijos funkcijos (žr. 12 pav.)

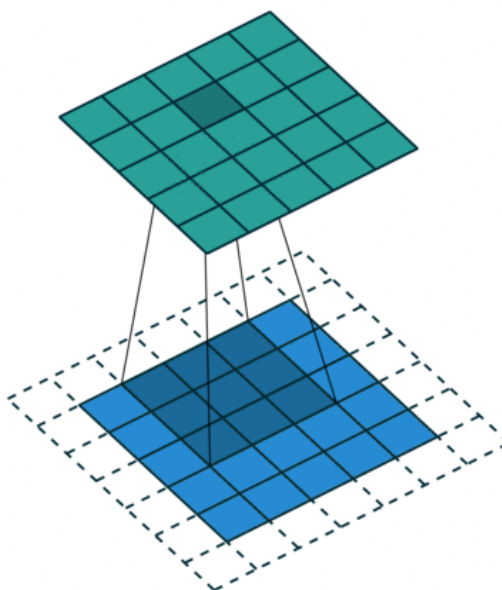


12 pav. *ReLU* (a) bei *sigmoid* (b) aktyvacijos funkcijų pavydžiai [154]

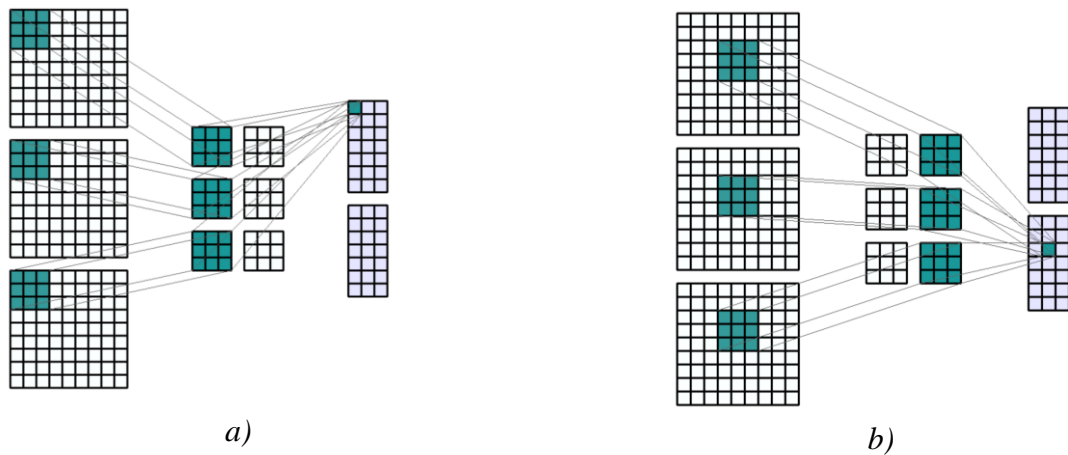
$$f(x) = \max(0, x) \quad (4)$$

$$f(x) = \frac{1}{1 + e^{-x}} \quad (5)$$

Norint tinkamai valdyti tarpinių matricių dimensijas, konvoliuciniuose sluoksniuose naudojamas užpildymo parametras (angl. *padding*), kuriuo tam tikra reikšme dirbtinai padidinamas įvesties matricos dydis pilnam filtro perėjimui (žr. 13 pav.) bei žingsnio parametras (angl. *stride*), reguliuojantis perėjimo žingsnio dydį (žr. 14 pav.)

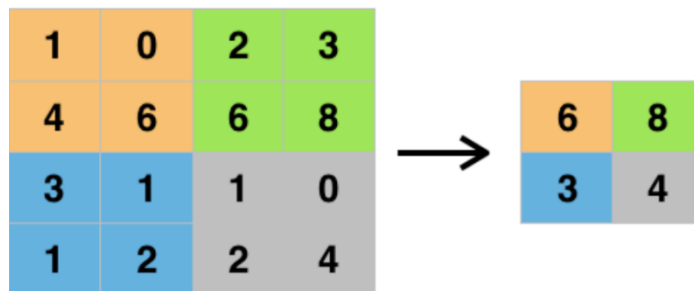


13 pav. Konvoliucijos operacija su užpildymu (brūkšniuotų linijų nurodomas plotas) [155]



14 pav. Pavyzdinis išvesties skaičiavimas trijų kanalų matricai su dviem konvoliuciniais filtrais (3x3x3), kai užpildymas = 0, žingsnis = (1, 3)¹⁰

Dimensionalumo mažinimui konvoliuciniuose tinkluose naudojami vadinamieji telkimo (angl. *pooling*) sluoksniai. Šių sluoksnių funkcija – sumažinti informacijos kiekį kaupiant požymių žemėlapių aktyvacijų grupes. Dažniausiai naudojama maksimalaus telkimo (angl. *max pooling*) konfigūracija, kuomet nepersidengiančioms požymių sritims pritaikomas maksimalaus įverčio filtras (žr. 15 pav.) Šios operacijos ne tik sumažina skaičiavimo kaštus dėl mažesnio tolesnių sluoksnių parametrų skaičiaus, tačiau ir padeda išvengti modelių persimokymo.



15 pav. Maksimalaus telkimo (angl. *max pooling*) operacijos iliustracija¹¹

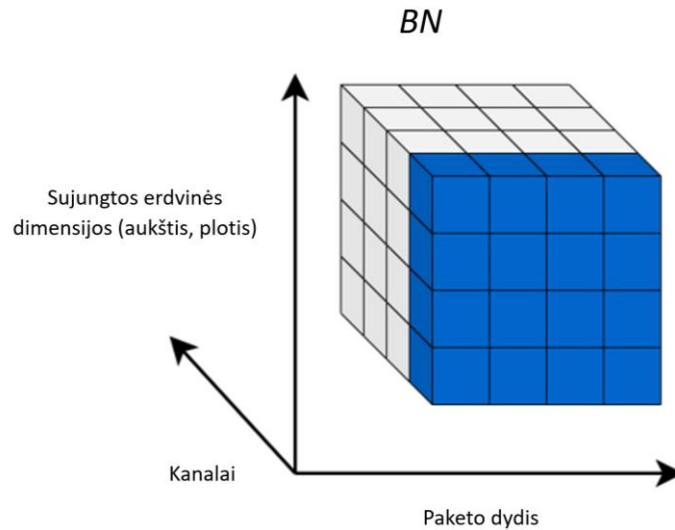
CNN tinklo spėjimų tikslumo gerinimui pasitelkiama paketinės normalizacijos (angl. *batch normalization*) metodika [156]. Neuroninių tinklų, taip pat ir CNN, mokymosi metu kiekvieno sluoksnio įvesčių pasiskirstymas keičiasi (angl. *shift*) nuo pradinio skirstinio dėl to, jog yra modifikuojami ankstesnių sluoksnių parametrai, o tai sunkina ir lėtina modelio mokymą (vadinamasis vidinės kovariacijos poslinkis (angl. *internal covariate shift*)). Paketinės normalizacijos metodas šią problemą sprendžia pagal (6) formulę normalizuojant kiekvieno pritaikomo sluoksnio įvestis (žr. 16 pav.) Mokymosi metu normalizacijos sluoksnis skaičiuoja kiekvieno paketo vidurkį bei dispersiją ir išsaugo šių parametrų sumą su inercijos (angl. *momentum*) parametru. Normalizuotos ir paslinktos reikšmės tuomet dauginamos ir pridedamos prie sluoksnio parametrų, o tai leidžia modeliui išmokti sudėtingesnes duomenų reprezentacijas. Modelio mokymo metu šio sluoksnio sukauptą sumą taip pat gali būti naudojama modelio svorių įšaldymui valdant inercijos parametru.

¹⁰ <https://towardsdatascience.com/conv2d-to-finally-understand-what-happens-in-the-forward-pass-1bbaafb0b148>.

¹¹ <https://deepai.org/machine-learning-glossary-and-terms/max-pooling>

$$y = \frac{x - E(x)}{\sqrt{Var(x) + \varepsilon}} \cdot \gamma + \beta \quad (6)$$

čia $E(x)$ – paketo vidurkis, $Var(x)$ – paketo dispersija, ε – pagalbiniis dalybos parametras, γ , β – išmokstami parametrai.



16 pav. Paketinės normalizacijos operacijos taikymo konteksto iliustracija¹²

Pati svorių įšaldymo metodika pritaikoma kaip specifinių (CNN – dažniausiai užkoduotojo) sluoksnių parametrų visiškai (angl. *full*) arba dalinis (angl. *partial*) fiksavimas (dalinis keitimas atliekamas su tam tikru inercijos parametru) klaidos skleidimo atgal (angl. *backpropagation*) etape [157].

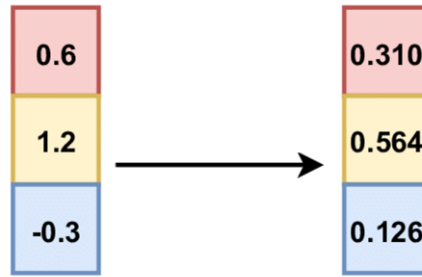
Paskutinis modelio sluoksnis yra skirtas konfigūruotam įvesties klasifikavimui. Konvoliucinių tinklų architektūrose visiškai sujungti (angl. *fully connected*) sluoksniai dažnai keičiami 1x1 dydžio konvoliuciniu filtru (angl. *fully convolutional*) [158]. Modelio optimizavimui svarbu gauti logitus, naudojamus nuostolio funkcijos skaičiavime kartu su žymėtų duomenų vektoriumi: šiam tikslui gali būti taikoma klasių tikimybių skirstinio normalizacija naudojant *softmax* funkciją (žr. 17 pav.) Gautiems logitams segmentavimo uždaviniuose dažnai naudojama kryžminės entropijos (angl. *cross entropy*) klaidos funkcija (7).

$$L_{CE} = -X \cdot Z + \log(1 + e^X) \quad (7)$$

čia X – gaunamas logitų vektorius, Z – žymėjimų vektorius.

Gaunama klaida naudojama skleidimu atgal derinant modelių parametrus [67]. Klasikiniuose tinkluose naudojamas stochastinio gradientinio nusileidimo (angl. *stochastic gradient descent*, SGD) optimizatorius [159] bei įvairios jo modifikacijos.

¹² <https://tek4.vn/batch-norm-trong-pytorch-lap-trinh-neural-network-voi-pytorch>



17 pav. Softmax veikimo iliustracija [160]

2.2. Modelių mokymas

Prižiūrimo mokymosi metu pagal duomenų rinkinį, turintį įvesčių ir išvesčių aibes, mašininio mokymosi algoritmai siekia išmokti geriausią būdą įvesčių-išvesčių sąsajai gauti¹³. Jei teigiama, jog egzistuoja tam tikra funkcija (angl. *mapping function*), nusakanti, kaip įvestys siejasi su išvestimis, mokymosi su mokytoju algoritmų tikslas – gauti geriausiai šią sąsają aproksimuojančią funkciją. Vienas populiariausių būdų šiam tikslui pasiekti yra neuroninių tinklų – universalių aproksimatorių – mokymas [67]. Mokymosi metu naudojama nuostolio funkcija mažiausią reikšmę įgauna tuo atveju, kai neuroninio tinklo išvestys sutampa su duomenų rinkinio išvestimis.

Klasikiniu atveju neuroniniai tinklai aproksimuoti mokosi minimizuojant nuostolio funkciją stochastinio gradientinio nusileidimo (SGD) [159] algoritmu. Teigiant, jog siekiama minimizuoti tam tikrą funkciją $J(\Theta)$, parametrizuotą modelio svoriais $\Theta \in \mathbb{R}^d$, gradientinio nusileidimo metodas tai atlieka atnaujinant svorius Θ priešinga funkcijos gradientui $\nabla_{\Theta} J(\Theta)$ kryptimi [161]. Įvedant mokymosi parametą α valdoma, kokio dydžio diskrečiais žingsniais atliekamas minėtas parametru atnaujinimas (sekama funkcijos hiperpaviršiaus nuolydžiu žemyn, kol pasiekiamas (lokalus) slėnis). Jei gradientinis nusileidimas vykdomas kas n mokymosi atvejų, atnaujinimą galima išreikšti (8) formule toliau:

$$\Theta = \Theta - \alpha \cdot \nabla_{\Theta} J(\Theta; x^{(i:i+n)}, z^{(i:i+n)}) \quad (8)$$

čia $x^{(i:i+n)}, z^{(i:i+n)}$ – žingsnyje i atitinkamai modelio išvesčių bei rinkinio žymėjimų vektoriai.

SGD algoritmas funkciją sunkiau minimizuoja tais atvejais, kai vienos dimensijos statusas yra daug didesnis už kitų (vadinamieji lokalaus minimumo taškai). Nusileidimo su pagreičiu (angl. *momentum*) modifikacija leidžia SGD greičiau leistis reikiama kryptimi ir mažina galimai dėl lokalaus minimumo atsirandančius svyravimus [161]. Įvedamas parametras γ , nurodantis, kokia ankstesnio atnaujinimo vektoriaus dalis pridedama prie žingsnio atnaujinimo (žr. (9) formulę).

$$v_t = \gamma v_{t-1} + \alpha \nabla_{\Theta} J(\Theta) \quad (9)$$

$$\Theta = \Theta - v_t$$

Tokiu nuo kalno riedančio kamuoliuko analogija paaiškinamu algoritmu greičiau pasiekiamas funkcijos konvergavimas ir sumažinami svyravimai tarp iteracijų.

¹³ <https://machinelearningmastery.com/neural-networks-are-function-approximators>

Bendroju atveju neuroninių tinklų efektyvumas gali būti neigiamai paveiktas persimokymo (angl. *overfitting*), kuomet modelis atsižvelgia į daugumą ar visus įvesties požymius nepaisant to, koki poveikį (mažą ar didelį) jie daro galutinei modelio išvesčiai [162]. Naudojant regularizaciją (angl. *regularization*) įvedama baudos reikšmė, kuria siekiama minimizuoti visų modelio svorių reikšmes ir taip išvengti persimokymo. Pagrindiniai regularizacijos metodai, naudojami neuroninių tinklų mokyme, yra $L1$, $L2$ regularizacijos bei išmetimo (angl. *dropout*) metodas.

$L1$ regularizacijos (dar vadinama *Lasso* regresija) metu prie nuostolių funkcijos pridedama baudos reikšmė, atitinkanti absoliučių svorių verčių sumą (žr. (10) formulę).

$$\Omega(\Theta) = \|\theta\|_1 = \sum_i |\theta_i| \quad (10)$$

Taikant $L1$ tam, kad būtų minimizuota nuostolių funkcija, kai kurie požymiai ignoruojami (svoriai nustatomi į 0).

Naudojant $L2$ regularizaciją (dar vadinama *Ridge* regresija) prie nuostolių funkcijos pridedama baudos reikšmė, atitinkanti euklidinį atstumą (žr. (11) formulę).

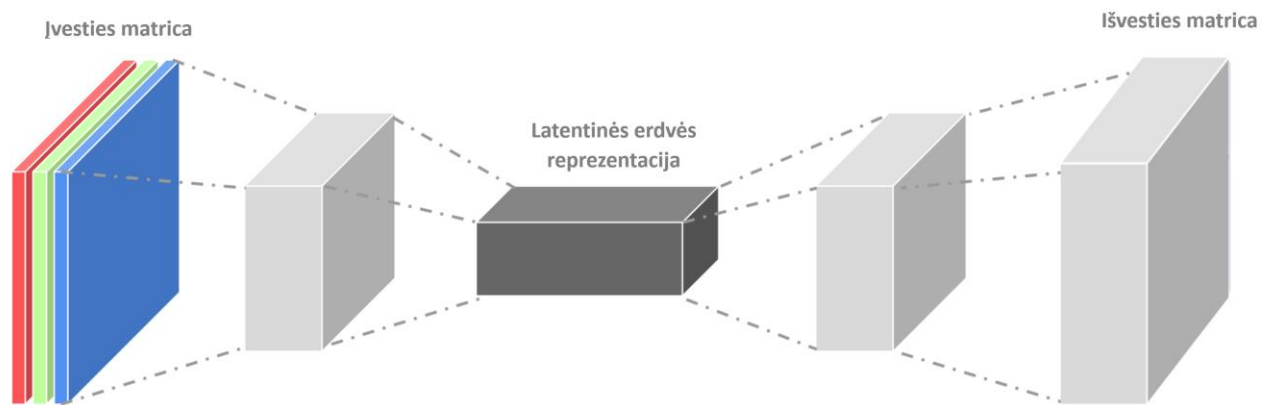
$$\Omega(\Theta) = \|\theta\|_2 = \sqrt{\sum_i \theta_i^2} \quad (11)$$

Taikant $L2$ regularizaciją modelis optimizuojamas priskiriant mažesnius svorius, o didesnes svorių vertes įgauna tik požymiai, reikšmingai veikiantys nuostolių funkcijos reikšmę.

Išmetimo metodika (angl. *dropout*) paremta laikinu neuroninio tinklo struktūros keitimu. Kiekvienos iteracijos metu sluoksniui priskirto hiperparametro reikšmė nustatoma tikimybė, jog atskira sluoksnio išvestis bus ignoruojama (išmestų neuronų svorių gradientai skleidimo atgal metu optimizuojant pagal (9) formulę atitinka 0). Šia metodika pasiekama, jog būtų sumažinta priklausomybė tarp atskirų neuronų (aprosimuojamas tinklų su išmestais neuronais spėjimų vidurkio efektas [162]).

2.3. Modelių konfigūracija

Visi projekto tyrimų kontekste naudojami modeliai pagrįsti užkodavimo-dekodavimo architektūra (angl. *encoder-decoder*). Šio tipo modeliai transformuoja įvestis į norimų dimensijų išvestis pirmiausia užkoduojuant matricas į žemesnio dimensionalumo latentinę erdvę (mažinimas (angl. *downsample*)) ir dekoduojuant gautą požymių žemėlapi į reikšmių srities erdvę (didinimas (angl. *upsample*)) (žr. 18 pav.)



18 pav. Abstrakti užkoduotojo-dekoduotojo architektūros schema¹⁴

Užkodavimo dalis įvesties matricas transformuoja į mažesnės erdvinės rezoliucijos, tačiau „gilesnius“ (didinama > 2 dimensija) požymių žemėlapius. Tai pasiekama įvairiai jungiant konvoliucinius, telkimo, visiškai sujungtus bei kitų tipų sluoksnius.

Užkodavimo dalies rezultatams – požymių žemėlapiui – gauti naudojami vadinamieji pagrindiniai, baziniai (angl. *backbone*) modeliai. Šiam tikslui dažnai taikomi segmentuojant duomenų rinkinius SOTA rezultatus pasiekę CNN modeliai, neįtraukiant paskutinių modelio sluoksnių („segmentacijos galvos“, angl. *segmentation head*), transformuojančių požymių žemėlapius į sukonfigūruotą segmentavimo rezultatų pateikimo reprezentaciją. Užkoduotojo modelių svoriai paprastai neinicijuoja atsitiktinai, o priskiriami iš standartinio (pvz., ImageNet [140]) rinkinio mokymosi metu gautų svorių.

Tolesniuose skyreliuose bus trumpai aptariamos naudojamų modelių architektūros.

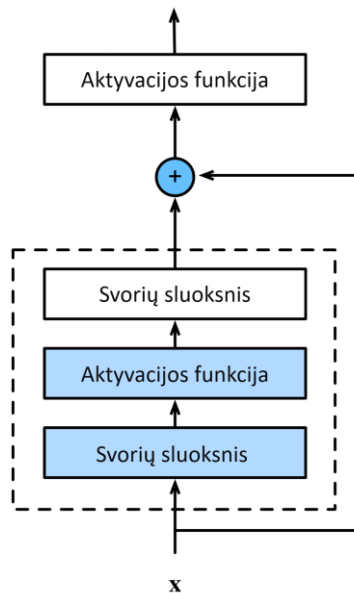
2.3.1. Modelių architektūros

Projekto tyrimams atlikti buvo naudotos penkios pagrindinės giliųjų konvoliucinių neuroninių tinklų architektūros – PSPNet (angl. *Pyramid Scene Parsing Network*) [163], LinkNet [164], FPN (angl. *Feature Pyramid Network*) [165], PAN (angl. *Pyramid Attention Network*) [166], Deeplab („Deeplabv3+“ tipas) [167]. Užkoduotojo dalies modeliams naudotos *SEResNeXt-50* [168], *EfficientNetB3* [169] bei ResNet (angl. *Residual Network*) (konfigūracijos ResNet-50 ir ResNet-18) [136]. Toliau bus aptariami pagrindiniai dekodavimo dalyje naudojamų architektūrų bei bazinio modelio ResNet ypatumai.

2.3.1.1. ResNet

ResNet (angl. *Residual Network*) – neuroninių tinklų architektūra, naudojanti liekamuosius (angl. *residual*) ryšius nykstančiojo gradiento (angl. *vanishing gradient*) problemai spręsti. Architektūroje naudojami ryšiai leidžia informacijai tiesiogiai pasiekti kitus sluoksnius, o tai įgalina modelį lengviau optimizuoti savo parametrus. Naudojant šiuos ryšius galima mokyti gilesnius, daugiau sluoksnių turinčius neuroninius tinklus. ResNet pavadinime dažniausiai nurodoma, kiek sluoksnių turi tinklas, pvz. ResNet-50 arba ResNet-101. ResNet struktūros pagrindinė idėja pateikiama 19 pav. toliau:

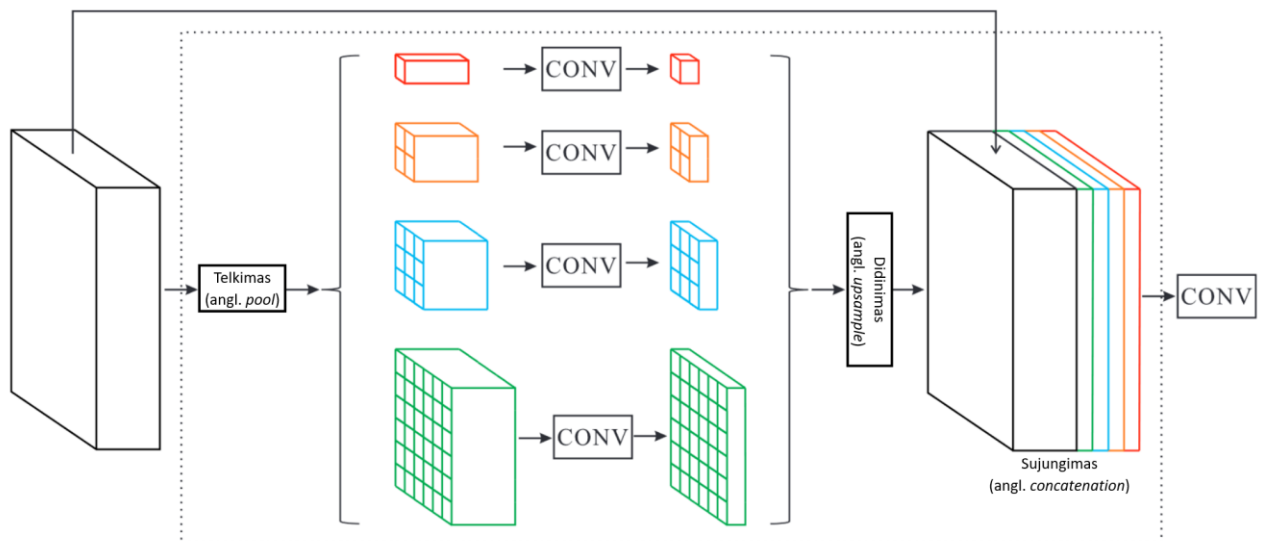
¹⁴ <https://hackernoon.com/latent-space-visualization-deep-learning-bits-2-bd09a46920df>



19 pav. ResNet modelio bloko algoritmo principo iliustracija¹⁵

2.3.1.2. PSPNet

PSPNet – 2017 m. *Zhao et al.* pasiūlyta giliojo neuroninio tinklo architektūra, pasižyminti į piramidę panašia struktūra. PSPNet modeliai pristatymo laikotarpiu pasiekė SOTA rezultatus segmentuojant Cityscapes [32] bei *PASCAL VOC* [28] duomenų rinkinius. Pagrindinis architektūros elementas – piramidės telkimo modulis (angl. *pyramid pooling module*), kuriuo architektūra išgauna globalaus konteksto informaciją agreguojant atskirų regionų kontekstus (prielaida – kartu naudojami bendri (globalūs) bei lokalūs požymiai gerina segmentavimo efektyvumą). Naudojant keturis atskirus skirtingų dydžių telkimo sluoksnius pirmiausia išgaunama skirtingų mastelių kontekstinė informacija, kuri vėliau jungiama ir perduodama konvoliuciniams sluoksniams, išvedantiems globalią informaciją apibendrinančią požymių žemėlapią (žr. 20 pav.)

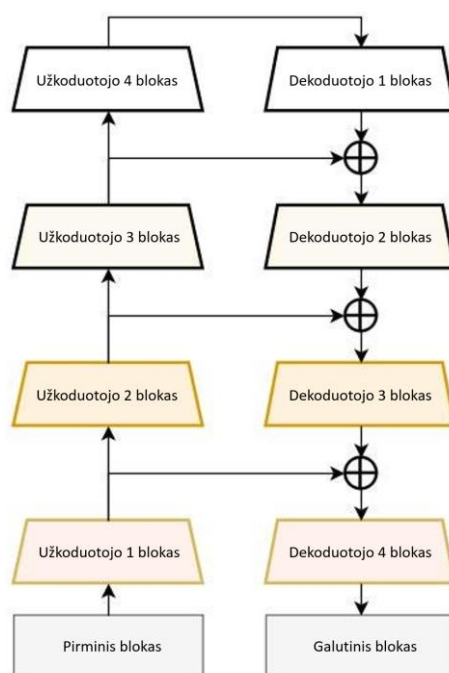


20 pav. PSPNet architektūra [163]

¹⁵ https://d2l.ai/chapter_convolutional-modern/resnet.html

2.3.1.3. LinkNet

Sprendžiant semantinio segmentavimo užduotis autonominio vairavimo ar panašiuose kontekstuose svarbus yra ne tik modelių segmentavimo efektyvumas, tačiau ir skaičiavimo resursų poreikis. Šiam tikslui Purdue (JAV) universiteto mokslininkai suprojektavo lengvasvorę (~11,5 mln. parametru) LinkNet architektūrą, galinčią atlikti segmentavimo užduotis pasitelkiant integruotų prietaisų (angl. *embedded device*) skaičiavimo resursus. Šioje architektūroje po kiekvieno užkodavimo dalies dimensionalumo mažinimo (angl. *downsampling*) bloko išmokta informacija dalinamasi su atitinkamu dekodavimo dalies bloku naudojantis šoniniais sujungimais (angl. *lateral, skip connections*) (žr. 21 pav.) Toks dalinimasis leidžia pasiekti gerus segmentavimo rezultatus bei sumažinti parametru skaičių sujungiant aukšto ir žemo lygio požymių informaciją.

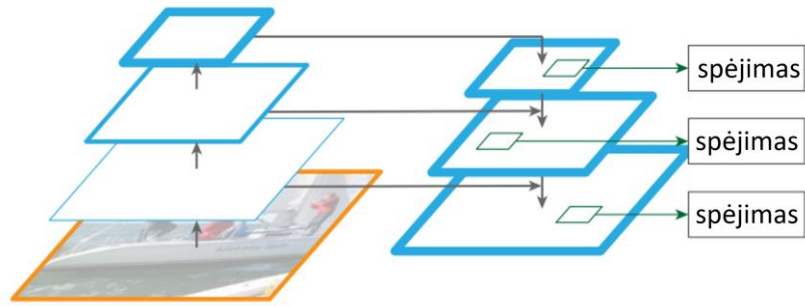


21 pav. LinkNet architektūra [164]

LinkNet užkodavimo dalis naudoja į ResNet panašius konvoliucinius blokus, tačiau kaip ir kitoms užkodavimo-dekodavimo architektūroms, šiam modeliui taip pat galima pritaikyti įvairius bazinius modelius.

2.3.1.4. FPN

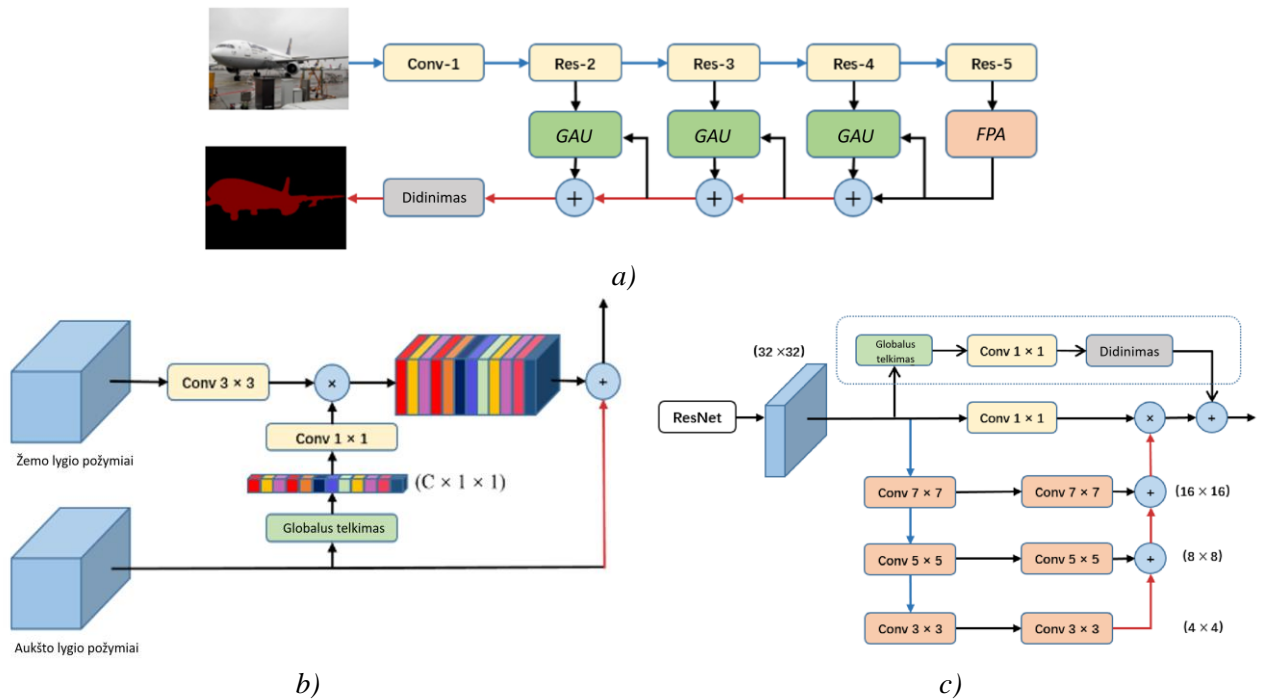
Semantinio segmentavimo ir objektų aptikimo užduotims Lin'o et al. pristatyta FPN architektūra bei įvairios jos konfigūracijos iki šiol plačiai taikomos kompiuterinės regos aplikacijoms. Šio tipo architektūra panaudoja žemos rezoliucijos, tačiau semantiškai stiprių indikacijų požymius bei semantiškai silpnus aukštos rezoliucijos požymius siekiant suprasti tiek bendrą paveikslėlio kontekstą, tiek jo detales pagal požymių piramidę. Modelį sudaro du pagrindiniai informacijos sklaidos keliai: „iš apačios į viršų“ (angl. *bottom-up*) (užkodavimo dalis) ir „iš viršaus į apačią“ (angl. *top-down*) (dekodavimo dalis) (žr. 22 pav.) „Iš apačios į viršų“ dalis sudaryta iš konvoliucinių sluoksnių sekos, skirtos skirtingų rezoliucijų požymių žemėlapiams sudaryti. Šie požymiai jungiami su „iš viršaus į apačią“ dalies sluoksniais, generuojančiais didesnės rezoliucijos požymių žemėlapius, pasitelkiant šoninius jungimus. Modelio spėjimas sudaromas skirtingose požymių piramidės lygiuose, o galutinė išvestis gaunama šias dalis sujungiant.



22 pav. FPN architektūros principo iliustracija [165]

2.3.1.5. PAN

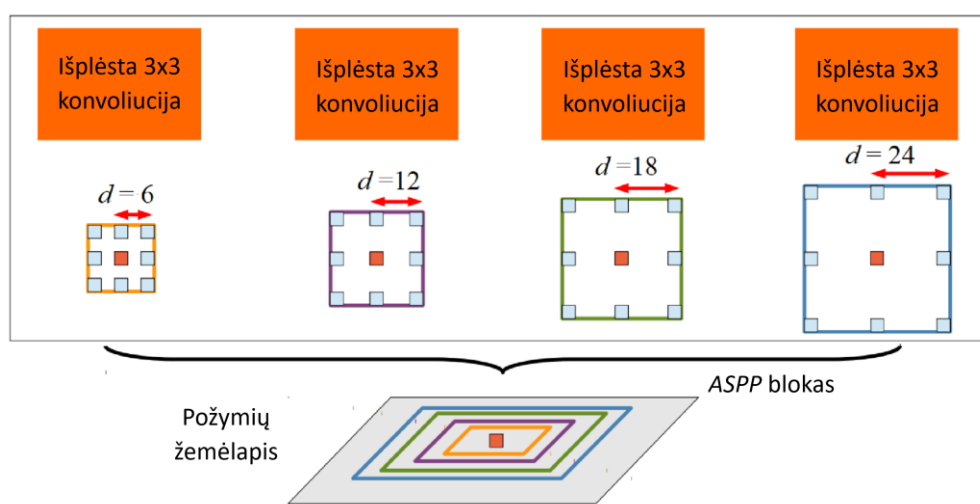
PAN architektūra, Wang 'o et al. pristatyta 2018 m., taip pat pagrįsta požymių piramidės – skirtingų mastelių požymių žemėlapių rinkinio – principu. Skirtingai nuo kitų modelių, PAN naudoja inovatyvius požymių piramidės dėmesio (angl. *feature pyramid attention module, FPA*) bei globalaus dėmesio didinimo (angl. *global attention upsample, GAU*) modulius selektyviam skirtingų lygių informacijos išrinkimui. *FPA* modulio tikslas – taškų lygmens dėmesio informacijos parinkimas iš užkoduotojo išvesties. Modulis sujungia trijų skirtingų lygių požymių piramidės dalis (panašiai kaip FPN), ši struktūra integruojama ir sudauginama su pradine užkoduotojo išvestimi. Paskutinis modulio žingsnis – gautosios struktūros sudėtis su pradinio užkoduotojo išvesties globalaus telkimo (angl. *global pooling*), konvoliucijos bei dimensijos didinimo sluoksnių rezultatais. *GAU* moduliui siekiama pateikti globalų kontekstą (aukštesnio lygio informacija – dekoduoja blokai) kaip nuorodą žemesnio lygio (užkoduotojo blokų) požymių žemėlapiams (žr. 23 pav.) Autoriai pademonstravo, jog ši konfigūracija pasiekia puikius rezultatus segmentuojant *PASCAL VOC* bei *Cityscapes* duomenų rinkinius.



23 pav. PAN architektūra: modelis (a), GAU blokas (b), FPA modulis (c) [166]

2.3.1.6. Deeplab

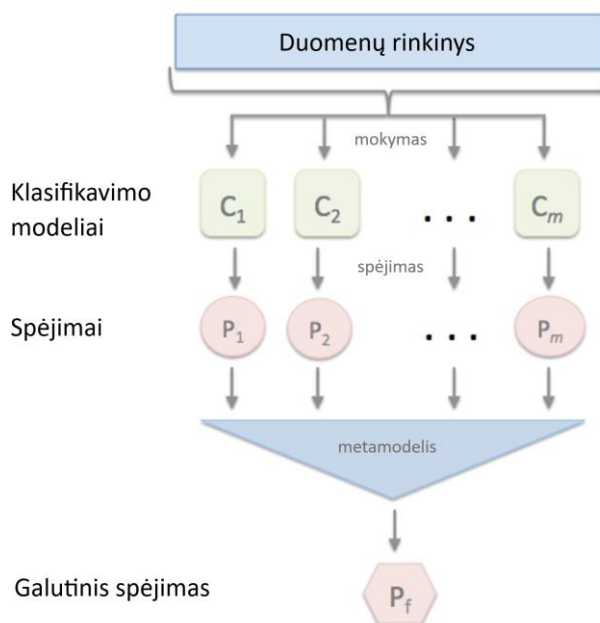
„Deeplabv3“ (Deeplab) – viena naujausių „Deeplab“ giliojo mokymosi architektūrų šeimos konfigūracijų. Šis modelis pritaiko įvairių mastelių kontekstinės informacijos kodavimą bei įvairių rezoliucijų požymių išskyrimą naudojant išplėstus konvoliucinius filtrus (angl. *dilated convolutions*). Tokia konvoliucijos operacija leidžia tarsi padidinti patį filtrą (filtro „renkamą“ informaciją) nepaveikiant parametrų skaičiaus. Pagrindinė struktūrinė tinklo dalis – išplėsto erdvinės piramidės sutelkimo modulis (angl. *atrous spatial pyramid pooling*, *ASPP*) (žr. 24 pav.) *ASPP* bloką sudaro kelių lygių (skirtingų dimensijų) požymių žemėlapio apdorojimas išplėstų konvoliucijų filtrais su skirtingu išplėtimo koeficientu. Tokia architektūra padeda modeliui išgauti požymių žemėlapio informaciją skirtingose dimensijose neprarandant erdvinės rezoliucijos.



24 pav. *ASPP* blokas [167]

2.4. Modelių kolektyvai

Modelių jungimas į daugybines struktūras (angl. *stacking*) ansamblių sudarymui – vienas paprasčiausių ir populiariausių būdų sujungti modelių įgytas žinias ir (tikėtina) gauti geresnius rezultatus dėl šių žinių apibendrinimo [170]. Jungimo metodika pagrįsta tuo, jog daugelio modelių, ypač naudojančių skirtingas architektūras ar hiperparametrus, sujungimas padeda išnaudoti kiekvieno individualaus modelio stiprybes ir mažinti atskirų modelio klaidų poveikį. Paprasčiausiu atveju kiekvienas modelis daro sprendimą duotajai įvesčiai, tuomet modelių išvestys apibendrinamos ir daromas galutinis sprendimas. Priklausomai nuo poreikio ir / ar užduoties konteksto, gali būti taikomi skirtingi apibendrinimo mechanizmai, pavyzdžiui rezultatų vidurkinimas, balsavimas (paprastosios daugumos, pasirenkant tam tikrą slenkstį) ir pan.



25 pav. Modelių jungimo algoritmo iliustracija¹⁶

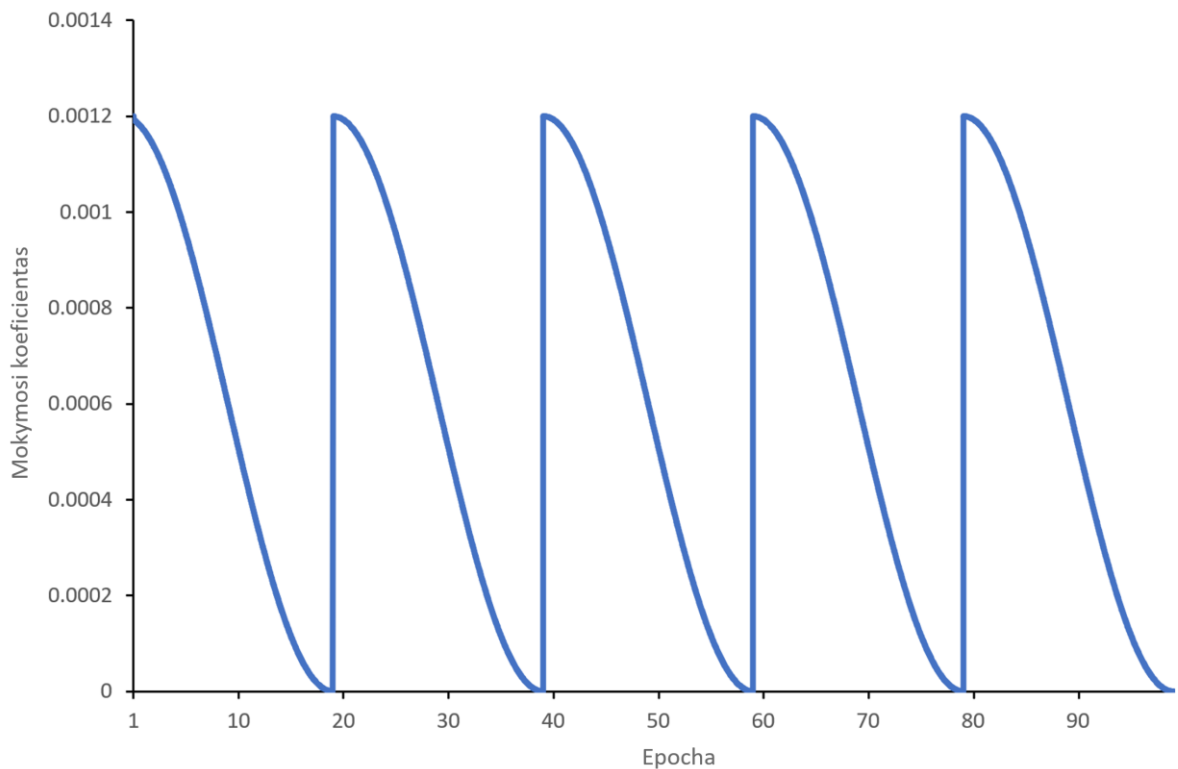
Modelių jungimo metodika (žr. 25 pav.) projekte taikoma sudaryti kolektyvą iš kelių (dviejų bei penkių) modelių spėjimų: šio etapo metu į grafinės plokštės atmintį vienas po kito įkraunami skirtingų architektūrų modeliai, atliekamas spėjimas paketais, rezultatai išsaugomi ir vidurkinami (metamodelis – visų spėjimų vienam įvesties taškui aritmetinio vidurkio skaičiavimo algoritmas). Sujungtų modelių rezultatų vidurkis priskiriamas galutiniam segmentavimo rezultatui.

2.4.1. Momentinis ansambliavimas

Momentinės kopijos ansambliavimo idėja (angl. *snapshot ensemble*), pristatyta Huang 'o et al. [171], pasitelkia kosinusinio atvėsimo (angl. *cosine annealing*) techniką modelio mokymosi koeficientui valdyti taip, jog būtų gaunami keli pilnai paruošti modeliai vienos mokymosi sekos metu. Ši metodika vieno mokymosi etapo metu cikliška mažina mokymosi koeficiento reikšmę po kiekvienos epochos ir grąžina reikšmę į pradinę po tam tikro jų skaičiaus. Mokymosi koeficiento režimo (angl. *learning rate schedule*) metu gaunama į kosinuso funkciją panaši kreivė (žr. 26 pav.) Pasitelkiant tokį koeficiento keitimą norima skatinti modelį konverguoti lokalaus minimumo taške po tam tikro epochų skaičiaus, o pakeitus mokymosi koeficiento reikšmę modelį tarsi sutrikdyti ir išstumti iš šio minimumo taško: procesą kartojant tikimasi gauti kelis konvergavimo taškus.

Pats momentinis ansamblis gaunamas išsaugant modelio svorius prieš kiekvieną mokymosi koeficiento padidinimą. Kartojant ciklą m kartų tikimasi turėti m momentinių modelio kopijų, kurias galima jungti į kolektyvą. Huang 'as et al. ansambliavimą testavimo metu pateikė kaip n ($n \leq m$) paskutinių gautų momentinių modelio kopijų *softmax* aktyvacijos funkcijos išvesčių vidurkį. Ši metodika bus taikoma ir projekto tyrimų metu.

¹⁶ https://rasbt.github.io/mlxtend/user_guide/classifier/StackingClassifier/

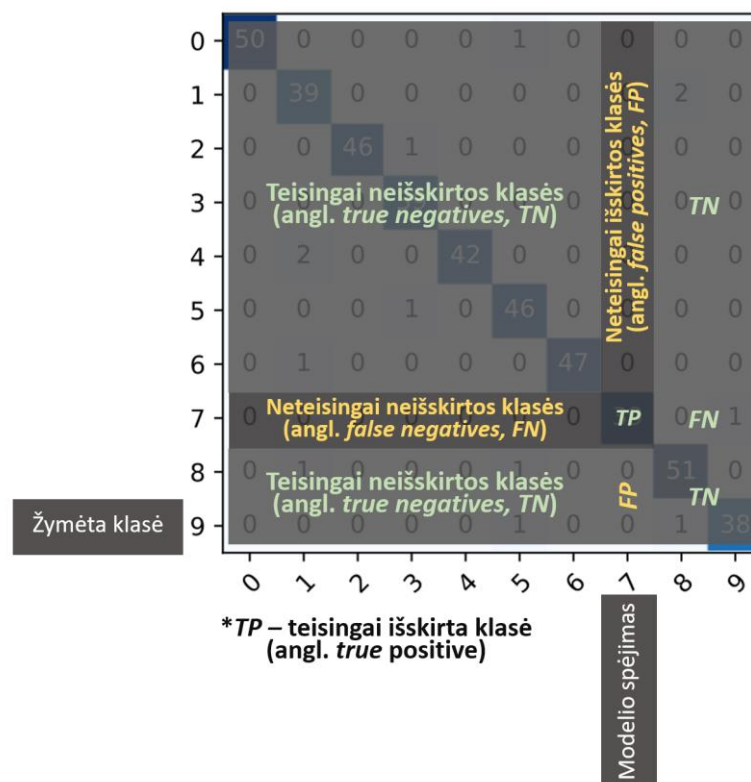


26 pav. Mokymosi koeficiento grafikas pritaikius kosinusinį atvėsimą (iš viso 100 epochų kas 20 epochų išsaugant momentinį modelį)

2.4.2. Vertinimo rodikliai

Semantinio segmentavimo efektyvumui vertinti naudojamos kiekybinės metrikos, nurodančios modelio gebėjimą teisingai priskirti atskirą įvesties paveikslėlio tašką tam tikrai klasei. Toliau bus aptariamai pagrindiniai šiame darbe naudojami segmentavimo vertinimo rodikliai.

Semantinio segmentavimo (taip pat ir kitų mašininio mokymosi bei statistinių klasifikacijų) metrikas galima skaičiuoti naudojant sumaišymo matricą (angl. *confusion matrix*). Kiekviena šios matricos eilutė išskiria žymėtas klases, kiekvienas stulpelis – modelio spėtas klases. Eilučių ir stulpelių susikirtimuose kiekvienai klasei (klasių rinkiniui) išskiriami keturi taškai (jų rinkiniai): teisingai išskirta klasė (angl. *true positive*), neteisingai išskirta klasė (angl. *false positive*), teisingai neišskirta klasė (angl. *true negative*), neteisingai neišskirta klasė (angl. *false negative*). Segmentavimo atveju stulpelių ir eilučių sandūra fiksuoja žymėtų ir nuspėtų taškų tai klasei skaičių (vienos įvesties, arba (sudedant) viso duomenų rinkinio) (žr. 27 pav.) Pavyzdžiui, matricos įstrižainės suma – visų klasių teisingai išskirtų taškų skaičius.



27 pav. Sumaišymo matricos pavyzdys kelių klasių spėjimams

Pagal sumaišymo matricą galima skaičiuoti algoritmų rezultatyvumo metrikas. Segmentavimo efektyvumui vertinti dažniausiai naudojamas *Intersection-Over-Union* (IOU) rodiklis, dar vadinamas Džakardo indeksu (angl. *Jaccard index*). Šis kriterijus suprantamas kaip rinkinio žymėjimų ir modelio spėjimų persidengimo sritis, padalyta iš jų sąjungos (žr. (12) formulę).

$$IOU = \frac{|A \cap B|}{|A \cup B|} \quad (12)$$

čia A – modelio nuspėto segmentavimo sritis, B – žymėta segmentavimo sritis.

Semantiniam segmentavimui (12) galima perrašyti (13) formule toliau:

$$IOU = \frac{TP}{TP + FN + FP} \quad (13)$$

čia (atskiroms klasėms) TP – teisingai modelio priskirtų taškų skaičius, FN – neteisingai modelio nepriskirtų taškų skaičius, FP – neteisingai modelio priskirtų taškų skaičius.

Segmentuojant įvestį su keliomis klasėmis naudojamas vidutinio IOU (angl. *mean Intersection-Over-Union*, mIOU) įvertis. Ši metrika skaičiuojama kaip vertinamų klasių (galima pasirinkti, kurių klasių neįtraukti; darbo metu nevertinama foninė klasė) IOU rodiklio vidurkis:

$$mIOU = \frac{\sum_{i=1}^n IOU_i}{n} \quad (14)$$

čia IOU_i – i -osios klasės IOU rodiklis.

Taip pat segmentavimo efektyvumą galima vertinti ir vadinamuoju F1 įverčiu (*Dice* koeficientu) [172]. Šis rodiklis dažnai naudojamas, kai duomenų klasės yra išbalansuotos. Metrikos skaičiavimas pateikiamas (15) formule.

$$F1 = \frac{2TP}{2TP + FN + FP} \quad (15)$$

čia (atskiroms klasėms) TP – teisingai modelio priskirtų taškų skaičius, FN – neteisingai modelio nepriskirtų taškų skaičius, FP – neteisingai modelio priskirtų taškų skaičius.

Atvejais, kuomet semantinis segmentavimas atliekamas su daugiau nei dviem klasėmis, galima skaičiuoti kelis metrikų vidurkių (klasių atžvilgiu) variantus: mikro- bei makrovidurkį (žr. (16, 17) formules). Makrovidurkis skaičiuojamas pagal atskirus kiekvienos klasės rodiklius, mikrovidurkis gaunamas agreguojant visų klasių rezultatus ir skaičiuojant bendrąją metriką. Šiame projekte naudojami abu rodiklių vidurkių variantai.

$$Metrika_{makro} = \frac{\sum_{i=1}^n Metrika_i}{n} \quad (16)$$

čia $Metrika_i$ – i -osios klasės rodiklis.

$$Metrika_{mikro} = Metrika \left(\sum_{i=1}^n TP_i, \sum_{i=1}^n TN_i, \sum_{i=1}^n FP_i, \sum_{i=1}^n FN_i \right) \quad (17)$$

čia TP_i, TN_i, FP_i, FN_i – modelio pataikymai ir klaidos iš sumaišymo matricos i -ajai klasei.

2.5. Modelių kolektyvo distiliavimas

Viena paprasčiausių žinių distiliavimo metodikų – tai modelių koncentravimas mokytojo-studento (angl. *teacher-student*) principu. Tokio mokymosi metu kartu naudojami du modeliai: mokytojas, dažniausiai sudėtingesnis, didesnis parametų kiekiu algoritmas, skirtas padėti mokyti (perduoti žinias) mažesnį modelį (studentą). Šios technikos tikslas – panaudoti naudingą informaciją (klasių pasiskirstymus skirtingame (idealiu atveju – optimaliame) konvergavimo taške) modeliui-studentui neruošiant naujų duomenų rinkinio elementų. Modelis-studentas mokomas mažinti skirtumus tarp savo ir modelio-mokytojo išvesčių toms pačioms įvestims. Šis skirtumas išreiškiamas modifikuota klaidos funkcija, kuomet iš dalies atsižvelgiama į modelio-mokytojo ir modelio-studentų išvesčių – klasių pasiskirstymų, dažniausiai išreiškiamų „minkštomis“ tikimybėmis (angl. *soft targets*) – neatitikimus. Tokia klaidos funkcija pateikiama (18) formule toliau:

$$LOSS = (1 - \alpha) \cdot TARGET_LOSS + \alpha \cdot DIST_LOSS(y_{student_softmax}, y_{teacher_softmax}) \quad (18)$$

čia $TARGET_LOSS$ – modelio-studento klaidos reikšmė, $DIST_LOSS$ – distiliavimo klaida, α – parametras, valdantis klaidų proporciją.

Egzistuoja daug įvairių distiliavimo klaidos konfigūracijų, viena tokių – Kulbako (Kullback) ir Leiblerio (Leibler) funkcija paremta klaida (angl. *Kullback-Leibler divergence loss*), dar vadinama santykinės entropijos klaida. Bendroju atveju ši klaida išreiškia skirtumą tarp tikimybių skirstinių P ir Q ir yra interpretuojama kaip tikėtina papildoma (angl. *excess*) Šenono informacija modeliuojant Q skirstiniu, kai tikrasis skirstinys yra P . Bendroju atveju semantiniam segmentavimui šios klaidos

skaičiavimas apibrėžiamas pagal (19) formulę. Projekte taikant modelių distiliavimą naudojama būtent šio tipo klaida.

$$L(y_{pred}, y_{true}) = y_{true} \cdot \log \frac{y_{true}}{y_{pred}} = y_{true} \cdot (\log y_{true} - \log y_{pred}) \quad (19)$$

čia y_{pred} – modelio spėjimai, y_{true} – tikrosios klasių žymos.

3. Tyrimų rezultatai

Šiame skyriuje aprašomi naudoti skaičiavimo ištekliai, programiniai įrankiai ir segmentavimo modelių paruošimui naudoti duomenų rinkiniai. Tolesniuose poskyriuose pristatomi giliojo mokymosi modelių duomenų rinkinių segmentavimo efektyvumai atskirų modelių, modelių ansamblių bei distiliuotų modelių ansamblių konfigūracijoms.

3.1. Uždavinys, programinė bei techninė įranga

Projekto metu sprendžiamas semantinio segmentavimo uždavinys, kuomet siekiama priskirti teisingą klasės žymą (angl. *label*) kiekvienam 2D paveikslėlio taškui. CNN siekia „suprasti“ paveikslėlių taškų lygiu ir atskirti semantines klases viena nuo kitos kiekvieną tašką susiejant su atskira klase ir taip sukuriant atskirus semantinius regionus. Magistrinio projekto kontekste semantinis segmentavimas suprantamas kaip prižiūrimo mokymosi (angl. *supervised learning*) uždavinys, kurio metu 3 kanalų (*RGB* formato (angl. *red green blue*) paveikslėliai segmentuojami taškų lygmeniu be persidengimo, priklausomai nuo konfigūracijos naudojant vieną arba kelias objektų žymes.

Eksperimentams atlikti buvo naudojamos kelios techninės įrangos konfigūracijos:

- stacionarus kompiuteris su 32 GB darbinės atminties, *i7-6700K* procesoriumi, *Nvidia GTX 1080* (8 GB vaizdinės atminties) grafiniu procesoriumi;
- stacionarus kompiuteris su 64 GB darbinės atminties, *i9-12900K* procesoriumi, *Nvidia GTX 3090* (24 GB vaizdinės atminties) grafiniu procesoriumi;
- serveris su 527 GB darbinės atminties, *AMD EPYC 7452* procesoriumi, dviejais *Nvidia Ampere A100* (40 GB vaizdinės atminties) grafiniais procesoriais.

Eksperimentams atlikti buvo naudoti šie programiniai įrankiai:

- *Microsoft Visual Studio Code* programavimo aplinka, *Python* programavimo kalbos 3.8.2 versija;
- *Keras* (versija 2.8.0) ir *Pytorch* (versija 1.11.0 (*Cuda 11.3*)) programiniai karkasai panaudoti neuroninių tinklų konfigūracijų sudarymui, mokymui ir testavimui, *albuementations* (versija 1.1.0) biblioteka pasitelkiama vaizdų augmentavimui.

Modeliams sudaryti buvo naudojamos jų pristatymo straipsniais (tam, kad gautos architektūros būtų panašios į pirminius modelius) bei toliau išvardintomis programinėmis bibliotekomis:

- FPN buvo kuriamas pagal *segmentation_models_pytorch* (versija 0.3.2) bibliotekos variantą;
- LinkNet pagal bibliotekos kodą¹⁷ ir *segmentation_models_pytorch*;
- PAN pagal bibliotekos kodą¹⁸ ir *segmentation_models_pytorch*;
- PSPNet pagal bibliotekos kodą¹⁹ ir *segmentation_models_pytorch*;
- ResNet pagal bibliotekos kodą²⁰ ir *segmentation_models_pytorch*;

¹⁷ <https://github.com/e-lab/pytorch-linknet/blob/master/models/>

¹⁸ <https://github.com/WenmuZhou/PAN.pytorch/blob/master/models>

¹⁹ <https://github.com/hszhao/PSPNet>

²⁰ <https://github.com/pytorch/vision/blob/main/torchvision/models>

- DeepLab pagal bibliotekos kodą²¹ bei *segmentation_models_pytorch*.

Pagrindinių modelių struktūros su Cityscapes segmentavimo pavyzdžiu pateikiamos 8-12 Prieduose.

3.2. Duomenų rinkiniai

Tyrimams atlikti naudoti dviejų tipų duomenys: vienas specializuotų duomenų (jūros dugno mozaikų) bei trys standartiniai semantinio segmentavimo aplikacijoms skirti vaizdų rinkiniai. Visi naudoti duomenys yra prieinami viešai.

Jūros dugno mozaikų rinkinys [173] sudarytas iš Svalbarde (Norvegija) povandeniniu dronu 2-65 m. gylyje filmuotų transektų²², suklijuotų specializuota programine įranga, bei segmentavimo kaukių, tyrėjų sužymėtų naudojantis *LabelBox* [174] platforma (toliau – Arktinis rinkinys). Segmentavimo kaukių informacija nuskaityta bei formatuojama: kiekvienos atskiros klasės atveju mozaikų kaukės parsisiunčiamos, konvertuojamos į tinkamą vaizdinės matricos formatą, suklijuojamos į vientisą mozaiką ir koduojamos vadinamuoju vienietinės aktyvacijos (angl. *one hot*) principu [175]. Gautos suklijuotos mozaikos dalinamos į matricų režius (angl. *patches*) ir išsaugomos. Šių režių masyvai tuomet naudojami modeliui mokytis, testuoti, mozaikų atvaizdavimui sudaryti bei klasių aktyvacijų statistikai skaičiuoti. Projekto metu iš Arktinio rinkinio naudojamos transektos su keturiomis segmentacijos klasėmis: jūrinėmis ofiūromis Ophiuroidea (jūros žvaigždės tipas), žieduotosiomis kirmėlėmis Polychaeta (angl. *tube-dwelling polychaete*), kirmėlinėmis aktinijomis Ceriantharia, rudaisiais dumbliais Laminaria.

Naudojamo Arkties vaizdų duomenų rinkinio pagrindinė informacija pateikiama 10 lentelėje toliau:

10 lentelė. Arkties vaizdų rinkinio informacija (skliausteliuose nurodomas atskirų klasių žymėjimų skaičius)

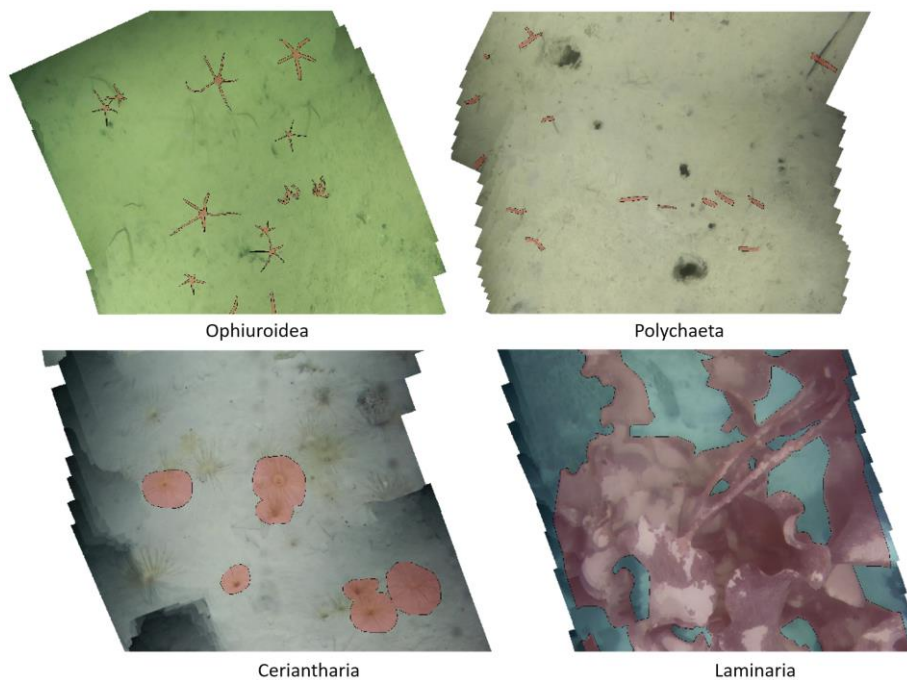
Klasė	Mokymo mozaikos	Testavimo mozaikos
Ophiuroidea	(578) B5_0102_30s (252) B5_0032_30s (46) B6_0215_30s (40) B6_0311_30s (25) B6_0409_30s	(18) B6_0040_30s (3) B2_0458_27s (2) B3a_0536_14s (2) B1_0332_30s
Polychaeta	(138) SJ(2)_1140_30s (101) HB03_0524_30s (81) G3_0928_30s (73) G3_0601_30s (68) G3_0821_30s	(65) G3_0857_30s (59) G3_0705_30s (45) G3_0100_30s (37) G3_0458_30s
Ceriantharia	(64) SJ(1)_0216_30s (41) B4_1509_30s (33) SJ(1)_0324_30s (26) SJ(1)_0250_30s (21) SJ(1)_0738_30s	(16) SJ(2)_0536_30s (9) SJ(2)_1140_30s (7) B7_0438_30s (6) B7_0237_30s (5) D1(2)_1144_30s
Laminaria	(19) D1(1)_0928_30s (13) G3_0100_30s (7) G3_0234_30s	(3) D1(1)_0822_30s (3) SJ(2)_0536_30s (2) B7_0237_30s

²¹ <https://github.com/VainF/DeepLabV3Plus-Pytorch>

²² ekologijoje ir aplinkotyroje augalijos linija arba profilis, pasirinktas geobotaninei analizei arba kartografavimui [181]

Klasė	Mokymo mozaikos	Testavimo mozaikos
	(7) SJ(1)_0738_30s (4) D1(1)_0855_30s	(1) B4_1509_30s

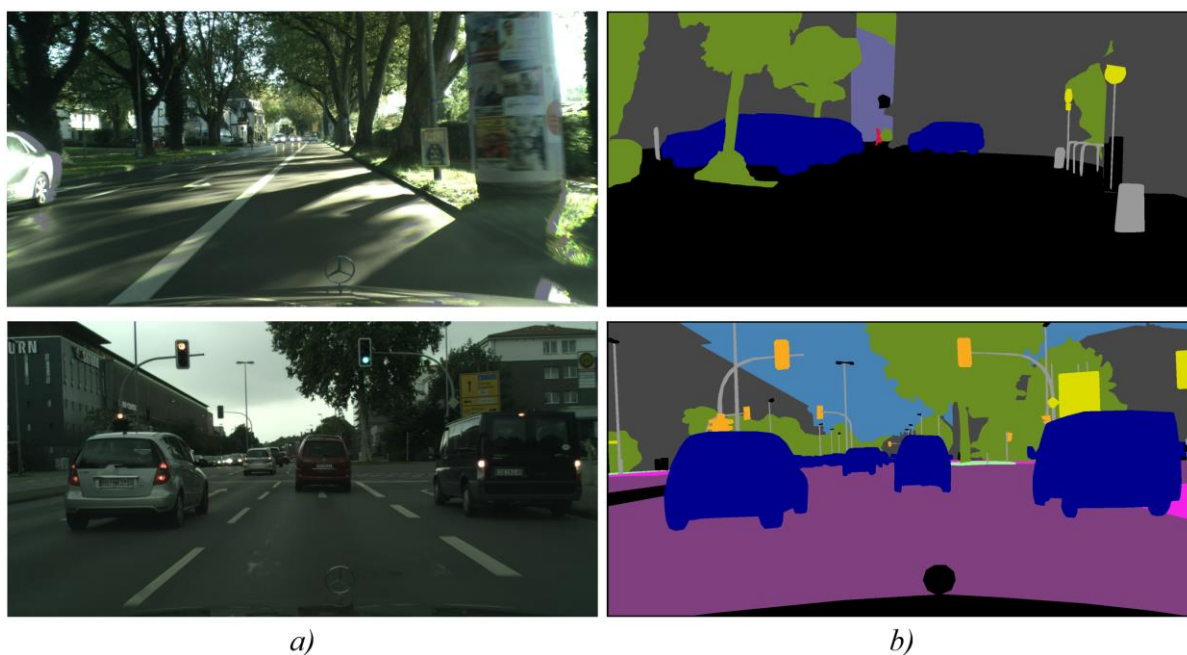
Duomenų rinkinio vaizdų pavyzdžiai pateikiami 28 pav. toliau:



28 pav. Pavyzdiniai sužymėtos transektos keturioms biologinės klasėms

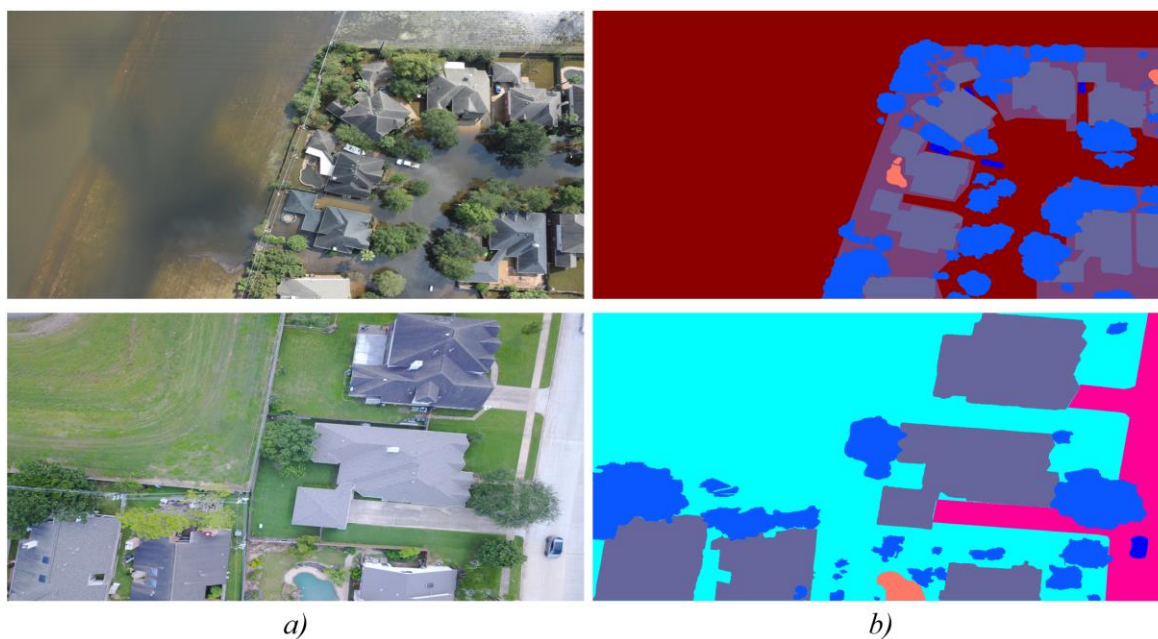
Projekto metu taip pat naudoti trys standartiniai vaizdų rinkiniai – Cityscapes [32], Floodnet [176], Camvid [33]. Toliau šie rinkiniai trumpai apibūdinami.

Cityscapes – penkiasdešimties miestų (daugiausia Vokietijos ir kaimyninių valstybių) gatvių filmuotas duomenų rinkinys [32], skirtas semantiniam miesto erdvių suvokimui kompiuterinės regos aplikacijose gerinti (pvz., autonomiam vairavimui) (žr. 29 pav.) Duomenų rinkinį sudaro 5000 paveikslėlių su taško lygmens kaukėmis (papildomi 20 tūkst. paveikslėlių sužymėti grubiais daugiakampiais), suskirstytų į mokymosi, validavimo bei testavimo aibes. Duomenys filmuoti naudojant ant automobilių prietaiso skydelio sumontuotas aukštos rezoliucijos kameras įvairiais metų laikais (pavasaris, vasara, rudenį), dienos metu bei esant geroms ir vidutinėms oro sąlygoms. Originalus duomenų rinkinys turi 30 atskirų semantinių klasių, tačiau projekte naudojama 19+1 klasių schema (19 atskirų objektų žymėjimų (žr. 13 Priedą) ir viena foninė (angl. *background*) klasė). Dėl to, jog testavimo duomenų aibė nėra viešai prieinama, darbo metu testavimui buvo naudota validavimo aibė.



29 pav. Cityscapes duomenų rinkinio pavyzdžiai: mokymo (pirma eilutė) ir testavimo (antra eilutė) vaizdai (a) ir segmentavimo kaukės (b)

Floodnet – po uragano „Harvis“ (2017 m.) bepilote sistema filmuotų aukštos rezoliucijos vaizdų rinkinys [176], skirtas stichinių nelaimių scenarijų semantiniam segmentavimui (pvz., nuostolių vertinimui (angl. *damage assessment*)). Duomenų rinkinį sudaro 2343 vaizdai iš uragano paveiktų ir nepaveiktų vietovių, filmuotų Teksaso valstijos Bendo apylinkėje (angl. *Bend County*) (žr. 30 pav.) Rinkinį sudaro 9 semantinės klasės (žr. 13 Priedą) ir papildoma foninė klasė, duomenys padalinti į mokymo (~60 %), validavimo (~20 %) bei testavimo (~20 %) aibes. Naudojant Floodnet rinkinį, validavimo bei testavimo rinkiniai sujungiami ir naudojami kaip viena testavimo aibė.



30 pav. Floodnet duomenų rinkinio pavyzdžiai: mokymo (pirma eilutė) ir testavimo (antra eilutė) vaizdai (a) ir segmentavimo kaukės (b)

Camvid (angl. *Cambridge-driving Labeled Video Database*) – semantinio segmentavimo vaizdų duomenų bazė [33], gauta iš ant automobilio prietaisų skydelio sumontuotos kameros Kembridžo (Jungtinės Karalystė) gatvėse nufilmuotų penkių 960x720 tšk. rezoliucijos vaizdo įrašų sekų. Šios sekos buvo padalintos vieno (keturios sekos) ir penkiolikos (viena seka) kadru per sekundę dažniais ir sužymėtos paruošiant 32 klasių semantines kaukes (žr. 31 pav.) Darbo metu panašiausios klasės buvo sujungtos (pagal programinę biblioteką²³) ir gauta 11 (ir viena foninė) semantinių klasių (žr. 13 Priedą).



31 pav. Camvid duomenų rinkinio pavyzdžiai: mokymo (pirma eilutė) ir testavimo (antra eilutė) vaizdai (a) ir segmentavimo kaukės (b)

Tolesniuose poskyriuose aptariami projekto metu gauti rezultatai.

3.3. Empiriniai rezultatai

Atliktus tyrimus galima skirstyti į kelias pagrindines dalis pagal naudotas technikas:

- giliojo mokymosi modeliai mokomi naudojant keturis duomenų rinkinius – Arkties jūros dugno vaizdų, Cityscapes, Floodnet bei Camvid;
- sudaromi heterogeniniai modelių ansambliai (jungiamos skirtingos architektūros);
- modeliai ansambluojami homogeniškai (panaudojant konsinusišią atvėsimą jungiamos identišką architektūrą);
- pritaikomas modelių distiliavimas – studento-mokytojo modeliavimas (kolektyvas distiliuojamas į atskirus modelius).

Tolesniuose skyreliuose pateikiamos modelių mokymo konfigūracijos bei empiriniai eksperimentų rezultatai atskiriems rinkiniams.

²³ <https://github.com/lih627/CamVid>

3.3.1. Arkties duomenų rinkinys

Pirmiesiems eksperimentams naudojamos dvi jūros dugno mozaikos ir mokoma atpažinti vieną biologinę klasę – Ophiuroidea. Mozaikų dydžiai – 1487x6775 tšk. („B5_0032_30s“, toliau – „mozaika_1“) bei 1488x7862 tšk. („B5_0102_30s“, toliau – „mozaika_2“). Mokymo rinkiniui paruošti mozaikos skaidomos į 288x288 tšk. dydžio režius su 144 tšk. persidengimu, galutiniams rezultatams formuoti suprogramuotas algoritmas nepersidengiantiems rezultatams gauti (pradiniai persidengimai suvidurkinami ir konstruojama visos transektos segmentavimo kaukė).

11 lentelė. Arkties duomenų mokymo parametrai vienai biologinei klasei

Parametras	Reikšmė
Išmetimo koeficientas	0,25
Mokymosi koeficientas	0,00012
Aktyvacijos funkcija	<i>sigmoid</i>
Nuostolio funkcija	<i>binary focal loss</i>
Paketo dydis	8
Įvestis	288x288 tšk. režis
Įvesčių skaičius	2 transektos
Vaizdų augmentacija	Horizontalus, vertikalus pasukimas, <i>RoWS</i> algoritmas [177]
Optimizavimo algoritmas	SGD [178]
Mokymo epochos	100
Užkoduotojo modelio svoriai	Imagenet

Pirmųjų eksperimentų metu mokyti PSPNet, FPN bei LinkNet modeliai, užkoduotojo modeliui naudojamos ResNet-50, taip pat *SEResNeXt-50* bei *EfficientNetB3* architektūros.

Atskirų modelių testavimo rezultatai sudarytai konfigūracijai (žr. 11 lentelę) po 100 mokymo epochų pateikiami 12 lentelėje toliau:

12 lentelė. Arkties rinkinio testavimo rezultatai vienai biologinei klasei

Architektūra	Bazinis modelis	Mokymosi mozaika	Testavimo mozaika	Ophiuroidea klasės IOU
PSPNet	<i>ResNet-50</i>	mozaika_1	mozaika_2	0,578
		mozaika_2	mozaika_1	0,527
	<i>SEResNeXt-50</i>	mozaika_1	mozaika_2	0,584
		mozaika_2	mozaika_1	0,526
	<i>EfficientNetB3</i>	mozaika_1	mozaika_2	0,590
		mozaika_2	mozaika_1	0,522
LinkNet	<i>ResNet-50</i>	mozaika_1	mozaika_2	0,620
		mozaika_2	mozaika_1	0,568
	<i>SEResNeXt-50</i>	mozaika_1	mozaika_2	0,613
		mozaika_2	mozaika_1	0,568
	<i>EfficientNetB3</i>	mozaika_1	mozaika_2	0,618
		mozaika_2	mozaika_1	0,568

Architektūra	Bazinis modelis	Mokymosi mozaika	Testavimo mozaika	Ophiuroidea klasės IOU
		mozaika_2	mozaika_1	0,560
FPN	ResNet-50	mozaika_1	mozaika_2	0,617
		mozaika_2	mozaika_1	0,547
	SEResNeXt-50	mozaika_1	mozaika_2	0,615
		mozaika_2	mozaika_1	0,550
	EfficientNetB3	mozaika_1	mozaika_2	0,610
		mozaika_2	mozaika_1	0,550

Pagal 12 lentelėje pateiktus rezultatus matoma, jog visi modeliai pasiekia patenkinamą segmentavimo rezultatą (>50 % IOU). Efektyviausiai „mozaika_2“ segmentuojama naudojant LinkNet architektūrą bei ResNet-50 bazinį modelį (61,2 % IOU). Prasčiausią segmentavimo rezultatą pademonstravo PSPNet modelis su ResNet-50 bazine struktūra: mozaika „mozaika_2“ buvo segmentuota su 57,8 % IOU efektyvumu. „Mozaika_1“ transektai fiksuoti dar prastesni rezultatai: LinkNet su *SEResNeXt-50* baziniu modeliu pasiekė 56,8 % IOU, PSPNet ir *EfficientNetB3* bazinis modelis – 52,2 % IOU. Pastebima, jog efektyviausias bei prasčiausias segmentavimo rezultatas abiem mozaikų mokymo-testavimo konfigūracijoms buvo pasiektas su atitinkamai LinkNet bei PSPNet modeliais.

Dvi geriausiai pasirodžiusių modelių konfigūracijos toliau testuojamos juos sujungiant į heterogeninį ansamblį (naudojamos dvi efektyviausiai mozaiką segmentuojančios architektūros LinkNet bei FPN, išvestys vidurkinamos). Testavimo rezultatai pateikiami 13 lentelėje toliau:

13 lentelė. Arkties rinkinio segmentavimo ansambliu rezultatai vienai biologinei klasei

Architektūra – modelis (bazinis modelis)	Mokymosi mozaika	Testavimo mozaika	Ophiuroidea klasės IOU
<i>LinkNet (ResNet-50) + FPN (ResNet-50)</i>	mozaika_1	mozaika_2	0,632
	mozaika_2	mozaika_1	0,547
<i>LinkNet (ResNet-50) + FPN (SEResNeXt-50)</i>	mozaika_1	mozaika_2	0,629
	mozaika_2	mozaika_1	0,548
<i>LinkNet (ResNet-50) + FPN (EfficientNetB3)</i>	mozaika_1	mozaika_2	0,627
	mozaika_2	mozaika_1	0,547
<i>LinkNet (EfficientNetB3) + FPN (EfficientNetB3)</i>	mozaika_1	mozaika_2	0,623
	mozaika_2	mozaika_1	0,545
<i>LinkNet (EfficientNetB3) + FPN (ResNet-50)</i>	mozaika_1	mozaika_2	0,631
	mozaika_2	mozaika_1	0,541
<i>LinkNet (EfficientNetB3) + FPN (SEResNeXt-50)</i>	mozaika_1	mozaika_2	0,626
	mozaika_2	mozaika_1	0,543
<i>LinkNet (SEResNeXt-50) + FPN (SEResNeXt-50)</i>	mozaika_1	mozaika_2	0,623
	mozaika_2	mozaika_1	0,550
<i>LinkNet (SEResNeXt-50) + FPN (ResNet-50)</i>	mozaika_1	mozaika_2	0,629
	mozaika_2	mozaika_1	0,546
<i>LinkNet (SEResNeXt-50) + FPN (EfficientNetB3)</i>	mozaika_1	mozaika_2	0,623
	mozaika_2	mozaika_1	0,548

Pagal gautus rezultatus matoma, jog homogeninis modelis pagerino konfigūracijos „mozaika_1“ (mokymas), „mozaika_2“ (testavimas) rezultatus: pasiekiamas maksimalus Ophiuroidea klasės

segmentavimo IOU buvo 63,2 % naudojant LinkNet bei FPN kombinaciją ir ResNet-50 užkoduotojo modelį. Taip pat pastebima, jog dviejų modelių ansamblis nepasiekė geresnių rezultatų antrajai konfigūracijai: aukščiausias IOU efektyvumas siekė 55 % naudojant LinkNet bei FPN su *SEResNeXt-50* baziniu modeliu. Tačiau galima pastebėti, jog kiekvienas ansamblis segmentavo efektyviau už visas pavienių modelių konfigūracijas, išskyrus naudojant geriausiai pasirodžiusį LinkNet modelį.

Tolesniame etape buvo vertinama kosinusinio atvėsimo technika homogeninio ansamblio sudarymui. Šiam tikslui mokytas pračiuose etapuose geriausiai pasirodęs modelis – LinkNet su ResNet-50 užkoduotojo struktūra. Mokymosi metu modelių svoriai išsaugomi 4 kartus (kas 25 epochas, iš viso mokant 100 epochų). Ansamblis sudaromas iš šių keturių modelių. Rezultatai pateikiami 14 lentelėje toliau:

14 lentelė. Momentinio ansamblio testavimo rezultatai biologinei klasei

Architektūra ir bazinis modelis	Mokymosi mozaika	Testavimo mozaika	Ophiuroidea klasės IOU
LinkNet (ResNet-50)	mozaika_1	mozaika_2	0,609
	mozaika_2	mozaika_1	0,532

Pagal gautus rezultatus pastebima, jog homogeninis ansamblis nepralenkė heterogeninio dviejų modelių ansamblio nei vienai konfigūracijai, o geriausias rezultatas pasiektas „mozaika_1“ (mokymas) ir „mozaika_2“ (testavimas) konfigūracijai su 0,609 IOU.

Tęsiant eksperimentus buvo lyginami pavienių modelių, homogeninių bei heterogeninių ansamblių rezultatai skirtingų jūros dugno klasių segmentavimui. Pasitelkiamos penkios skirtingos architektūros: FPN, PAN, PSPNet, LinkNet bei Deeplab su ResNet-18 baziniu modeliu. Mokymo konfigūracija pateikiama 15 lentelėje toliau.

15 lentelė. Arkties rinkinio keturių biologinių klasių mokymo parametrai

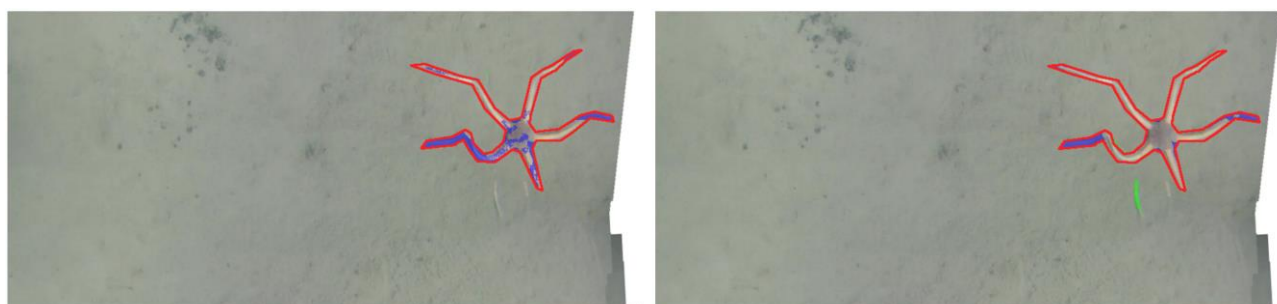
Parametras	Reikšmė
Išmetimo koeficientas	0,3
Mokymosi koeficientas	0,00012
Aktyvacijos funkcija	<i>sigmoid</i>
Nuostolio funkcija	<i>Binary focal loss</i>
Paketo dydis	120
Įvestis	288x288 tšk. režis
Įvesčių skaičius	Klasės transektos pagal 10 lentelę
Vaizdų augmentacija	Horizontalus, vertikalus pasukimas, <i>RoWS</i> algoritmas (angl. <i>removal of water scattering</i>) [177]
Optimizavimo algoritmas	SGD [178]
Mokymo epochos	100
Užkoduotojas	ResNet-18
Užkoduotojo modelio svoriai	Imagenet
Kosinusinio atvėsimo epochos	20

16 lentelėje pateikiami pavienių modelių ir jų homogeninių ansamblių, taip pat heterogeninio modelio segmentavimo efektyvumai, išreikšti IOU, kiekvienai testavimo rinkinio transektai.

16 lentelė. Modelių bei momentinių ansamblių segmentavimo rezultatai (IOU) Ophiuroidea klasei (x5 – homogeninis kolektyvas)

Architektūra	B6_0040	B2_0458	B3a_0536	B1_0332	Bendras
FPN	0,403	0,223	0,159	0,648	0,304
FPN x5	0,400	0,272	0,170	0,666	0,323
PAN	0,410	0,242	0,119	0,666	0,305
PAN x5	0,381	0,249	0,219	0,707	0,329
PSPNet	0,335	0,309	0,193	0,639	0,322
PSPNet x5	0,388	0,288	0,223	0,653	0,339
Deeplab	0,373	0,293	0,210	0,698	0,337
Deeplab x5	0,449	0,226	0,223	0,675	0,342
LinkNet	0,352	0,265	0,160	0,594	0,301
LinkNet x5	0,413	0,296	0,184	0,730	0,349
Ansamblis	0,405	0,270	0,162	0,701	0,327

Gautieji Ophiuroidea klasės segmentavimo rezultatai nebuvo optimalūs (<0,5 IOU). Geriausias apibendrintas segmentavimo rezultatas buvo pasiektas naudojant homogeninį ansamblį LinkNet pagrindu (34,9 % IOU). Kitų modelių segmentavimo efektyvumai buvo panašūs (~0,3 IOU). Pastebėta, jog visos (išskyrus FPN) architektūros modelius jungiant į homogeninius ansamblius pasiekė geresnius rezultatus už heterogeninio ansamblio 32,7 % IOU. Taip pat galima matyti, jog homogeniniai modelių ansambliai pasirodo geriau už pavienių modelius: daugumai atskirų mozaikų segmentacijų (15 atvejų iš 20) homogeniniai ansambliai suveikė efektyviau už vieno modelio segmentavimą. Galima daryti išvadą, jog Ophiuroidea klasei homogeniniai ansambliai turėjo teigiama poveikį stebimam segmentavimo efektyvumui. Pagal 32 pav. pastebima, jog homogeninis LinkNet ansamblis geriau atpažįsta ofiūros dalis ir daro mažiau neteisingai nepažymėto tipo klaidų už atskirą LinkNet modelį. Tačiau ansamblis klaidingai atpažįsta papildomus, į ofiūrų „kojas“ panašius darinius smėlyje.



LinkNet

LinkNet x5

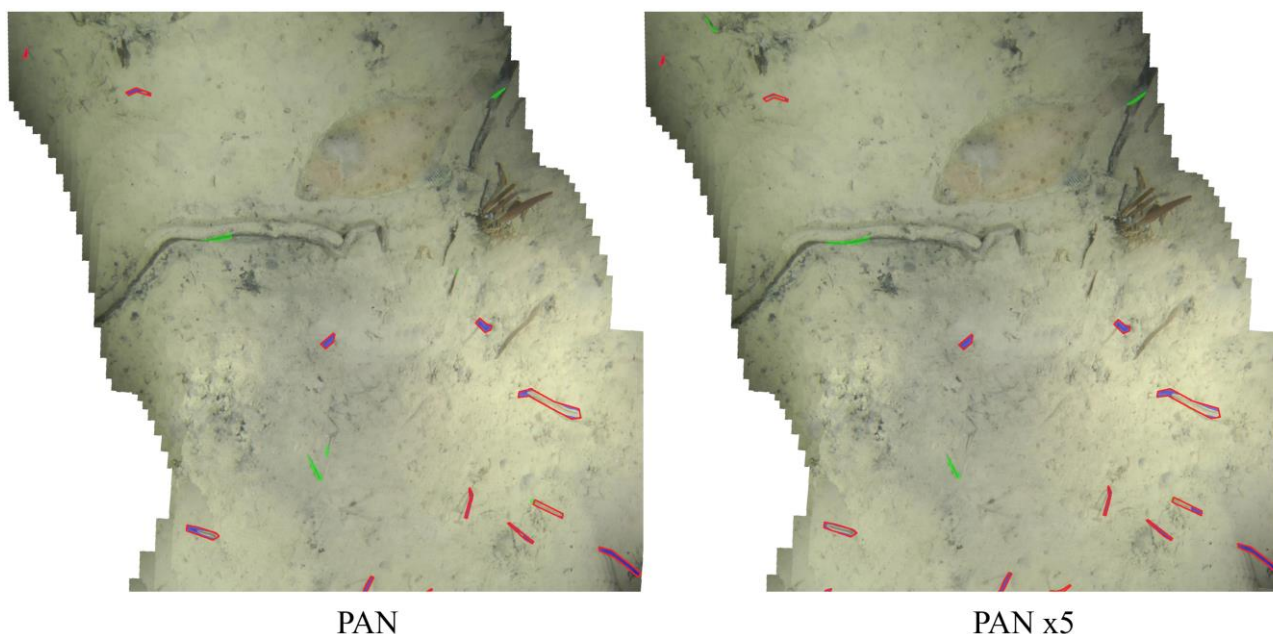
32 pav. „B1_0332“ transektos Ophiuroidea klasės segmentavimas LinkNet modeliu bei homogeniniu ansambliu (x5) (raudonai vaizduojamas ekspertinio žymens apvadas, žaliai – neteisingai žymėta klasė, mėlynai – neteisingai nepažymėta klasė)

Toliau atliekamas Polychaeta klasės segmentavimas. Rezultatai pateikiami 17 lentelėje toliau:

17 lentelė. Modelių bei momentinių ansamblių segmentavimo rezultatai (IOU) Polychaeta klasei (x5 – homogeninis kolektyvas)

Architektūra	G3_0857	G3_0705	G3_0100	G3_0458	Bendras
Deeplab	0,137	0,131	0,145	0,215	0,148
Deeplab x5	0,102	0,127	0,156	0,146	0,128
FPN	0,165	0,212	0,195	0,205	0,193
FPN x5	0,111	0,130	0,176	0,108	0,130
LinkNet	0,219	0,236	0,197	0,212	0,218
LinkNet x5	0,152	0,144	0,188	0,137	0,158
PSPNet	0,201	0,198	0,175	0,168	0,188
PSPNet x5	0,182	0,145	0,183	0,141	0,164
PAN	0,187	0,172	0,179	0,189	0,181
PAN x5	0,174	0,191	0,202	0,198	0,189
Ansamblis	0,157	0,183	0,180	0,212	0,179

Pagal Polychaeta klasės rezultatus pastebima, jog ši klasė segmentuojama palyginus prastai (rezultatai svyruoja nuo ~0,1 iki ~0,2 IOU). Geriausias pasiektas rezultatas – 0,218 IOU segmentuojant LinkNet modeliu. Homogeninio ansamblio formavimas nepasiteisino visoms architektūroms išskyrus PAN (pastebimas minimalus bendrasis IOU padidėjimas nuo 0,181 iki 0,189) tiek atsižvelgiant į bendrus, tiek į atskirus mozaikų segmentavimo rezultatus. Klasės segmentavimui atskiri modeliai (išskyrus PAN atveju) veikė efektyviau už homogeninį ansamblį ir tik segmentuojant „G3_0100“ transektą buvo pastebėtas homogeninių ansamblių pranašumas. Prasti rezultatai gali būti paaiškinami tuo, jog ši semantinė klasė turi klaidinančią šešėlių, panašų į pačią klasę, taip pat ekspertai pažymėjo ne visus klasės objektus, todėl modelis labiau klydo net žymint teisingas klases (žr. 33 pav.)

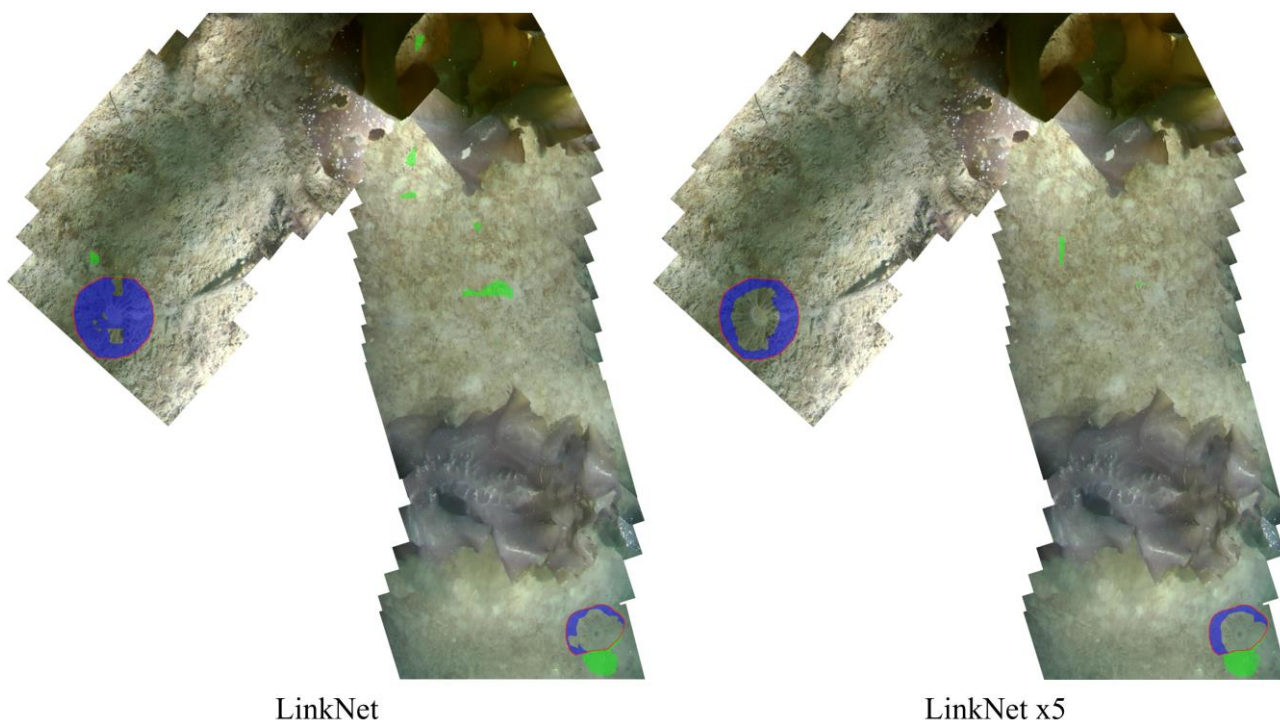


33 pav. „G3_0100“ transektos Polychaeta klasės segmentavimas PAN modeliu bei homogeniniu ansambliu (x5) (raudonai vaizduojamas ekspertinio žymens apvadas, žaliai – neteisingai žymėta klasė, mėlynai – neteisingai nepažymėta klasė)

18 lentelė. Modelių bei momentinių ansamblių segmentavimo rezultatai (IOU) Ceriantharia klasei (x5 – homogeninis kolektyvas)

Architektūra	SJ(2)_0536	SJ(2)_1140	B7_0438	B7_0237	D1(2)_1144	Bendras
PSPNet	0,288	0,258	0,336	0,266	0,052	0,276
PSPNet x5	0,298	0,205	0,315	0,298	0,102	0,275
Deeplab	0,383	0,276	0,500	0,418	0,494	0,418
Deeplab x5	0,322	0,168	0,218	0,307	0,384	0,277
PAN	0,431	0,325	0,321	0,294	0,405	0,361
PAN x5	0,401	0,234	0,244	0,354	0,457	0,335
FPN	0,341	0,246	0,288	0,352	0,524	0,334
FPN x5	0,429	0,207	0,432	0,429	0,410	0,399
LinkNet	0,348	0,246	0,432	0,378	0,511	0,376
LinkNet x5	0,437	0,215	0,435	0,473	0,440	0,416
Ansamblis	0,386	0,238	0,384	0,337	0,377	0,355

Ceriantharia klasės atveju (žr.18 lentelę) geriausi segmentavimo rezultatai buvo pasiekti individualiu Deeplab modeliu (0,418 IOU) bei LinkNet homogeniniu ansambliu (0,416 IOU). Heterogeninis ansamblis pasiekė prastesnius rezultatus ir segmentavo 35,5 % IOU efektyvumu, aplenkdamas PSPNet, FPN atskirus modelius bei PSPNet, Deeplab ir PAN homogeninius kolektyvus. Daugumai modelių konfigūracijų (PAN, Deeplab, PSPNet) homogeninio ansamblio sudarymas nepasiteisino, tačiau LinkNet homogeninis ansamblis pasiekė gana sėkmingus bendrus rezultatus ir pagerino segmentavimo efektyvumą (žr. 34 pav.)

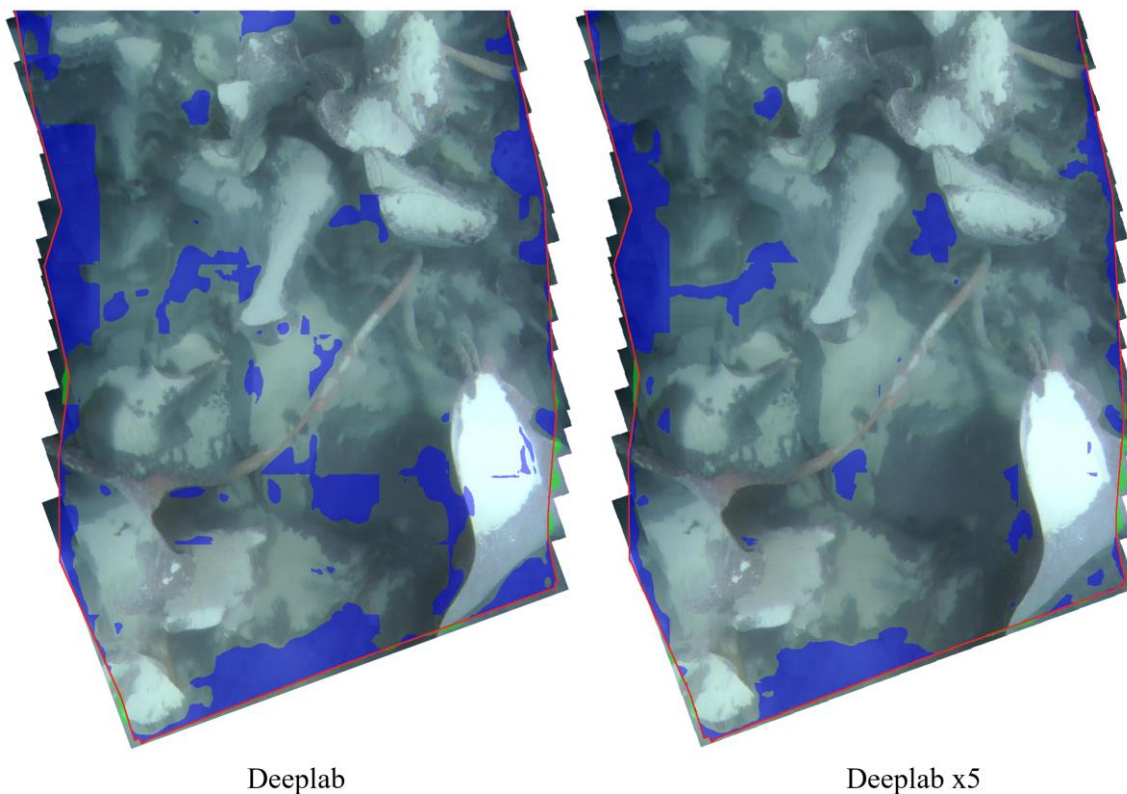


34 pav. „B7_0237“ transektos Ceriantharia klasės segmentavimas LinkNet modeliu bei homogeniniu ansambliu (x5) (raudonai vaizduojamas ekspertinio žymens apvadas, žaliai – neteisingai žymėta klasė, mėlynai – neteisingai nepažymėta klasė)

19 lentelė. Modelių bei momentinių ansamblių segmentavimo rezultatai (IOU) Laminaria klasei (x5 – homogeninis kolektyvas)

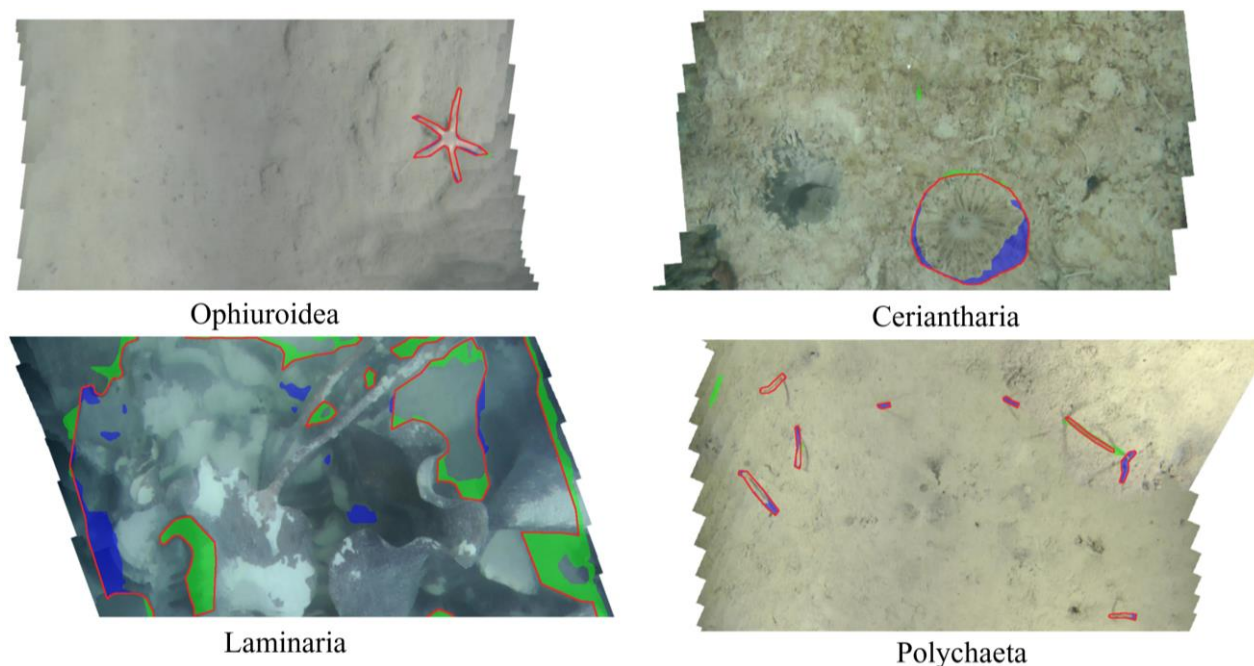
Architektūra	D1(1)_0822	SJ(2)_0536	B7_0237	B7_1509	Bendras
PSPNet	0,736	0,474	0,319	0,229	0,587
PSPNet x5	0,732	0,722	0,345	0,294	0,610
LinkNet	0,796	0,537	0,370	0,352	0,646
LinkNet x5	0,791	0,607	0,409	0,427	0,667
PAN	0,741	0,611	0,348	0,500	0,622
PAN x5	0,785	0,738	0,482	0,223	0,677
FPN	0,800	0,543	0,407	0,423	0,668
FPN x5	0,798	0,718	0,487	0,321	0,693
Deeplab	0,766	0,609	0,428	0,329	0,652
Deeplab x5	0,819	0,648	0,449	0,346	0,694
Ansamblis	0,786	0,686	0,386	0,359	0,659

Geriausi segmentavimo rezultatai šiam duomenų rinkiniui buvo pasiekti Laminaria klasei (žr. 19 lentelę). Tai gali būti iš dalies paaiškinta aukštu klasės padengimo (palyginti su foniniais taškais) procentu. Deeplab modelis pasiekė geriausius rezultatus su 69,4 % IOU, panašiai pasirodė ir FPN su 69,3 % IOU. Visų modelių segmentavimas (išskyrus PSPNet), lyginant su heterogeniniu ansambliu, pagerėjo ruošiant homogeninį kolektyvą. Prasčiausiai pasirodė PSPNet modelis (0,587 IOU vieno modelio atveju, 0,610 IOU homogeninio ansamblio atveju). Taip pat pastebimas santykinai (lyginant su kitų klasių segmentavimu) mažesnis Linknet efektyvumas. Efektyviausiai segmentuojančio Deeplab ansamblio ir atskiro modelio palyginimas pateiktas 35 pav. toliau:



35 pav. „D1(1)_0822“ transektos Laminaria klasės segmentavimas Deeplab modeliu bei homogeniniu ansambliu (x5) (raudonai vaizduojamas ekspertinio žymens apvadas, žaliai – neteisingai žymėta klasė, mėlynai – neteisingai nepažymėta klasė)

Remiantis atliktų eksperimentų jūros dugno vaizdų rinkiniui rezultatais pastebimą tendenciją galima apibendrinti taip: sudėtingų klasių, tokių kaip Polychaeta bei Ceriantharia, segmentavimui bendri atskirų modelių rezultatai lenkė momentinių homogeninių bei heterogeninių ansamblių rezultatus, o Ophiuroidea bei Laminaria klasėms homogeninių ansamblių sudarymas pasiteisino. Segmentuojant Ophiuroidea bei Laminaria klases homogeniniai ansambliai daugumai atvejų lenkė atskirų bei heterogeninių ansamblių efektyvumą, o geriausiai veikė LinkNet (Ophiuroidea) bei Deeplab (Laminaria) homogeniniai ansambliai (žr. 36 pav.)



36 pav. Sėkmingų biologinių klasių segmentavimo Deeplab homogeniniu ansambliu modeliu pavyzdžiai (raudonai vaizduojamas ekspertinio žymens apvadas, žaliai – neteisingai žymėta klasė, mėlynai – neteisingai nepažymėta klasė)

Distiliavimo efektyvumui Arkties rinkinio segmentavime testuoti buvo pasirinkti du pirminių tyrimų metu naudoti modeliai PAN ir FPN, turintys mažiausią parametrų kiekį (PAN ~24,3 mln., FPN~ 26,3 mln. parametrų), taip pat LinkNet modelis. Mokymo metu naudota Kulbako (*Kullback*) ir Leiblerio (*Leibler*) funkcija paremta klaida.

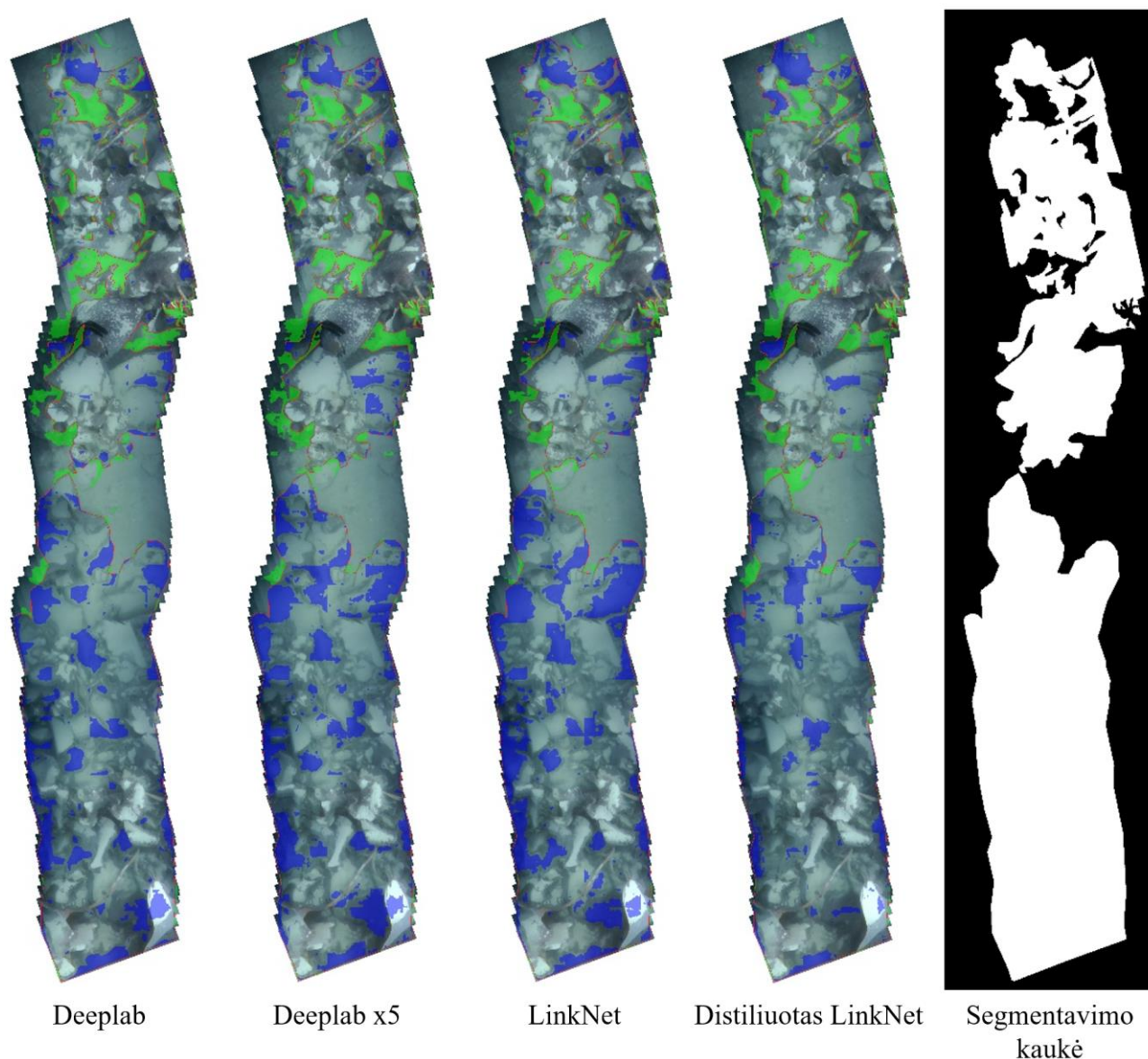
Pirmųjų distiliavimo eksperimentų metu kaip modelį-mokytoją pasirinkta naudoti geriausiai rinkinyje Laminaria klasės segmentavimą atlikusį Deeplab modelį. Dėl padidėjusių resursų reikalavimo buvo pasirinkta mokytį LinkNet su *EfficientNetB3* baziniu modeliu, tačiau mokymo epochų skaičius padidintas iki 200, klaidos balanso parametras α nustatomas į 0,25. Šios konfigūracijos rezultatai pateikiami 20 lentelėje toliau:

20 lentelė. Distiliavimo rezultatai (IOU) Laminaria klasei (x5 – homogeninis kolektyvas, dist. – distiliuotas modelis)

Laminaria mozaikos	Deeplab	Deeplab x5	LinkNet dist.	LinkNet
D1(1)_0822	0,753	0,815	0,781	0,744
B7_0237	0,391	0,554	0,424	0,392
SJ(2)_0536	0,605	0,634	0,546	0,507

Laminaria mozaikos	Deeplab	Deeplab x5	LinkNet dist.	LinkNet
B4_1509	0,278	0,315	0,395	0,357

Matoma, jog geriausi rezultatai visoms transektoms buvo pasiekti naudojant homogeninį Deeplab ansamblį, efektyviausiai segmentuojant „D1(1)_0822“ su 81,5 % IOU. Taip pat pastebima, jog distiliavimas Laminaria klasei pasiteisino: visais keturiais atvejais stebimas LinkNet segmentavimo efektyvumo padidėjimas; didžiausias teigiamas poveikis stebimas segmentuojant „SJ(2)_0536“ mozaiką, kuomet LinkNet atskiro modelio rezultatai palyginus su distiliuotu modeliu padidėja nuo 50,7 % IOU iki 54,6 % IOU. Bendrai, distiliuotas modelis-studentas pasiekia geresnius rezultatus už įprastu būdu mokyta modelį. Galima teigti, jog distiliavimas šiai ribotai klasei segmentavimo konfigūracijai buvo sėkmingas (žr. 37 pav.)



37 pav. „D1(1)_0822_30s“ transektos *Laminaria* klasės segmentavimas Deeplab modeliu, Deeplab homogeniniu ansambliu, LinkNet bei distiliuotu LinkNet modeliais (raudonai vaizduojamas ekspertinio žymens apvadas, žaliai – neteisingai žymėta klasė, mėlynai – neteisingai nepažymėta klasė)

3.3.2. Cityscapes duomenų rinkinys

Tęsiant eksperimentus standartiniams duomenų rinkiniams buvo mokomi penkių architektūrų modeliai (PSPNet, FPN, PAN, LinkNet ir Deeplab) bei ruošiami jų momentiniai homogeniniai ansambliai. Modelių rengimo konfigūracija pateikiama 21 lentelėje toliau:

21 lentelė. Cityscapes duomenų mokymo parametrai

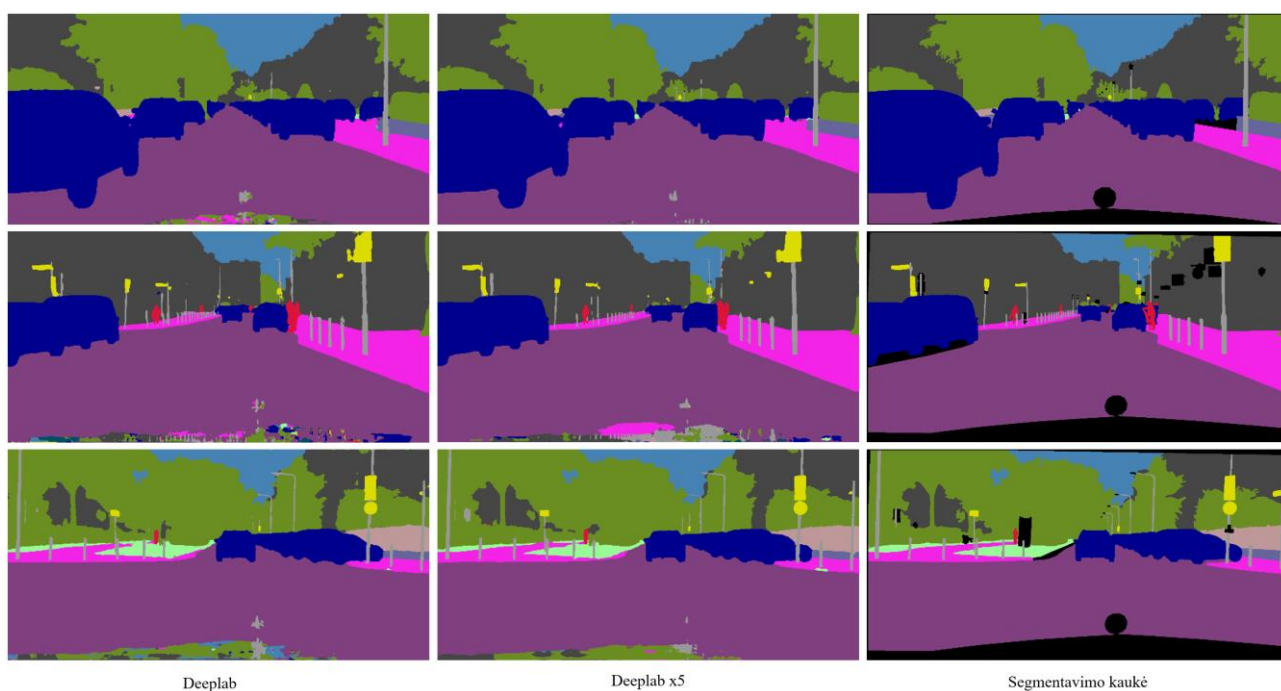
Parametras	Reikšmė
Išmetimo koeficientas	0,1
Mokymosi koeficientas	0,1
Aktyvacijos funkcija	<i>ReLU</i>
Nuostolio funkcija	<i>Cross Entropy Loss</i>
Paketo dydis	8 (7 distiliuojant modelius)
Įvestis	2048x1048 tšk.
Įvesčių skaičius	2975 mokymui, 500 testavimui
Vaizdų augmentacija	Dydžio keitimas (fiksotas), horizontalus, vertikalus pasukimas (50 % tikimybė)
Optimizavimo algoritmas	SGD [178] (<i>momentum (L2) = 0,9, weight decay (L1) = 0,0001</i>)
Mokymo epochos	200
Užkoduotojas	ResNet-50
Užkoduotojo modelio svoriai	Imagenet (užkoduotojo svoriai dalinai iššaldomi naudojant paketo normalizacijos inercijos parametą 0,01, mokymosi koeficientas dauginamas iš 0,01)
Kosinusinio atvėsimo epochos	40

Segmentavimo rezultatai atskiriems modeliams, momentiniam bei heterogeniniam ansambliams testuojant su Cityscapes rinkiniu pateikiami 22 lentelėje toliau:

22 lentelė. Segmentavimo rezultatai Cityscapes duomenų rinkiniui (x5 – homogeninis ansamblis)

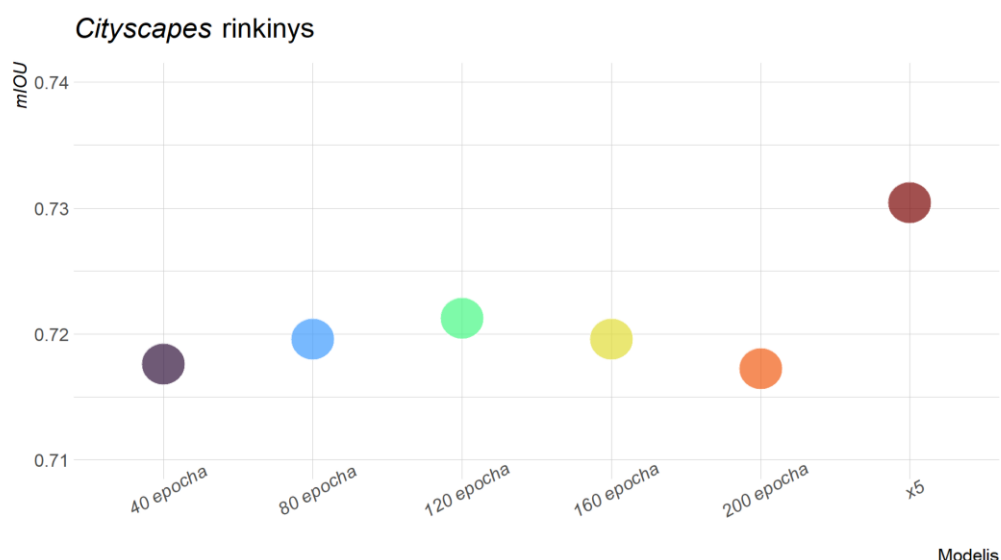
Architektūra	mIOU	mikroIOU	mikroF1	F1
PAN	0,642	0,889	0,762	0,941
PAN x5	0,652	0,892	0,770	0,943
PSPNet	0,643	0,878	0,767	0,935
PSPNet x5	0,662	0,884	0,782	0,939
LinkNet	0,665	0,896	0,780	0,945
LinkNet x5	0,682	0,903	0,794	0,949
FPN	0,667	0,898	0,783	0,946
FPN x5	0,689	0,904	0,801	0,950
Deeplab	0,704	0,907	0,816	0,918
Deeplab x5	0,730	0,914	0,834	0,955
Ansamblis	0,711	0,912	0,819	0,954

Pagal gautus rezultatus galima pastebėti, jog visų penkių naudojamų architektūrų atveju homogeninis ansamblis lenkė atskirų modelių segmentavimo rezultatus. Didžiausi skirtumai tarp atskiro modelio bei momentinio ansamblio efektyvumą pastebėti mokant FPN (0,667 mIOU ir 0,689 mIOU) bei Deeplab (0,704 mIOU ir 0,730 mIOU). Prasčiausiai testavimo duomenų rinkinį segmentavo PAN modelis, pasiekęs 64,2 % mIOU, taip pat prastai pasirodė ir iš šio modelio narių sudarytas homogeninis ansamblis (65,2 % mIOU). Panašų efektyvumą demonstravo PSPNet modelis, tačiau šios architektūros konfigūracija daugiau pagerėjo mokant homogeninį ansamblį (0,662 mIOU ir 0,643 mIOU). Deeplab atveju, homogeninis kolektyvas pralenkė penkių narių heterogeninį ansamblį (0,730 mIOU ir 0,711 mIOU). Pastebima, jog mikro- ir makrorodikliai skiriasi: tai rodo skirtingą atskirų klasių segmentavimo efektyvumą. Pavyzdiniai Deeplab modelio bei homogeninio kolektyvo segmentavimo vaizdai pateikti 38 pav. toliau:



38 pav. Cityscapes testavimo rinkinio segmentavimai Deeplab modeliu ir homogeniniu ansambliu (x5)

39 pav. pateikiamas atskirų Deeplab homogeninio ansamblio narių efektyvumas. Pagal gautus rezultatus matoma, jog homogeninio ansamblio nariai neviršija paties ansamblio efektyvumo (tam tikras narys nesuteikia pranašumo veikdamas labai efektyviai), o geri momentinio ansamblio rezultatai pasiekiami per skirtingų narių spėjimų vidurkinimą.



39 pav. Deeplab homogeninio ansamblio narių efektyvumai (*Cityscapes*)

Eksperimentai tęsiami Deeplab modelių ansamblį naudojant kaip modelį-mokytoją dviejų mažesnių modelių – FPN ir PAN – distiliavimui. Papildoma konfigūracija eksperimentams pateikiama 23 lentelėje toliau:

23 lentelė. Distiliavimo testavimo standartiniams rinkiniams konfigūracija

Konfigūracijos parametras	Reikšmė
Modelis-mokytojas	Deeplab homogeninis ansamblis
Modelis-studentas	FPN, PAN
Parametrai	$\alpha = 0,25; 0,5; 0,75$

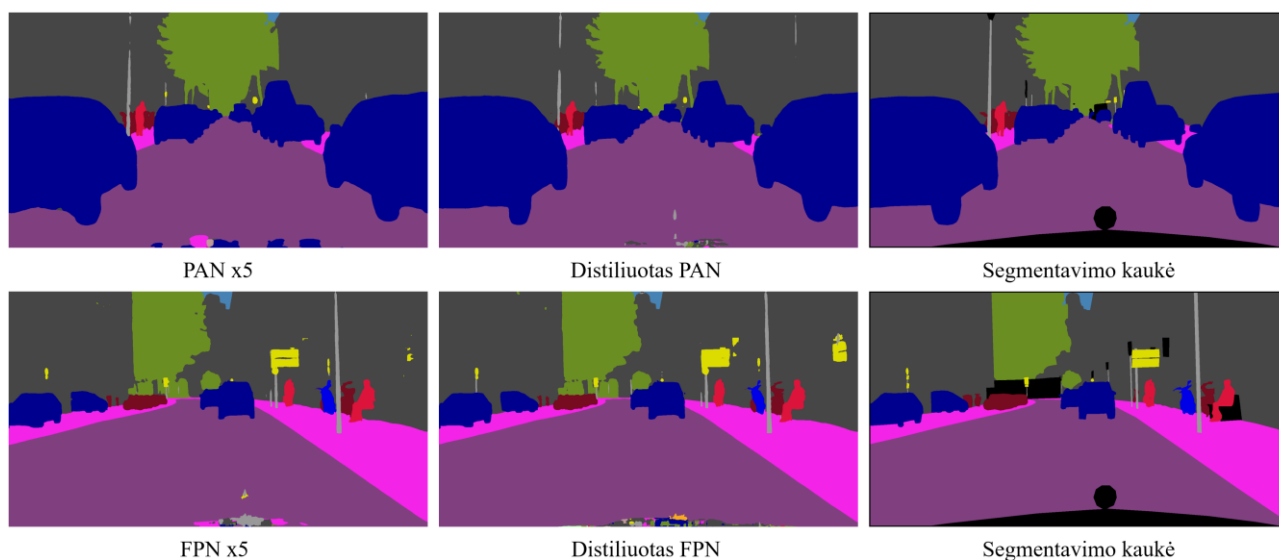
Eksperimentų rezultatai *Cityscapes* rinkiniui pateikiami 24 lentelėje toliau:

24 lentelė. Distiliuotų modelių segmentavimo rezultatai *Cityscapes* duomenų rinkiniui (distiliuotiems modeliams nurodoma mokymosi etapo α reikšmė, x5 – homogeninis kolektyvas, pilkame fone pateikti pradiniai (žr. 22 lentelę) rezultatai)

Architektūra	mIOU	mikroIOU	mikroF1	F1
PAN	0,642	0,889	0,762	0,941
PAN x5	0,652	0,892	0,770	0,943
PAN ($\alpha = 0,25$)	0,635	0,888	0,940	0,757
PAN ($\alpha = 0,5$)	0,632	0,885	0,939	0,753
PAN ($\alpha = 0,75$)	0,631	0,884	0,938	0,753
FPN	0,667	0,898	0,783	0,946
FPN x5	0,689	0,904	0,801	0,950
FPN ($\alpha = 0,25$)	0,666	0,898	0,946	0,782
FPN ($\alpha = 0,5$)	0,666	0,897	0,946	0,781
FPN ($\alpha = 0,75$)	0,666	0,897	0,946	0,781

Pagal rezultatus matoma, jog PAN modelio distiliavimas nepagerina *Cityscapes* duomenų rinkinio segmentavimo efektyvumo. Aukščiausias distiliuoto modelio pasiekiamas rezultatas lygus 0,635

mIOU (įprastos konfigūracijos PAN pasiekė 0,642 mIOU) su parametru $\alpha = 0,25$. Prasčiausias rezultatas gaunamas mokymo konfigūracijoje su $\alpha = 0,75$, kai testavimo rinkinys segmentuojamas 0,631 mIOU tikslumu. FPN atveju su visomis α konfigūracijomis gaunami labai panašūs rezultatai (segmentuojama su 0,666 mIOU) ir šis efektyvumas tik šiek tiek nusileidžia kosinusinio atvėsimo ansambliavimo technikai, kuomet gaunamas 0,667 mIOU. Gauti rezultatai galimai rodo, jog pasiekiami tam tikra modelio segmentavimo limitacija ir tolesnis segmentavimo efektyvumas nėra gerinamas derinant hiperparametrus. Pavyzdiniai segmentavimai PAN, FPN homogeniniais ansambliais bei jų distiliuotais modeliais pateikiami 40 pav. toliau:



40 pav. Cityscapes testavimo rinkinio segmentavimai homogeniniais ansambliais (x5) ir distiliuotais modeliais PAN ir FPN architektūroms

3.3.3. Floodnet duomenų rinkinys

Floodnet rinkinio segmentavimas atliekamas tokia pačia konfigūracija, kaip ir Cityscapes (žr. 21 lentelę), skiriasi tik įvesčių skaičius, pateikiamas 25 lentelėje.

25 lentelė. Floodnet duomenų parametrai

Parametras	Reikšmė
Įvesčių skaičius	1445 mokymui, 448 testavimui

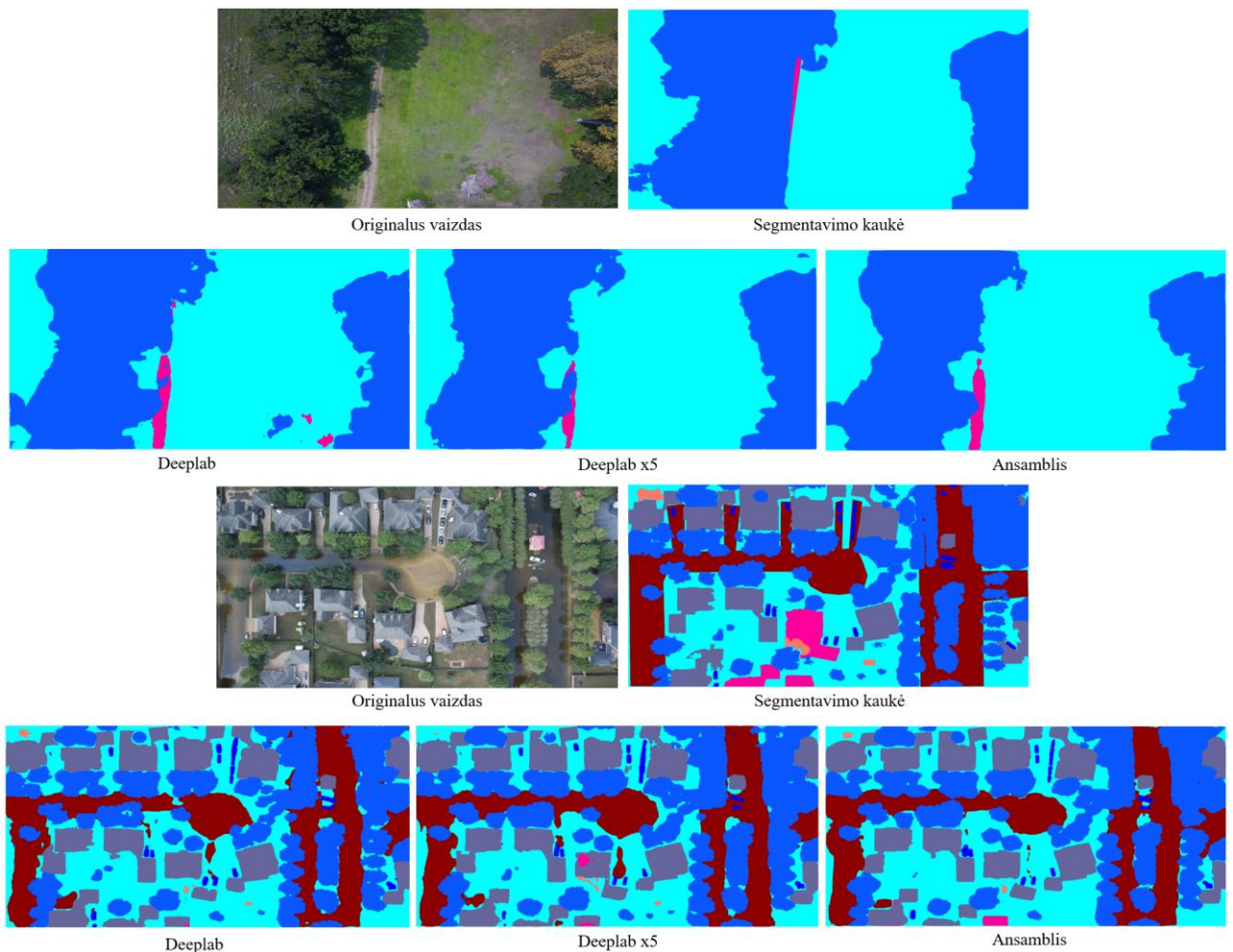
Šios konfigūracijos segmentavimo rezultatai Floodnet testavimo rinkiniui pateikiami 26 lentelėje toliau:

26 lentelė. Segmentavimo rezultatai Floodnet duomenų rinkiniui (x5 – homogeninis kolektyvas)

Architektūra	mIOU	mikroIOU	mikroF1	F1
LinkNet	0,620	0,794	0,885	0,746
LinkNet x5	0,612	0,799	0,888	0,734
FPN	0,643	0,803	0,891	0,771
FPN x5	0,618	0,791	0,883	0,749
PAN	0,603	0,774	0,873	0,741
PAN x5	0,628	0,803	0,891	0,755
PSPNet	0,641	0,812	0,896	0,767

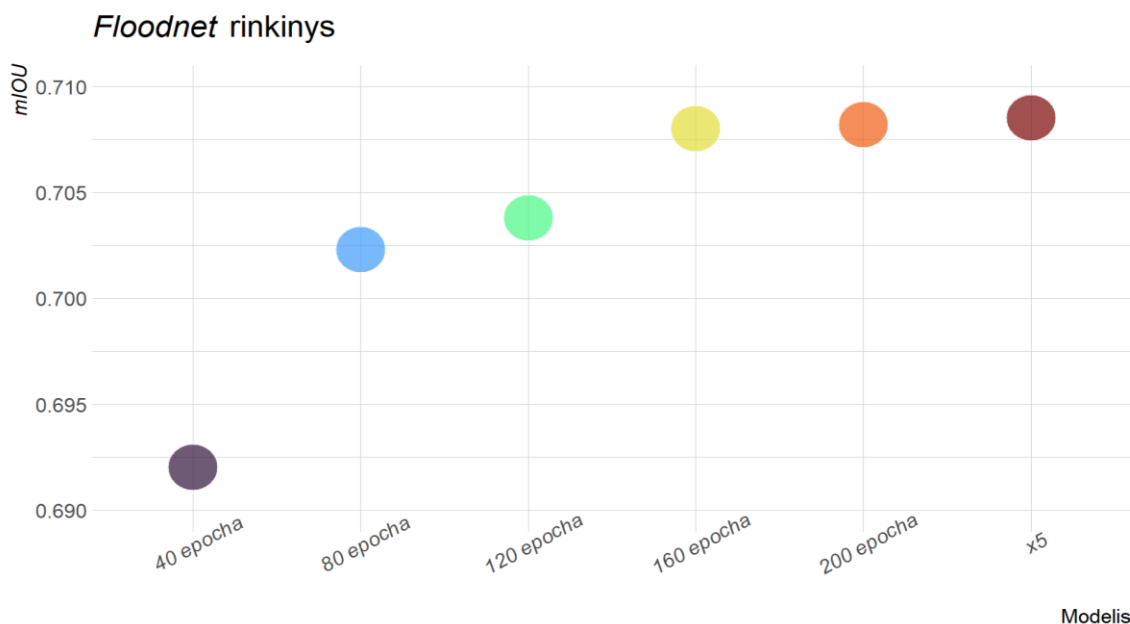
Architektūra	mIOU	mikroIOU	mikroF1	F1
PSPNet x5	0,650	0,828	0,906	0,772
Deeplab	0,703	0,849	0,915	0,807
Deeplab x5	0,709	0,856	0,922	0,819
Ansamblis	0,699	0,835	0,910	0,814

Pagal eksperimentų rezultatus Floodnet rinkiniui pastebima, jog homogeninių ansamblių efektyvumas, palyginti su atitinkamais atskirais modeliais, buvo mažesnis FPN bei LinkNet atvejais. Kaip ir Cityscapes atveju, geriausiai Floodnet rinkinį segmentavo Deeplab homogeninis ansamblis, pasiekęs 70,9 % mIOU ir taip pralenkęs heterogeninį penkių modelių ansamblį (69,9 % mIOU). Kaip jau minėta, mokant segmentuoti Floodnet rinkinį homogeninio kolektyvo kūrimas nepasiteisino LinkNet bei FPN architektūroms: LinkNet atveju atskiro modelio efektyvumas siekė 0,620 mIOU bei 0,612 mIOU homogeniniam ansamblui, naudojant FPN atskiras modelis segmentavo 0,643 mIOU efektyvumu, homogeninis ansamblis – 0,618 mIOU. PAN, PSPNet modeliams sudarant momentinius ansamblius stebimas segmentavimo pagerėjimas: atskiro PAN efektyvumas siekė 0,603 mIOU, o homogeninio ansamblio – 0,628 mIOU; PSPNet atskiras modelis segmentavo su 64,1 % mIOU, kolektyvas – 65 % mIOU. Pastebima, jog segmentavimo mikro- bei makrorodikliai skiriasi mažiau nei testuojant Cityscapes rinkinį, o tai rodo tolygesnę skirtingų klasių segmentavimo efektyvumą. Pavyzdiniai segmentavimai Floodnet rinkiniui efektyviausiai pasirodžiusiems homogeniniam Deeplab bei atskiram Deeplab modeliams bei heterogeniniam ansamblui pateikiami 41 pav. toliau:



41 pav. Floodnet testavimo rinkinio segmentavimai Deeplab modeliu ir homogeniniu ansambliu (x5)

Pagal 42 pav. matoma, jog nėra stebimas didesnis homogeninio ansamblio efektyvumo pagerėjimas mokant Floodnet duomenų rinkiniu, t.y. paskutinių narių rezultatai yra panašūs į ansamblio segmentavimą. Nuo prasčiausio (pirmojo) ansamblio nario bendras rezultatas skiriasi per 0,017 mIOU (2,23 %), pastebimas didesnis šio modelio atotrūkis nuo kitų ansamblio narių. Iš dalies tokie rezultatai gali būti paaiškinami tuo, jog pirmieji nariai nespėja pilnai išmokti mažesnio duomenų rinkiniu požymių, todėl testavimo rinkinys segmentuojamas ne taip efektyviai.



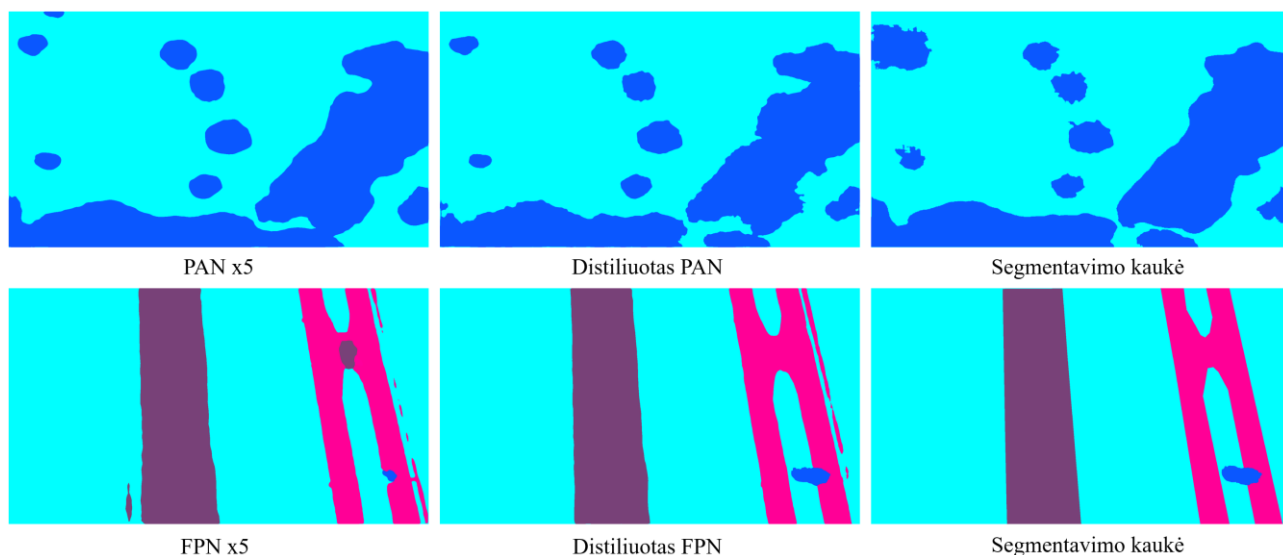
42 pav. Deeplab homogeninio ansamblio narių efektyvumai (Floodnet)

Modelio distiliavimas Floodnet atliekamas naudojantis tokia pačia, kaip ir Cityscapes atveju, konfigūracija (žr. 23 lentelę). Segmentavimo rezultatai distiliuotiems modeliams pateikiami 27 lentelėje toliau:

27 lentelė. Distiliuotų modelių segmentavimo rezultatai Floodnet duomenų rinkiniui (distiliuotiems modeliams nurodoma mokymosi etapo α reikšmė, x5 – homogeninis kolektyvas, pilkame fone pateikti pradiniai (žr. 26 lentelę) rezultatai)

Architektūra	mIOU	mikroIOU	mikroF1	F1
PAN	0,603	0,774	0,873	0,741
PAN x5	0,628	0,803	0,891	0,755
PAN ($\alpha = 0,25$)	0,638	0,803	0,891	0,766
PAN ($\alpha = 0,5$)	0,672	0,903	0,903	0,793
PAN ($\alpha = 0,75$)	0,679	0,838	0,912	0,796
FPN	0,643	0,803	0,891	0,771
FPN x5	0,618	0,791	0,883	0,749
FPN ($\alpha = 0,25$)	0,651	0,810	0,895	0,776
FPN ($\alpha = 0,5$)	0,677	0,820	0,901	0,797
FPN ($\alpha = 0,75$)	0,671	0,820	0,901	0,792

Pagal eksperimentų rezultatus pastebima, jog segmentuojant Floodnet rinkinį PAN bei FPN modelių efektyvumas naudojant distiliavimo techniką pagerėja ne tik palyginus su atskirų modelių segmentavimu (0,603-0,679 mIOU PAN ir 0,643-0,677 mIOU FPN) bet ir lyginant su homogeniniais ansambliais (0,628-0,679 mIOU PAN ir 0,618-0,677 mIOU FPN). Efektyviausias α parametras FPN modeliui buvo 0,5, o PAN mokymuisi – 0,75. Pastebima, jog skirtingai nuo Cityscapes rinkinio, PAN modeliui efektyvesnis segmentavimas pasiekiamas klaidos funkcijoje didžiaja dalimi (kai $\alpha = 0,75$) atsižvelgiant į studento-mokytojo klasių išvesčių skirtumus. Toliau 43 pav. pateikiami pavyzdiniai Floodnet segmentavimai PAN, FPN momentiniais ansambliais bei distiliuotais modeliais.



43 pav. Floodnet testavimo rinkinio segmentavimai homogeniniais ansambliais (x5) ir distiliuotais PAN ir FPN modeliais

3.3.4. Camvid duomenų rinkinys

Ketvirtasis eksperimentų ciklas atliekamas su Camvid duomenų rinkiniu. Eksperimentų metu naudojama tokia pati, kaip ir Cityscapes bei Floodnet atvejais, mokymosi konfigūracija (žr. 21 lentelę), skiriasi tik modelių įvesčių dydis bei skaičius (žr. 28 lentelę).

28 lentelė. Camvid duomenų mokymo parametrai

Parametras	Reikšmė
Įvestis	1920x960 tšk.
Įvesčių skaičius	369 mokymui, 232 testavimui

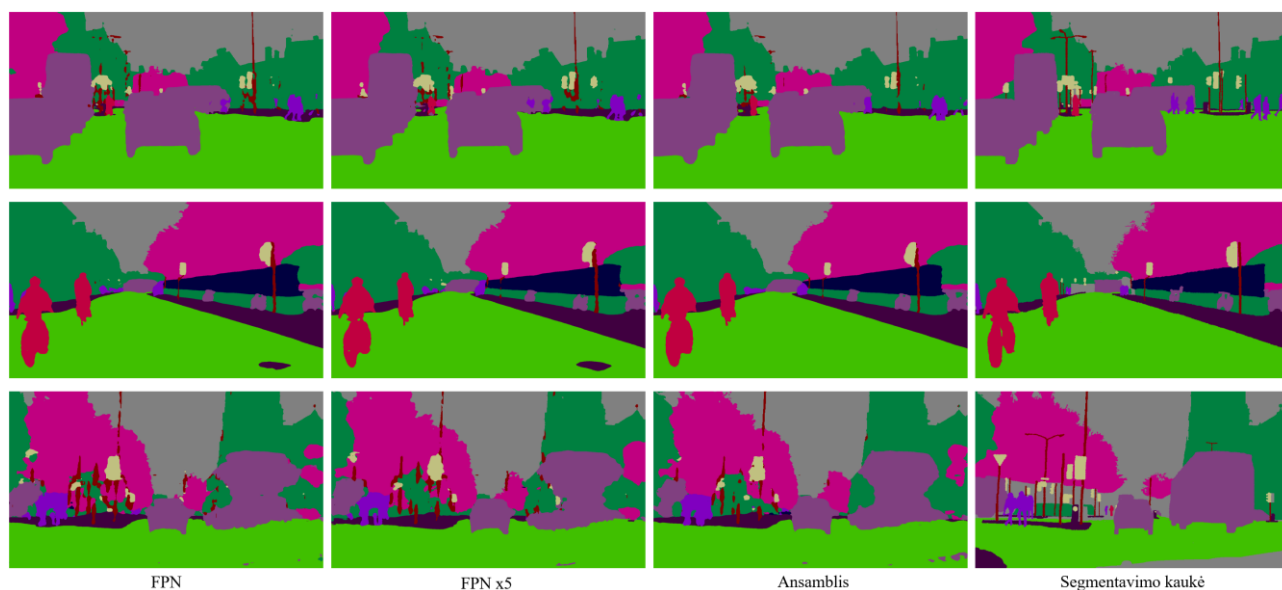
Penkių giliojo mokymosi architektūrų atskirų modelių, jų momentinių ansamblių bei heterogeninio ansamblio Camvid testavimo rezultatai pateikiami 29 lentelėje toliau:

29 lentelė. Segmentavimo rezultatai Camvid duomenų rinkiniui (x5 – homogeninis ansamblis)

Architektūra	mIOU	mikroIOU	mikroF1	F1
PSPNet	0,614	0,828	0,906	0,723
PSPNet x5	0,612	0,830	0,907	0,720
PAN	0,635	0,838	0,912	0,750
PAN x5	0,622	0,834	0,909	0,737
LinkNet	0,676	0,861	0,925	0,785

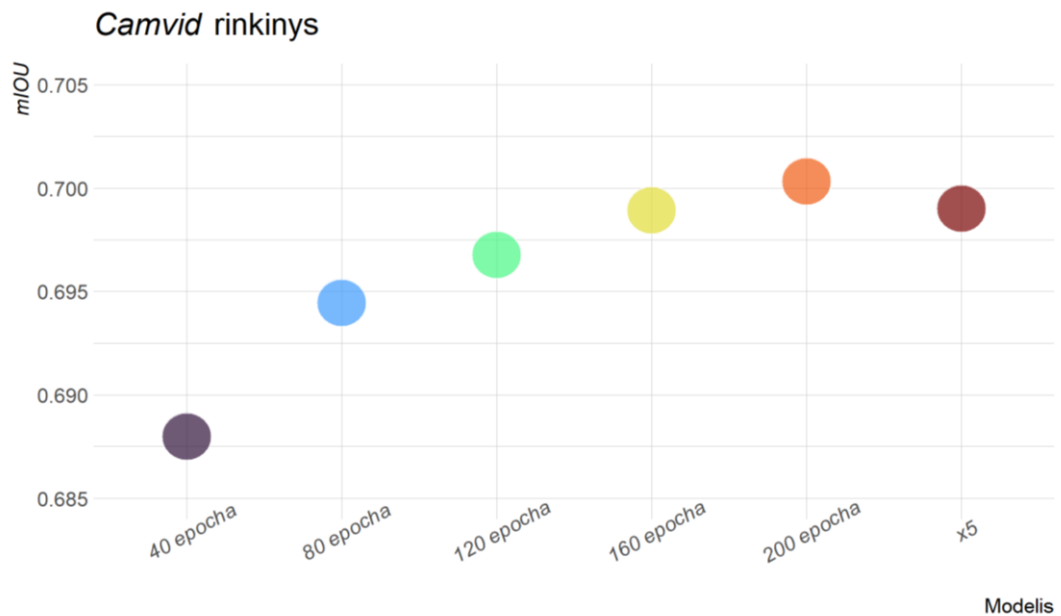
Architektūra	mIOU	mikroIOU	mikroF1	F1
LinkNet x5	0,677	0,863	0,926	0,785
FPN	0,689	0,861	0,925	0,791
FPN x5	0,693	0,865	0,927	0,800
Deeplab	0,700	0,869	0,930	0,809
Deeplab x5	0,699	0,870	0,930	0,805
Ansamblis	0,709	0,897	0,936	0,812

Pagal gautus duomenis analizuojant Camvid duomenų segmentavimą galima pastebėti, jog homogeninio ansamblio sudarymas segmentavimo efektyvumo nepagerino trimis iš penkių testuotų konfigūracijų. Svarbu paminėti, jog atvejais, kuomet segmentavimo efektyvumas pagerėjo (FPN bei LinkNet), šis pagerėjimas siekė 0,001 mIOU (LinkNet atveju) bei 0,004 mIOU (FPN atveju), t.y. buvo labai nežymus. Geriausią segmentavimo rezultatą šiam testavimo rinkiniui pasiekė heterogeninis ansamblis su 70,9 % mIOU. Didžiausias rezultatų pablogėjimas naudojant momentinį kolektyvą pastebimas PAN architektūrai: atskiras modelis testavimo rinkinį segmentavo 0,635 mIOU, kolektyvas – 0,622 mIOU. Bendrai prasčiausius rezultatus demonstravo PSPNet modelis (0,614 mIOU) bei šios architektūros homogeninis ansamblis (0,612 mIOU). Pavyzdiniai segmentavimai pateikiami 44 pav. toliau:



44 pav. Camvid testavimo rinkinio segmentavimai FPN modeliu ir homogeniniu ansambliu (x5)

Camvid standartinio rinkinio atveju homogeninio Deeplab kolektyvo gaunamų rezultatų pasiskirstymas panašus į Floodnet segmentavimų efektyvumus (žr. 45 pav.) Camvid atveju homogeninis kolektyvas netgi demonstruoja mažesnę efektyvumą nei paskutinis (mokomas 200 epochų) ansamblio narys. Gautus rezultatus (kaip ir segmentuojant Floodnet rinkinį) galima būtų paaiškinti modelio negebėjimu išmokti pakankamą kiekį rinkinio specifikos naudojant ribotą mokymosi aibės duomenų kiekį.



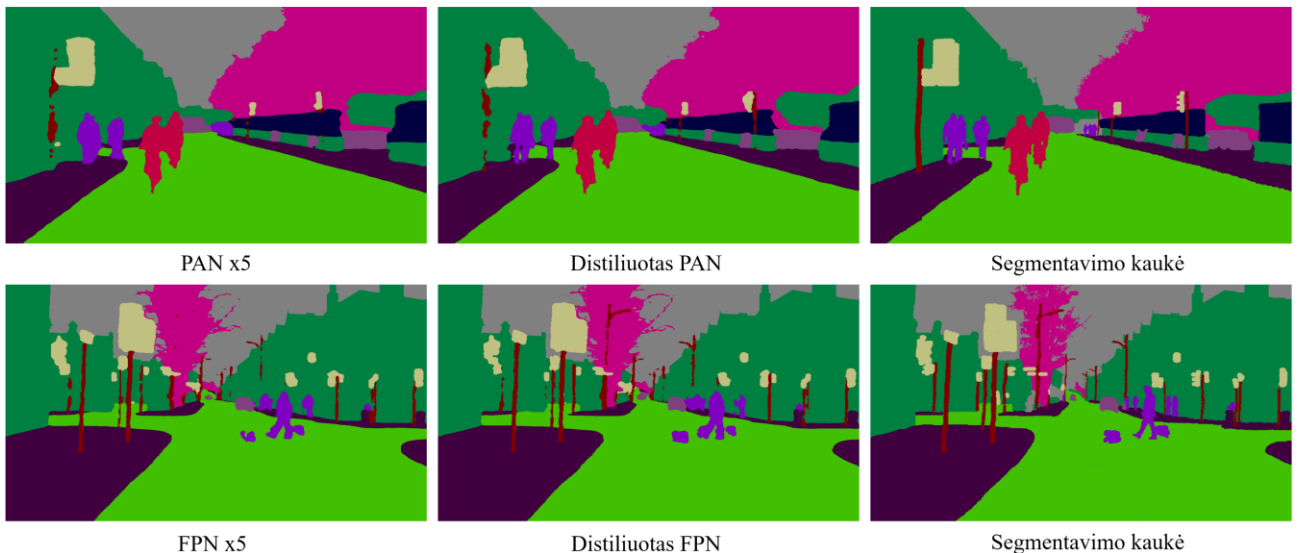
45 pav. Deeplab homogeninio ansamblio narių efektyvumai (Camvid)

Camvid rinkiniui pritaikius žinių distiliavimą PAN bei FPN modeliams gauti rezultatai pateikiami 30 lentelėje toliau:

30 lentelė. Distiliuotų modelių segmentavimo rezultatai Camvid duomenų rinkiniui (distiliuotiems modeliams nurodoma mokymosi etapo α reikšmė, x5 – homogeninis kolektyvas, pilkame fone pateikti pradiniai (žr. 29 lentelę) rezultatai)

Architektūra	mIOU	mikroIOU	mikroF1	F1
PAN	0,635	0,838	0,912	0,50
PAN x5	0,622	0,834	0,909	0,737
PAN ($\alpha = 0,25$)	0,647	0,844	0,915	0,761
PAN ($\alpha = 0,5$)	0,650	0,848	0,918	0,762
PAN ($\alpha = 0,75$)	0,648	0,843	0,915	0,761
FPN	0,689	0,861	0,925	0,791
FPN x5	0,693	0,865	0,927	0,800
FPN ($\alpha = 0,25$)	0,686	0,860	0,925	0,794
FPN ($\alpha = 0,5$)	0,678	0,859	0,924	0,788
FPN ($\alpha = 0,75$)	0,672	0,856	0,922	0,782

Pagal gautus rezultatus pastebima, jog segmentuojant Camvid rinkinį distiliavimo technika pagerino rezultatus tik PAN architektūrai. Šiuo atveju, geriausias rezultatas (0,650 mIOU) pasiekiamas su $\alpha = 0,5$ parametro reikšme ir lenkia atskiro modelio (0,635 mIOU) bei homogeninio ansamblio (0,622 mIOU) rezultatus. Distiliavimą taikant didesniai FPN modeliui, rezultatai nepagerėja: efektyvumas koncentruojant su $\alpha = 0,25$ išlieka panašus tiek distiliuotam (0,686 mIOU), tiek atskiram modeliui (0,689 mIOU) bei homogeniniam ansamblui (0,693 mIOU). Pastebima, jog kuomet distiliavimas efektyvina segmentavimą naudojant PAN modelį, geresni rezultatai pasiekiami su didesnėmis $\alpha = 0,5$ bei $0,75$ – reikšmėmis. Pavyzdiniai segmentavimai homogeniniams kolektyvams bei distiliuotiems modeliams FPN ir PAN pateikiami 46 pav. toliau:



46 pav. Camvid testavimo rinkinio segmentavimai homogeniniais ansambliais (x5) ir distiliuotais PAN ir FPN modeliais

3.4. Ilgesnis modelių mokymas

Paskutiniame tyrimo etape bandoma testuoti, ar tikslinga naudoti modelių efektyvumo gerinimo technikas (momentinį ansambliavimą panaudojant kosinusinį atvėsimą bei modelių distiliavimą). Šių technikų taikymo metu pasitelkiami sudėtingesni, iš daugiau etapų sudaryti algoritmai, naudojama daugiau resursų, pavyzdžiui modeliui-mokytojui paruošti bei modeliui-studentui mokytis pasitelkiamas didesnis epochų skaičius. Hipotetizuojama, jog galbūt galima panašius rezultatus gauti tiesiog mokant įprastus atskirus modelius didesnę epochų skaičių. Taip pat tikrinama, ar ilgiau mokant momentinius ansamblius ir juos distiliuojant į modelius per tą patį epochų skaičių pasiekiamas didesnis segmentavimo efektyvumas.

Eksperimentų eigai pasirinkta ši konfigūracija:

- PAN bei FPN modeliai mokomi 1000 epochų (tiek epochų reiktų ruošti tradicinį heterogeninį penkių modelių ansamblių);
- Deeplab modeliui taikoma kosinusinio atvėsimo technika, momentiška fiksuojant modelio svorius kas 200 epochų (iš viso 1000 epochų), o PAN ir FPN modeliai (distiliuojant) mokomi 200 epochų.

31 lentelėje toliau pateikiami Deeplab momentinio ansamblio rezultatus naudotiems standartiniams – Cityscapes, Floodnet bei Camvid – duomenų rinkiniams.

31 lentelė. Atskirų momentinio modelių ansamblio narių rezultatai (mokant 1000 epochų)

Duomenų rinkinys	Ansamblio narys (epocha)	mIOU
Cityscapes	200	0,709
	400	0,710
	600	0,706
	800	0,708
	1000	0,711
	Ansamblis	0,730

Duomenų rinkinys	Ansamblio narys (epocha)	mIOU
Floodnet	200	0,706
	400	0,710
	600	0,712
	800	0,708
	1000	0,706
	Ansamblis	0,720
Camvid	200	0,711
	400	0,712
	600	0,711
	800	0,709
	1000	0,707
	Ansamblis	0,715

Pagal gautus rezultatus galima matyti, jog visų rinkinių atvejais atskiri momentinio ansamblio nariai nepralenkė bendro modelių ansamblio efektyvumo. Palyginus su 200 epochų mokytais modeliais galima pastebėti, jog Cityscapes atveju nėra pastebima jokio segmentavimo pagerėjimo – efektyvumas išlieka 0,730 mIOU. Floodnet bei Camvid rinkiniai segmentuojami geriau: Floodnet atveju homogeninio ansamblio, mokyto 1000 epochų, rezultatai pagerėja, tačiau nežymiai (0,720 mIOU ir 0,709 mIOU); Camvid atveju taip pat stebimas nedidelis efektyvumo padidėjimas – 0,715 mIOU 1000 epochų ansamblui ir 0,699 mIOU 200 epochų ansamblui.

Distiliavimui naudojant 1000 epochų ruošą Deeplab momentinį ansamblį ir 200 epochų mokant PAN ir FPN modelius gaunami rezultatai pateikiami 32 lentelėje toliau:

32 lentelė. Distiliavimo rezultatai standartiniams rinkiniams (pilkame fone pateikti pradiniai (žr. 22, 26, 29 lenteles) rezultatai, modeli-mokytoją ruošiant 200 epochų)

Duomenų rinkinys	Architektūra	mIOU	mikroIOU	mikroF1	F1
Cityscapes	PAN ($\alpha = 0,25$)	0,634	0,886	0,940	0,755
	PAN ($\alpha = 0,25$)	0,635	0,888	0,940	0,757
	FPN ($\alpha = 0,25$)	0,669	0,898	0,784	0,946
	FPN ($\alpha = 0,25$)	0,666	0,898	0,946	0,782
Floodnet	PAN ($\alpha = 0,25$)	0,600	0,773	0,872	0,735
	PAN ($\alpha = 0,25$)	0,638	0,803	0,891	0,766
	FPN ($\alpha = 0,25$)	0,657	0,812	0,896	0,782
	FPN ($\alpha = 0,25$)	0,651	0,810	0,895	0,776
Camvid	PAN ($\alpha = 0,25$)	0,628	0,835	0,910	0,745
	PAN ($\alpha = 0,25$)	0,647	0,844	0,915	0,761
	FPN ($\alpha = 0,25$)	0,686	0,861	0,925	0,794
	FPN ($\alpha = 0,25$)	0,686	0,860	0,925	0,794

Pagal gautus rezultatus pastebima, jog modeliai-studentai, distiliuojami iš penkis kartus ilgiau mokytojo homogeninio Deeplab modelio su $\alpha = 0,25$ klaidos koeficientu, bendrai nepagerina segmentavimo efektyvumo. Labai nedidelis rezultatų pagerėjimas stebimas Cityscapes duomenų rinkinio segmentavimui su FPN modeliu (0,666 ir 0,669 mIOU). Camvid bei Floodnet testavimo metu pastebimas segmentavimo degradavimas naudojant PAN architektūrą bei labai panašus FPN modelio segmentavimo efektyvumas lyginant ilgiau ir trumpiau mokomus modelius-mokytojus.

3.5. Diskusija

Bendri tyrimo metu atlikti eksperimentai rodo kelis panaudotų ansambliavimo bei distiliavimo metodikų aspektus:

- Homogeninis ansambliavimas specializuotam Arkties duomenų rinkiniui pasiteisino vertinant paprastesnes, sėkmingiau (su didesniu IOU) segmentuojamas klases (Ophiuroidea ir Laminaria), o sudėtingos klasės buvo efektyviau atpažįstamos naudojant atskirų modelių konfigūracijas. Distiliavimas riboto testavimo metu pagerino LinkNet modelio rezultatus;
- Homogeninis ansambliavimas buvo efektyvus daug duomenų turinčiam (Cityscapes) rinkiniui: homogeniniai ansambliai visoms architektūroms pasiekė geresnius rezultatus nei pavieniai modeliai, o efektyviausio Deeplab homogeninio kolektyvo segmentavimas pralenkė heterogeninį kolektyvą. Distiliavimas šiam rinkiniui nepasiteisino ir rezultatai išliko panašūs (FPN) arba suprastėjo (PAN);
- Homogeninis ansambliavimas vidutinio dydžio rinkiniui (Floodnet) segmentavimo rezultatus pagerino ne visais atvejais (3/5), tačiau geriausias momentinis ansamblis lenkė heterogeninio ansamblio efektyvumą. Žinių distiliavimas šiam rinkiniui pasiteisino ir buvo pagerinti abiejų testuojamų architektūrų tiek pavienių modelių, tiek homogeninių ansamblių rezultatai;
- Homogeninis ansambliavimas mažo dydžio rinkiniui (Camvid) nebuvo efektyvus ir rezultatų nepagerino trimis atvejais iš penkių, o gaunamas pagerėjimas buvo nežymus. Šiuo atveju pasiteisino distiliavimo technika ir buvo gauti panašūs (FPN) arba efektyvesni (PAN) segmentavimo rezultatai;
- Ilgiau mokomas modelis-mokytojas riboto testavimo metu neturėjo teigiamos įtakos distiliuoto modelio rezultatų efektyvumui.

Su skirtingomis konvoliucinių neuroninių tinklų architektūromis atlikti bandymai rodo homogeninių ansamblių efektyvumą didesniems duomenų rinkiniams ir aiškiau atskiriamoms segmentavimo klasėms bei didesni modelių distiliavimo našumą mažesniems duomenų rinkiniams. Svarbu paminėti, jog rezultatai skiriasi priklausomai nuo modelio architektūros, distiliavimo konfigūracijos, duomenų rinkinio dydžio, o eksperimentų metu buvo panaudota ribota paveikslėlių dydžio keitimo bei augmentavimo algoritmų aibė, pasitelkti tik trys standartiniai rinkiniai.

Tęsiant eksperimentavimą būtina išbandyti didesnę standartinių ir specializuotų semantinio segmentavimo rinkinių skaičių. Toks testavimas būtų labai naudingas gauti stiprų statistinį pagrindimą, jog modelių distiliavimas labiau tinka ruošiant modelius mažesniems, galbūt specifinių panaudojimo atvejų duomenų rinkiniams, o homogeninių ansamblių kūrimas labiau pasiteisina didesniems rinkiniams ir įvairesniems duomenims. Taip pat svarbu išbandyti skirtingus heterogeninio ansamblio sudarymo aspektus ir juos palyginti su homogeninio kolektyvo efektyvumu.

Kitas svarbus tyrimo apribojimas – naudotų mokymosi epochų konfigūracija. Mokymas naudojant didesnę (didesnių rinkinių atveju – mažesnę) epochų skaičių gali būti svarbus homogeninių ansamblių

sudarymui. Svarbu pasiekti mokymosi pusiausvyrą: ruošiant momentinius ansamblius pirmieji nariai turi pasiekti pakankamai gerą lokalaus optimumo tašką tam, jog galutinis viso ansamblio rezultatas nemažėtų dėl vieno ar kelių narių neefektyvumo. Siekiant nuodugniai ištirti pagrindinius homogeninių ansamblių privalumus taip pat būtina atsižvelgti į modelio svorių inicializavimą po kiekvieno mokymosi koeficiento perstatymo į pradinę reikšmę. Tam būtų galima taikyti triukšmo pridėjimo ar panašias technikas modelio svorių keitimui tikintis, jog kito mokymosi ciklo metu (per nustatytą kosinusinio atvėsimo epochų skaičių) algoritmas pasieks skirtingą, galbūt dar efektyvesnę lokalų minimumą ir taip bus gerinamas galutinio homogeninio ansamblio segmentavimo efektyvumas.

Tyrimų metu taip pat svarbu išbandyti skirtingus distiliavimo parametrus naudojant studento-mokytojo techniką: svarbu testuoti įvairias hiperparametro α konfigūracijas, taip pat parametrinės temperatūros T [5] reikšmes.

Siekiant bendrų rekomendacijų semantinio segmentavimo rinkiniams būtina testuoti ir kitas architektūras bei skirtingas užkoduotojo dalies modelių konfigūracijas. Būtų naudinga žinoti, ar mokant segmentuoti didesnius / mažesnius (ar sudėtingesnius / lengvesnius) duomenų rinkinius efektyviau naudoti masyvesnius (daugiau parametrų turinčius) užkoduotojo dalies bazinius modelius ir lengvesnes dekoduoja architektūras, ar atvirkščiai. Be to, reikalingas ir platesnis pradinės modelių svorių inicializavimo / fiksavimo mokymosi metu tyrimas: galbūt skirtingos šio etapo metodikos padėtų pasiekti geresnius rezultatus priverčiant modelį siekti kitų konvergavimo taškų.

3.6. Ekonominės rekomendacijos

Kuriant, mokant bei tobulinant giliųjų neuroninių tinklų architektūras tikimasi, jog jos ateityje bus naudingos tam tikrais panaudojimo atvejais ir neš ekonominę naudą. Svarbu paminėti, jog šie modeliai reikalauja daug skaičiavimo resursų (ypač tai aktualu konvoliuciniams tinklams), todėl metodai, kuriais naudojantis būtų galima teikti tiek pat efektyvius (ar bent jau nedaug suprastėjusius tikslumo aspektu) sprendimus su mažesniu operacijų skaičiumi yra itin svarbūs. Eksperimentų metu pademonstruota, jog galima gauti tiek efektyvesnius ansamblius (pvz., ruošiant momentinį ansamblį, pasirodantį efektyviau už heterogenines konstrukcijas), tiek atskirus modelius juos distiliuojant mokytojo-studento technika.

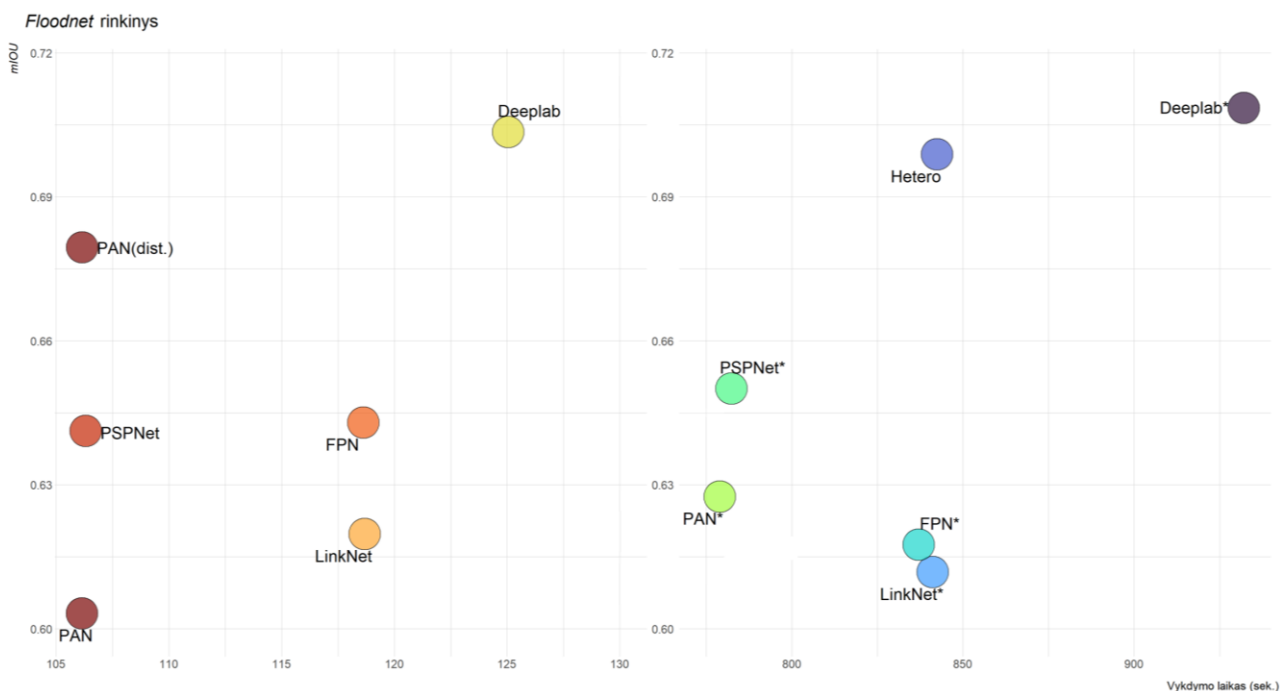
Pastaraisiais metais sparčiai augant giliojo mokymosi taikymui įvairiose srityse, neišvengiamai augo ir reikalaujami skaičiavimo pajėgumai [179]. Kaip jau minėta, didžioji dalis (~90 %) skaičiavimų, atliekamų daugiausiai duomenų centrų serveriuose, yra sunaudojama būtent modelių spėjimų etape [153]. Šios operacijos, dažniausiai vykdomos pasitelkiant *Amazon* ar *Google* infrastuktūros galimybes, kainuoja, o tai gali iš esmės riboti tam tikrų modelių naudojimą priklausomai nuo projekto ar panaudojimo atvejo biudžeto galimybių.

Projekto metu testuotos žinių distiliavimo metodikos galimai ekonominei naudai įvertinti buvo skaičiuojamas paruoštų modelių ansamblių bei distiliuotų modelių spėjimų sudarymo laikas. Toliau 33 lentelėje pateikiama pagrindinė konfigūracijos, su kuria buvo atliekamas testavimas, informacija.

33 lentelė. Laiko testavimo konfigūracija

Parametras	Reikšmė
Duomenų rinkinys	Floodnet
Grafikos procesorius	RTX 3090
Testavimo paketo dydis	12
Pradinis duomenų augmentavimas	Dydžio keitimas (angl. <i>resize</i>)

Rezultatų stabilumui užtikrinti kiekvienas paveikslėlių rinkinio spėjimo etapas buvo kartojamas 5 kartus. Kiekvieno modelio spėjimo rezultatus Floodnet rinkiniui galima matyti 47 pav. toliau:



47 pav. Modelių segmentavimo trukmė bei efektyvumas Floodnet rinkiniui (* – homogeninis kolektyvas)

Galima pastebėti, jog ilgiausiai spėjimo procesas trunka naudojant Deeplab homogeninį ansamblį, tačiau šis kolektyvas taip pat demonstruoja geriausias segmentavimo rezultatus. Naudojant debesų kompiuterijos resursus, RTX 3090 grafikos procesoriaus galiai arčiausias pavyzdys yra AWS (angl. *Amazon Web Services*) „g5.4xlarge“ virtualios mašinos resursų vienetas (angl. *instance*) [180]. Teigiant, jog viena šio debesijos grafikos procesoriaus naudojimo valanda kainuoja 0,88 EUR (pagal 2023 m. balandžio duomenis), visas Floodnet mokymo rinkinio segmentavimas Deeplab ansambliu truktų 0,26 val. ir kainuotų maždaug $0,26 \text{ val.} \cdot 0,88 \text{ EUR/val.} = 0,23 \text{ EUR}$. Šio rinkinio atveju distiliuoto PAN modelio segmentavimo efektyvumas ne daug nusileidžia geriausiajam modeliui (0,709 mIOU ir 0,679 mIOU (žr. 26, 27 lenteles)). Vieną kartą segmentuoti Floodnet mokymosi rinkinį PAN modeliu kainuotų $0,029 \text{ val.} \cdot 0,88 \text{ EUR/val.} = 0,026 \text{ EUR}$. Matoma, jog naudojant distiliuotą PAN modelį būtų galima sutaupyti net 88 % skaičiavimų kainos. Taigi, vaizdų segmentavimo uždaviniui spręsti galima pasirinkti efektyvesnius, bet daugiau resursų reikalaujančius modelius arba, pritaikant tokias technikas kaip žinių distiliavimas, paruošti modelius, kurie teikia panašius segmentavimo rezultatus, tačiau naudoja mažesnę skaičiavimo resursų kiekį.

Išvados

1. Atlikus literatūros analizę nustatyta, kad egzistuoja įvairios giliojo mokymosi modelių ansambliavimo bei distiliavimo technikos, leidžiančios ne tik efektyvinti modelių rezultatus, bet ir sutaupyti skaičiavimo laiko. Dėl didelio resursų poreikio ansambliavimas rečiau taikomas vaizdams segmentuoti, todėl modelių efektyvinimas, taigi ir žinių distiliavimas, šiai užduočiai tampa ypač aktualūs. Tyrimui atlikti pasirinktos heterogeninio ansambliavimo, homogeninių kolektyvų sudarymo naudojant kosinusinį atvėsimą bei žinių distiliavimo metodikos.
2. Pagrindinėms giliųjų konvoliucinių tinklų konfigūracijoms bei naudojamoms technikoms testuoti pasirinkti Klaipėdos universiteto tyrėjų surinkti Arkties jūros dugno vaizdai bei trys standartiniai semantinio segmentavimo duomenų rinkiniai – Cityscapes, Floodnet bei Camvid.
3. Konvoliucinių tinklų realizacijai naudoti penki užkoduotojo-dekoduotojo idėją taikantys modeliai: FPN, PSP, LinkNet, PAN bei Deeplab. Modeliai mokyti ir testuoti tiek atskirai, tiek sudarant heterogeninius bei homogeninius kolektyvus.
4. Segmentuojant specializuotą Arkties duomenų rinkinį geriausi rezultatai buvo pasiekti naudojant homogeninius ansamblius Ophiuroidea ir Laminaria klasėms: gauti atitinkamai 0,349 IOU (LinkNet) ir 0,694 IOU (Deeplab) segmentavimo efektyvumai. Sudėtingų klasių (Ceriantharia ir Polychaeta) atpažinimui efektyviau veikė atskiri modeliai, ypač PAN ir LinkNet architektūros. Homogeninių kolektyvų sudarymas Cityscapes rinkiniui pasiteisino ir visų naudotų architektūrų ansambliai pasiekė geresnius rezultatus nei pavieniai modeliai, o Deeplab kolektyvo segmentavimo tikslumas (0,730 mIOU) pralenkė heterogeninį kolektyvą (0,711 mIOU). Floodnet rinkiniui pritaikant homogeninį ansambliavimą segmentavimo rezultatai pagerėjo daugumai (trims iš penkių) testuotų modelių, o geriausias rezultatas siekė 0,709 mIOU segmentuojant homogeniniu Deeplab kolektyvu ir lenkė heterogeninį ansamblį (0,699 mIOU). Rinkiniui Camvid homogeninis ansambliavimas nepasirodė efektyvus ir rezultatai suprastėjo trimis atvejais iš penkių, o gautas pagerėjimas buvo nežymus.
5. Modelių distiliavimo technika nepasiteisino didžiausiam iš testuotų – Cityscapes – duomenų rinkiniui. Arkties rinkinio testavimo metu distiliavimas pagerino LinkNet modelio rezultatus Laminaria klasės segmentavime. Modelius distiliuojant Floodnet segmentavimui pagerinti testuotų PAN ir FPN architektūrų rezultatai, Camvid rinkinio atveju gauti panašūs FPN bei efektyvesni PAN modelių segmentavimai.
6. Eksperimentų su įvairiomis konvoliucinių neuroninių tinklų architektūromis rezultatai indikuoja homogeninių ansamblių efektyvumą didesniems duomenų rinkiniams bei labiau išreikštomis segmentavimo klasėms, o distiliavimas labiau pasiteisina mažesnėms duomenų aibėms.

Literatūros sąrašas

1. ZHOU, Zhi-Hua. Learnware: On the Future of Machine Learning. *Frontiers of Computer Science*. 2016, 10 (4), 589-590. ISSN 2095-2236. doi 10.1007/s11704-016-6906-3. Prieiga per: <https://doi.org/10.1007/s11704-016-6906-3>
2. YANG, Yang, ZHAN, De-Chuan, FAN, Ying, JIANG, Yuan, ZHOU, Zhi-Hua. Deep Learning for Fixed Model Reuse. Iš: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, February 4–9, 2017, San Francisco, California, US. AAAI Press, 2017. pp. 2831–2837. doi 10.1609/aaai.v31i1.10855. Prieiga per: <https://doi.org/10.1609/aaai.v31i1.10855>
3. DWORK, Cynthia, et al. Guilt-Free Data Reuse. *Commun.ACM*. 2017, 60 (4), 86–93. ISSN 0001-0782. doi 10.1145/3051088. Prieiga per: <https://doi.org/10.1145/3051088>
4. WOLPERT, David H. Stacked Generalization. *Neural Networks*. 1992, 5 (2), 241-259. ISSN 0893-6080. doi 10.1016/S0893-6080(05)80023-1. Prieiga per: [https://doi.org/10.1016/S0893-6080\(05\)80023-1](https://doi.org/10.1016/S0893-6080(05)80023-1)
5. HINTON, Geoffrey E., VINYALS, Oriol, DEAN, Jeffrey. Distilling the Knowledge in a Neural Network. *arXiv*. 2015, abs/1503.02531. doi 10.48550/arXiv.1503.02531. Prieiga per: <https://doi.org/10.48550/arXiv.1503.02531>.
6. BUCILUUNDEFINED, Cristian, CARUANA, Rich, NICULESCU-MIZIL, Alexandru. Model Compression. Iš: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Philadelphia, PA, US. New York: Association for Computing Machinery, 2006. pp. 535–541. ISBN 1595-933395. doi 10.1145/1150402.1150464. Prieiga per: <https://doi.org/10.1145/1150402.1150464>
7. ZHUANG, Fuzhen, et al. A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*. 2021, 109 (1), 43-76. ISSN 1558-2256. doi 10.1109/JPROC.2020.3004555. Prieiga per: <https://doi.org/10.1109/JPROC.2020.3004555>.
8. LONG, Jonathan, SHELHAMER, Evan, DARRELL, Trevor. Fully Convolutional Networks for Semantic Segmentation. Iš: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015*. pp. 3431-3440. ISBN 1063-6919. doi 10.1109/CVPR.2015.7298965. Prieiga per: <https://doi.org/10.1109/CVPR.2015.7298965>
9. LATEEF, Fahad and RUICHEK, Yassine. Survey on Semantic Segmentation using Deep Learning Techniques. *Neurocomputing*. 2019, 338, 321-348. ISSN 0925-2312. doi 10.1016/j.neucom.2019.02.003. Prieiga per: <https://doi.org/10.1016/j.neucom.2019.02.003>
10. MARTIN, Thoma. A Survey of Semantic Segmentation. *arXiv*. 2016. doi 10.48550/arXiv.1602.06541. Prieiga per: <https://doi.org/10.48550/arXiv.1602.06541>
11. ANTWI-BEKOE, Eldad, LIU, Guisong, AINAM, Jean-Paul, SUN, Guolin, XIE, Xiurui. A Deep Learning Approach for Insulator Instance Segmentation and Defect Detection. *Neural Computing and Applications*. 2022, 34 (9), 7253-7269. ISSN 1433-3058. doi 10.1007/s00521-021-06792-z. Prieiga per: <https://doi.org/10.1007/s00521-021-06792-z>
12. WU, Haitao, et al. Combining Computer Vision with Semantic Reasoning for on-Site Safety Management in Construction. *Journal of Building Engineering*. 2021, 42, 103036. ISSN 2352-7102. doi 10.1016/j.jobbe.2021.103036. Prieiga per: <https://doi.org/10.1016/j.jobbe.2021.103036>
13. KAR, Mithun K., NATH, Malaya K., NEOG, Debanga R. A Review on Progress in Semantic Image Segmentation and its Application to Medical Images. *SN Computer Science*. 2021, 2 (5), 397.

ISSN 2661-8907. doi 10.1007/s42979-021-00784-5. Prieiga per: <https://doi.org/10.1007/s42979-021-00784-5>

14. SAMARASINGHE, Gihan, et al. Deep Learning for Segmentation in Radiation Therapy Planning: A Review. *Journal of Medical Imaging and Radiation Oncology*. 2021, 65 (5), 578-595. doi 10.1111/1754-9485.13286. Prieiga per: <https://doi.org/10.1111/1754-9485.13286>

15. WEI, Yuchen, TRAN, Son, XU, Shuxiang, KANG, Byeong, SPRINGER, Matthew. Deep Learning for Retail Product Recognition: Challenges and Techniques. *Computational Intelligence and Neuroscience*. 2020, 2020, 8875910. ISSN 1687-5265. doi 10.1155/2020/8875910. Prieiga per: <https://doi.org/10.1155/2020/8875910>

16. LI, Jingyu, et al. Lane-DeepLab: Lane Semantic Segmentation in Automatic Driving Scenarios for High-Definition Maps. *Neurocomputing*. 2021, 465, 15-25. ISSN 0925-2312. doi 10.1016/j.neucom.2021.08.105. Prieiga per: <https://doi.org/10.1016/j.neucom.2021.08.105>

17. FENG, Di, et al. Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges. *IEEE Transactions on Intelligent Transportation Systems*. 2021, 22 (3), 1341-1360. ISSN 1524-9050. doi 10.1109/TITS.2020.2972974. Prieiga per: <https://doi.org/10.1109/TITS.2020.2972974>

18. NGUYEN, Minh, YAN, Wei Q., HO, Harvey. MamboNet: Adversarial Semantic Segmentation for Autonomous Driving. Iš: *First International Symposium, ISGV 2021, January 28-29, 2021 Switzerland: Auckland, New Zealand. Switzerland: Springer International Publishing AG, 2021. pp. 354-362. ISBN 9783-030720735. doi 10.1007/978-3-030-72073-5_27. Prieiga per: https://doi.org/10.1007/978-3-030-72073-5_27*

19. WANG, Wenfu, FU, Yongjian, PAN, Zhijie, LI, Xi, ZHUANG, Yueting. Real-Time Driving Scene Semantic Segmentation. *IEEE Access*. 2020, 8, 36776-36788. ISSN 2169-3536. doi 10.1109/ACCESS.2020.2975640. Prieiga per: <https://doi.org/10.1109/ACCESS.2020.2975640>

20. BOSILJ, Petra, APTOULA, Erchan, DUCKETT, Tom, CIELNIAK, Grzegorz. Transfer Learning between Crop Types for Semantic Segmentation of Crops Versus Weeds in Precision Agriculture. *Journal of Field Robotics*. 2019, 37 (1), 7-19. doi 10.1002/rob.21869. Prieiga per: <https://doi.org/10.1002/rob.21869>

21. ANAND, Tanmay, SINHA, Soumendu, MANDAL, Murari, CHAMOLA, Vinay, YU, Fei. AgriSegNet: Deep Aerial Semantic Segmentation Framework for IoT-Assisted Precision Agriculture. *IEEE Sensors Journal*. 2021, 21 (16), 17581-17590. ISSN 2379-9153. doi 10.1109/JSEN.2021.3071290. Prieiga per: <https://doi.org/10.1109/JSEN.2021.3071290>

22. MILIOTO, Andres, LOTTES, Philipp, STACHNISS, Cyrill. Real-Time Semantic Segmentation of Crop and Weed for Precision Agriculture Robots Leveraging Background Knowledge in CNNs. Iš: *2018 IEEE International Conference on Robotics and Automation (ICRA) 2018. pp. 2229-2235. ISBN 2577-087X. doi 10.1109/ICRA.2018.8460962. Prieiga per: https://doi.org/10.1109/ICRA.2018.8460962*

23. YOU, Jie, LIU, Wei, LEE, Joonwhoan. A DNN-Based Semantic Segmentation for Detecting Weed and Crop. *Computers and Electronics in Agriculture*. 2020, 178, 105750. ISSN 0168-1699. doi 10.1016/j.compag.2020.105750. Prieiga per: <https://doi.org/10.1016/j.compag.2020.105750>

24. ALI, Omar, et al. A Systematic Literature Review of Artificial Intelligence in the Healthcare Sector: Benefits, Challenges, Methodologies, and Functionalities. *Journal of Innovation &*

- Knowledge. 2023, 8 (1), 100333. ISSN 2444-569X. doi 10.1016/j.jik.2023.100333. Prieiga per: <https://doi.org/10.1016/j.jik.2023.100333>
25. KAUL, Vivek, ENSLIN, Sarah, GROSS, Seth A. History of Artificial Intelligence in Medicine. *Gastrointestinal Endoscopy*. 2020, 92 (4), 807-812. ISSN 0016-5107. doi 10.1016/j.gie.2020.06.040. Prieiga per: <https://doi.org/10.1016/j.gie.2020.06.040>
26. WANG, Risheng, et al. Medical Image Segmentation using Deep Learning: A Survey. *IET Image Processing*. 2022, 16 (5), 1243-1267. doi 10.1049/ipr2.12419. Prieiga per: <https://doi.org/10.1049/ipr2.12419>
27. KILLOCK, David. AI Outperforms Radiologists in Mammographic Screening. *Nature Reviews Clinical Oncology*. 2020, 17 (3), 134. ISSN 1759-4782. doi 10.1038/s41571-020-0329-7. Prieiga per: <https://doi.org/10.1038/s41571-020-0329-7>
28. EVERINGHAM, Mark, GOOL, Luc V., WILLIAMS, Christopher K. I., WINN, John M., ZISSERMAN, Andrew. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*. 2010, 88, 303-338. doi 10.1007/s11263-009-0275-4. Prieiga per: <https://doi.org/10.1007/s11263-009-0275-4>
29. MOTTAGHI, Roozbeh, et al. The Role of Context for Object Detection and Semantic Segmentation in the Wild. Iš: 2014 IEEE Conference on Computer Vision and Pattern Recognition, June 23-28, 2014, Columbus, OH, US. IEEE, 2014. pp. 891-898. ISBN 1063-6919. doi 10.1109/CVPR.2014.119. Prieiga per: <https://doi.org/10.1109/CVPR.2014.119>
30. HARIHARAN, Bharath, ARBELAEZ, Pablo, BOURDEV, Lubomir, MAJI, Subhansu, MALIK, Jitendra. Semantic Contours from Inverse Detectors. Iš: 2011 International Conference on Computer Vision, November 6-13, 2011, Barcelona, Spain. IEEE, 2011. pp. 991-998. ISBN 1550-5499. doi 10.1109/ICCV.2011.6126343. Prieiga per: <https://doi.org/10.1109/ICCV.2011.6126343>
31. LIN, Tsung-Yi, et al. Microsoft COCO: Common Objects in Context. Iš: 13th European Conference on Computer Vision, September 6-12, 2014, Zurich, Switzerland. Springer, 2014. ISBN 9783-319106021. doi 10.1007/978-3-319-10602-1_48. Prieiga per: https://doi.org/10.1007/978-3-319-10602-1_48
32. CORDTS, Marius, et al. The Cityscapes Dataset for Semantic Urban Scene Understanding. Iš: 2016 IEEE Conference on Computer Vision and Pattern Recognition, June 27-30, 2016, Las Vegas, NV, US. IEEE, 2016. pp. 3213-3223. doi 10.1109/CVPR.2016.350. Prieiga per: <https://doi.org/10.1109/CVPR.2016.350>
33. BROSTOW, Gabriel J., FAUQUEUR, Julien, CIPOLLA, Roberto. Semantic Object Classes in Video: A High-Definition Ground Truth Database. *Pattern Recognition Letters*. 2009, 30 (2), 88-97. ISSN 0167-8655. doi 10.1016/j.patrec.2008.04.005. Prieiga per: <https://doi.org/10.1016/j.patrec.2008.04.005>
34. GOULD, Stephen, FULTON, Richard, KOLLER, Daphne. Decomposing a Scene into Geometric and Semantically Consistent Regions. Iš: 2009 IEEE 12th International Conference on Computer Vision, September 29 - October 2, 2009, Kyoto, Japan. IEEE, 2009. pp. 1-8. ISBN 2380-7504. doi 10.1109/ICCV.2009.5459211. Prieiga per: <https://doi.org/10.1109/ICCV.2009.5459211>
35. LIU, Ce, YUEN, Jenny, TORRALBA, Antonio. SIFT Flow: Dense Correspondence Across Scenes and its Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2011, 33 (5), 978-994. ISSN 1939-3539. doi 10.1109/TPAMI.2010.147. Prieiga per: <https://doi.org/10.1109/TPAMI.2010.147>

36. JAIN, Suyog and GRAUMAN, Kristen. Supervoxel-Consistent Foreground Propagation in Video. Iš: 13th European Conference of Computer Vision, September 6-12, 2014, Zurich, Switzerland. 2014. pp. 656-671. ISBN 9783-319105925. doi 10.1007/978-3-319-10593-2_43. Prieiga per: https://doi.org/10.1007/978-3-319-10593-2_43
37. SHEN, Xiaoyong, et al. Automatic Portrait Segmentation for Image Stylization. Computer Graphics Forum. 2016, 35, 93-102. doi 10.1111/cgf.12814. Prieiga per: <https://doi.org/10.1111/cgf.12814>
38. BELL, Sean, UPCHURCH, Paul, SNAVELY, Noah, BALA, Kavita. Material Recognition in the Wild with the Materials in Context Database. Iš: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 07-12, 2015, Boston, MA, US. IEEE, 2015. pp. 3479-3487. ISBN 1063-6919. doi 10.1109/CVPR.2015.7298970. Prieiga per: <https://doi.org/10.1109/CVPR.2015.7298970>
39. SILBERMAN, Nathan, HOIEM, Derek, KOHLI, Pushmeet, FERGUS, Rob. Indoor Segmentation and Support Inference from RGBD Images. Iš: 12th European Conference on Computer Vision, October 7-13, 2012, Florence, Italy. Springer, 2012. pp. 746–760. doi 10.1007/978-3-642-33715-4_54. Prieiga per: https://doi.org/10.1007/978-3-642-33715-4_54
40. ZHOU, Bolei, et al. Scene Parsing through ADE20K Dataset. Iš: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI, US. IEEE, 2017. pp. 5122-5130. ISBN 1063-6919. doi 10.1109/CVPR.2017.544. Prieiga per: <https://doi.org/10.1109/CVPR.2017.544>
41. CAESAR, Holger, UIJLINGS, Jasper, FERRARI, Vittorio. COCO-Stuff: Thing and Stuff Classes in Context. Iš: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 18-23, 2018, Salt Lake City, UT, US. Los Alamitos, CA, US: IEEE Computer Society, 2018. pp. 1209-1218. doi 10.1109/CVPR.2018.00132. Prieiga per: <https://doi.org/10.1109/CVPR.2018.00132>
42. CHEN, Xianjie, et al. Detect what You can: Detecting and Representing Objects using Holistic Models and Body Parts. Iš: 2014 IEEE Conference on Computer Vision and Pattern Recognition, June 23-28, 2014, Columbus, OH, US. IEEE, 2014. pp. 1979-1986. doi 10.1109/CVPR.2014.254. Prieiga per: <https://doi.org/10.1109/CVPR.2014.254>
43. ZAMIR, Syed W., et al. iSAID: A Large-Scale Dataset for Instance Segmentation in Aerial Images. Iš: IEEE Conference on Computer Vision and Pattern Recognition Workshops, June 16-20, 2019, Long Beach, CA, US. IEEE, 2019. pp. 29-37. Prieiga per: <https://dblp.org/rec/conf/cvpr/ZamirAGKSK00XB19.html>
44. KUZNETSOVA, Alina, et al. The Open Images Dataset V4: Unified Image Classification, Object Detection, and Visual Relationship Detection at Scale. International Journal of Computer Vision. 2020, 128, 1956–1981. doi 10.1007/s11263-020-01316-z. Prieiga per: <https://doi.org/10.1007/s11263-020-01316-z>
45. HANI, Nicolai, ROY, Pravakar, ISLER, Volkan. MinneApple: A Benchmark Dataset for Apple Detection and Segmentation. IEEE Robotics and Automation Letters. 2019, 5 (2), 852-858. doi 10.1109/LRA.2020.2965061. Prieiga per: <https://doi.org/10.1109/LRA.2020.2965061>
46. GUPTA, Agrim, DOLLÁR, Piotr, GIRSHICK, Ross. LVIS: A Dataset for Large Vocabulary Instance Segmentation. Iš: 2019 IEEE/CVF Conference on Computer Vision and Pattern

- Recognition, June 15-20, 2019, Long Beach, CA, US. IEEE, 2019. pp. 5351-5359. ISBN 2575-7075. doi 10.1109/CVPR.2019.00550. Prieiga per: <https://doi.org/10.1109/CVPR.2019.00550>
47. CIOCCA, Gianluigi, NAPOLETANO, Paolo, SCHETTINI, Raimondo. Food Recognition: A New Dataset, Experiments and Results. *IEEE Journal of Biomedical and Health Informatics*. 2017, 21 (3), 588-598. doi 10.1109/JBHI.2016.2636441. Prieiga per: <https://doi.org/10.1109/JBHI.2016.2636441>
48. GEYER, Jakob, et al. A2D2: Audi Autonomous Driving Dataset. *arXiv*. 2020, abs/2004.06320. doi 10.48550/arXiv.2004.06320. Prieiga per: <https://doi.org/10.48550/arXiv.2004.06320>.
49. CHEBROLU, Nived, et al. Agricultural Robot Dataset for Plant Classification, Localization and Mapping on Sugar Beet Fields. *The International Journal of Robotics Research*. 2017. doi 10.1177/0278364917720510.
50. DIETTERICH, Thomas G. Ensemble Methods in Machine Learning. Iš: *First International Workshop on Multiple Classifier Systems*, June 21-23, 2000, Cagliari, Italy. Springer, 2000. pp. 1-15. doi 10.1007/3-540-45014-9_1. Prieiga per: https://doi.org/10.1007/3-540-45014-9_1
51. ROCCA, Joseph. Ensemble Methods: Bagging, Boosting and Stacking, 2019 [žiūrėta 2021-04-20]. Prieiga per: <https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205>
52. YANG, Yongquan, LV, Haijun, CHEN, Ning. A Survey on Ensemble Learning Under the Era of Deep Learning. *Artificial Intelligence Review*. 2022, 56 (6), 5545-5589. ISSN 1573-7462. doi 10.1007/s10462-022-10283-5. Prieiga per: <https://doi.org/10.1007/s10462-022-10283-5>
53. BROWNLEE, Jason. Stacking Ensemble for Deep Learning Neural Networks in Python, 2018 [žiūrėta 2021-03-15]. Prieiga per: <https://machinelearningmastery.com/stacking-ensemble-for-deep-learning-neural-networks/>
54. GANAIE, M. A., HU, Minghui, TANVEER, Muhammad, SUGANTHAN, Ponnuthurai N. Ensemble Deep Learning: A Review. *Engineering Applications of Artificial Intelligence*. 2021, 115, 105151. ISSN 0952-1976. doi 10.1016/j.engappai.2022.105151. Prieiga per: <https://doi.org/10.1016/j.engappai.2022.105151>
55. SCHAPIRE, Robert E. Empirical Inference. Explaining Adaboost. Berlin, Heidelberg: Springer, 2013, pp. 37-52. ISBN 9783642411359.
56. FRIEDMAN, Jerome H. Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*. 2001, 29 (5), 1189. doi 10.1214/aos/1013203451. Prieiga per: <https://doi.org/10.1214/aos/1013203451>
57. LAAN, Mark, POLLEY, Eric C., HUBBARD, Alan E. Super Learner. *Statistical Applications in Genetics and Molecular Biology*. 2007, 6 (1). doi 10.2202/1544-6115.1309. Prieiga per: <https://doi.org/10.2202/1544-6115.1309>
58. KIM, Myeongso, LEE, Minyoung, AN, Minjeong, LEE, Hong-Chul. Effective Automatic Defect Classification Process Based on CNN with Stacking Ensemble Model for TFT-LCD Panel. *Journal of Intelligent Manufacturing*. 2020, 31 (5), 1165-1174. ISSN 1572-8145. Prieiga per: <https://doi.org/10.1007/s10845-019-01502-y>
59. ROY, Saikat, DAS, Arindam, BHATTACHARYA, Ujjwal. Generalized Stacking of Layerwise-Trained Deep Convolutional Neural Networks for Document Image Classification. Iš: *2016 23rd International Conference on Pattern Recognition (ICPR)*, December 4-8, 2016, Cancun, Mexico.

2016. pp. 1273-1278. doi 10.1109/ICPR.2016.7899812. Prieiga per: <https://doi.org/10.1109/ICPR.2016.7899812>
60. YAZDIZADEH, Ali, PATTERSON, Zachary, FAROOQ, Bilal. Ensemble Convolutional Neural Networks for Mode Inference in Smartphone Travel Survey. *IEEE Transactions on Intelligent Transportation Systems*. 2020, 21 (6), 2232-2239. ISSN 1558-0016. doi 10.1109/TITS.2019.2918923. Prieiga per: <https://doi.org/10.1109/TITS.2019.2918923>
61. YANG, Bin, YAN, Junjie, LEI, Zhen, LI, Stan Z. Convolutional Channel Features. Iš: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, December 7-13, 2015, Santiago, Chile. US: IEEE Computer Society, 2015. pp. 82-90. ISBN 9781-467383912. doi 10.1109/ICCV.2015.18. Prieiga per: <https://doi.org/10.1109/ICCV.2015.18>
62. LIU, Y. and YAO, X. Ensemble Learning Via Negative Correlation. *Neural Networks*. 1999, 12 (10), 1399-1404. ISSN 0893-6080. doi 10.1016/S0893-6080(99)00073-8. Prieiga per: [https://doi.org/10.1016/S0893-6080\(99\)00073-8](https://doi.org/10.1016/S0893-6080(99)00073-8)
63. SRIVASTAVA, Nitish, HINTON, Geoffrey E., KRIZHEVSKY, Alex, SUTSKEVER, Ilya, SALAKHUTDINOV, Ruslan. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The Journal of Machine Learning Research*. 2014, 15 (1), 1929-1958. ISSN 1532-4435. doi 10.5555/2627435.2670313. Prieiga per: <https://dl.acm.org/doi/10.5555/2627435.2670313>
64. HUANG, Gao, SUN, Yu, LIU, Zhuang, SEDRA, Daniel, WEINBERGER, Kilian Q. Deep Networks with Stochastic Depth. Iš: *14th European Conference on Computer Vision*, October 11-14, 2016, Amsterdam, The Netherlands. Springer International Publishing, 2016. pp. 646-661. ISBN 9783-319464930. doi 10.1007/978-3-319-46493-0_39. Prieiga per: https://doi.org/10.1007/978-3-319-46493-0_39
65. SINGH, Saurabh, HOIEM, Derek, FORSYTH, David. Swapout: Learning an Ensemble of Deep Architectures. Iš: *30th International Conference on Neural Information Processing Systems*, December 5-10, 2016, Barcelona, Spain. Red Hook, NY, US: Curran Associates Inc, 2016. pp. 28-36. ISBN 9781-510838819
66. XIE, Lingxi, WANG, Jingdong, WEI, Zhen, WANG, Meng, TIAN, Qi. DisturbLabel: Regularizing CNN on the Loss Layer. Iš: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 27-30, 2016 Los Alamitos, CA, US: IEEE Computer Society, 2016. pp. 4753-4762. ISBN 1063-6919. doi 10.1109/CVPR.2016.514. Prieiga per: <https://doi.org/10.1109/CVPR.2016.514>
67. GOODFELLOW, Ian J., BENGIO, Yoshua, COURVILLE, Aaron. *Deep Learning*. Cambridge, MA, US: MIT Press, 2016. ISBN 0262035618.
68. KUNCHEVA, Ludmila. *Combining Pattern Classifiers: Methods and Algorithms: Second Edition. Bagging and Boosting*. John Wiley & Sons, 2014, pp. 203-235. ISBN 9780471660262.
69. LOWD, Daniel and DOMINGOS, Pedro. Naive Bayes Models for Probability Estimation. Iš: *22nd International Conference on Machine Learning*, August 7 - 11, 2005, Bonn, Germany. New York, US: Association for Computing Machinery, 2005. pp. 529-536. doi 10.1145/1102351.1102418. Prieiga per: <https://doi.org/10.1145/1102351.1102418>
70. XU, Lei, KRZYZAK, Adam, SUEN, Ching. Methods of Combining Multiple Classifiers and their Applications to Handwriting Recognition. *IEEE Transactions on Systems, Man, and Cybernetics*. 1992, 22 (3), 418-435. ISSN 2168-2909. doi 10.1109/21.155943. Prieiga per: <https://doi.org/10.1109/21.155943>

71. JU, Cheng, BIBAUT, Aurelien, LAAN, Mark. The Relative Performance of Ensemble Methods with Deep Convolutional Neural Networks for Image Classification. *Journal of Applied Statistics*. 2017, 45 (15), 2800-2818. doi 10.1080/02664763.2018.1441383. Prieiga per: <https://doi.org/10.1080/02664763.2018.1441383>
72. PALADINI, Emanuela, et al. Two Ensemble-CNN Approaches for Colorectal Cancer Tissue Type Classification. *Journal of Imaging*. 2021, 7 (3), 51. doi 10.3390/jimaging7030051. Prieiga per: <https://doi.org/10.3390/jimaging7030051>
73. SYED, Muslim, HASHMANI, Manzoor, REHMAN, Mobashar, BUDIMAN, Arif. Adaptive CNN Ensemble for Complex Multispectral Image Analysis. *Complexity*. 2020, 2020, 1-21. ISSN 1076-2787. doi 10.1155/2020/8361989. Prieiga per: <https://doi.org/10.1155/2020/8361989>
74. HAN, Shizhong, MENG, Zibo, KHAN, Ahmed-Shehab, TONG, Yan. Incremental Boosting Convolutional Neural Network for Facial Action Unit Recognition. Iš: 30th International Conference on Neural Information Processing Systems, December 5-10, 2016, Barcelona, Spain. Red Hook, NY, US: Curran Associates Inc., 2016. pp. 109-117
75. MOGHIMI, Mohammad, et al. Boosted Convolutional Neural Networks. Iš: British Machine Vision Conference, September 19-22, 2016, York, UK. BMVA Press, 2016. pp. 24.1-24.13. doi 10.5244/C.30.24. Prieiga per: <https://doi.org/10.5244/C.30.24>
76. BRAHIMI, Sourour, BEN AOUN, Najib, BEN AMAR, Chokri. Boosted Convolutional Neural Network for Object Recognition at Large Scale. *Neurocomputing*. 2019, 330, 337-354. ISSN 0925-2312. doi 10.1016/j.neucom.2018.11.031. Prieiga per: <https://doi.org/10.1016/j.neucom.2018.11.031>
77. LI, Hongyang, OUYANG, Wanli, WANG, Xiaogang. Multi-Bias Non-Linear Activation in Deep Neural Networks. Iš: 33rd International Conference on Machine Learning, June 19 - 24, 2016, New York, US. JMLR.org, 2016. pp. 221-229
78. OPITZ, Michael, WALTNER, Georg, POSSEGGGER, Horst, BISCHOF, Horst. BIER — Boosting Independent Embeddings Robustly. Iš: 2017 IEEE International Conference on Computer Vision (ICCV), October 22-29, 2017, Venice, Italy. IEEE, 2017. pp. 5199-5208. ISBN 2380-7504. doi 10.1109/ICCV.2017.555. Prieiga per: <https://doi.org/10.1109/ICCV.2017.555>
79. YUAN, Yuhui, YANG, Kuiyuan, ZHANG, Chao. Hard-Aware Deeply Cascaded Embedding. Iš: 2017 IEEE International Conference on Computer Vision (ICCV), October 22-29, 2017, Venice, Italy. IEEE, 2017. pp. 814-823. ISBN 2380-7504. doi 10.1109/ICCV.2017.94. Prieiga per: <https://doi.org/10.1109/ICCV.2017.94>
80. SERBETCI, Ayse and AKGUL, Yusuf S. End-to-End Training of CNN Ensembles for Person Re-Identification. *Pattern Recognition*. 2020, 104, 107319. ISSN 0031-3203. doi 10.1016/j.patcog.2020.107319. Prieiga per: <https://doi.org/10.1016/j.patcog.2020.107319>
81. BROWN, Tom, et al. Language Models are Few-Shot Learners. Iš: 34th International Conference on Neural Information Processing Systems, December 6-12, 2020, Vancouver, Canada. New York, US: Curran Associates Inc., 2020. pp. 1877–1901.
82. GREENE, Tristan. Google's New Trillion-Parameter AI Language Model is almost 6 Times Bigger than GPT-3, 2021 [žiūrėta 2022-03-09]. Prieiga per: <https://thenextweb.com/news/googles-new-trillion-parameter-ai-language-model-is-almost-6-times-bigger-than-gpt-3>
83. PETERSON, Hannah. An Overview of Model Compression Techniques for Deep Learning in Space, 2020 [žiūrėta 2022-03-09]. Prieiga per: <https://medium.com/gsi-technology/an-overview-of-model-compression-techniques-for-deep-learning-in-space-3fd8d4ce84e5>.

84. CHENG, Yu, WANG, Duo, ZHOU, Pan, ZHANG, Tao. Model Compression and Acceleration for Deep Neural Networks: The Principles, Progress, and Challenges. *IEEE Signal Processing Magazine*. 2018, 35 (1), 126-136. ISSN 1558-0792. doi 10.1109/MSP.2017.2765695. Prieiga per: <https://doi.org/10.1109/MSP.2017.2765695>
85. HANSON, Stephen and PRATT, Lorien. Comparing Biases for Minimal Network Construction with Back-Propagation. Iš: 1st International Conference on Neural Information Processing Systems, January 1, 1988, Denver, US. Cambridge, US: MIT Press, 1988. pp. 177–185.
86. HASSIBI, Babak and STORK, David G. Second Order Derivatives for Network Pruning: Optimal Brain Surgeon. Iš: 5th International Conference on Neural Information Processing Systems, November 30 - December 3, 1992, Denver, US. San Francisco, US: Morgan Kaufmann Publishers Inc., 1992. pp. 164–171.
87. GONG, Yunchao, LIU, L., YANG, Ming, BOURDEV, Lubomir D. Compressing Deep Convolutional Networks using Vector Quantization. *ArXiv*. 2014, abs/1412.6115. doi 10.48550/arXiv.1412.6115. Prieiga per: <https://doi.org/10.48550/arXiv.1412.6115>.
88. CHEN, Wenlin, WILSON, James T., TYREE, Stephen, WEINBERGER, Kilian Q., CHEN, Yixin. Compressing Neural Networks with the Hashing Trick. Iš: 32nd International Conference on International Conference on Machine Learning, July 6-11, 2015, Lille, France. *JMLR.org*, 2015. pp. 2285–2294.
89. HAN, Song, POOL, Jeff, TRAN, John, DALLY, William J. Learning both Weights and Connections for Efficient Neural Network. Iš: 28th International Conference on Neural Information Processing Systems, December 7 - 12, 2015, Montreal, Canada. Cambridge, US: MIT Press, 2015. pp. 1135–1143.
90. SRINIVAS, Suraj and BABU, R. V. Data-Free Parameter Pruning for Deep Neural Networks. Iš: British Machine Vision Conference 2015, September 7-9, 2015, Swansea, UK. *BMVA Press*, 2015.
91. COURBARIAUX, Matthieu, BENGIO, Yoshua, DAVID, Jean-Pierre. BinaryConnect: Training Deep Neural Networks with Binary Weights during Propagations. Iš: 28th International Conference on Neural Information Processing Systems, December 7 - 12, 2015, Montreal, Canada. Cambridge, US: MIT Press, 2015. pp. 3123–3131.
92. GUPTA, Suyog, AGRAWAL, Ankur, GOPALAKRISHNAN, K., NARAYANAN, Pritish. Deep Learning with Limited Numerical Precision. Iš: 32nd International Conference on International Conference on Machine Learning, July 6-11, 2015, Lille, France. *JMLR.org*, 2015. pp. 1737–1746.
93. WEN, Wei, WU, Chunpeng, WANG, Yandan, CHEN, Yiran, LI, Hai H. Learning Structured Sparsity in Deep Neural Networks. Iš: 30th International Conference on Neural Information Processing Systems, December 5-10, 2016, Barcelona, Spain. Red Hook, New York, United States: Curran Associates Inc., 2016. pp. 2082–2090.
94. ZHOU, Hao, ALVAREZ, Jose M., PORIKLI, Fatih. Less is More: Towards Compact CNNs. Iš: European Conference on Computer Vision 2016.
95. LEBEDEV, Vadim and LEMPITSKY, Victor. Fast ConvNets using Group-Wise Brain Damage. Iš: 2016 IEEE Conference on Computer Vision and Pattern Recognition, June 27-30, 2016, Las Vegas, NV, US. 2016. pp. 2554-2564. ISBN 1063-6919. doi 10.1109/CVPR.2016.280. Prieiga per: <https://doi.org/10.1109/CVPR.2016.280>
96. MEROLLA, Paul, APPUSWAMY, Rathinakumar, ARTHUR, John V., ESSER, Steven K., MODHA, Dharmendra S. Deep Neural Networks are Robust to Weight Binarization and Other Non-

- Linear Distortions. arXiv. 2016, abs/1606.01981. doi 10.48550/arXiv.1606.01981. Prieiga per: <https://doi.org/10.48550/arXiv.1606.01981>
97. RASTEGARI, Mohammad, ORDONEZ, Vicente, REDMON, Joseph, FARHADI, Ali. XNOR-Net: ImageNet Classification using Binary Convolutional Neural Networks. Iš: Computer Vision - ECCV 2016 Cham: Springer International Publishing, 2016. pp. 525-542. ISBN 9783-319464930.
98. WU, J., LENG, C., WANG, Y., HU, Q., CHENG, J. Quantized Convolutional Neural Networks for Mobile Devices. Iš: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), jun Los Alamitos, CA, US: IEEE Computer Society, 2016. pp. 4820-4828. ISBN 1063-6919. doi 10.1109/CVPR.2016.521.
99. LI, Hao, KADAV, Asim, DURDANOVIC, Igor, SAMET, Hanan, GRAF, Hans P. Pruning Filters for Efficient ConvNets. arXiv. 2016, abs/1608.08710. doi 10.48550/arXiv.1608.08710. Prieiga per: <https://doi.org/10.48550/arXiv.1608.08710>
100. ULLRICH, Karen, MEEDS, Edward, WELLING, Max. Soft Weight-Sharing for Neural Network Compression. arXiv. 2017, abs/1702.04008. doi 10.48550/arXiv.1702.04008. Prieiga per: <https://doi.org/10.48550/arXiv.1702.04008>
101. HOU, Lu, YAO, Quanming, KWOK, James T. Loss-Aware Binarization of Deep Networks. arXiv. 2017. doi 10.48550/arXiv.1611.01600. Prieiga per: <https://doi.org/10.48550/arXiv.1611.01600>.
102. ZHU, Chenzhuo, HAN, Song, MAO, Huizi, DALLY, William J. Trained Ternary Quantization. arXiv. 2016, abs/1612.01064. doi 10.48550/arXiv.1612.01064. Prieiga per: <https://doi.org/10.48550/arXiv.1612.01064>
103. CHOI, Yoojin, EL-KHAMY, Mostafa, LEE, Jungwon. Towards the Limit of Network Quantization. arXiv. 2016, abs/1612.01543. doi 10.48550/arXiv.1612.01543. Prieiga per: <https://doi.org/10.48550/arXiv.1612.01543>
104. SIRONI, Amos, TEKIN, Bugra, RIGAMONTI, Roberto, LEPETIT, Vincent, FUA, Pascal. Learning Separable Filters. Iš: 2013 IEEE Conference on Computer Vision and Pattern Recognition, June 23-28, 2013, Portland, OR, US. IEEE, 2013. pp. 2754-2761. ISBN 1063-6919. doi 10.1109/CVPR.2013.355. Prieiga per: <https://doi.org/10.1109/CVPR.2013.355>.
105. DENIL, Misha, SHAKIBI, Babak, DINH, Laurent, RANZATO, Marc'Aurelio, FREITAS, Nando d. Predicting Parameters in Deep Learning. Iš: 26th International Conference on Neural Information Processing Systems, December 5 - 10, 2013, Lake Tahoe, NV, US. Curran Associates Inc., 2013. pp. 2148-2156.
106. SAINATH, Tara, KINGSBURY, Brian, SINDHWANI, Vikas, ARISOY, Ebru, RAMABHADRAN, Bhuvana. Low-Rank Matrix Factorization for Deep Neural Network Training with High-Dimensional Output Targets. Iš: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, May 26-31, 2013, Vancouver, Canada. IEEE, 2013. pp. 6655-6659. ISBN 2379-190X. doi 10.1109/ICASSP.2013.6638949. Prieiga per: <https://doi.org/10.1109/ICASSP.2013.6638949>
107. DENTON, Emily L., ZAREMBA, Wojciech, BRUNA, Joan, LECUN, Yann, FERGUS, Rob. Exploiting Linear Structure within Convolutional Networks for Efficient Evaluation. arXiv. 2014, abs/1404.0736. doi 10.48550/arXiv.1404.0736. Prieiga per: <https://doi.org/10.48550/arXiv.1404.0736>

108. JADERBERG, Max, VEDALDI, Andrea, ZISSERMAN, Andrew. Speeding Up Convolutional Neural Networks with Low Rank Expansions. arXiv. 2014, abs/1405.3866. doi 10.48550/arXiv.1405.3866. Prieiga per: <https://doi.org/10.48550/arXiv.1405.3866>
109. LEBEDEV, Vadim, GANIN, Yaroslav, RAKHUBA, Maksim, OSELEDETS, I., LEMPITSKY, Victor S. Speeding-Up Convolutional Neural Networks using Fine-Tuned CP-Decomposition. arXiv. 2014, abs/1412.6553. doi 10.48550/arXiv.1412.6553. Prieiga per: <https://doi.org/10.48550/arXiv.1412.6553>
110. TAI, Cheng, XIAO, Tong, WANG, Xiaogang, WEINAN, E. Convolutional Neural Networks with Low-Rank Regularization. arXiv. 2015. doi 10.48550/arXiv.1511.06067. Prieiga per: <https://doi.org/10.48550/arXiv.1511.06067>
111. COHEN, Taco and WELLING, Max. Group Equivariant Convolutional Networks. Iš: 33rd International Conference on Machine Learning, June 19 - 24, 2016, New York, US. JMLR.org, 2016. pp. 2990-2999.
112. ZHAI, Shuangfei, CHENG, Yu, ZHANG, Zhongfei, LU, Weining. Doubly Convolutional Neural Networks. Iš: 30th International Conference on Neural Information Processing Systems, December 5 - 10, 2016, Barcelona, Spain. Curran Associates Inc., 2016. pp. 1090–1098.
113. SHANG, Wenling, SOHN, Kihyuk, ALMEIDA, Diogo, LEE, Honglak. Understanding and Improving Convolutional Neural Networks Via Concatenated Rectified Linear Units. Iš: 33rd International Conference on International Conference on Machine Learning, June 19 - 24, 2016, New York, US. JMLR.org, 2016. pp. 2217–2225.
114. DIELEMAN, Sander, FAUW, Jeffrey D., KAVUKCUOGLU, Koray. Exploiting Cyclic Symmetry in Convolutional Neural Networks. Iš: 33rd International Conference on International Conference on Machine Learning, June 19 - 24, 2016, New York, US. JMLR.org, 2016. pp. 1889–1898.
115. SZEGEDY, Christian, IOFFE, Sergey, VANHOUCHE, Vincent, ALEMI, Alexander. Inception-V4, Inception-ResNet and the Impact of Residual Connections on Learning. Iš: Thirty-First AAAI Conference on Artificial Intelligence, February 4 - 9, 2017, San Francisco, CA, US. AAAI Press, 2016. pp. 4278–4284. doi 10.1609/aaai.v31i1.11231. Prieiga per: <https://doi.org/10.1609/aaai.v31i1.11231>
116. WU, Bichen, IANDOLA, Forrest, JIN, Peter, KEUTZER, Kurt, WAN, Alvin. SqueezeDet: Unified, Small, Low Power Fully Convolutional Neural Networks for Real-Time Object Detection for Autonomous Driving. Iš: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), July 21-26, 2017, Honolulu, HI, US. IEEE, 2017. pp. 446-454. ISBN 2160-7516. doi 10.1109/CVPRW.2017.60. Prieiga per: <https://doi.org/10.1109/CVPRW.2017.60>
117. BA, Jimmy and CARUANA, Rich. Do Deep Nets really Need to be Deep? Iš: 27th International Conference on Neural Information Processing Systems, December 8 - 13, 2014, Montreal, Canada. Cambridge, MA, US: MIT Press, 2014. pp. 2654–2662.
118. ROMERO, Adriana, et al. FitNets: Hints for Thin Deep Nets. arXiv. 2014, abs/1412.6550. doi 10.48550/arXiv.1412.6550. Prieiga per: [10.48550/arXiv.1412.6550](https://doi.org/10.48550/arXiv.1412.6550)
119. KORATTIKARA, Anoop, RATHOD, Vivek, MURPHY, Kevin, WELLING, Max. Bayesian Dark Knowledge. 2015.
120. LUO, Ping, ZHU, Zhenyao, LIU, Ziwei, WANG, Xiaogang, TANG, Xiaoou. Face Model Compression by Distilling Knowledge from Neurons. Iš: Thirtieth AAAI Conference on Artificial

- Intelligence, February 12 - 17, 2016, Phoenix, AZ, US. AAAI Press, 2016. pp. 3560–3566. doi 10.1609/aaai.v30i1.10449. Prieiga per: <https://doi.org/10.1609/aaai.v30i1.10449>
121. CHEN, Tianqi, GOODFELLOW, Ian J., SHLENS, Jonathon. Net2Net: Accelerating Learning Via Knowledge Transfer. arXiv. 2015, abs/1511.05641. doi 10.48550/arXiv.1511.05641. Prieiga per: <https://doi.org/10.48550/arXiv.1511.05641>
122. ZAGORUYKO, Sergey and KOMODAKIS, Nikos. Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks Via Attention Transfer. arXiv. 2016, abs/1612.03928. doi 10.48550/arXiv.1612.03928. Prieiga per: <https://doi.org/10.48550/arXiv.1612.03928>
123. Kelvin. Distilling the Knowledge in a Neural Network, 2020 [žiūrėta 2021-05-31]. Prieiga per: <https://towardsdatascience.com/distilling-knowledge-in-neural-network-d8991faa2cdc>
124. DENG, Li. The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. IEEE Signal Processing Magazine. 2012, 29 (6), 141-142. ISSN 1558-0792. doi 10.1109/MSP.2012.2211477.
125. WANG, Kuan-Chieh J., et al. Adversarial Distillation of Bayesian Neural Network Posteriors. Iš: 35th International Conference on Machine Learning, July 10-15, 2018, Stockholm, Sweden. PMLR, 2018. pp. 5190-5199.
126. WELLING, Max and TEH, Yee W. Bayesian Learning Via Stochastic Gradient Langevin Dynamics. Iš: 28th International Conference on International Conference on Machine Learning, June 28 - July 2, 2011, Bellevue, WA, US. Madison, WI, US: Omnipress, 2011. pp. 681–688.
127. WONG, Jeremy and GALES, M. J. F. Multi-Task Ensembles with Teacher-Student Training. Iš: 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), December 16-20, 2017, Okinawa, Japan. IEEE, 2017. pp. 84-90. doi 10.1109/ASRU.2017.8268920. Prieiga per: <https://doi.org/10.1109/ASRU.2017.8268920>
128. MALININ, Andrey, MLODOZENIEC, Bruno, GALES, Mark J. F. Ensemble Distribution Distillation. arXiv. 2019, abs/1905.00076. doi 10.48550/arXiv.1905.00076. Prieiga per: <https://doi.org/10.48550/arXiv.1905.00076>
129. COVER, Thomas M. and THOMAS, Joy A. Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing). US: Wiley-Interscience, 2006. ISBN 0471241954.
130. KRIZHEVSKY, Alex. Learning Multiple Layers of Features from Tiny Images. 2009, 32-33. Prieiga per: <https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf>.
131. LE, Y. and YANG, X .S. Tiny ImageNet Visual Recognition Challenge, 2015.
132. SIMONYAN, Karen and ZISSERMAN, Andrew. Very Deep Convolutional Networks for Large-Scale Image Recognition. Iš: 3rd International Conference on Learning Representations, May 7-9, 2015, San Diego, CA, US. 2015. pp. 1-14. Prieiga per: <http://arxiv.org/abs/1409.1556>
133. GAROFOLO, John S. TIMIT Acoustic-Phonetic Continuous Speech Corpus. 1993. doi: 10.35111/17gk-bn40.
134. ALOYSIUS, Neena and MADATHILKULANGARA, Geetha. Image Classification using an Ensemble-Based Deep CNN. Iš: 5th International Conference on Advanced Computing, Networking, and Informatics, June 1-3, 2017, Goa, India. 2018. pp. 445-456. doi 10.1007/978-981-10-8633-5_44. Prieiga per: https://doi.org/10.1007/978-981-10-8633-5_44

135. KIM, Wonsik, GOYAL, Bhavya, CHAWLA, Kunal, LEE, Jungmin, KWON, Keunjoo. Attention-Based Ensemble for Deep Metric Learning. Iš: 15th European Conference of Computer Vision, September 8-14, 2018, Munich, Germany. Springer, 2018. pp. 760–777. ISBN 9783-030012465. doi 10.1007/978-3-030-01246-5_45. Prieiga per: https://doi.org/10.1007/978-3-030-01246-5_45
136. HE, Kaiming, ZHANG, X., REN, Shaoqing, SUN, Jian. Deep Residual Learning for Image Recognition. Iš: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 27-30, 2016, Las Vegas, NV, US. IEEE, 2015. pp. 770-778. doi 10.1109/CVPR.2016.90. Prieiga per: <https://doi.org/10.1109/CVPR.2016.90>
137. HOLLIDAY, Andrew, BAREKATAIN, Mohammadamin, LAURMAA, Johannes, KANDASWAMY, Chetak, PRENDINGER, Helmut. Speedup of Deep Learning Ensembles for Semantic Segmentation using a Model Compression Technique. *Computer Vision and Image Understanding*. 2017, 164, 16-26. ISSN 1077-3142. doi 10.1016/j.cviu.2017.05.004. Prieiga per: <https://doi.org/10.1016/j.cviu.2017.05.004>
138. KRIZHEVSKY, Alex, SUTSKEVER, Ilya, HINTON, Geoffrey E. ImageNet Classification with Deep Convolutional Neural Networks. *Communications of the ACM*. 2017, 60 (6), 84–90. ISSN 0001-0782. doi 10.1145/3065386. Prieiga per: <https://doi.org/10.1145/3065386>
139. LECUN, Yann, BOTTOU, Leon, BENGIO, Y., HAFFNER, Patrick. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*. 1998, 86 (11), 2278-2324. ISSN 1558-2256. doi 10.1109/5.726791. Prieiga per: <https://doi.org/10.1109/5.726791>
140. DENG, Jia, et al. ImageNet: A Large-Scale Hierarchical Image Database. Iš: 2009 IEEE Conference on Computer Vision and Pattern Recognition, June 20-25, 2009, Miami, FL, US. IEEE, 2009. pp. 248-255. ISBN 1063-6919. doi 10.1109/CVPR.2009.5206848. Prieiga per: <https://doi.org/10.1109/CVPR.2009.5206848>
141. MALININ, Andrey and GALES, Mark J. F. Predictive Uncertainty Estimation Via Prior Networks. Iš: 32nd International Conference on Neural Information Processing Systems, December 3 - 8, 2018, Montreal, Canada. New York, US: Curran Associates Inc., 2018. pp. 7047–7058.
142. TEICHMANN, Marvin T. T. and CIPOLLA, Roberto. Convolutional CRFs for Semantic Segmentation. *arXiv*. 2018, abs/1805.04777. doi 10.48550/arXiv.1805.04777. Prieiga per: <https://doi.org/10.48550/arXiv.1805.04777>
143. KR AHENB UHL, Philipp and KOLTUN, Vladlen. Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. Iš: 24th International Conference on Neural Information Processing Systems, December 12 - 15, 2011, Granada, Spain. New York, US: Curran Associates, Inc, 2011.
144. SARKAR, Dipanjan. A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning, 2018 [žiūrėta 2022-05-02]. Prieiga per: <https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a>
145. PAN, Sinno and YANG, Qiang. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*. 2010, 22 (10), 1345-1359. ISSN 2326-3865. doi 10.1109/TKDE.2009.191. Prieiga per: <https://doi.org/10.1109/TKDE.2009.191>
146. LI, Xuhong, GRANDVALET, Yves, DAVOINE, Franck. Explicit Inductive Bias for Transfer Learning with Convolutional Networks. Iš: 35th International Conference on Machine Learning, July 10-15, 2018, Stockholm, Sweden. PMLR, 2018. pp. 2830-2839.

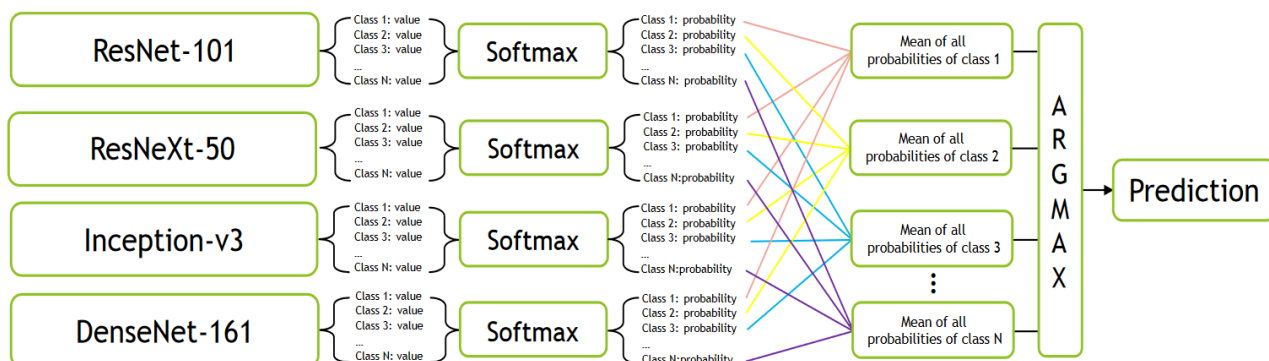
147. WANG, K., et al. Pay Attention to Features, Transfer Learn Faster CNNs. 2020.
148. LIU, Zhuang, et al. Learning Efficient Convolutional Networks through Network Slimming. Iš: 2017 IEEE International Conference on Computer Vision (ICCV), October 22-29, 2017, Venice, Italy. Los Alamitos, CA, US: IEEE Computer Society, 2017. pp. 2755-2763. ISBN 2380-7504. doi 10.1109/ICCV.2017.298. Prieiga per: <https://doi.org/10.1109/ICCV.2017.298>
149. HE, Yang, KANG, Guoliang, DONG, Xuanyi, FU, Yanwei, YANG, Yi. Soft Filter Pruning for Accelerating Deep Convolutional Neural Networks. Iš: Proceedings of the 27th International Joint Conference on Artificial Intelligence, July 13 - 19, 2018, Stockholm, Sweden. AAAI Press, 2018. pp. 2234–2240. ISBN 9780-999241127
150. SUSILO, Marco. Machine Learning Model Deployment Options, 2021 [žiūrėta 2022-05-03]. Prieiga per: <https://towardsdatascience.com/machine-learning-model-deployment-options-47c1f3d77626>
151. BILOGUR, Aleksey. MLOps Concepts for Busy Engineers: Model Serving, 2020 [žiūrėta 2021-05-03]. Prieiga per: <https://spell.ml/blog/mlops-concepts-model-serving-X385IREAACcAAGzS>
152. LUBLINKSY, Boris. Serving Machine Learning Models. O'Reilly Media, Inc., 2017. ISBN 9781492024088.
153. DESISLAVOV, Radosvet, MARTÍNEZ-PLUMED, Fernando, HERNÁNDEZ-ORALLO, José. Trends in AI Inference Energy Consumption: Beyond the Performance-Vs-Parameter Laws of Deep Learning. Sustainable Computing: Informatics and Systems. 2023, 38, 100857. ISSN 2210-5379. doi 10.1016/j.suscom.2023.100857. Prieiga per: <https://doi.org/10.1016/j.suscom.2023.100857>
154. FENG, Junxi, et al. Reconstruction of Porous Media from Extremely Limited Information using Conditional Generative Adversarial Networks. Phys.Rev.E. Sep. 2019, 100 (3), 033308. doi 10.1103/PhysRevE.100.033308. Prieiga per: <https://doi.org/10.1103/PhysRevE.100.033308>
155. PRÖVE, Paul-Louis. An Introduction to Different Types of Convolutions in Deep Learning, 2017 [žiūrėta 2023-04-23]. Prieiga per: <https://towardsdatascience.com/types-of-convolutions-in-deep-learning-717013397f4d>
156. IOFFE, Sergey and SZEGEDY, Christian. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Iš: 32nd International Conference on Machine Learning, July 6 - 11, 2015, Lille, France. PMLR, 2015. pp. 448-456.
157. IMAN, Mohammadreza, RASHEED, Khaled M., ARABNIA, Hamid R. A Review of Deep Transfer Learning and Recent Advancements. Technologies. 2023, 11 (2). doi 10.3390/technologies11020040. Prieiga per: <https://doi.org/10.3390/technologies11020040>
158. GU, Jiuxiang, et al. Recent Advances in Convolutional Neural Networks. Pattern Recognition. 2018, 77, 354-377. ISSN 0031-3203. doi 10.1016/j.patcog.2017.10.013. Prieiga per: <https://doi.org/10.1016/j.patcog.2017.10.013>
159. ROBBINS, Herbert and MONRO, Sutton. A Stochastic Approximation Method. Annals of Mathematical Statistics. 1951, 22, 400-407. doi 10.1214/aoms/1177729586. Prieiga per: <https://doi.org/10.1214/aoms/1177729586>
160. GADDIPATI, Sasi, PLÖGER, Paul, HOCHGESCHWENDER, Nico and METZLER, Tim. Automatic Formative Assessment for Students' Short Text Answers through Feature Extraction. 2021.

161. RUDER, Sebastian. An Overview of Gradient Descent Optimization Algorithms. arXiv. 2016, abs/1609.04747. doi 10.48550/arXiv.1609.04747. Prieiga per: <https://doi.org/10.48550/arXiv.1609.04747>
162. YING, Xue. An Overview of Overfitting and its Solutions. Journal of Physics: Conference Series. 2019, 1168 (2), 022022. doi 10.1088/1742-6596/1168/2/022022. Prieiga per: <https://doi.org/10.1088/1742-6596/1168/2/022022>
163. ZHAO, Hengshuang, SHI, Jianping, QI, Xiaojuan, WANG, Xiaogang, JIA, Jiaya. Pyramid Scene Parsing Network. Iš: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI, US. IEEE, 2017. pp. 6230-6239. ISBN 1063-6919. doi 10.1109/CVPR.2017.660. Prieiga per: <https://doi.org/10.1109/CVPR.2017.660>
164. CHAURASIA, Abhishek and CULURCIELLO, Eugenio. LinkNet: Exploiting Encoder Representations for Efficient Semantic Segmentation. Iš: 2017 IEEE Visual Communications and Image Processing (VCIP), December 10-13, 2017, St. Petersburg, FL, US. IEEE, 2017. pp. 1-4. doi 10.1109/VCIP.2017.8305148. Prieiga per: <https://doi.org/10.1109/VCIP.2017.8305148>
165. LIN, Tsung-Yi., et al. Feature Pyramid Networks for Object Detection. Iš: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI, US. Los Alamitos, CA, US: IEEE Computer Society, 2017. pp. 936-944. ISBN 1063-6919. doi 10.1109/CVPR.2017.106. Prieiga per: <https://doi.org/10.1109/CVPR.2017.106>
166. LI, Hanchao, XIONG, Pengfei, AN, Jie, WANG, Lingxue. Pyramid Attention Network for Semantic Segmentation. arXiv. 2018, abs/1805.10180. doi 10.48550/arXiv.1805.10180. Prieiga per: <https://doi.org/10.48550/arXiv.1805.10180>
167. CHEN, Liang-Chieh, PAPANDREOU, George, KOKKINOS, Iasonas, YUILLE, Alan. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2018, 40 (4), 834-848. ISSN 1939-3539. doi 10.1109/TPAMI.2017.2699184. Prieiga per: <https://doi.org/10.1109/TPAMI.2017.2699184>
168. HU, Jie, SHEN, Li, SUN, Gang. Squeeze-and-Excitation Networks. Iš: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT, US. IEEE, 2018. pp. 7132-7141. ISBN 1063-6919. doi 10.1109/CVPR.2018.00745.
169. TAN, Mingxing and LE, Quoc. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Iš: 36th International Conference on Machine Learning, June 9-15, 2019, Long Beach, CA, US. PMLR, 2019. pp. 6105-6114.
170. BARTON, Maxwell and LENNOX, Barry. Model Stacking to Improve Prediction and Variable Importance Robustness for Soft Sensor Development. Digital Chemical Engineering. 2022, 3, 100034. ISSN 2772-5081. doi 10.1016/j.dche.2022.100034. Prieiga per: <https://doi.org/10.1016/j.dche.2022.100034>
171. HUANG, Gao, et al. Snapshot Ensembles: Train 1, Get M for Free. arXiv. 2017, abs/1704.00109. doi 10.48550/arXiv.1704.00109. Prieiga per: <https://doi.org/10.48550/arXiv.1704.00109>
172. DICE, Lee R. Measures of the Amount of Ecologic Association between Species. Ecology. 1945, 26 (3), 297-302. doi 10.2307/1932409. Prieiga per: <https://doi.org/10.2307/1932409>
173. ŠIAULYS, Andrius, et al. A Fully-Annotated Imagery Dataset of Sublittoral Benthic Species in Svalbard, Arctic. Data in Brief. 2021, 35, 106823. ISSN 2352-3409. doi 10.1016/j.dib.2021.106823. Prieiga per: <https://doi.org/10.1016/j.dib.2021.106823>

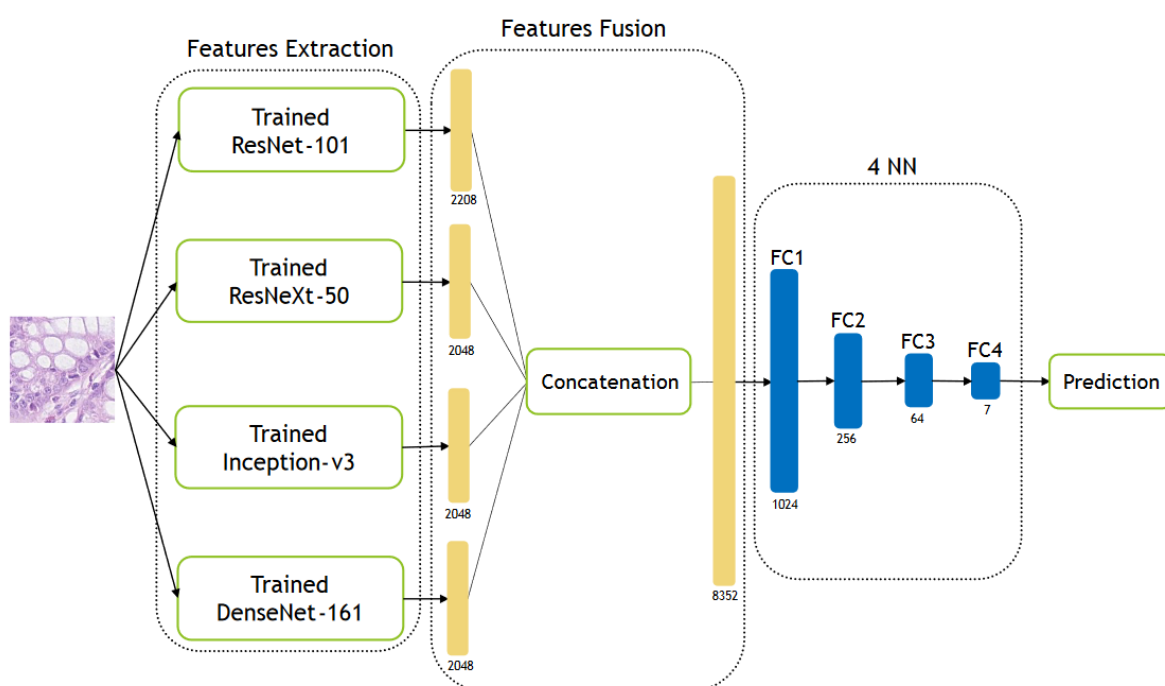
174. Labelbox. Labelbox, 2023. Prieiga per: <https://labelbox.com>.
175. ALKHARUSI, Hussain. Categorical Variables in Regression Analysis: A Comparison of Dummy and Effect Coding. *International Journal of Education*. 2012, 4 (2), 202-210. doi 10.5296/ije.v4i2.1962. Prieiga per: <https://doi.org/10.5296/ije.v4i2.1962>
176. RAHNEMOONFAR, Maryam, et al. FloodNet: A High Resolution Aerial Imagery Dataset for Post Flood Scene Understanding. *IEEE Access*. 2020, 9, 89644-89654. doi 10.1109/ACCESS.2021.3090981. Prieiga per: <https://doi.org/10.1109/ACCESS.2021.3090981>
177. CHAO, Liu and WANG, Meng. Removal of Water Scattering. Iš: 2nd International Conference on Computer Engineering and Technology, April 16-18, 2010, Chengdu, China. IEEE, 2010. pp. 2-39. doi 10.1109/ICCET.2010.5485339. Prieiga per: <https://doi.org/10.1109/ICCET.2010.5485339>
178. LIN, Tsung-Yi, GOYAL, Priya, GIRSHICK, Ross B., HE, Kaiming, DOLLAR, Piotr. Focal Loss for Dense Object Detection. Iš: 2017 IEEE International Conference on Computer Vision (ICCV), October 22-29, 2017, Venice, Italy. IEEE, 2017. pp. 2999-3007.
179. RODRIGUEZ-CONDE, Ivan, CAMPOS, Celso, FDEZ-RIVEROLA, Florentino. Horizontally Distributed Inference of Deep Neural Networks for AI-Enabled IoT. *Sensors*. 2023, 23 (4) ISSN 1424-8220. doi 10.3390/s23041911. Prieiga per: <https://doi.org/10.3390/s23041911>
180. KORNEEV, Roman and DOERNING, Philip. How do the AWS Instances Compare to Commercially Available Graphic Cards? 2022 [žiūrėta 2022-04-29]. Prieiga per: <https://www.tecracer.com/blog/2022/02/how-do-the-aws-instances-compare-to-commercially-available-graphic-cards.html>
181. PAULAUSKAS, Algimantas. Ekologijos Terminų Aiškinamasis Žodynas - Lietuvių-Anglų-Vokiečių-Rusų Kalbomis. Vilnius: Viešoji įstaiga Grunto valdymo technologijos, 2008. ISBN 9786099504407.

Priedai

1 priedas. Ansamblio sudarymo schemas



Mean-ensemble-CNN ansamblio sudarymas [72]



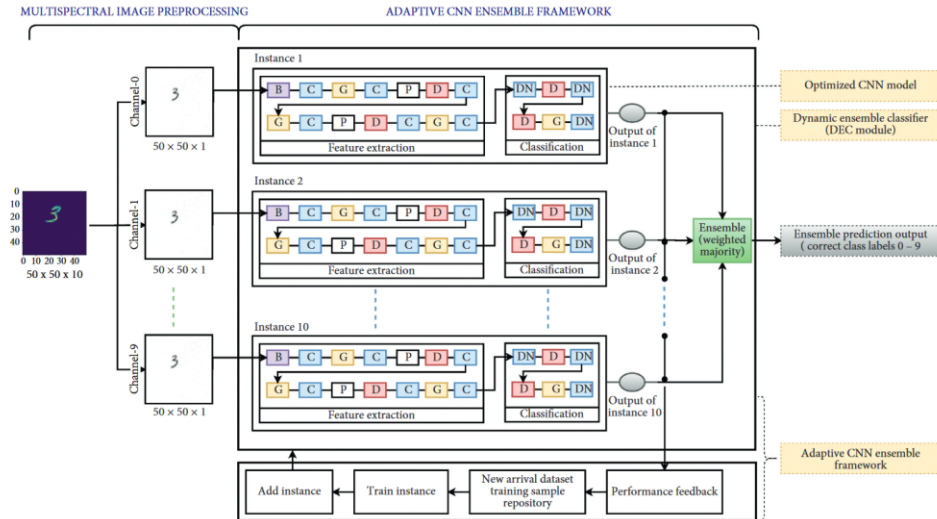
NN-ensemble-CNN ansamblio sudarymas [72]

2 priedas. Kather-CRC-2016 testavimo rezultatai

Architecture	Fold 1 (%)	Fold 2 (%)	Fold 3 (%)	Fold 4 (%)	Fold 5 (%)	Mean (%)
ResNet-101	95.80	94.30	96.10	97.50	95.90	95.92
ResNeXt-50	96.20	94.00	95.80	97.10	95.60	95.74
Inception-v3	94.80	92.00	93.80	95.80	93.50	93.98
DenseNet-161	95.90	94.10	95.90	96.90	95.20	95.60
Mean-Ensemble-CNNs	96.20	95.00	96.30	97.50	95.80	96.16
NN-Ensemble-CNNs	96.40	95.60	96.00	97.10	95.60	96.14

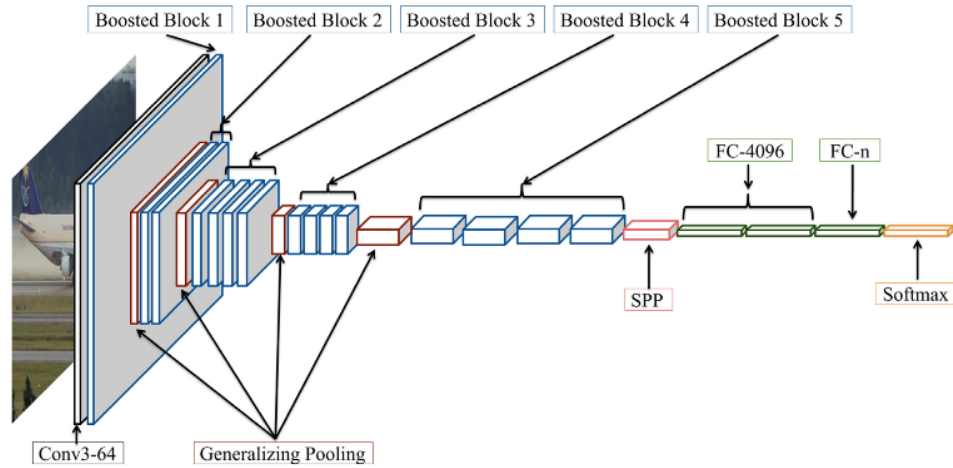
Mean-ensemble-CNN ir NN-ensemble-CNN testavimo rezultatai naudojantis Kather-CRC-2016 duomenimis [72]

3 priedas. Adaptīvais ansambļa struktūra



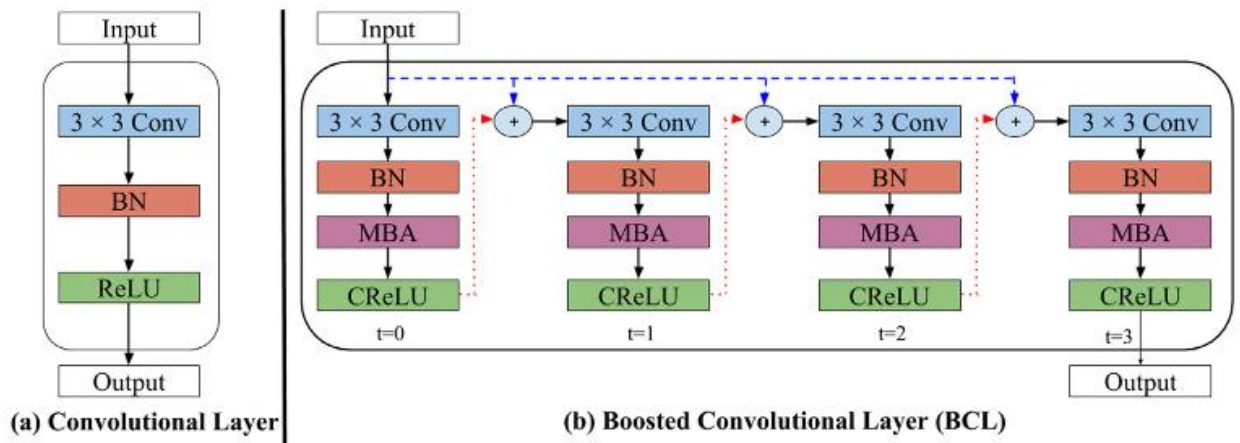
Adaptīvais ansambļa struktūra [73]

4 priedas. Boosted CNN arhitektūra



Boosted CNN arhitektūra [76]

5 priedas. BCL sandara



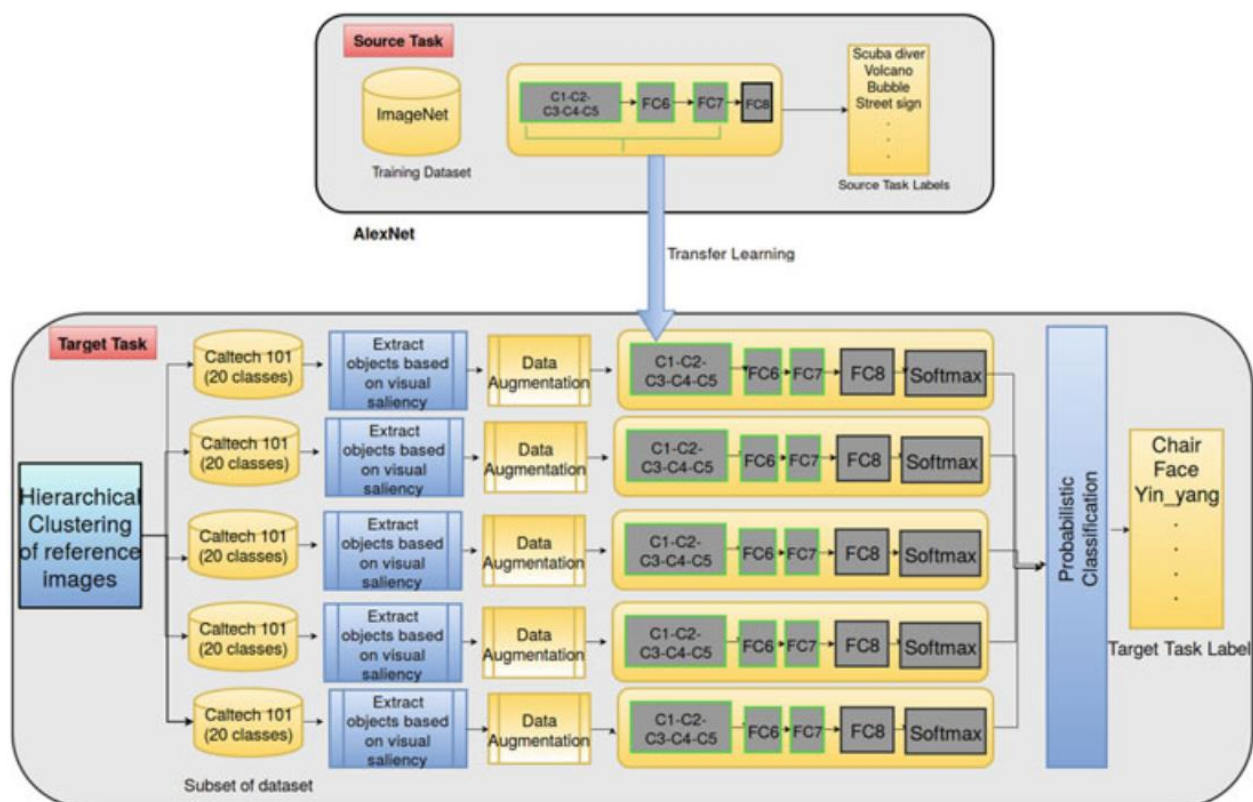
BCL sandara [151]

6 priedas. Modelių-specialistų testavimo rezultatai

# of specialists covering	# of test examples	delta in top1 correct	relative accuracy change
0	350037	0	0.0%
1	141993	+1421	+3.4%
2	67161	+1572	+7.4%
3	38801	+1124	+8.8%
4	26298	+835	+10.5%
5	16474	+561	+11.1%
6	10682	+362	+11.3%
7	7376	+232	+12.8%
8	4703	+182	+13.6%
9	4706	+208	+16.6%
10 or more	9082	+324	+14.1%

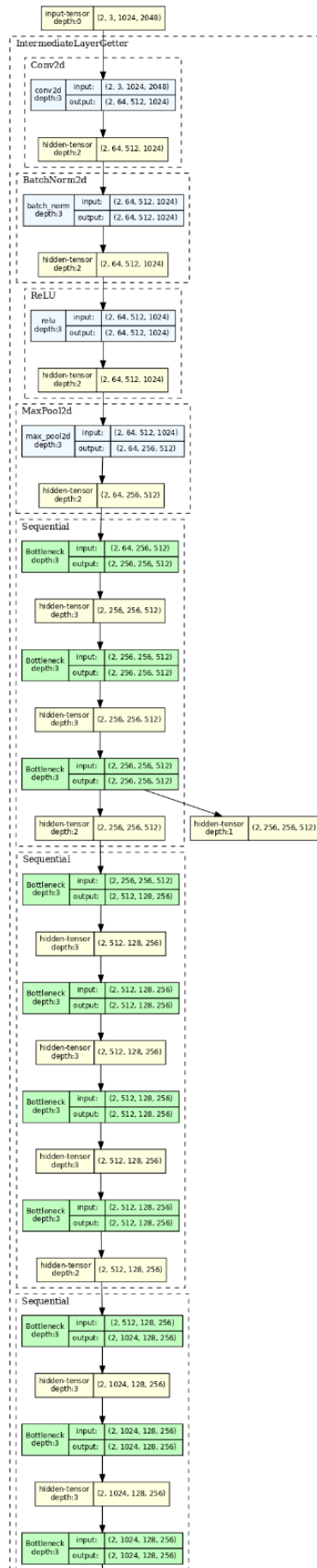
Geriausius rezultatus pasiekiančių modelių specialistų testavimo rezultatai [5]

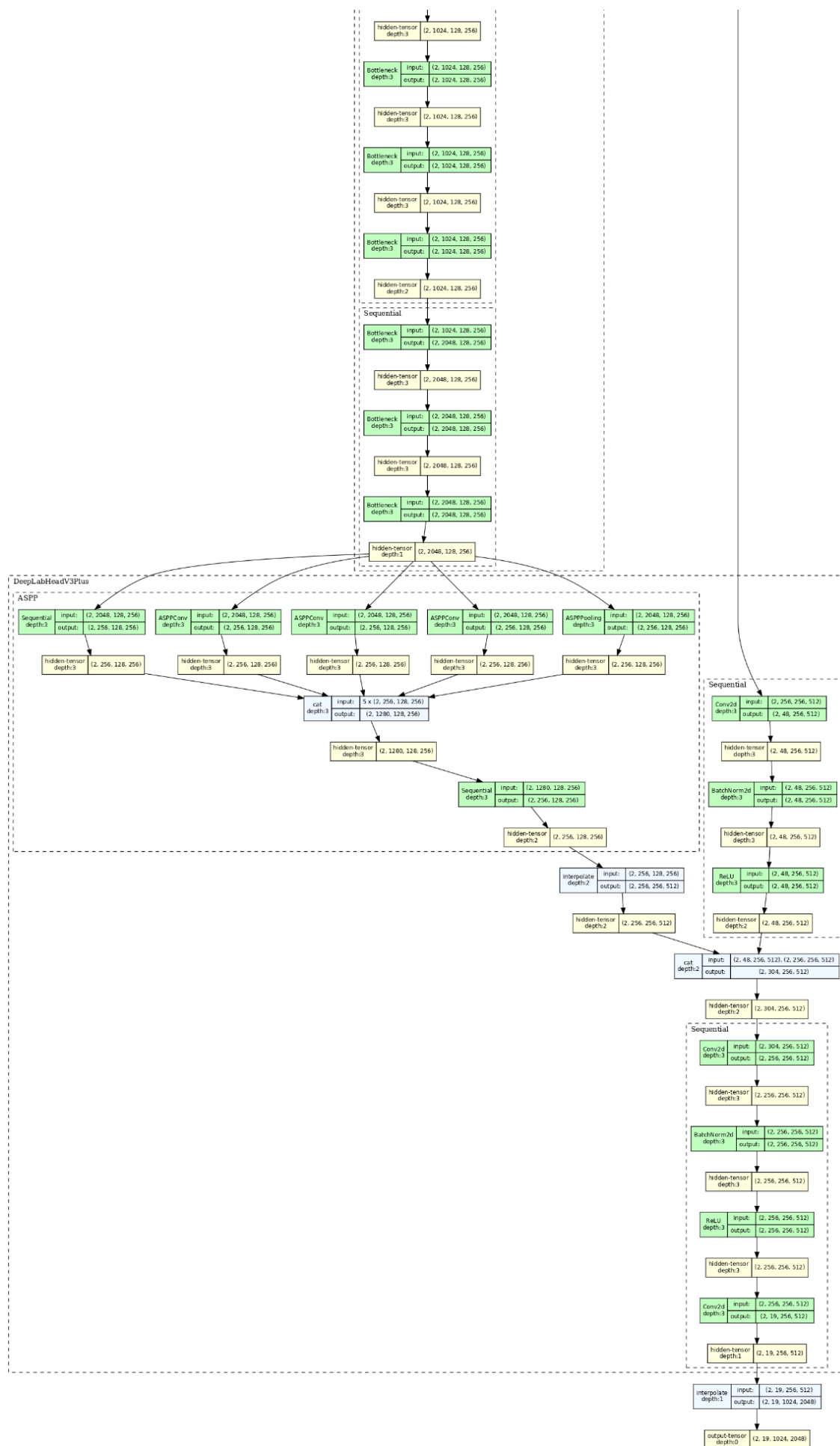
7 priedas. CNN ansamblio architektūra objektų atpažinimui



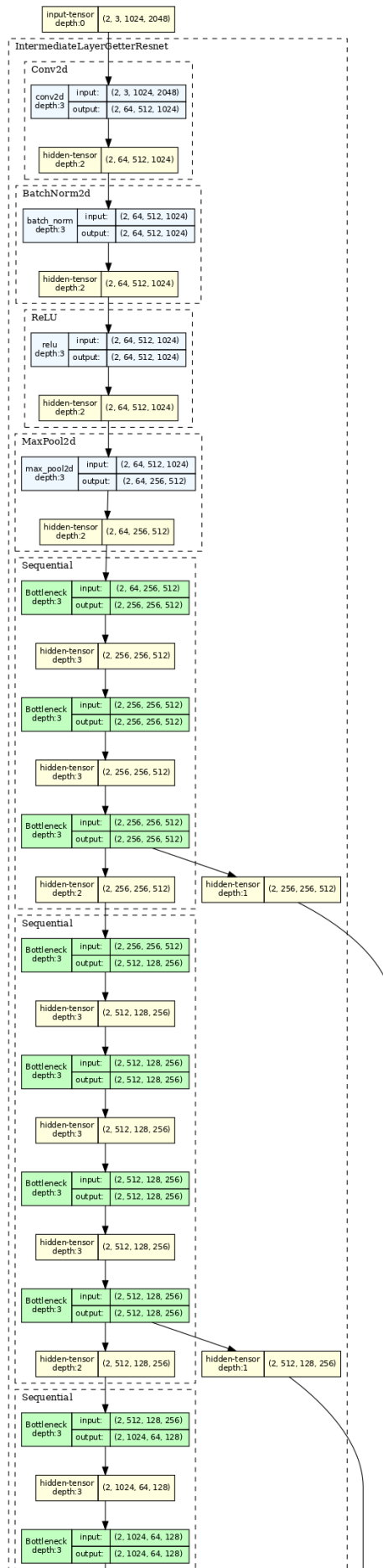
Ansamblio architektūra [134]

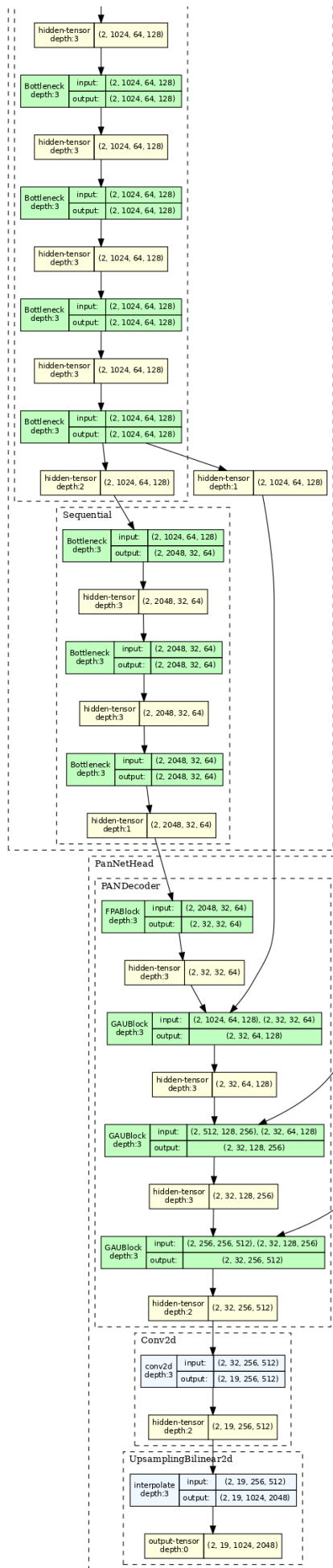
8 priedas. Deeplab modelio struktūra (Cityscapes segmentavimo pavyzdys)



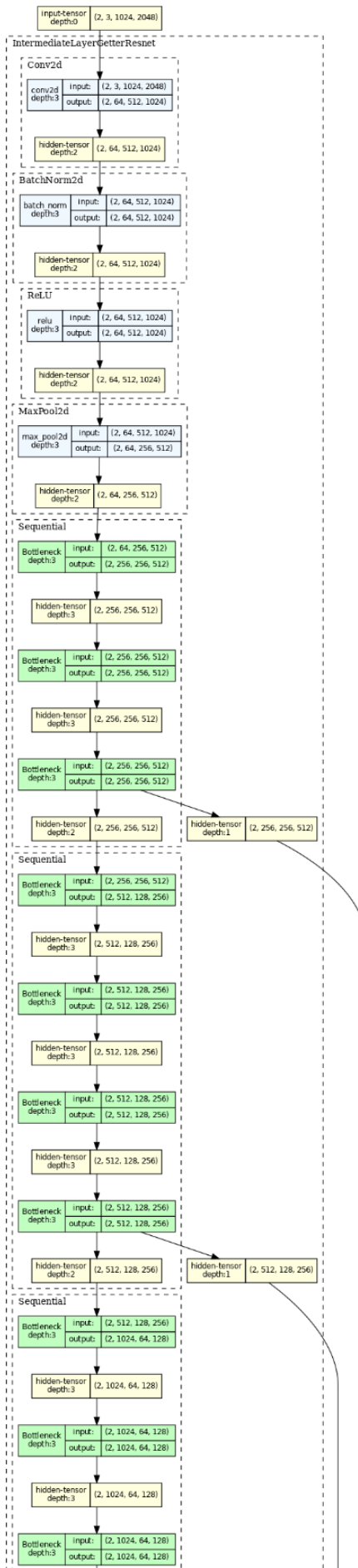


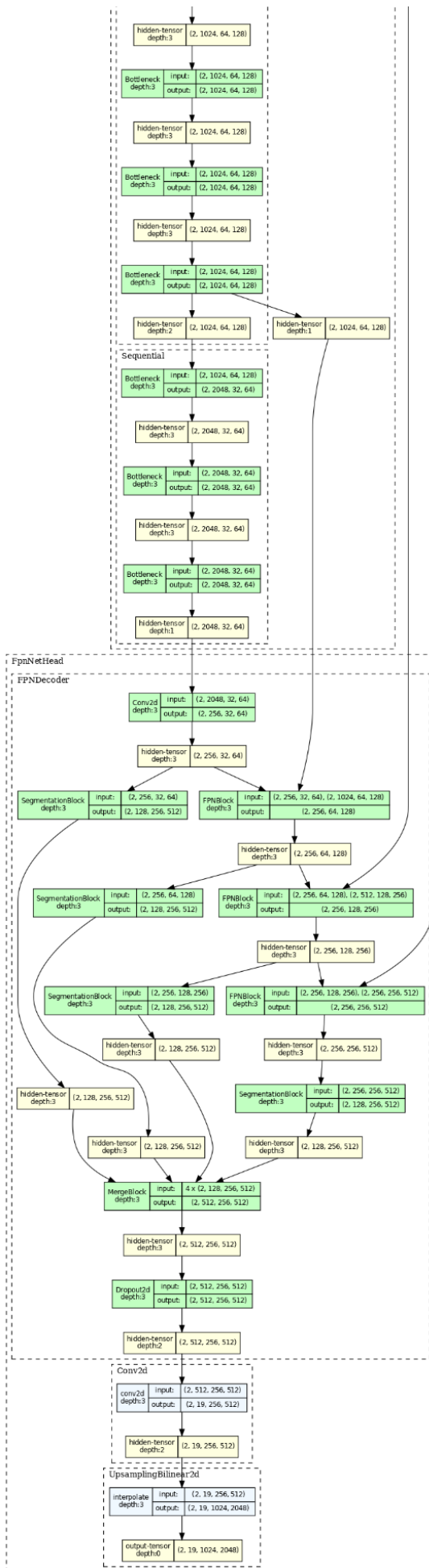
9 priedas. PAN modelio struktūra (Cityscapes segmentavimo pavyzdys)



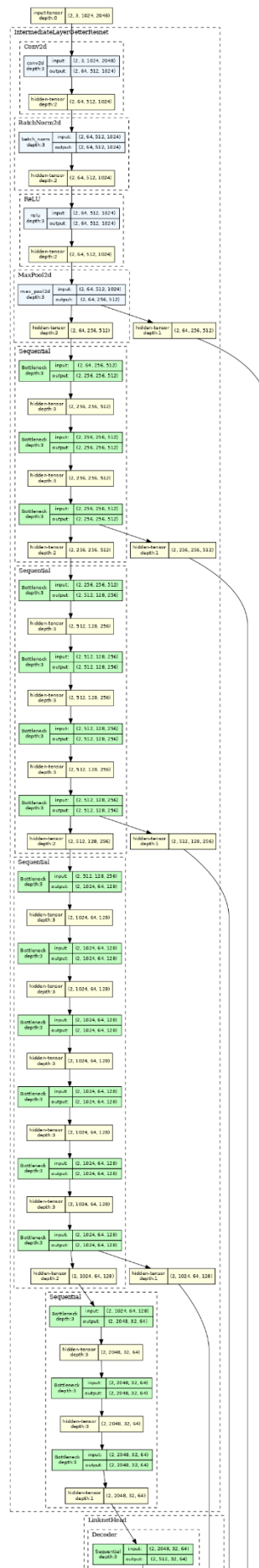


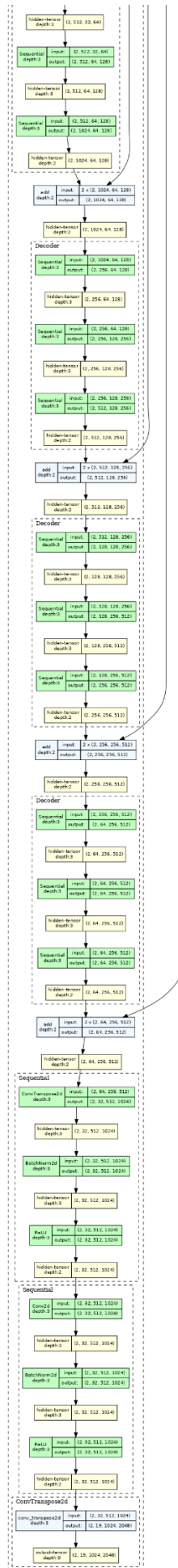
10 priedas. FPN modelio struktūra (Cityscapes segmentavimo pavyzdys)



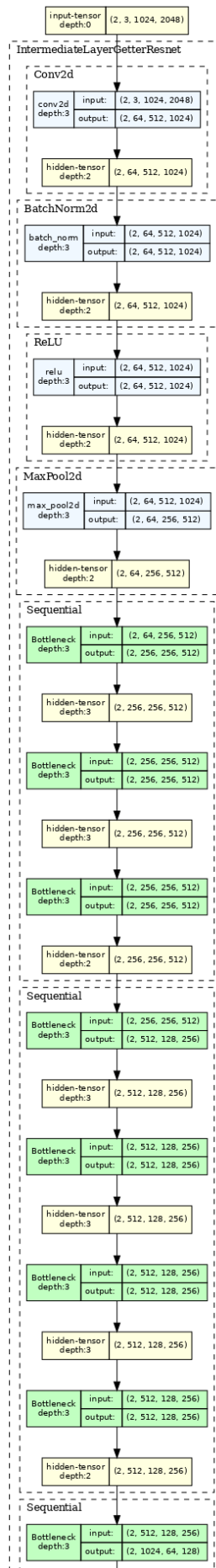


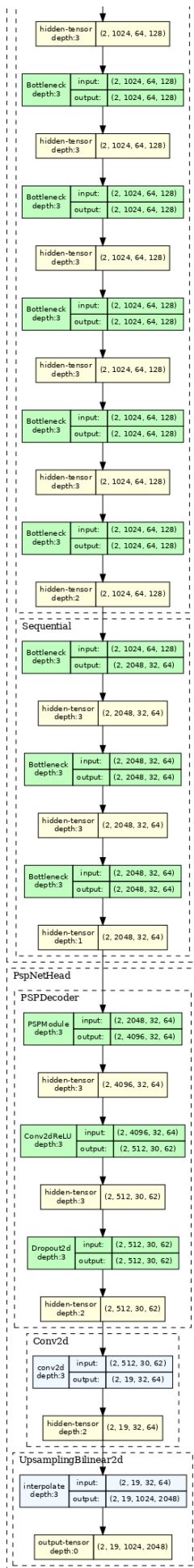
11 priedas. LinkNet modelio struktūra (Cityscapes segmentavimo pavyzdys)





12 priedas. PSPNet modelio struktūra (Cityscapes segmentavimo pavyzdys)





13 priedas. Standartinių duomenų rinkinių semantinės klasės

- a)*
- road
 - sidewalk
 - building
 - wall
 - fence
 - pole
 - traffic light
 - traffic sign
 - vegetation
 - terrain
 - sky
 - person
 - rider
 - car
 - truck
 - bus
 - train
 - motorcycle
 - bicycle
- b)*
- Building-flooded
 - Building-non-flooded
 - Road-flooded
 - Road-non-flooded
 - Water
 - Tree
 - Vehicle
 - Pool
 - Grass
- c)*
- sky
 - building
 - column pole
 - road
 - pavement
 - tree
 - signSymbol
 - fence
 - Car
 - Pedestrian
 - Bicyclist

Standartinių duomenų rinkinių Cityscapes (*a*), Floodnet (*b*) ir Camvid (*c*) semantinės klasės ir spalvinis kodavimas