

ŠIAULIŲ UNIVERSITETAS  
REGIONŲ PLĖTROS INSTITUTAS

Arnoldas Bialoglovis

ŽIEDADULKIŲ RŪŠIŲ ATPAŽINIMAS KOMPIUTERINE REGA

Magistro darbas

**Vadovas**  
prof. dr. G. Daunys

ŠIAULIAI, 2019

ŠIAULIŲ UNIVERSITETAS  
REGIONŲ PLĖTROS INSTITUTAS

TVIRTINU  
Inžinerinių studijų programų  
komiteto pirmininkas

\_\_\_\_\_  
(parašas)

doc. dr. D. Balbonas

2019 m. \_\_\_\_\_ mėn. \_\_\_\_ d.

Arnoldas Bialoglovis

ŽIEDADULKIŲ RŪŠIŲ ATPAŽINIMAS KOMPIUTERINE REGA

Magistro darbas

**Recenzentas**

doc. dr. D. Dervinis

2019

**Vadovas**

prof. dr. G. Daunys

2019

**Atliko**

RM-17 gr. stud.

A. Bialoglovis

2019

## SANTRAUKA

Bialoglovis A., Žiedadulkių rūšių atpažinimas kompiuterine rega: Išmaniosios gamybos inžinerijos magistro baigiamasis darbas / mokslinis vadovas prof. dr. G. Daunys; Šiaulių Universitetas, Regionų plėtros institutas, Šiauliai 2017. 69p.

Žiedadulkių rūšių atpažinimas yra sudėtingas uždavinys, kuris dažniausiai reikalauja operatoriaus. Visiškas šio uždavinio automatizavimas padėtų sutaupyti laiko ir resursų bei suteiktų galimybę tobulinti informacines sistemas susijusias su alergijomis. Šiuo metu žiedadulkėms identifikuoti naudojamos sistemos yra paremtos optiniais įtaisais, jos yra tikslios, pateikia daug informacijos, tačiau yra brangios, užima nemažai vietos ir reikalauja specialios įrangos gautiems duomenims apdoroti. Konvoliuciniai neuroniniai tinklai yra geriausias pasirinkimas šiai užduočiai kaip alternatyva žmogaus akių. Jų yra daug, kyla klausimas kuri iš jų ir kokiomis sąlygomis pateikia geriausius rezultatus žiedadulkių rūšių klasifikavime. Šiam tikslui pasiekti buvo apmokytos pasirinktos architektūros, kurios vėliau išbandytos su mokyme nenaudotais duomenimis ir stebimas kiekvienos situacijos rezultatas.

Darbo tikslas – ištirti ir nustatyti, kuri iš pasirinktų konvoliucinių neuroninių tinklų architektūrų ir kokiomis sąlygomis yra geriausia siekiant sukurti žiedadulkių klasifikatorių.

Šiam tyrimui buvo pasirinktos 18 skirtingų jau apmokytų ir lengvai prieinamų architektūrų modelių, kurios buvo pritaikytos atpažinti žiedadulkes remiantis mokėjimo perkėlimu. Galutiniai rezultatai parodė, kad MobileNet apmokytas su rinkiniu, kuriame yra po 1000 klasės pavyzdžių, geriausiai tinka žiedadulkių atpažinimo uždavinyje.

## SUMMARY

Bialoglovis A., Pollen Type Recognition by Computer Vision: Master thesis of Smart manufacturing engineering / research advisor prof. dr. G. Daunys; Šiauliai University, Institute of Regional Development, - Šiauliai, 2019. 69p.

Pollen type recognition is a difficult task, which requires operator. Complete automatization of this task would save time and resources also gives opportunity to improve systems related to allergies. Currently systems made for identification of pollen are based on optical devices, all those systems are very accurate, provides lots of information, but takes up a lot of space, requires special equipment for data analysis and are expensive. Convolutional neural networks are the best option as an alternative for human eyes. There are many architectures that are used, but which one of them and in what conditions presents the best results in classification of pollen. For the achievement of this purpose selected architectures have been trained, which were tested on the data that hasn't been related to the training and every case results were analyzed.

The aim of this study to investigate, which one of the selected convolutional neural network architectures and in what conditions is the best option to develop pollen classifier.

In this study eighteen different pretrained and easily available architecture models, which were trained based on transfer learning for pollen recognition. The results confirmed that MobileNet, which has been trained on dataset that had 1000 examples of every class, is the most suitable for pollen recognition task.

## TURINYS

IVADAS .....	9
2. Literatūros analizė .....	10
2.1. Žiedadulkių morfologija ir sandara.....	10
2.1.1. Žiedadulkės morfologija .....	10
2.1.2. Žiedadulkių sandara .....	10
2.2. Žiedadulkių segmentavimas .....	11
2.2.1. Regionu besiremiantis segmentavimas .....	11
2.2.1.1. Slenkstinis .....	11
2.2.1.2. Regioninio auginimo segmentavimas .....	12
2.2.2. Kraštų aptikimo segmentavimas .....	12
2.2.2.1. Sobel operatorius.....	12
2.2.2.2. Laplaso operatorius .....	13
2.2.3. Segmentavimas paremtas klasterizavimu.....	13
2.3. Kompiuterinė rega .....	13
2.4. Objektų aptikimas .....	14
2.5. Mašininis mokymas .....	14
2.5.1. Prižiūrimas .....	14
2.5.2. Neprižiūrimas .....	15
2.6. Mokėjimo perkėlimas .....	15
2.7. Vaizdų klasifikavimas.....	15
2.8. Konvoliuciniai neuroniniai tinklai .....	15
2.8.1. Konvoliucinio neuroninio tinklo sluoksniai.....	16
2.8.2. Hiperparametrai.....	16
2.8.2.1. Mokymosi koeficientas .....	16
2.8.2.2. Epochų skaičius.....	16
2.8.2.3. Paketo dydis .....	16
2.8.2.4. Aktyvinimo funkcija .....	17
2.8.2.5. Paslėptų sluoksnių skaičius .....	17
2.8.2.6. Sviurių inicializavimas .....	17
2.8.3. Konvoliucinių neuroninių tinklų architektūros .....	17
2.8.3.1. LeNet-5.....	17
2.8.3.2. AlexNet .....	20
2.8.3.3. VGG-16, VGG-19.....	22
2.8.3.4. GoogLeNet.....	24
2.8.3.5. ResNet .....	26
2.8.3.6. DenseNet .....	27
2.9. Tinklo persimokymas .....	27
2.10. Būdai išvengti persimokymo .....	28
2.11. Duomenų augmentacija .....	28
2.11.1. Vaizdų augmentacija.....	28
2.12. Duomenų rinkiniai: .....	29
2.12.1. Duomenų rinkinio sandara .....	29
2.12.2. Duomenų rinkinio paskirstymas.....	30
2.12.3. Duomenų rinkinio disbalansas .....	30
2.12.3.1. Duomenų rinkinio balansavimo būdai .....	30
3. Tyrimo metodika .....	31
3.1. Tyrimo eiga ir jos sudarymas.....	31
3.2. Duomenų rinkinio sugeneravimas .....	33
3.3. Pasirinktos konvoliucinių tinklų architektūros .....	34
3.4. Įranga naudota tyrimo metu .....	35
3.4.1. Techninė įranga .....	35

3.4.2. Programinė įranga .....	35
4. Tyrimo rezultatų analizė.....	36
4.1. Individualių konvoliucinių neuroninių tinklų rezultatų analizė.....	36
4.1.1. DenseNet.....	36
4.1.1.1. DenseNet121 .....	36
4.1.1.2. DenseNet169 .....	38
4.1.1.3. DenseNet201 .....	39
4.1.2. ResNet.....	41
4.1.2.1. ResNet50 .....	41
4.1.2.2. ResNet101 .....	43
4.1.2.3. ResNet152 .....	44
4.1.3. ResNetV2 .....	45
4.1.3.1. ResNet50V2 .....	46
4.1.3.2. ResNet101V2 .....	47
4.1.3.3. ResNet152V2 .....	49
4.1.4. ResNeXt .....	50
4.1.4.1. ResNeXt50 .....	50
4.1.4.2. ResNeXt101 .....	52
4.1.5. VGG .....	53
4.1.5.1. VGG16 .....	54
4.1.5.2. VGG19 .....	55
4.1.6. Inception.....	56
4.1.6.1. InceptionV3 .....	57
4.1.6.1. InceptionResNetV2 .....	58
4.1.7. MobileNet.....	60
4.1.7.1. MobileNet.....	60
4.1.7.2. MobileNetV2.....	62
4.1.8. Xception .....	63
4.2. Bendra konvoliucinių neuroninių tinklų rezultatų analizė.....	64
5. Išvados.....	67
<b>LITERATŪRA .....</b>	<b>68</b>

## PAVEIKSLĖLIAI

1 pav. Žiedadulkės sandara.....	11
2 pav. LeNet architektūra (GradientBased Learning Applied to Document Recognition, 1998) .....	18
3 pav. AlexNet architektūra (Krizhevsky, Sutskever, & Hinton, 2012).....	21
4 pav. VGG16-19 modelio architektūra (Rosebrock, 2017) .....	23
5 pav. GoogLeNet (InceptionV3) modelio architektūra (Google, 2019) .....	24
6 pav. Modelio apmokymo algoritmas .....	32
7 pav. Pradinio duomenų rinkinio pasiskirstymas.....	33
8 pav. DenseNet121, DenseNet169, DenseNet201 modelių palyginimas .....	36
9 pav. DenseNet121 modelių apmokymo priklausomybė nuo duomenų rinkinio dydžio .....	37
10 pav. DenseNet169 modelių apmokymo priklausomybė nuo duomenų rinkinio dydžio .....	38
11 pav. DenseNet201 modelių apmokymo priklausomybė nuo duomenų rinkinio dydžio .....	40
12 pav. ResNet101, ResNet152, ResNet50 modelių palyginimas .....	41
13 pav. ResNet50 modelių apmokymo priklausomybė nuo duomenų rinkinio dydžio .....	42
14 pav. ResNet101 modelių priklausomybė nuo duomenų rinkinio dydžio .....	43
15 pav. ResNet152 modelių priklausomybė nuo duomenų rinkinio dydžio .....	44
16 pav. ResNet101V2, ResNet152V2, ResNet50V2 modelių palyginimas.....	46
17 pav. ResNet50V2 modelių priklausomybė nuo duomenų rinkinio dydžio .....	46
18 pav. ResNet101V2 modelių priklausomybė nuo duomenų rinkinio dydžio .....	48
19 pav. ResNet152V2 modelių priklausomybė nuo duomenų rinkinio dydžio .....	49
20 pav. ResNeXt101 ir ResNeXt50 modelių palyginimas.....	50
21 pav. ResNeXt50 priklausomybė nuo duomenų rinkinio dydžio .....	51
22 pav. ResNeXt101 modelių priklausomybė nuo duomenų rinkinio dydžio .....	52
23 pav. VGG16 ir VGG19 modelių palyginimas.....	53
24 pav. VGG16 modelių priklausomybė nuo duomenų rinkinio dydžio .....	54
25 pav. VGG19 modelių priklausomybė nuo duomenų rinkinio dydžio .....	55
26 pav. InceptionResNetV2 ir InceptionV3 modelių palyginimas.....	57
27 pav. InceptionV3 modelių priklausomybė nuo duomenų rinkinio dydžio .....	57
28 pav. InceptionResNetV2 modelių priklausomybė nuo duomenų rinkinio dydžio .....	59
29 pav. MobileNetV2 ir MobileNet modelių palyginimas.....	60
30 pav. MobileNet modelių priklausomybė nuo duomenų rinkinio dydžio.....	61
31 pav. MobileNetV2 modelių priklausomybė nuo duomenų rinkinio dydžio.....	62
32 pav. Xception modelių priklausomybė nuo duomenų rinkinio dydžio .....	63
33 pav. Geriausių modelių palyginimas .....	65
34 pav. MobileNet maišos matrica.....	66
35 pav. ResNet152V2 maišos matrica.....	66
36 pav. Xception maišos matrica.....	66
37 pav. InceptionResNetV2 maišos matrica.....	66

## LENTELĖS

Lentelė 1 Kiekvienos klasės vaizdų kiekis pradiniam duomenų rinkinyje.....	33
Lentelė 2 Duomenų rinkinių pasiskirtymas.....	34
Lentelė 3 DenseNet121 modelių rezultatai .....	37
Lentelė 4 DenseNet169 modelių rezultatai .....	38
Lentelė 5 DenseNet201 modelių rezultatai .....	40
Lentelė 6 ResNet50 modelių rezultatai .....	42
Lentelė 7 ResNet101 modelių rezultatai .....	43
Lentelė 8 ResNet152 modelių rezultatai .....	45
Lentelė 9 ResNet50V2 modelių rezultatai.....	47
Lentelė 10 ResNet101V2 modelių rezultatai.....	48
Lentelė 11 ResNet152V2 modelių rezultatai.....	49
Lentelė 12 ResNeXt50 modelių rezultatai.....	51
Lentelė 13 ResNeXt101 modelių rezultatai.....	52
Lentelė 14 VGG16 modelių rezultatai .....	54
Lentelė 15 VGG19 modelių rezultatai .....	56
Lentelė 16 InceptionV3 modelių rezultatai .....	58
Lentelė 17 InceptionResNetV2 modelių rezultatai .....	59
Lentelė 18 MobileNet modelių rezultatai.....	61
Lentelė 19 MobileNetV2 modelių rezultatai.....	62
Lentelė 20 Xception modelių rezultatai.....	64
Lentelė 21 Visų tinklų, geriausiai apmokytų, modelių palyginimas.....	65



## IVADAS

Žiedadulkių rūšių atpažinimas yra sudėtingas uždavinys, kuris dažniausiai reikalauja operatoriaus. Visiškas šio uždavinio automatizavimas padėtų sutaupyti laiko ir resursų, bei suteiktų galimybę tobulinti informacines sistemas susijusias su alergijomis.

Šiuo metu žiedadulkėms identifikuoti naudojamos sistemos yra paremtos optiniais įtaisais, jos yra tikslios, pateikia daug informacijos, tačiau yra brangios, užima nemažai vietos ir reikalauja specialios įrangos gautiems duomenims apdoroti. Kadangi kompiuteriai, mobilieji telefonai ar mikroprocesoriai šiais laikais tobulėja labai sparčiai ir geba atlikti užduotis, kurias seniau ribodavo resursų, pajėgumų stygius, šių įrenginių pritaikymas žiedadulkių klasifikavimo uždavinyje leistų sumažinti įrangos kaštus, padėtų praplėsti pritaikymo mastą įvairiems poreikiams bei padaryti sistemas kilnojamąs. Pradedant sistemomis stebinčiomis žiedadulkių aktyvumą atokiose vietose, baigiant pilnai automatizuotomis apjungtomis informavimo sistemomis, kurios gebėtų pačios apdoroti informaciją ir ją perduoti į kitus tinklus ar įrenginius.

Konvoliuciniai neuroniniai tinklai yra geriausias pasirinkimas šiai užduočiai kaip alternatyva žmogaus akiai:

- Jie geba išmokti svarbius klasių požymius daugeliu atveju efektyviau įsisavina pateiktų duomenų savybes nei žmogus;
- Jau apmokytus tinklus galima pritaikyti naujiems uždaviniams ir išlaikyti gerą tinklo tikslumą;

Konvoliucinių neuroninių tinklų architektūrų yra daug, kyla klausimas kuri iš jų ir kokiomis sąlygomis pateikia geriausius rezultatus žiedadulkių rūšių klasifikavime. Šiam tikslui pasiekti buvo apmokytos pasirinktos architektūros, kurios vėliau išbandytos su mokyme nenaudotais duomenimis ir stebimas kiekvienos situacijos rezultatas.

**Darbo tikslas** – ištirti ir nustatyti, kuri iš pasirinktų konvoliucinių neuroninių tinklų architektūrų ir kokiomis sąlygomis yra geriausia siekiant sukurti žiedadulkių klasifikatorių.

### **Darbo uždaviniai:**

- Paruošti turimą duomenų rinkinį konvoliucinių tinklų apmokymui;
- Parengti konvoliucinių tinklų architektūras ir jas apmokyti;
- Palyginti gautus rezultatus skirtingomis sąlygomis ir pateikti išvadas, rekomendacijas;

## 2. Literatūros analizė

### 2.1. Žiedadulkių morfologija ir sandara

Žiedadulkės – mikroskopinės, mikrometrų eilės, augalinės dalelės naudojamos augalų apdulkinimui, pakeltos vėjo sklinda oru arba yra pernešamos vabzdžių. Plika akimi individualių žiedadulkių neįmanoma pamatyti, jos pastebimos kaip spalvotas dulkių debesėlis tik esant dideliame tankiui ore.

Žiedadulkių morfologija yra svarbi siekiant vizualizuoti bendrus jų požymius, tokius kaip: simetrija, forma, dydis, angų kiekį ir jų vietas, taip pat ornamentika. Morfologija yra būtinybė norint klasifikuoti žiedadulkes į rūšis (Halbritter, et al., 2018).

#### 2.1.1. Žiedadulkės morfologija

Identifikuojant žiedadulkės yra trys charakteristikos į kurias reikia atsižvelgti:

- Dydis – priklauso nuo jos hidratacijos ir paruošimo metodų, jos skirstomos į:
  - Labai mažos (<10 μm);
  - Mažos (10-25 μm);
  - Vidutinio dydžio (26-50 μm);
  - Didelės (51-100 μm);
  - Labai didelės (>100 μm);
- Forma – priklauso nuo žiedadulkės poliškumo ašies ir pusiaujo ašies santykio, pavyzdžiai gali būti:
  - Sferoidas;
  - Kubas;
  - Tetraedras;
  - Penkiakampė prizmė;
- Struktūra ir pačios žiedadulkės sandara;

#### 2.1.2. Žiedadulkių sandara

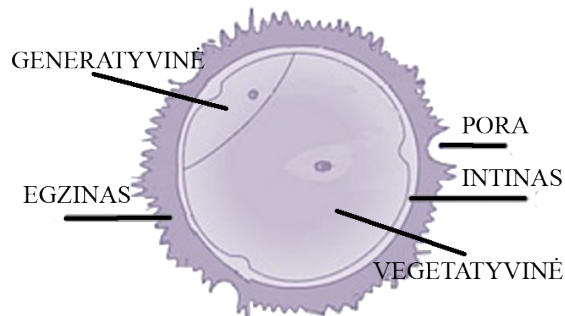
Žiedadulkės sandara, pavaizduota 1 paveiksle, ji susideda iš dviejų sluoksnių išorinės sienos, kurią sudaro:

- Egzinas – kietas išorinis sluoksnis, susidedantis iš sporopollenino, kuris yra vienas iš labiausiai atsparių žinomų organinių medžiagų.
- Gemalo poros – egzino tarpai, kuriose sporopollenino nėra.

- Intinas – vidinė plona ir vientisa žiedadulkės siena, susidedanti iš celiuliozės ir pektino, supanti žiedadulkės citoplazmą.

Subrendusios žiedadulkės viduje yra dvi ląstelės:

- Vegetatyvinė – didesnė ląstelė turinti gausias maisto atsargas ir didelį netaisyklingos formos branduolį.
- Generatyvinė – mažesnė suklio formos ląstelė su tankia citoplazma ir branduoliu, plaukiojanti vegetatyvinės ląstelės citoplazmoje.



1 pav. Žiedadulkės sandara

## 2.2. Žiedadulkių segmentavimas

Vaizdai savyje turi daug svarbios informacijos, informacijos ištraukimas iš vaizdo yra svarbi skaitmeninių vaizdų technologijų pritaikymo sritis. Praktikoje, ne viskas vaizde domina, todėl segmentavimas yra vienas iš būdų vaizdų apdorojime bei kompiuterinėje regoje, kuris yra naudojamas vaizdų analizavimui ir požymių išskyrimui ar atpažinimui. Segmentavimo pritaikymas yra platus, pradedant automatizuotose surinkimo linijose, baigiant medicinoje.

Žiedadulkių segmentavimas dažnai yra atliekamas rankiniu būdu, tai reikalauja laiko ir yra subjektyvus uždavinys. Yra įvairių segmentavimo technikų, tačiau nėra tokios, kuri būtų universali visoms įmanomoms situacijoms ar vaizdams, kadangi algoritmai naudoja specifinius vaizdų formavimo modelius.

### 2.2.1. Regionu besiremiantis segmentavimas

Regionu besiremiantis segmentavimas veikia grupuodamas kaimyninius pikselius, kurie turi panašias vertes, taip juos atskirdamas į pikselių grupes, kurių vertės yra heterogeninės.

#### 2.2.1.1. Slenkstinis

Paprasčiausias segmentavimo metodas, nors ir dažniausiai naudojamas. Tiesiogiai dalindamas pilkos spalvos informacijos apdorojimas besiremiantis skirtingų objektų pilkos spalvos verte. Slenkstinis segmentavimas gali būti dalinamas į vietinį, kuris pasirenka keletą slenksčių ir

sudalina vaizdą į keletą objektų regionų ir fonų remiantis bendrą atitikimų matricą, ir visuminį, kuris padalina vaizdą į du regionus: fono ir objekto, naudodamas vieną slenkstį, kuris gali būti nustatytas remiantis histograma arba pačiu vaizdu.

Dažniausiai naudojamas slenkstinio segmentavimo metodas – Otsu. Šis metodas pasirenka globaliai optimalų slenkstį maksimaliai padidinant dispersiją tarp klasių.

Slenkstinio segmentavimo metodo privalumas yra paprastas skaičiavimas ir didelis operacijų atlikimo greitis. Trūkumas – sunkus išskyrimas vaizde, kuriame nėra žymaus pilkos spalvos verčių skirtumų arba plataus persidengimo, taip pat jautrus triukšmams ir pilkos spalvos netolygumui. Dažnai šis metodas yra sujungiamas su kitais metodais siekiant kompensuoti jo trūkumus.

### **2.2.1.2. Regioninio auginimo segmentavimas**

Šis metodas remiasi idėja, kad aplinkiniai pikseliai turintys vienodas savybes suformuoja vientisą regioną. Pasirinkus norimą pikselį metodas apjungia pikselius esančius aplink pasirinktąjį ir panašius į jį.

Remiantis kriterijumi, kad absoliutinis pilkos vertės skirtumas tarp kaimyninio pikselio ir pasirinkto pikselio yra mažesnis už tam tikrą slenkstinę vertę, pikselis bus įtraukiamas į regioną, kuriame yra pasirinktasis pikselis, todėl pasirinkti tinkamą slenkstinę vertę yra labai svarbu.

Šio metodo privalumas yra metodo idėjos paprastumas, kadangi tereikia pasirinkti kelis pikselius, gebėjimas gerai atskirti sujungtus regionus su panašiomis charakteristikomis naudojantis vieną ar keletą kriterijų, pateikti gerą ribų informaciją ir segmentavimo rezultatus. Trūkumas tas, kad skaičiavimams reikia daug resursų, triukšmai ir pilkos spalvos nelygumai lemia tuštumas ir perteklius, taip pat šešėlių efektas vaizde dažnai yra negeras.

### **2.2.2. Kraštų aptikimo segmentavimas**

Kraštai visada yra tarp dviejų gretimų regionų, kurie nėra panašūs. Ne vientisumas gali būti aptinkamas skaičiuojant išvestines, kurios gali būti apskaičiuojamos skirtingais operatoriais, tokiais kaip Sobel ar Laplaso. Lygiagrečių kraštų aptikimas dažnai atliekamas erdvės diferenciniais operatoriais norint atlikti vaizdų segmentavimą atliekant konvoliuciją naudojant šabloną ir vaizdą.

#### **2.2.2.1. Sobel operatorius**

Dažniausiai naudojamas kraštų aptikimui ir yra išvestinių operatorius naudojamas skaičiuoti vaizdo šviesumo funkcijos gradiento aproksimaciją. Kraštų radimui naudoja pirmos eilės išvestines, tai padaro triukšmus glotnius, kad jie galėtų būti efektyviai panaikinti. Sobel operatorius susideda iš dviejų trys prie trijų matricų, kurios yra skersinis ir išilginis šablonai.

### 2.2.2.2. Laplaso operatorius

Laplaso operatorius yra izotropinis operatorius, remiasi antros eilės išvestinėmis, jis labiau tinka norint gauti krašto padėtį neatsižvelgiant į pilkos spalvos skirtumus aplink jį. Prieš panaudojant Laplaso operatorių pirma reikia pritaikyti žemo dažnio filtrą, kuris pašalintų triukšmus, kadangi Laplaso operatoriaus pritaikomas tik vaizdams be triukšmų, nes atsakas į izoliuotą pikselį yra didesnis nei į kraštą.

Šis operatorius yra paprasčiausias izotropinis diferencialinis operatorius su sukimosi invariancija. Dviejų dimensijų vaizdo Laplaso transformacija yra izotropinė antro laipsnio išvestinė, kuri yra labiau priderinama skaitmeninių vaizdų apdorojimui. (Yuheng & Hao, 2017)

### 2.2.3. Segmentavimas paremtas klasterizavimu

K-Means klasterizavimas yra vienas iš plačiausiai naudojamų algoritmų. Šio metodo esmė yra surinkti pavyzdžius į skirtingus klasterius pagal nuotolį, kuo du taškai arčiau vienas kito, tuo jie bus surišti taip formuojant klasterius.

K-Means pritaikymas išreikštas (Liu, Soran, Petrie, & Shapiro, 2019):

1. Atsitiktinai pasirenkami klasterių centrai;
2. Apskaičiuojami atstumai nuo kiekvieno klasterio centro ir mėginiai prijungiami prie arčiausiai jų esančio centro;
3. Apskaičiuojami nauji klasterių centrai;
4. Procesas kartojamas kol klasterio centras nebesikeičia arba pasiektas maksimalus iteracijų skaičius;

Šio metodo privalumas, kad algoritmas yra greitas, paprastas ir efektyvus, taip pat lankstus didelių duomenų rinkiniams. Trūkumai, kad šio algoritmo klasterių skaičius neturi aiškių pasirinkimo kriterijų, kiekviena iteracija peržvelgia visą duomenų masyvą – laiko gaištis ir netinkamas neišgaubtų duomenų klasterių klasterizavimui.

## 2.3. Kompiuterinė rega

Kompiuterinės regos tikslas yra apibūdinti pasaulį, kurį mes matome, viename ar daugiau vaizdų ir atkurti jo savybes, tokias kaip forma, apšvietimas, spalva, pasiskirstymas. Žmonės informaciją stebėdami aplinką suvokia trijų dimensijų struktūra. Iš trimatės struktūros gali pasakyti kokia objekto forma, jo skaidrumas, apšvietimas, šešėliavimas, be didelių vargų atskirti patį objektą nuo fono ar šalimais esančių kitų objektų. Taip pat gali lengvai suskaičiuoti objektus, juos identifikuoti, galbūt net nusakyti objekto būseną.

Nors kompiuterinė rega vis dar tobulėja, tačiau svajonė, kad kompiuteris interpretuotu vaizdus taip pat kaip žmogus vis dar nepasiekta. Dalinai todėl, kad rega yra atvirkštinė problema, kurioje siekiama atkurti keletą nežinomųjų pasitelkus nepakankamą kiekį informacijos pilnai nurodyti sprendinį (Szeliski, 2010). Taip pat kadangi vaizdinė informacija nėra pastovi ir gali bet kuriuo laiko momentu stipriai pasikeisti, todėl kyla sunkumų norint aprašyti jos apdorojimo metodiką bei sprendimus tam tikriems scenarijams.

## **2.4. Objektų aptikimas**

Kompiuterinės regos technika, kuri aptinka semantinius tam tikros klasės objektų atvejus nuotraukose ar vaizdo įrašuose. Plačiausiai naudojama veidų aptikime ir atpažinime, taip pat naudojama judančių objektų sekime, tokių kaip futbolo kamuolys rungtynių metu ar einantis gatvę žmogus.

## **2.5. Mašininis mokymas**

Mašininis mokymas paremtas taisyklėmis, kurias kompiuteris sukuria apdorojęs pateiktus duomenis ir atsakymus, apmokytas kompiuteris galės būti panaudotas naujų duomenų apdorojimui. Tai skiriasi nuo klasikinio programavimo, kuriame pagrindinis tikslas yra pateiktus taisykles ir duomenis gauti atsakymą.

Mašininis mokymas gali būti prižiūrimas ir neprižiūrimas. Pagrindinis skirtumas tarp dviejų mokymo tipų yra tas, kad prižiūrimas mokymas yra atliekamas su „ground truth“, iš anksto žinant koks turėtų būti tinklo spėjimas. Tai lemia, kad prižiūrimo mokymo tikslas yra išmokti funkciją, taisyklę pagal kurią iš duotų duomenų ir norimų išėjimų geriausiai apibrėžiama priklausomybė tarp įėjimo ir išėjimo pastebimą iš duomenų. Neprižiūrimas mokymas neturi sužymėtų išėjimų, todėl jis daro išvadą iš pateiktų duomenų natūralios struktūros.

Galimos problemos:

- Neteisingai parinkti apmokymo duomenų hiperparametrai;
- Prasta duomenų naudotų apmokymui, bandymui kokybė;
- Neteisingai pasirinktos duomenų paruošimo technikos;

### **2.5.1. Prižiūrimas**

Duomenys yra sužymėti ir iš jų yra sukuriamos taisyklės kaip prognozuoti žymes naujiems duomenims. Šio mokymo tikslas yra klasifikacija arba regresija.

### **2.5.2. Neprižiūrimas**

Duomenys nėra sužymėti, todėl šio mokymo pritaikymas yra grupavimui arba taisyklių atradimui.

### **2.6. Mokėjimo perkėlimas**

Mašininio mokėjimo technika kurios pagrindinis tikslas yra pritaikyti vienai užduočiai sukurtą modelį kaip kitos užduoties modelio pradinį tašką. Mokėjimo perkėlimas leidžia greitą apmokymą arba pagerintą našumą modeliuojant naują užduotį. Dažniausiai naudojami modeliai būna apmokyti su ImageNet duomenų rinkiniu, kurį sudaro 20000 įvairių kategorijų, kiekviena iš jų turi po keletą šimtų savo klasės pavyzdžių.

### **2.7. Vaizdų klasifikavimas**

Pagrindinė vaizdų klasifikavimo problema yra priskirti duotam vaizdai vieną etiketę iš fiksuoto klasių rinkinio, neatsižvelgiant į jos paprastumą turi aibę praktinio pritaikymo galimybių. (Johnson & Karpathy, 2015)

### **2.8. Konvoliuciniai neuroniniai tinklai**

Daugiasluoksniai neuroniniai tinklai sukurti apdoroti duomenis. Turintys bent vieną konvoliucinį sluoksnį. Dauguma surenkamų duomenų yra sudaryti iš kelių dimensijų eilučių:

- Vienos dimensijos eilutės signalams ir sekoms;
- Dviejų – paveikslams ir garso spektrogramoms;
- Trijų – video arba tūriniams paveikslėliams;

Plačiausiai naudojamas vaizdų atpažinimui ir identifikavimui, atpažinti vaizdinius šablonus tiesiai iš vaizdo su minimaliais išankstiniais apdorojimais.

Konvoliuciniai neuroniniai tinklai remiasi keturiomis idėjomis:

1. Vietinėmis jungtimis;
2. Bendrais svoriais;
3. Raiškos sumažinimu;
4. Daugybės sluoksnių naudojimu;

### 2.8.1. Konvoliucinio neuroninio tinklo sluoksniai

- Įėjimo sluoksnis – laiko neapdorotą vaizdo pikselių informaciją: aukšti, plotį ir spalvinių kanalų informaciją;
- Konvoliucijos sluoksnis – pagrindinis sluoksnis naudojamas konvoliucinio tinklo kūrimui, kuris atlieka daugiausiai skaičiavimų. Šio sluoksnio parametrai sudaro rinkinį mokomųjų filtrų, kurie užima nedaug erdvės, tačiau išsiplečia į visą įėjimo apimties gylį. Šis sluoksnis sukuria dviejų dimensijų aktyvavimo žemėlapi, kuris pateikia to filtro atsaką kiekvienoje erdvinėje padėtyje.
- Sujungimo sluoksnis – dažnai įterpiamas tarp konvoliucinių sluoksnių, naudojamas progresyviai sumažinti vaizdo erdvinį dydį taip sumažinant parametru ir skaičiavimų kiekį tinkle, taip kontroliuojant persimokymą;
- ReLU sluoksnis – pritaiko elementarią aktyvavimo funkciją, tokią kaip  $\max(0, x)$  pritaikant nulio ribinę vertę;
- Pilnai sujungtas sluoksnis – neuronai šiame sluoksnyje turi sujungimus su visomis prieš tai buvusio sluoksnio aktyvavimo funkcijomis.

### 2.8.2. Hiperparametrai

Pasirinkti tinkamus konvoliucinio tinklo hiperparametrus yra sudėtingas ir daug laiko reikalaujantis procesas, kadangi jų pasirinkimas nulemia galutinio modelio tikslumą, kai kuriais atvejais jis gali pasikeisti nuo kelių procentų iki keliasdešimt procentų.

#### 2.8.2.1. Mokymosi koeficientas

Mokymosi koeficientas – kontroliuoja po kiek reikia atnaujinti svorį optimizavimo algoritme. Jis gali būti fiksuotas, laipsniškai mažėjantis arba adaptyvus atsižvelgiant į pasirinktą optimizavimo metodą (SGD, Adam, Adagrad, AdaDelta ar RMSProp).

#### 2.8.2.2. Epochų skaičius

Epochų skaičius – skaičius iteracijų, kurių metu apmokymo duomenų rinkinys yra praleidžiamas per tinklą. Epochų skaičius yra didinamas tol kol pastebimas mažas pokytis tarp apmokymo ir testavimo paklaidų.

#### 2.8.2.3. Paketo dydis

Visų duomenų iškarto negalima pateikti tinklui, kadangi jie reikalauja daug resursų, maži paketai yra parankesni konvoliucinio tinklo mokymo procese lyginant su visu duomenų rinkiniu. Konvoliucinis tinklas yra jautrus paketų dydžiui, dažniausiai pasirenkamas dydis esantis tarp 16 ir 128.



#### **2.8.2.4. Aktyvinimo funkcija**

Ji įtraukia netiesiškumą į modelį, dažniausiai naudojamas ReLU, taip pat alternatyvos yra sigmoidė, hiperbolinio tangento funkcija ir kitos aktyvinimo funkcijos remiantis esama užduotimi.

#### **2.8.2.5. Paslėptų sluoksnių skaičius**

Dažnai sluoksniai pridedami tol kol testavimo paklaida nebe gerėja, tačiau tai prideda papildomus skaičiavimus apmokymo metu, tai reikalauja daugiau skaičiavimo resursų bei laiko.

#### **2.8.2.6. Sviurių inicializavimas**

Tinklo apmokymui inicializuojami maži atsitiktiniai skaičiai siekiant išvengti mirusių neuronų, bet ne per maži siekiant išvengti nulinio gradiento, dažnai naudojamas tolygiai pasiskirstęs skirstinys.

### **2.8.3. Konvoliucinių neuroninių tinklų architektūros**

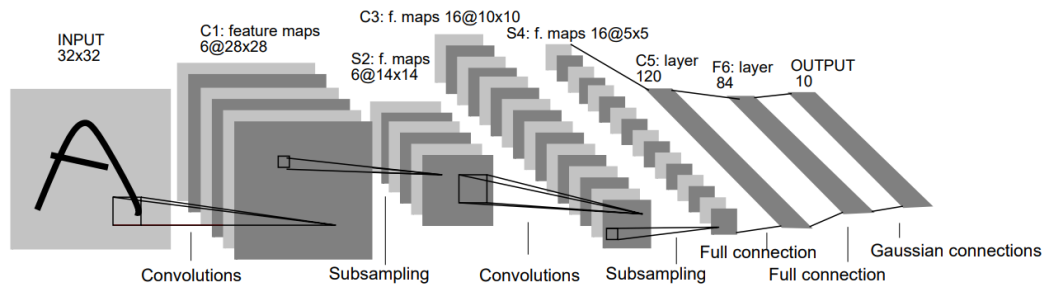
Konvoliuciniai tinklai yra gana nauja sritis, labiausiai žinomos architektūros:

1. LeNet-5 – 1998
2. AlexNet – 2012
3. VGG-16, VGG-19 – 2014
4. GoogLeNet – 2014
5. ResNet-50, ResNet-101, ResNet-152 – 2015
6. DenseNet – 2017
7. NASNet – 2018
8. SENet – 2018

#### **2.8.3.1. LeNet-5**

LeNet buvo pirmasis konvoliucinis neuroninis tinklas, sukurtas Y. LeCun, kuris pagyvino konvoliucinių neuroninių tinklų sritį. Tai yra konvoliucinių tinklų pradžia: konvoliuciniai sluoksniai naudoja prieš tai buvusio lygio sluoksnio kanalų pogrupį kiekvienam filtrui taip siekiant sumažinti skaičiavimų kiekį tinkle, pateikta 2 paveiksle. LeNet architektūra pagrinde buvo naudojama raidžių ir skaičių atpažinimo uždaviniuose kaip pašto kodų, skaičių ar kt. atskaitymui.

Po LeNet atsirado daug naujų architektūrų, kurios buvo tobulesnės, tačiau jos visos naudojo LeNet pagrindines idėjas. Galima sakyti, kad LeNet yra modernių konvoliucinių architektūrų ištakos, suprantant LeNet veikimą galima suprasti ir visas kitas architektūras.



2 pav. LeNet architektūra (*GradientBased Learning Applied to Document Recognition, 1998*)

Šį tinklą sudaro:

- 60000 mokomųjų parametrų;
- 340908 jungčių;
- Keturios pagrindinės operacijos:
  - Konvoliucija (konvoliucijos sluoksnis)
  - Netiesiškumas (aktyvinimo funkcija)
  - Sujungimas (sujungimo sluoksnis)
  - Klasifikavimas (pilnai sujungtas sluoksnis)

Modelyje panaudotos operacijos yra pagrindiniai blokai naudojami kiekviename konvoliuciniame tinkle. Todėl kiekvienos iš šių operacijų supratimas yra vienas iš svarbiausių žingsnių konvoliucinių tinklų kūrimo.

Konvoliucija – pagrindinė funkcija, jos tikslas yra išskirti požymius ir atpažinti erdvinius šablonus iš vaizdo išlaikant erdvinę priklausomybę tarp pikselių. Mokantis požymius naudojamas mažesnis nei vaizdas šablonas, dar kitaip vadinamas filtru, kuris yra stumiamas per patį vaizdą ir kiekviena pozicija yra apskaičiuojama atliekant daugybą kiekvieno elemento atžvilgiu ir sudedant dauginimo išėjimus. Taip gaunama vieną vertę, kuri sudaro išėjimo matricos, dar kitaip vadinamos požymių žemėlapiu, vienetą. Praktikoje filtro vertės nėra deklaruojamos, jos gaunamos besimokant pačiam tinklui, tačiau reikia deklaruoti:

- Filtro skaičių sluoksnyje;
- Filtro dydį;

Požymių žemėlapio dydis yra kontroliuojamas trijų parametrų:

- Gylio – atitinka filtrų, naudojamų konvoliucijos operacijos metu, skaičių. Šie filtrai sudarys skirtingus požymių žemėlapius, kuriuos galima įsivaizduoti kaip sujungtas dviejų dimensijų matricas.

- Žingsnio – skaičius pikselių kuriuo yra paslenkamas filtras. Kuo didesnis žingsnis tuo mažesni požymių žemėlapiai sudaromi, kadangi filtras persöks daugiau pikselių.
- Nulių pridėjimo – pridedant nulius aplink patį vaizdą galima valdyti požymių žemėlapio dydį.

Netiesiškumas arba aktyvinimo funkcija – operacija pritaikoma kiekvienam pikseliui, jei jis neigiamos vertės tai prilyginamos nuliui. Tai daroma siekiant pritaikyti konvoliucinį tinklą netiesiškiems duomenims, kadangi pati konvoliucijos operacija yra tiesinė. LeNet naudojo hiperbolinį tangentą visuose sluoksniuose išskyrus klasifikavimo sluoksnyje, ten buvo naudojama sigmoidinė funkcija. Lygintuvo funkcija (ReLU) nebuvo naudojama LeNet laikais, tačiau dabar ReLU yra viena geriausių aktyvinimo funkcijų, kuri paspartina konvergenciją mokymo metu (Agarap, 2019).

Sujungimas – sumažina kiekvieno požymių žemėlapio dydį, tačiau išlaiko svarbiausią informaciją, atliekamas atskirai kiekvienam požymių žemėlapiui siekiant progresyviai sumažinti erdvinį įėjimo dydį. Praktikoje plačiausiai naudojamas maksimalus sujungimas, tačiau gali būti skirtingų tipų: maksimalus, vidutinis, suminis ir kt. (Zeiler & Fergus, 2013):

- Maksimalaus sujungimo – iš deklaruoto dydžio lango išrenkama didžiausia vertė;
- Vidutinis sujungimas – deklaruoto dydžio lange gaunamas bendras visų elementų vidurkis, kuris yra naudojamas toliau;
- Suminis sujungimas – deklaruoto dydžio lange visos vertės yra susumuojamos ir tas skaičius yra naudojamas toliau;

Iš esmės sujungimo sluoksnis sumažina parametrų ir atliekamų skaičiavimų kiekį pačiame tinkle, taip suvaldant tinklo persimokymą, taip pat padaro tinklą invariantišką mažoms vaizdų transformacijoms, iškraipymams ar perkėlimams.

Klasifikavimas (pilnai sujungtas sluoksnis) – konvoliucijos ir sujungimo išėjimai yra aukšto lygio vaizdo požymiai, klasifikavimo tikslas yra panaudoti tuos požymius siekiant priskirti apmokytą klasę į tinklą paduotam vaizdui, tam dažniausiai naudoja minkšto maksimumo funkciją, kuri gautas vertes paskirsto [0;1] ribose. Pilnai sujungtas sluoksnis gali būti naudojamas ne vien klasifikavimui, tai yra gan efektyvus būdas siekiant išmokti šių požymių netiesinį ryšį.

Pačio tinklo apmokymas gali būti apibendrintas taip:

- Inicializuojami atsitiktinių verčių filtrai, parametrai bei svoriai;
- Panaudojant apmokymui skirtus vaizdus tinklas apmokomas srautu į priekį, gaunamos išėjimo tikimybės kiekvienai klasei;

- Apskaičiuojamos bendras klaidų kiekis išėjimo sluoksnyje;
- Atgalinio sklidimo metu apskaičiuojami klaidų kiekių gradientai visų svorių atžvilgiu ir remiantis gradientinio nusileidimo metodu atnaujinamos visų filtrų vertės, svoriai ir parametrai siekiant sumažinti išėjimo klaidų kiekį. Svoriai koreguojami proporcingai jų įtakai bendram klaidų kiekiui, parametrai tokie kaip: filtrų skaičius, filtrų dydis, tinklo architektūra, deklaruoti pačioje pradžioje nekinta, kinta tik filtrų matricių vertės ir atnaujinami jungčių svoriai;

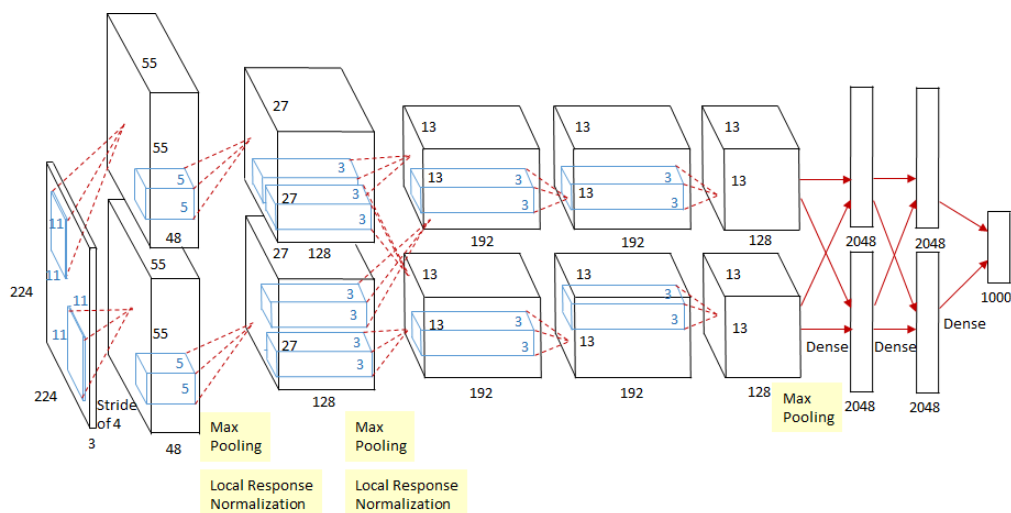
Pateikus tuos pačius atvaizdus bendras klaidų kiekis išėjimo sluoksnyje turėtų būti sumažėjęs, tai reiškia, kad tinklas išmoko klasifikuoti šiuos vaizdus pritaikęs savo svorius bei filtrų vertes. Šis procesas yra kartojamas kol yra panaudojami visi apmokymui skirti duomenys.

Pateikus dar nematytus vaizdus jau apmokytam tinklui jis pateiktų tikimybes kiekvienai klasei. Turint nemažą apmokymo duomenų rinkinį tinklas turėtų gerai apibendrinti naujus vaizdus ir juos klasifikuoti į teisingas kategorijas.

LeNet-5 apmokytas su MNIST-10 duomenų rinkiniu pasiekia 0,95% klaidų kiekį testavimo duomenims ir 0,35% klaidų kiekį mokymo duomenims (GradientBased Learning Applied to Document Recognition, 1998). Šiam tinklui dažnas persimokymas, kai mokymo klaidų kiekis krenta, bet testavimo klaidų kiekis pasiekęs savo minimumą pradeda didėti, tai gali lemti per mažas mokymo koeficientas. Palyginus LeNet tikslumus su skirtingų dydžių duomenų rinkiniais, didėjant duomenų rinkinio dydžiui gerėja tinklo apmokymo tikslumas.

### **2.8.3.2. AlexNet**

Toronto Universitete A. Krizhevsky sukurta architektūra, pateikta 3 paveiksle, laikoma pirmuoju moderniu konvoluciniu tinklu bei giliuoju neuroniniu tinklu. Laimėjusi ILSVRC-2012 konkursą išpopuliarino konvoliucinius tinklus. Modelio architektūra labai panaši į LeNet, tačiau autorius norėjo parodyti, kad yra įmanoma sėkmingai apmokyti didelės apimties konvoliucinį tinklą su dideliu kiekiu parametrų naudojant ImageNet duomenų rinkinį ir pasiekti gerą tikslumą.



3 pav. AlexNet architektūra (Krizhevsky, Sutskever, & Hinton, 2012)

Šį tinklą sudaro:

- 60 milijonų parametru;
- 650000 neuronų;
- 8 sluoksniai su svoriais:
  - 5 konvoliuciniai sluoksniai;
  - 3 pilnai sujungti sluoksniai;

Kiekvienam konvoliuciniam ir pilnai sujungtam sluoksniui yra taikoma netiesinė ReLU aktyvinimo funkcija, tai paspartina didelio tinklo su daug parametru mokymą keletą kartų lyginant su hiperbolinio tangento funkcija.

Po pirmo ir antro konvoliucinio sluoksnio, buvo panaudota vietinio atsako normalizacija, kuri autoriaus nuomone yra šiek tiek panaši į vietinio kontrasto normalizavimą. Bendra mintis yra ta, kad abi normalizacijos siekia įgyvendinti neuronų išėjimų konkurencingumą, skirtumas tik tas, kad vietinio atsako normalizacija atima vidurkį ir konkuruoja tik to paties sluoksnio gretutiniai neuronų išėjimai (Krizhevsky, Sutskever, & Hinton, 2012).

Maksimalaus sujungimo sluoksnis pritaikomas po abiejų vietinio atsako normalizacijos sluoksnių, taip pat kaip ir po paskutinio konvoliucinio sluoksnio.

Pilnai sujungtas išėjimo sluoksnis, naudojantis minkšto maksimumo aktyvinimo funkcija, pateikia skirstinį su 1000 klasių etiketėmis.

Šioje architektūroje panaudotos naujovės, kurių nebuvo LeNet-5:

- ReLU netiesiškumas – standartiškai buvo naudojami hiperbolinio tangento ir sigmoidės netiesiškumai. Esminė jų problema buvo nykstantys gradientai įsisotiniančiuose

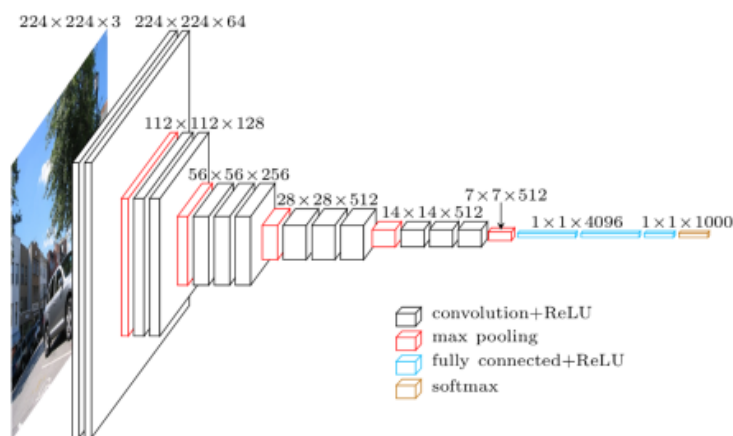
regionuose, kuriuose gradientų vertės tapdavo labai mažos, kas lemdavo gradientinio nusileidimo sulėtėjimą, o tai prailgindavo tinklo apmokymą. ReLU netiesiškumas padėjo ištaisyti gradientų problemą, nors gradientai tapdavo nuliais vietoj neigiamų verčių, tačiau tai neturėjo įtakos ir lėmė žymiai spartesnę tinklo mokymo procesą;

- Vietinio atsako normalizacija – šis sluoksnis padėjo pagerinti tinklo gebėjimą apibendrinti turimus duomenis, pagerina tikslumą ir sumažina klaidų skaičių. Šis sluoksnis nebėra naudojamas kitose architektūrose, kadangi jis turi minimalų poveikį ir dabar egzistuoja geresnės alternatyvos, tokios kaip batch normalizacija arba retinimas;
- Persidengiantys sujungimo sluoksniai – tipiška sujungimo langai nepersidengdavo, tačiau leidžiant sujungimo langams persidengti sumažino klaidų kiekį ir sumažino tikimybę modeliui persimokyti;
- Retinimas – tai buvo pirmasis retinimo sluoksnio pritaikymas tinkle, jis padeda tinklui nepersimokyti. Nustačius paslėptų neuronų išėjimą, su 50% tikimybe, į nulį, išmesti neuronai neprideda prie sklaidos į priekį ir atgal. Kiekvieną kartą į įėjimą pateikus duomenis tinklas atsitiktinai pasirenka skirtingą architektūrą, tačiau jos visos turi bendrus svorius. Tai lemia, kad kiekvienas neuronas yra verčiamas mažiau kliautis kitais neuronais ir pačiam įsisavinti tvirtus klasių požymius, kurie yra naudingi kartu su daugybe įvairių atsitiktinių kitų neuronų;
- Duomenų augmentacijos – padėjo išvengti persimokymo dirbtinai padidinus apmokymo duomenų rinkinį, tai sumažino klaidų kiekį 1%;
- Keleto GPU panaudojimas – paspartinę mokymą lyginant su vienos GPU panaudojimu padalinant krūvi keletui branduolių ir dalinantis bendra atmintimi;

Modelis buvo apmokomas naudojant stochastinį gradientinį nusileidimą su 128 batch dydžiu, mokymo koeficientas deklaruotas 0,01 ir sumažinamas dešimtimi, kai tikrinimo klaidos kiekis stabilizuodavosi. Apmokytas modelis su ILSVRC-2012 duomenų rinkiniu pasiekė 15,3% top-5 klaidų kiekį, kai antroje vietoje buvęs tinklo klaidų kiekis siekė 26,2%. Geriausi šio tinklo tikslumai yra 78,1% ir 60,9% (Krizhevsky, Sutskever, & Hinton, 2012).

### **2.8.3.3. VGG-16, VGG-19**

ILSVRC-2014 antros vietos architektūra, sukurta K. Simonyan ir A. Zisserman, pateikta 4 paveiksle. Pagrindinis šių architektūrų tikslas buvo parodyti, kad gilūs tinklai pagerina tikslumą ir kad keletas mažų filtrai yra geriau lyginant su vienu dideliu filtru. Skaičiai 16 ir 19 nurodo sluoksnių su svoriais skaičių pačiame tinkle.



4 pav. VGG16-19 modelio architektūra (Rosebrock, 2017)

Šiuos modelius sudaro:

- 138 ir 144 milijonai mokomųjų parametrų;
- VGG16 16 sluoksnių:
  - 13 konvoliucinių sluoksnių;
  - 3 pilnai sujungti sluoksniai;
- VGG19 19 sluoksnių:
  - 16 konvoliucinių sluoksnių;
  - 3 pilnai sujungti sluoksniai;

Visų konvoliucinių sluoksnių filtrų dydžiai 3x3, filtrų žingsnis yra 1 pikselis, taip pat visi įėjimai į konvoliucinius sluoksnius yra papildyti nuliais taip, kad po konvoliucijos žingsnio išėjimo dimensijos išliktų tokios pat kaip įėjimo, skiriasi tik filtrų kiekis sluoksnyje:

- VGG16 – pirmi du turi po 64 filtrus, trečias ir ketvirtas turi po 128, nuo penkto iki septinto po 256, nuo aštunto iki trylikto po 512;
- VGG19 – pirmi du turi po 64 filtrus, trečias ir ketvirtas turi po 128, nuo penkto iki aštunto po 256, nuo devinto iki šešiolikto po 512;

Naudojant keletą mažų filtrų nėra prarandama informacija, kadangi apjungus keletą mažų filtrų, jų jautrus plotas gaunamas didesnis nei jų pradinis. Taip pat tai suteikia keletą pranašumų:

- Sprendimo funkcija būna labiau diskriminuojanti, tai skatina sudėtingesnių požymių išmokimą, tai lemia geresnį vaizdų ir etikečių siejimą.
- Sumažėja parametrų naudojant rietuvę konvoliucinių sluoksnių;

- Sumažina modeliui reikalingos atminties kiekį ir atlieka tinklo reguliavimą, taip sumažinant tikimybę tinklu persimokyti;

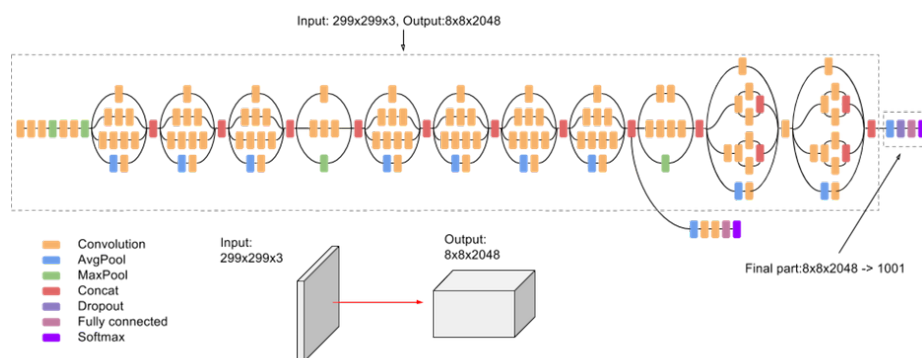
Mokymas paremtas mini-batch gradientiniu nusileidimu su 256 batch dydžiu ir tęsiasi 74 epochas, tai trunka gana ilgai. Retinimas yra pritaikomas tik pirmiems dviems pilnai sujungtiems sluoksniuais, su išretinimo tikimybe nustatyta 50%. Pradinis mokymo koeficientas 0,01 ir sumažinamas dešimtimi kiekvieną kartą kai tikrinimo tikslumas stabilizuojasi.

Modelio testavimui vaizdai būna su izotropiškai pakeista skale. Sumažintas paveikslėlis paduodamas tinklui, netikrinant ar jo dydis atitinka įėjimo dydį, kadangi konvoliuciniai sluoksniai yra invariantiški įėjimo dydžiui, todėl jų veikimas bus įprastas, tik skirsis požymių žemėlapių dydis. Problema atsiranda pilnai sujungtuose sluoksniuose, kadangi jų įėjimui reikia numatyto dydžio vektoriaus, tačiau tai ištaisoma pilnai sujungtą sluoksnį pakeičiant į konvoliucinį. Taip suteikiama daugiau informacijos apie testavimo vaizdus ir pritaiko tinklą prie didesnių vaizdų, taip padarant modelį labiau praktinį (Simonyan & Zisserman, 2015)

Apmokytas tinklas su ILSVRC-2014 siekia 92,7% top-5 tikslumo, 23,7% tikrinimo klaidų kiekis, bei 6,8% top-5 tikrinimo klaidų kiekis, taip pat 6,8% top-5 testavimo klaidų kiekis (Rosebrock, 2017).

### 2.8.3.4. GoogLeNet

ILSVRC-2014 Laimėtojai, pagerino konvoliucinius modelius naudodami dar didesnes architektūras. Tikslas su šiuo modeliu buvo pasiekti didelį tikslumą išlaikant mažą skaičiavimų kiekį ir turėtų praktinį pritaikymą. Yra keletas šios architektūros tęstinių versijų, paskiausia Inception-V5.



5 pav. GoogLeNet (InceptionV3) modelio architektūra (Google, 2019)

Modelio architektūra pateikta 5 paveiksle, ją sudaro:

- 22 sluoksniai, kuriuos sudaro:
  - Konvoliuciniai sluoksniai;
  - Inception modulių;



- Maksimalaus sujungimo sluoksniai;
- Priešpaskutinis pilnai sujungtas sluoksnis pakeičiamas vidurkio sujungimo sluoksniu;

Inception modulių sukūrimas dramatiškai sumažino parametrų skaičių tinkle, kadangi tinklas pasirinkdavo optimaliausią konvoliucinių sluoksnių išdėstymą. Tai pasiekama atliekant kiekvieną konvoliuciją paraleliai su tuo pačiu įėjimu, su tiek pat vaizdą papildančių nulių, kad išėjimas būtų tokio pat dydžio kaip ir įėjimas, ir susiejant visus požymių žemėlapius į vieną didelį, kuris yra paduodamas kitam inception moduliui. Šiame tinkle Inception moduliai yra ribojami su filtrais, kurių dydžiai (Szegedy, et al., 2014):

- 1x1
- 3x3
- 5x5

Maži filtrai padeda atpažinti vietines detales ir požymius, kai tuo metu dideli filtrai aptinka abstraktesnis arba labiau išsimėčiusius požymius. Taip pat 3x3 dydžio maksimalus sujungimas yra pritaikomas kiekvienam inception moduliui, kadangi jis paspartina tinklo veikimą.

Šio modulio trūkumas yra tas, kad jis reikalauja daug skaičiavimų, pvz.: 5x5 filtrui reikia 120 milijonų operacijų. Siekiant sumažinti skaičiavimų skaičių įtraukiamas dydžio sumažinimas atliekamas su 1x1 filtru prieš naudojant didesnius filtrus, tai atlikus skaičiavimų kiekis sumažėja iki 12 milijonų operacijų, taip pat prideda papildomą netiesiškumą.

Pakeistas vidurkio sujungimo sluoksnis sumažina parametrų skaičių ir skaičiavimų kiekį, beveik 49 kartus, taip pat pagerina tinklo tikslumą 0,6%.

GoogLeNet apmokymui pridama papildoma struktūra, kuri yra pašalinama testavimo metu, tai yra pagalbiniai klasifikatoriai prijungti prie vidurinio sluoksnio, kuris turi dvi paskirtis: padaro vidurinę sluoksnį labiau diskriminuojantį, kas lemia geresni požymių išskyrimą, ir susumuoja klaidų kiekį iš kiekvieno klasifikatoriaus, taip spartinant mokymą, nes gradientai iš pagrindinio klasifikatoriaus pasiekę pradinis sluoksnius būtų labai maži, tačiau pagalbinio klasifikatoriaus dėka pradinių sluoksnių gradientai tampa pakankamai dideli, kad būtų galima tęsti mokymą.

Šią struktūrą sudaro:

- Vidurkio sujungimo sluoksnis;
- Konvoliucinis sluoksnis su 128 1x1 dydžio filtrais;
- Retinimo sluoksnis su 70% tikimybe;
- Klasifikatorius su minkšto maksimumo funkcija ir tiek pat išėjimų kiek ir pagrindinis;

Pats modelis apmokomas asinchroniniu stochastiniu gradientiniu nusileidimu, kurio metu mokymo koeficientas yra mažinamas po 4% po 8 epochų, neatsižvelgiant ar tikslumas stabilizavosi ar ne, kaip dažniausiai yra daroma.

Inception-V3 architektūra su ImageNet apmokyto modelio tikslumas didesnis nei 78,1%, klaidų kiekis 4,2% ir 18,77% top-5 (Szegedy, Vanhoucke, Ioffe, & Shlens, 2015).

### 2.8.3.5. ResNet

Modelis laimėjęs ILSVRC-2015. Pagrindinis šio modelio tikslas buvo išsiaiškinti, kas lėmė gilių tinklų tikslumo nuostolį lyginant su jų negiliosiomis alternatyvomis, kadangi didinant tinklo gylį tikslumas neaugo, tačiau atvirkščiai – krito, tai buvo pastebima ne vien su tikrinimo ar testavimo, bet ir su mokymo. Giliųjų tinklų problema buvo optimizavimas, jie negalėjo susieti jėgimo vaizdo su klasės etiketėmis.

Tai padėjo ištaisyti liekamasis mokymas, kurio metu ResNet išmoksta liekamąsias jungtis. Liekamieji blokai yra šio tinklo pagrindas, kurių sudaro:

- Sluoksnių rietuvė;
- Praleidimo jungties;

Praleidimo jungčių pagrindinis tikslas yra praleisti keletą sluoksnių ir taip padaryti tinklą dinamiškesnį. Jos neprideda papildomų parametrų. Liekamasis mokymas yra pritaikomas šių sluoksnių rietuvei.

Realų modelį sudaro:

- 152 sluoksniai;
- 25 milijonai mokomųjų parametrų;

ResNet modelis yra tiesiog liekamieji blokai pakartoti keletą kartų, po kiekvieno iš jų yra panaudota batch normalizacija, kuri buvo prieš aktyvinimo funkciją. Konvoliuciniai sluoksniai esantys tinkle nepaveikia vaizdo dydžio, tačiau maksimalaus sujungimo sluoksnis juos dalina pusiau po kiekvieno konvoliucinio sluoksnio. Paskutinio konvoliucinio sluoksnio išėjimo požymių žemėlapis yra paduodamas per vidurkio sujungimo sluoksnį į pilnai sujungtą sluoksnį, kuris susideda iš tiek neuronų, kiek yra klasių.

Gilesni ResNet tinklai dar turi papildomą butelio kaklelio sluoksnį, panašius kokius naudojo GoogLeNet: pirma pritaikomas konvoliucinis filtras, kurio dydis 1x1, tada filtras, kurio dydis 3x3 ir pabaigoje pritaikomas dar vienas filtras, kurio dydis 1x1, taip siekiama padidinti arba sumažinti kanalų skaičių.

Tinklas mokomas stochastinių gradientiniu nusileidimu su 256 dydžio batch, mokymo koeficientas nustatytas 0,1 ir sumažinamas dešimtimi kai tikrinimo klaidos kiekis stabilizuojasi. Apmokyto tinklo klaidų kiekis siekia tik 3,57% top-5 ImageNet testavimo duomenų rinkiniu, ResNet50 klaidų kiekis siekia 22,85% ir 6,71% top-5, ResNet101 klaidų kiekis – 21,75% ir 6,05% top-5, ResNet152 klaidų kiekis – 21,43% ir 5,71% top-5 (He, Zhang, Ren, & Sun, 2015).

#### **2.8.3.6. DenseNet**

Šio modelio pagrindinis tikslas buvo išspręsti konvoliucinio tinklo gylio problemą, kadangi didinant konvoliucinių tinklų gylį informacija apie įėjimą arba gradientą pereinant per daug sluoksnių dingdavo.

Siekiant tai išspręsti buvo pritaikytas jungčių šablonas, kuris remiasi maksimalia informacijos tėkme tarp sluoksnių abiem kryptim. Šis tinklas sujungia kiekvieną sluoksnį su kiekvienu kitu sluoksniu esančiu prieš jį, taip kiekvienas sluoksnis gauna papildomus įėjimus iš prieš tai buvusių sluoksnių ir perduoda savo požymių žemėlapią į visus prieš jį esančius sluoksnius.

Tinklas susideda iš sluoksnių, kurie pritaiko netiesines transformacijas, kurios gali būti sudedamos su batch normalizacijos, ReLU, sujungimo ar konvoliucijos operacijomis. Vaizdams sumažinti visa tinklo architektūra yra padalinta į keletą tankiai sujungtų sluoksnių, tarp kurių yra pereinamieji sluoksniai, kurie atlieka konvoliuciją arba sujungimą.

Šio tinklo naudoti patobulinimai:

- Prieš tai buvusio sluoksnių požymiai į kitus sluoksnius perduodami apjungti, tai padidina vėlesnių įėjimų variaciją ir pagerina efektyvumą;
- Nykstančių gradientų problemą sprendžiama naudodami tankias jungtis;
- Skatinamas požymių daugkartinis panaudojimas;
- Pritaikyti 1x1 konvoliuciniai butelio kaklelio sluoksniai prieš kiekvieną 3x3 konvoliucinį sluoksnį sumažina įėjimų požymių žemėlapių dydį taip padidinant skaičiavimų efektyvumą;

Apmokytas modelis su CIFAR10 duomenų rinkiniu turi 3,46% klaidų kiekį ir 17,18% klaidų kiekis apmokius su CIFAR100 (Huang, Liu, & Maaten, 2018).

### **2.9. Tinklo persimokymas**

Modelio nemokėjimas apibendrinti duomenų iš apmokymo duomenų rinkinio į dar nematytus duomenis, tai dažna problema mašiniam mokymui.

Tinklo neapmokymas pasireiškia kai modelis yra per paprastas – apmokytas su per mažai požymių arba perreguluotas, tai padaro modelį nelankstų mokymui iš esamų duomenų.

Paprasti tinklai potencialiai linkę turėti mažas dispersijas jų spėjimuose, tačiau didelius nuokrypius į neteisingus rezultatus, kai sudėtingi tinklai linkę turėti didesnes dispersijas savo spėjimuose.

Mašininiam mokyme nustatyti tinklo persimokymą yra ganėtinai sudėtinga kol tinklas nėra išbandomas su tinklui dar nematytais duomenimis. Jei modelis turi daug didesnę tikslumą apmokymo duomenų rinkiniui nei testavimo duomenų rinkiniui, galima teigti, kad modelis persimokė.

## **2.10. Būdai išvengti persimokymo**

Norint išvengti persimokymo yra keletas būdų:

- Mokyti su daugiau duomenų – įtraukus daugiau duomenų, pasitelkus duomenų augmentaciją, algoritmas gali aptikti požymius geriau, tačiau informacija su triukšmais ne visada padeda;
- Ankstesnis apmokymo stabdymas – kiekviena iteracija tobulina modelį iki tam tikro iteracijų skaičiaus po kurio modelio gebėjimas apibendrinti duomenis silpsta ir tinkas pradeda persimokyti. Ankstesnis stabdymas nutraukia tinklo apmokymą prieš jam peržengiant tą ribą, dažniausiai naudojamas;
- Reguliavimas – įvairios technikos kurios dirbtinai priverčia modelį būti paprastesniam, šis metodas priklausomas nuo tinklo tipo. Dažnai reguliavimas yra hiperparametras, kurio reikšmė gali būti koreguojama kryžminių patvirtinimų;

## **2.11. Duomenų augmentacija**

Duomenų augmentacija yra vienas iš būdų kaip galima sumažinti modelio permokymą, kai padidinamas tik apmokomų duomenų kiekis. Ši metodika nėra nauja ir turi įvairių taikomųjų technikų, jos pagrindinis tikslas yra pridėti vertes prie bazinių duomenų gautų iš vidinių ir išorinių šaltinių. Duomenų augmentacija gali būti pritaikyta visokių formų duomenims

### **2.11.1. Vaizdų augmentacija**

- Perkėlimas – visi objekto taškai yra perkeliama tiesia linija ta pačia kryptimi, dydis, forma ir orientacija yra tokia pat kaip ir prieš tai.
- Atspindys – vienas iš plačiausiai naudojamų metodų dėl savo paprastumo, visi objekto taškai yra apverčiami pagal vieną arba dvi ašis, dydis ir forma nekinta, kinta tik orientacija.

- Pasukimas – visi objekto taškai yra pasukami aplink objekto centrą, dydis ir forma nekinta, kinta tik orientacija.
- Euklidinė transformacija – dažniausiai naudojama transformacija, kurios metu naudojamas perkėlimas, pasukimas arba atspindys, jos metu išlaikomi euklidiniai atstumai tarp visų taškų porų.
- Mastelio keitimas – visi objekto taškai yra sutraukiami arčiau prie objekto centro, forma ir orientacija nepakinta, pasikeičia dydis.
- Triukšmo įtraukimas – į vaizdą pridedami pikseliai atsitiktinėse vietose, objekto padėtis, dydis, forma ir orientacija nekinta.

## 2.12. Duomenų rinkiniai:

Didžiausia problema konvoliuciniams neuroniniams tinklams nėra patys tinklai, tačiau duomenų rinkinio sukūrimas, kuriame būtų tinkami duomenys tinkamame formate (SkyMind.ai, 2019). Gauti tinkamus duomenis reiškia surinkti arba identifikuoti duomenis kurie koreliuotų su rezultatais, kuriuos reikia nuspėti. Duomenys turi būti susieti su problema kurią reikia išspręsti, neturint tinkamų duomenų nepavyks apmokyti tinklo tinkamai.

Tinkamas formatas tinklo mokymui bendrai yra tenzorius arba kelių dimensijų masyvas. Duomenų srautas sukurtas konvoliuciniam tinklui konvertuos visus duomenis į vektorius arba tenzorius, kuriems tiesinės algebros veiksmai gali būti pritaikomi. Duomenys dažnai turi būti normalizuoti, standartizuoti ir išvalyti norint padidinti duomenų naudingumą.

Mašininiam mokymui reikalingas geras apmokymo duomenų rinkinys, kurio surinkimui ir sudarymui reikalingas laikas bei nemažas kiekis žinomų duomenų. Apmokymo duomenys prilygsta lyginamajai analizei pagal kurią tinklai yra apmokomi.

### 2.12.1. Duomenų rinkinio sandara

- Apmokymo duomenų rinkinys – didžiausia dalis viso duomenų rinkinio, kurio pagrindinis tikslas yra išmokyti tinklą pasverti skirtingus požymius, sureguliuoti jų koeficientus pagal jų tikimybę sumažinti rezultatų paklaidas;
- Bandymo duomenų rinkinys – naudojamas pateikti nešališką modelio prisitaikymo prie apmokymo duomenų įvertinimą kartu pakoreguojant jo hiperparamterus, tinklas mato šiuos duomenis, tačiau iš jų nesimoko;
- Testavimo duomenų rinkinys – naudojamas kai tinklas būna optimizuotas ir apmokytas pateikia nešališką galutinio modelio prisitaikymo prie apmokymo duomenų įvertinimą, gauti rezultatai parodo kokiu procentu tikslumu tinklas atpažįsta objektus iš dar nematytų duomenų;

### **2.12.2. Duomenų rinkinio paskirstymas**

Priklauso nuo turimo rinkinio apimties ir modelio, kurį reikės apmokyti. Vieniems modeliams reikia didelės apimties duomenų rinkinio apmokymui, šiuo atveju reikėtų optimizuoti didelės apimties apmokymo duomenų rinkinį. Modeliams turintiems mažai hiperparametrų bandymai ir reguliavimai bus lengvesni, todėl bandymo duomenų rinkinys gali būti sumažintas, jei turi daug hiperparametrų – reikalingas didelis. Modeliams neturintiems hiperparametrų bandymo duomenų rinkinys nėra būtinas.

### **2.12.3. Duomenų rinkinio disbalansas**

Dažnai pasitaikanti problema, kai apmokymui naudojami duomenys turi nevienodus kiekius klasės pavyzdžių. Tai gali nulemti tinklo optimizavimą nereikšmingiems požymiams pasirinktai užduočiai. Galima pasiekti aukštą tikslumą, tačiau pats modelis nebus tinkamas praktiniam panaudojimui, kadangi jis rinksis klasės, kurios yra daugiau duomenų rinkinyje, spėjimą. Klasių disbalansas yra žalingas veiksnys klasifikavimo našumui (Buda, Maki, & Mazurowski, 2018).

#### **2.12.3.1. Duomenų rinkinio balansavimo būdai**

- Klasės pavyzdžių pridėjimas – veikia tiesiogiai su duomenimis, atsitiktinai pasirinktos mažumos klasės pavyzdžių dubliavimas ar augmentavimas. Labiau pažengę metodai naudoja dirbtinių pavyzdžių, sukurtų interpoliuojant kaimyninius duomenis, augmentaciją (Chawla, Bowyer, Hall, & Kegelmeyer, 2002). Šios technikos privalumas yra tas, kad užtikrinamas vienodas kiekvienos klasės pasiskirstymas ir pasirinktų pavyzdžių kontroliavimas, tačiau įtraukus dublikatų galimas tinklo persimokymas;
- Klasės pavyzdžių atėmimas – veikia tiesiogiai su duomenimis, remiasi atsitiktinės daugumos klasės pavyzdžių pašalinimu kol visos klasės turi vienodą kiekį pavyzdžių. Privalumas yra tinklo persimokymo išvengimas, tačiau prarandami duomenys.
- Dviejų etapų mokymas – modelis pirma yra apmokomas su vienodu kiekiu klasės pavyzdžių, po to paskutinis sluoksnis prieš softmax yra patikslinamas su originaliu nesubalansuotu duomenų rinkiniu;
- Ribinės vertės įvedimas – koreguoja klasifikatoriaus sprendimo slenkstį testavimo metu ir remiasi išėjimo klasės tikimybės keitimu, atsižvelgiant į pačios klasės dažnumą duomenų rinkinyje;

### 3. Tyrimo metodika

Konvoliucinių tinklų srityje nėra specialių metodų, kuriais reikėtų remtis siekiant išsiaiškinti tinklo gebėjimą prisitaikyti prie tam tikrų uždavinių. Įvairūs tinklai išskiria skirtingus požymius ir skirtingai mokosi, todėl kiekvienas tinklas yra apmokomas su tam tikru duomenų rinkiniu esant tam tikroms nustatytoms sąlygoms. Svarbiausia tinklo apmokyme yra tai, kad reikia atsižvelgti koks yra naudojamo duomenų rinkinio dydis, kadangi jis lemia tinklo gebėjimą įsisavinti klasių požymius tai sudaro tinklo apmokymo pagrindą.

Analizuojant priežastis, kodėl konvoliuciniams neuroniniams tinklams duomenų rinkinio dydis yra svarbus faktorius, galima pastebėti, kad kintant duomenų rinkinio dydžiui kinta ir tinklo gebėjimas geriau įsisavinti klasių požymius. Taigi yra aišku, kad tyrimo metu tinklai apmokyti su mažais duomenų rinkiniais turės mažesnius tikslumus, nei tinklai apmokyti su dideliais duomenų rinkiniais.

Šio darbo metu buvo panaudotas mokėjimo perkėlimo metodas tinklų apmokymui, naudojantis dvidešimt skirtingų konvoliucinių tinklų architektūrų apmokytų „ImageNet“ duomenų rinkiniui.

#### 3.1. Tyrimo eiga ir jos sudarymas

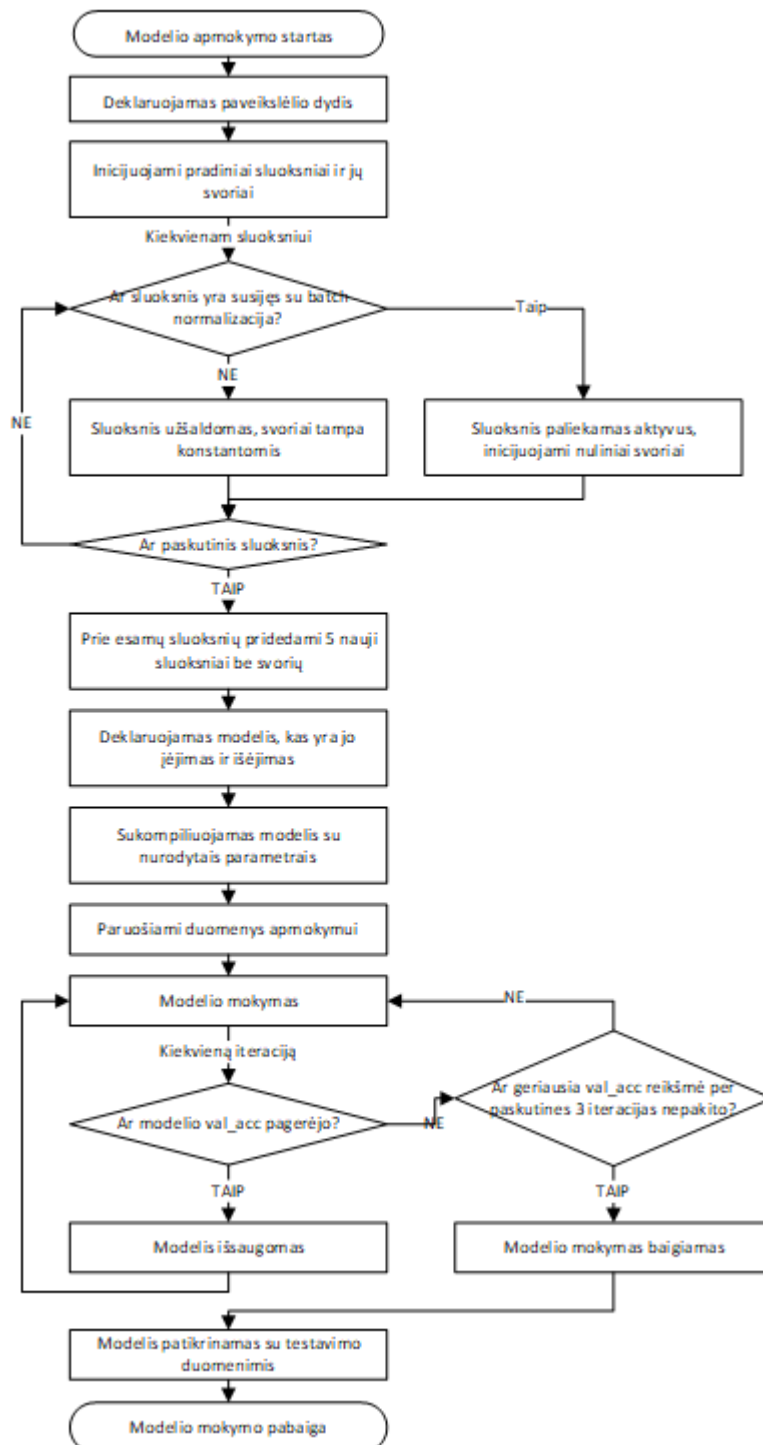
Kiekvienas tinklas apmokomas pagal algoritmą pateiktą 6 paveiksle. Įkeliamas jau apmokytas modelis, be paskutinių, spėjimą lemiančių, sluoksnių. Paskui seka sluoksnių užšaldymas, kuris nebeleidžia sluoksniams mokytis ir jų svoriai tampa konstantomis, tačiau ne visi sluoksniai yra užšaldomi. Sluoksniai, kurie yra susiję su batch normalizacija yra paliekami aktyvus su nuliniiais svoriais, tai padaryta dėl to, kad šie sluoksniai yra prisitaikę prie prieš tai naudoto duomenų rinkinio ir bandant apmokyti naują rinkinį jie neleisėtų tinklui mokytis, todėl visiems naujiems duomenų rinkiniams jie turi būti palikti aktyvūs. Kiekvienam modeliui buvo naudojami apsauginiai metodai, kad tinklas nepersimokytų šiuo atveju: jei mokant tinklą jo tikrinimo tikslumas per tris iteracijas nepagerėja mokymas yra nutraukiamas ir pradedamas testavimas.

Visi tinklai buvo apmokyti su kiekvienu rinkiniu po penkis kartus, gauti rezultatai buvo apdoroti ir išsaugoti.

Modelio mokymo metu panaudoti hiperparametrai:

- Epochų skaičius – 100 (mokymas stabdomas jei tikrinimo tikslumas nepakinta per 3 iteracijas);
- Batch dydis – 128 (MobileNet ir MobileNetV2 – 32-64);
- Nuostolio funkcija – kategorinė kryžminė entropija;

- Optimizavimo algoritmas – Adam
- Mokymo koeficientas –  $1e-4$

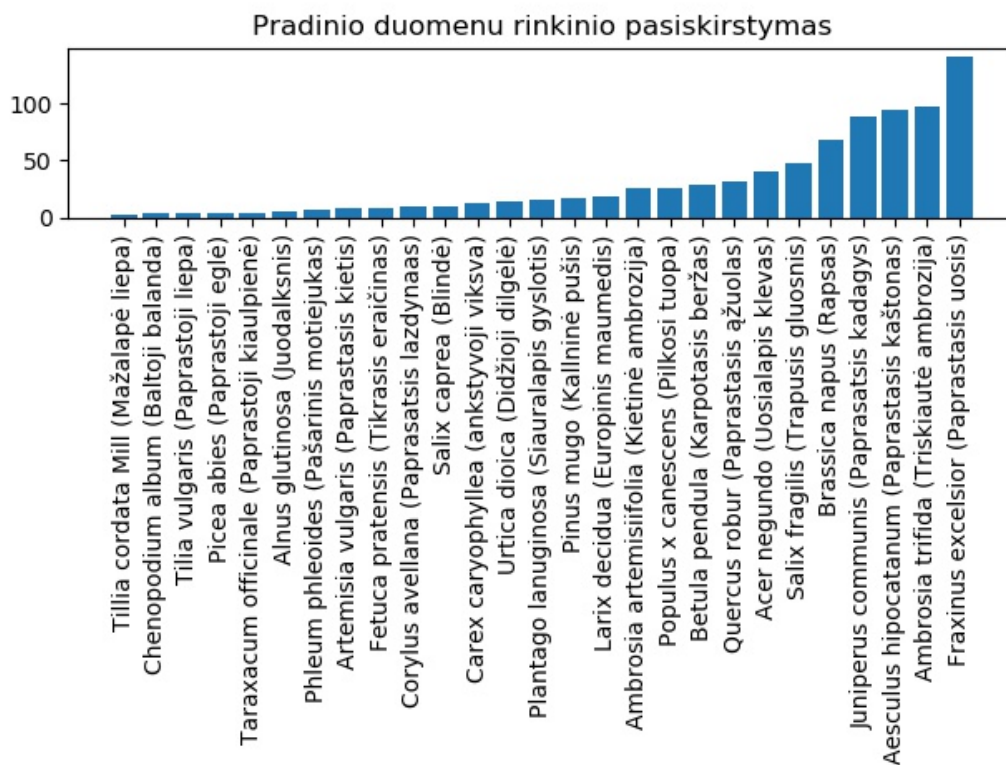


6 pav. Modelio apmokymo algoritmas



### 3.2. Duomenų rinkinio sugeneravimas

Pradinio duomenų rinkinio pasiskirstymas pavaizduotas 7 paveiksle ir pateiktas lentelėje 1, jame galima pastebėti, kad pradinis duomenų rinkinys buvo nesubalansuotas: vienos klasės turėjo žymiai daugiau klasės pavyzdžių nei kitos. Siekiant išvengti tinklo persimokymo buvo nuspręsta jų nenaudoti, tačiau sugeneruoti subalansuotus duomenų rinkinius naudojantis augmentavimo biblioteka.



7 pav. Pradinio duomenų rinkinio pasiskirstymas

Lentelė 1 Kiekvienos klasės vaizdų kiekis pradiniame duomenų rinkinyje

nr	Klasė	Kiekis
0	Acer negundo (Uosialapis klevas)	123
1	Aesculus hipocatanum (Paprastasis kaštonas)	285
2	Alnus glutinosa (Juodalksnis)	15
3	Ambrosia artemisiifolia (Kietinė ambrozija)	75
4	Ambrosia trifida (Triskiautė ambrozija)	291
5	Artemisia vulgaris (Paprastasis kietis)	24
6	Betula pendula (Karpotasis beržas)	84
7	Brassica napus (Rapsas)	204
8	Carex caryophylla (ankstyvoji viksva)	39
9	Chenopodium album (Baltoji balanda)	9
10	Corylus avellana (Paprastasis lazdynas)	27
11	Fetuca pratensis (Tikrasis eraičinas)	24
12	Fraxinus excelsior (Paprastasis uosis)	423
13	Juniperus communis (Paprastasis kadagys)	267

14	Larix decidua (Europinis maumedis)	54
15	Phleum phleoides (Pašarinis motiejukas)	18
16	Picea abies (Paprastoji eglė)	12
17	Pinus mugo (Kallninė pušis)	51
18	Plantago lanuginosa (Siauralapis gyslotis)	48
19	Populus x canescens (Pilkosi tuopa)	75
20	Quercus robur (Paprastasis ažuolas)	93
21	Salix caprea (Blindė)	30
22	Salix fragilis (Trapusis gluosnis)	141
23	Taraxacum officinale (Paprastoji kiaulpienė)	12
24	Tilia vulgaris (Paprastoji liepa)	9
25	Tilia cordata Mill (Mažalapė liepa)	6
26	Urtica dioica (Didžioji dilgėlė)	42

Tinklo apmokymui sugeneruoti penki duomenų rinkiniai, kurie yra paremti pradinio rinkinio duomenimis ir kuriuos sudaro 27 klasės su skirtingais kiekiais pavyzdžių. Šių rinkinių kiekvienos klasės duomenys buvo suskirstyti į 80% skirtų mokymui ir 20% skirtų tikrinimui mokymo metu. Testavimui buvo sukurtas atskiras duomenų rinkinys, kurį sudarė po šimtas kiekvienos klasės pavyzdžių, jis niekaip nebuvo susijęs su mokymo procesu. Duomenų pasiskirstymas matomas lentelėje 2.

Lentelė 2 Duomenų rinkinių pasiskirstymas

Klasė turi pavyzdžių	Mokymui skirta	Tikrinimui skirta	Iš viso mokymui	Iš viso tikrinimui	iš viso paveikslėlių	Testavimui skirta
50	40	10	1080	270	1350	100
100	80	20	2160	540	2700	
300	240	60	6480	1620	8100	
500	400	100	10800	2700	13500	
1000	800	200	21600	5400	27000	

Taip pat verta paminėti, kad duomenų rinkiniai tinklams buvo generuojami atsižvelgiant į mažiausią vaizdo dydį leistiną pačiam tinklui, šiuo atveju vaizdų dydžiai buvo:

- 32x32 px
- 71x71 px
- 75x75 px
- 128x128 px

### 3.3. Pasirinktos konvoliucinių tinklų architektūros

Šiam tyrimui buvo pasirinktos 18 skirtingų architektūrų:

- DenseNet121
- DenseNet169
- DenseNet201
- ResNet50
- ResNet101
- ResNet152
- ResNet50V2
- ResNet101V2
- ResNet152V2
- ResNeXt50
- ResNeXt101
- VGG16
- VGG19
- Xception
- InceptionV3
- InceptionResNetV2
- MobileNet
- MobileNetV2

Šios architektūros pasirinktos, kadangi jos jau turi apmokytus ir lengvai prieinamus modelius, kurių tikslumai svyruoja nuo 70,4% iki 80,3%, pateiktus Keras kūrėjų. Tai užtikrina identišką pradinių modelių apmokymo sąlygas ir neįvelia netikslumų juos pritaikant naujiems vaizdams

### **3.4. Įranga naudota tyrimo metu**

#### **3.4.1. Techninė įranga**

Tyrimo metu naudojamas personalinis kompiuteris, kurio procesorius *intel i7-6700K*, kurio taktinis dažnis 4,2 GHz. Operatyvioji atmintis 16 GB, vaizdo plokštė *NVIDIA GeForce GTX 960* vidinė atmintis 2 GB. Kompiuteryje suinstaliuota Windows 10 64-bit operacinė sistema.

- Vaizdo plokštės tvarkyklės versija: 430.64
- Operacinės sistemos komponavimo versija: 17763.529

#### **3.4.2. Programinė įranga**

- Miniconda:
  - Python – 3.6.8;

- Conda – 4.6.14;
- PyCharm – 2019.1.2 (Build #PY-191.7141.48);
- Keras – 2.2.4;
- TensorFlow – 1.13.1 (Base – 1.13.1; Estimator – 1.13.0; GPU – 1.13.1)

## 4. Tyrimo rezultatų analizė

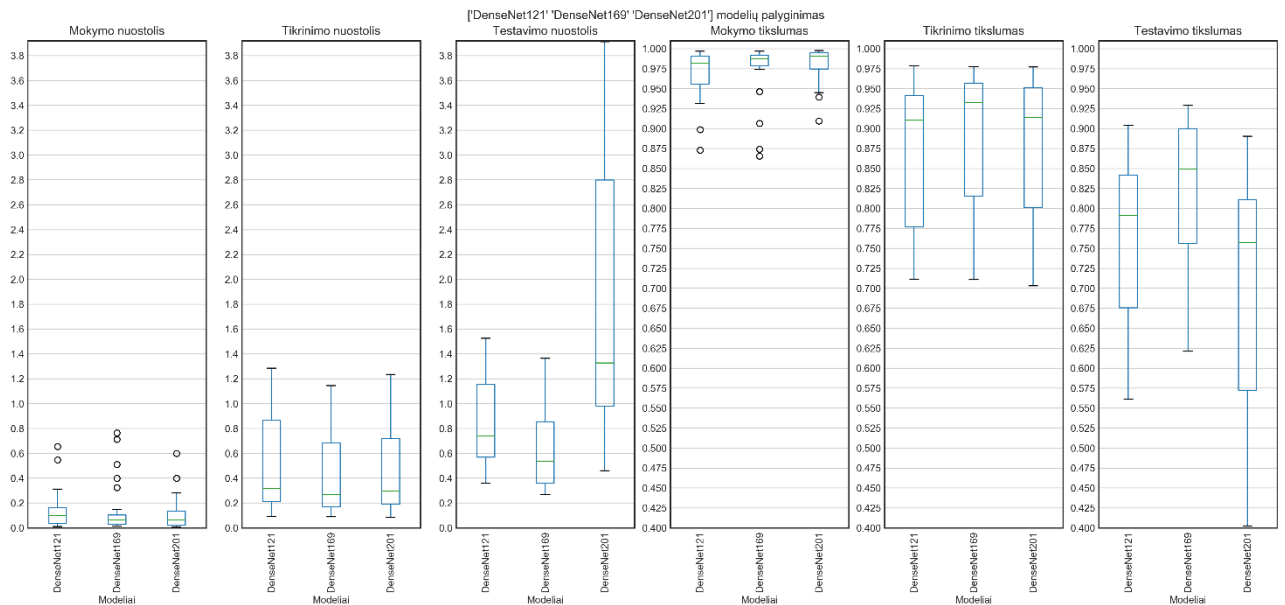
Visas duomenų apdorojimas buvo atliktas naudojantis „Python 3.6.8“ programinę įrangą. Skaičiavimai ir grafikai pateikti naudojantis „Numpy“ ir „Matplotlib“ bibliotekas. Visi rezultatai pateikiami neatsižvelgiant į jų apmokymo laiką.

### 4.1. Individualių konvoliucinių neuroninių tinklų rezultatų analizė

Kadangi yra ištirta iš viso yra apmokyta 18 tinklų, pirma apžvelgsiu kiekvienos architektūros ir jų tinklų rezultatus.

#### 4.1.1. DenseNet

DenseNet tinklai, apibendrinus visų gautų modelių rezultatus, pateiktus 8 paveiksle, neatsižvelgiant į jų apmokymui naudotus duomenų rinkinių dydžius, mokymo ir tikrinimo rezultatai yra panašūs ir patenka į tas pačias ribas. Testavimo metu DenseNet201 išsiskyrė, kadangi jo nuostolis testavimo duomenims buvo didžiausias, o tikslumas turėjo plačiausią imtį nuo 40% iki beveik 90%.



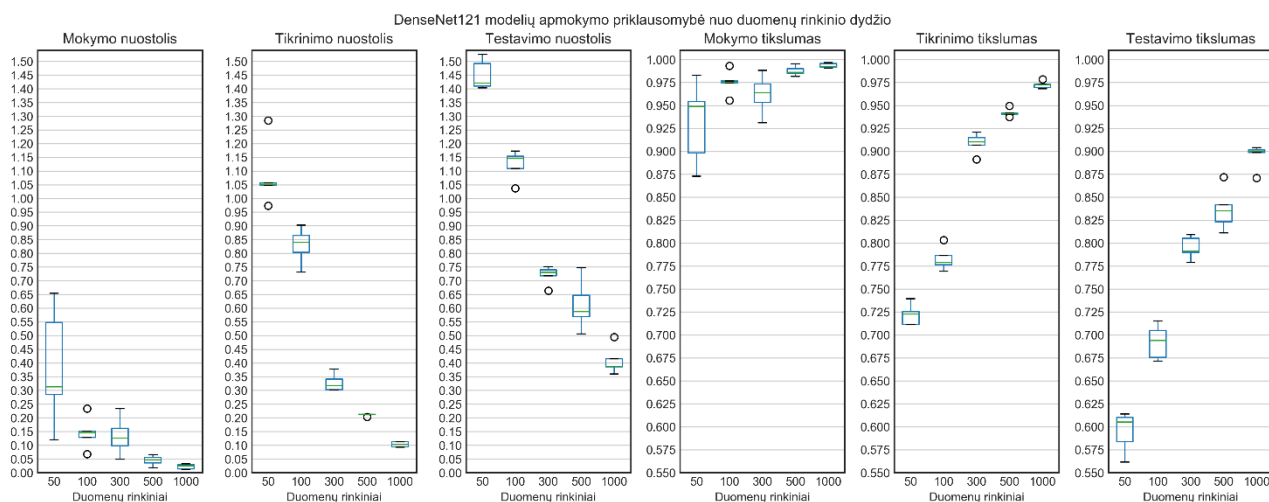
8 pav. DenseNet121, DenseNet169, DenseNet201 modelių palyginimas

##### 4.1.1.1. DenseNet121

Apmokius DenseNet121 iš gautų rezultatų pateiktų 9 paveiksle, galima pastebėti, kad didinant kiekvienos klasės pavyzdžių kiekį tinklo gebėjimas įsisavinti informaciją didėja. Su 50

kiekvienos klasės pavyzdžių apmokyto modelio tikslumas su testavimo duomenimis svyravo apie 60%, kai su 1000 – svyravo apie 90%.

Atsižvelgiant į kiekvieno modelio duomenis, pateiktus lentelėje 3, galima teigti, kad modeliai, kurie buvo apmokyti su 1000 kiekvienos klasės pavyzdžių buvo tiksliausi. Jų mokymo periodu atliktų tikrinimų tikslumai svyravo apie 97%, o testavimo tikslumas vidutiniškai siekė 90%. Ketvirtu bandymu mokytas tinklas (paryškinta eilutė) pasiekė geriausius rezultatus: tikrinimo tikslumas siekė 97,8%, o testavimo metu tikslumas siekė 90,4%.



9 pav. DenseNet121 modelių apmokymo priklausomybė nuo duomenų rinkinio dydžio

Lentelė 3 DenseNet121 modelių rezultatai

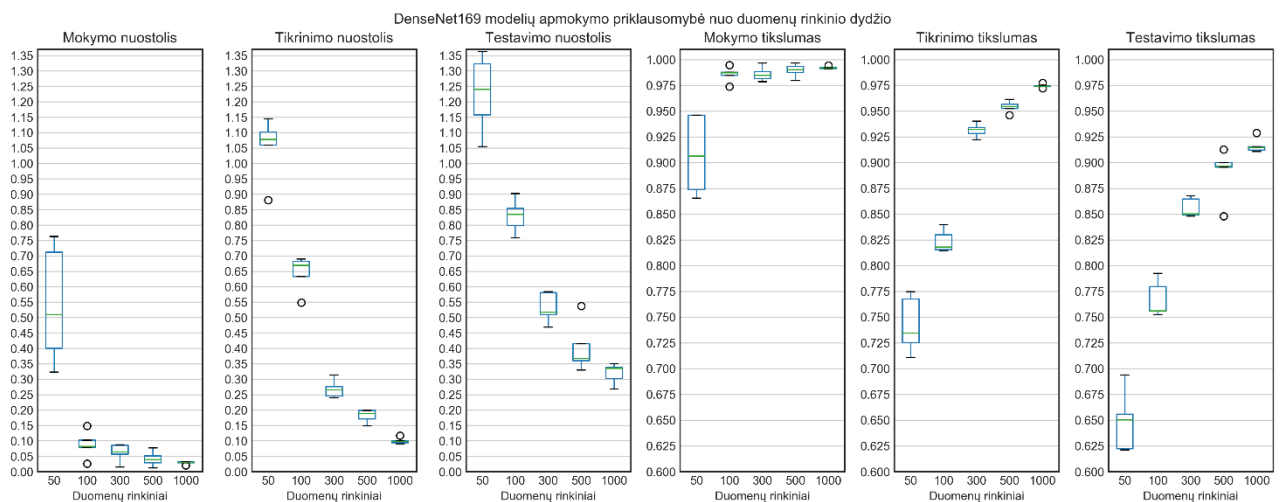
Klasė turi pavyzdžių	Mokymo nuostolis	Mokymo tikslumas	Tikrinimo nuostolis	Tikrinimo tikslumas	Testavimo nuostolis	Testavimo Tikslumas	Vaizdo dydis	Bandymas
50	0,1195	0,983	0,9736	0,7227	1,4038	0,61	32	1
	0,5474	0,8985	1,0485	0,7113	1,4921	0,5837		2
	0,654	0,873	1,2847	0,7113	1,5251	0,5614		3
	0,2851	0,9543	1,0571	0,7254	1,4091	0,614		4
	0,3131	0,949	1,0513	0,7394	1,4208	0,6051		5
100	0,1282	0,9755	0,8401	0,7767	1,1458	0,694		1
	0,2334	0,9556	0,8648	0,7791	1,1544	0,6714		2
	0,1461	0,9748	0,8038	0,8034	1,1725	0,6755		3
	0,0674	0,9931	0,7313	0,7864	1,0372	0,7151		4
	0,1511	0,9768	0,9026	0,7694	1,109	0,7048		5
300	0,1616	0,9534	0,3014	0,9149	0,7177	0,7914		1
	0,0488	0,9882	0,3172	0,9209	0,7301	0,8051		2
	0,2333	0,9312	0,3779	0,8914	0,7398	0,7792		3
	0,1262	0,9638	0,3023	0,9102	0,6631	0,8092		4
	0,0985	0,9736	0,3407	0,9068	0,7507	0,7899		5
500	0,055	0,985	0,2113	0,9409	0,5687	0,8418	1	

	0,0178	0,9952	0,2132	0,9495	0,5059	0,8718		2
	0,0658	0,9817	0,2038	0,9405	0,587	0,8351		3
	0,0464	0,986	0,2114	0,9413	0,6468	0,8233		4
	0,0357	0,9899	0,2131	0,9374	0,7469	0,8111		5
1000	0,0327	0,9906	0,1129	0,9683	0,4948	0,8711		1
	0,0289	0,9919	0,1129	0,9695	0,3597	0,8988		2
	0,0115	0,9966	0,1038	0,9727	0,4157	0,9014		3
	0,0134	0,9957	0,0934	0,9786	0,3864	0,904		4
	0,0252	0,9926	0,0933	0,9729	0,3875	0,8999		5

#### 4.1.1.2. DenseNet169

Apmokius DenseNet169 iš gautų rezultatų pateiktų 10 paveiksle, galima pastebėti, kad didinant kiekvienos klasės pavyzdžių kiekį tinklo gebėjimas įsisavinti informaciją didėja. Su 50 kiekvienos klasės pavyzdžių apmokyto modelio tikslumas su testavimo duomenimis svyravo apie 65%, kai su 1000 – svyravo apie 92,5%.

Atsižvelgiant į kiekvieno modelio duomenis, pateiktus lentelėje 4, galima teigti, kad modeliai, kurie buvo apmokyti su 1000 kiekvienos klasės pavyzdžių buvo tiksliausi. Jų mokymo periodu atliktų tikrinimų tikslumai svyravo apie 97%, o testavimo tikslumas vidutiniškai siekė 91,5%. Ketvirtu bandymu mokytas tinklas (paryškinta eilutė) pasiekė geriausius rezultatus: nors tikrinimo tikslumas siekė tik 97,4%, trečiu atveju buvo pasiekta 97,7%, tačiau testavimo metu tikslumas siekė 92,8%, lyginant su trečio atvejo 91,5%.



10 pav. DenseNet169 modelių apmokymo priklausomybė nuo duomenų rinkinio dydžio

Lentelė 4 DenseNet169 modelių rezultatai

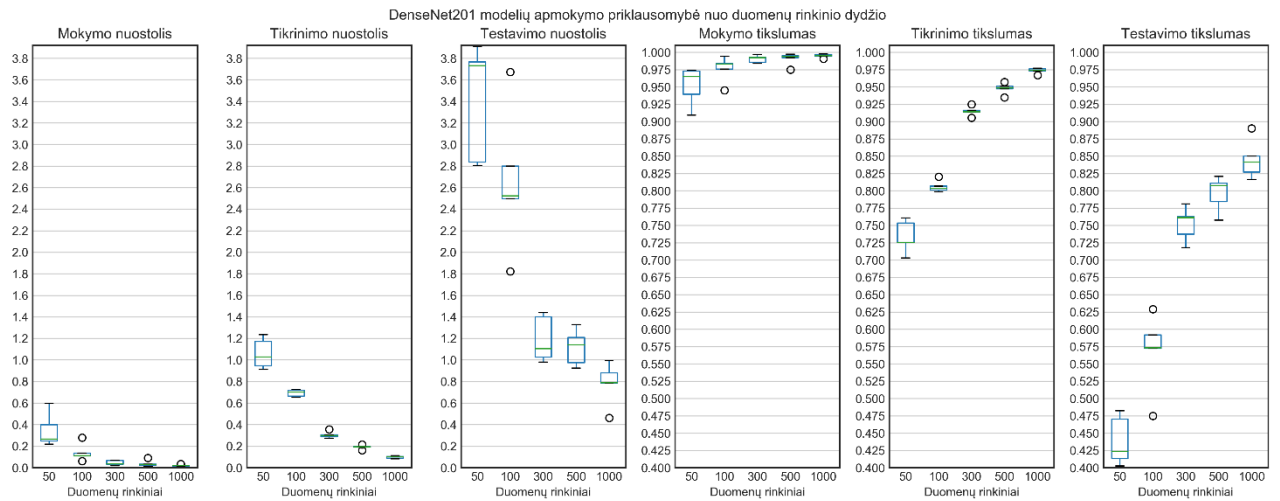
Klasė turi pavyzdžių	Mokymo nuostolis	Mokymo tikslumas	Tikrinimo nuostolis	Tikrinimo tikslumas	Testavimo nuostolis	Testavimo Tikslumas	Vaizdo dydis	Bandymas
50	0,7634	0,874	1,1442	0,7254	1,3225	0,6211	32	1

	0,7116	0,8655	1,1019	0,7676	1,2405	0,6503	2
	0,3996	0,9462	1,0776	0,7109	1,0542	0,694	3
	0,3234	0,9462	0,8813	0,7746	1,1575	0,6559	4
	0,5099	0,9065	1,0591	0,7344	1,3629	0,6222	5
100	0,078	0,9872	0,6894	0,8155	0,7585	0,7796	1
	0,0821	0,9877	0,669	0,8301	0,9019	0,7525	2
	0,1019	0,9848	0,6826	0,8145	0,8535	0,7559	3
	0,0257	0,9946	0,5481	0,8398	0,7984	0,7925	4
	0,1485	0,974	0,6327	0,818	0,8345	0,7562	5
300	0,0643	0,9848	0,2392	0,9323	0,5841	0,8481	1
	0,0872	0,9818	0,2759	0,9283	0,5095	0,8503	2
	0,0853	0,9785	0,3137	0,9223	0,4691	0,8648	3
	0,0566	0,9884	0,2452	0,9343	0,581	0,8492	4
	0,0145	0,9968	0,2655	0,9403	0,5169	0,8677	5
500	0,0388	0,9904	0,1707	0,9568	0,3599	0,8955	1
	0,0122	0,9966	0,1483	0,9615	0,4153	0,8999	2
	0,0766	0,9796	0,1987	0,946	0,5375	0,8481	3
	0,0515	0,9875	0,1987	0,9545	0,3295	0,9129	4
	0,0287	0,9934	0,1881	0,9526	0,3677	0,8966	5
1000	0,032	0,9919	0,0973	0,9723	0,302	0,9148	1
	0,03	0,9911	0,0993	0,9744	0,3355	0,9107	2
	0,0285	0,9926	0,0937	0,9774	0,3379	0,9155	3
	0,0311	0,9915	0,0901	0,9742	0,268	0,9288	4
	0,0206	0,9945	0,1166	0,9738	0,3503	0,9122	5

#### 4.1.1.3. DenseNet201

Apmokius DenseNet201 iš gautų rezultatų pateiktų 11 paveiksle, galima pastebėti, kad didinant kiekvienos klasės pavyzdžių kiekį tinklo gebėjimas įsisavinti informaciją didėja. Su 50 kiekvienos klasės pavyzdžių apmokyto modelio tikslumas su testavimo duomenimis svyravo apie 42,5%, kai su 1000 – svyravo apie 85%.

Atsižvelgiant į kiekvieno modelio duomenis, pateiktus lentelėje 5, galima teigti, kad modeliai, kurie buvo apmokyti su 1000 kiekvienos klasės pavyzdžių buvo tiksliausi. Jų mokymo periodu atliktų tikrinimų tikslumai svyravo apie 97%, o testavimo tikslumas vidutiniškai siekė 91,5%. Penktu bandymu mokytas tinklas (paryškinta eilutė) pasiekė geriausius rezultatus: tikrinimo tikslumas siekė 97,6%, testavimo metu tikslumas siekė 89%.



11 pav. DenseNet201 modelių apmokymo priklausomybė nuo duomenų rinkinio dydžio

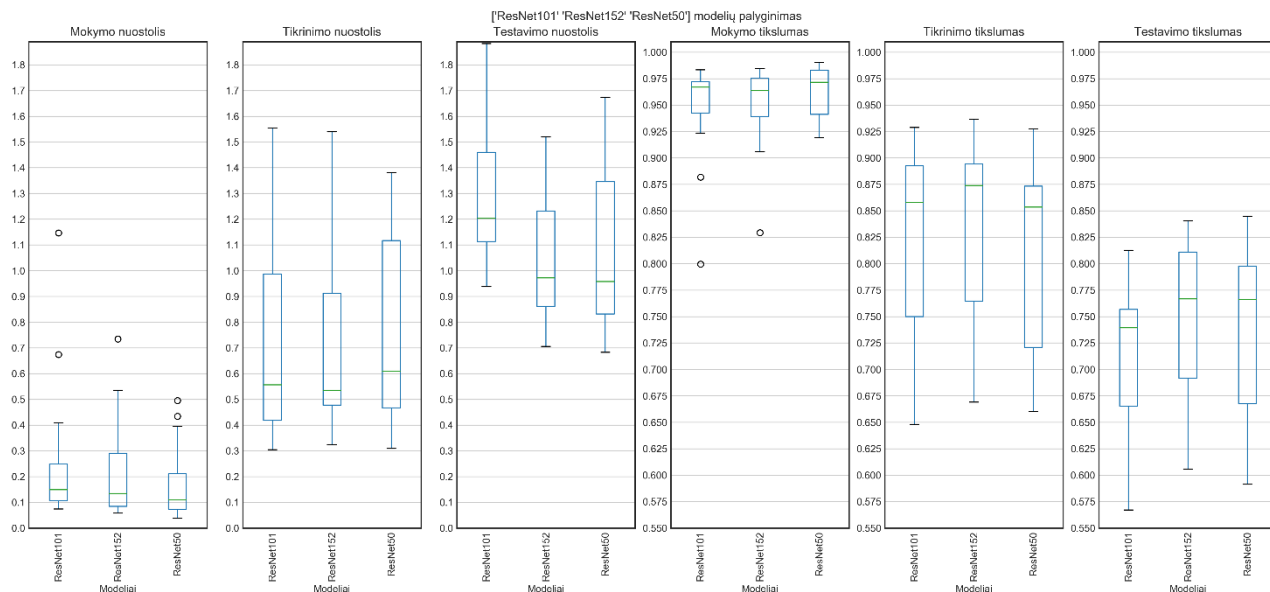
Lentelė 5 DenseNet201 modelių rezultatai

Klasė turi pavyzdžių	Mokymo nuostolis	Mokymo tikslumas	Tikrinimo nuostolis	Tikrinimo tikslumas	Testavimo nuostolis	Testavimo Tikslumas	Vaizdo dydis	Bandymas
50	0,266	0,973	0,9134	0,7606	2,8357	0,4822	32	1
	0,5985	0,9093	1,1725	0,7254	3,9111	0,4025		2
	0,2479	0,965	1,029	0,7254	3,7319	0,4237		3
	0,2185	0,9741	0,9445	0,7535	2,8042	0,4703		4
	0,3978	0,9393	1,2345	0,7031	3,7656	0,4133		5
100	0,2794	0,945	0,7258	0,801	3,6729	0,4748		1
	0,0615	0,9945	0,7003	0,8066	2,798	0,5725		2
	0,1353	0,9757	0,7171	0,7985	1,8218	0,6288		3
	0,1118	0,9833	0,6639	0,8034	2,5237	0,5733		4
	0,1097	0,9834	0,6521	0,8204	2,496	0,5918		5
300	0,0679	0,9856	0,2973	0,9135	1,0249	0,7625		1
	0,0332	0,9924	0,3061	0,9141	1,1036	0,7614		2
	0,0315	0,9926	0,2869	0,9162	1,4382	0,7177		3
	0,0664	0,9841	0,3566	0,9055	1,4001	0,7374		4
	0,0192	0,9966	0,2745	0,9249	0,9793	0,7811		5
500	0,0329	0,9924	0,194	0,9471	0,9734	0,8107		1
	0,0228	0,9951	0,1955	0,951	0,9231	0,8207		2
	0,0287	0,994	0,1906	0,9483	1,1382	0,8074		3
	0,0132	0,9972	0,1627	0,9572	1,2079	0,7844		4
	0,089	0,9747	0,214	0,9347	1,3272	0,7574		5
1000	0,0089	0,9976	0,0836	0,977	0,9934	0,827		1
	0,034	0,9906	0,1123	0,967	0,879	0,8162		2
	0,0191	0,9945	0,1036	0,9731	0,788	0,8414		3
	0,0175	0,9951	0,1021	0,974	0,7874	0,8499		4
	0,0111	0,9967	0,0866	0,9761	0,4611	0,8903		5



## 4.1.2. ResNet

Apmokius ResNet tinklus 12 paveiksle pastebime, kad mokymo tikslumai turi atvejus, kurie nepatenka į imtį, tikrinimo tikslumai išsidėstę su panašiais vidurkiais, svyruoja apie 86%, testavimo tikslumai irgi panašūs, svyruoja 76%.

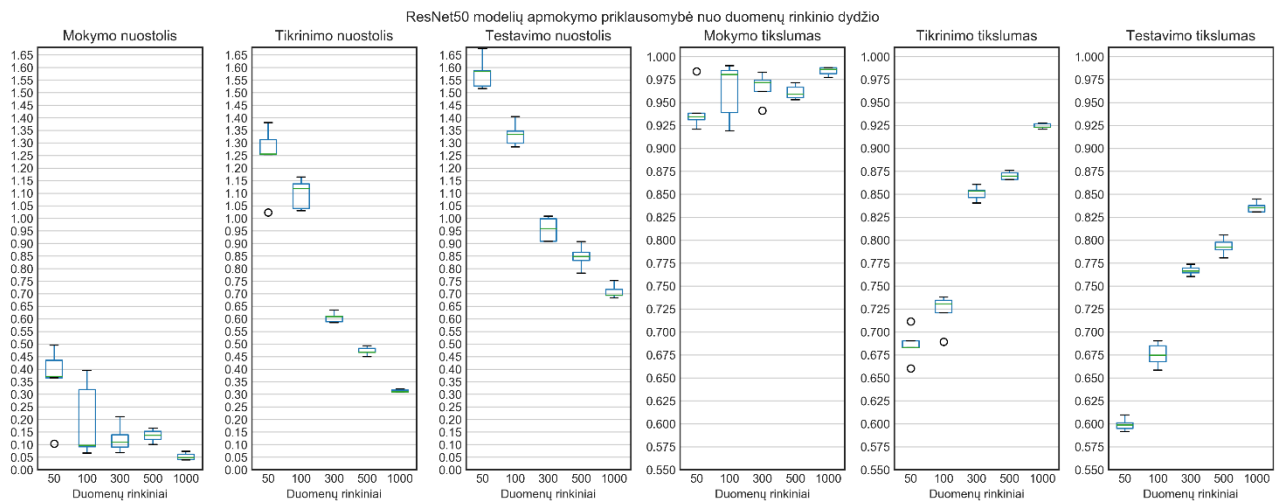


12 pav. ResNet101, ResNet152, ResNet50 modelių palyginimas

### 4.1.2.1. ResNet50

Apmokius ResNet50 iš gautų rezultatų pateiktų 13 paveiksle, galima pastebėti, kad didinant kiekvienos klasės pavyzdžių kiekį tinklo gebėjimas įsisavinti informaciją didėja. Su 50 kiekvienos klasės pavyzdžių apmokyto modelio tikslumas su testavimo duomenimis svyravo apie 60%, kai su 1000 – svyravo apie 83%.

Atsižvelgiant į kiekvieno modelio duomenis, pateiktus lentelėje 6, galima teigti, kad modeliai, kurie buvo apmokyti su 1000 kiekvienos klasės pavyzdžių buvo tiksliausi. Jų mokymo periodu atliktų tikrinimų tikslumai svyravo apie 92%, o testavimo tikslumas vidutiniškai siekė 83%. Ketvirtu bandymu mokytas tinklas (paryškinta eilutė) pasiekė geriausias rezultatus: tikrinimo tikslumas siekė 92,7%, testavimo metu tikslumas siekė 84,4%.



13 pav. ResNet50 modelių apmokymo priklausomybė nuo duomenų rinkinio dydžio

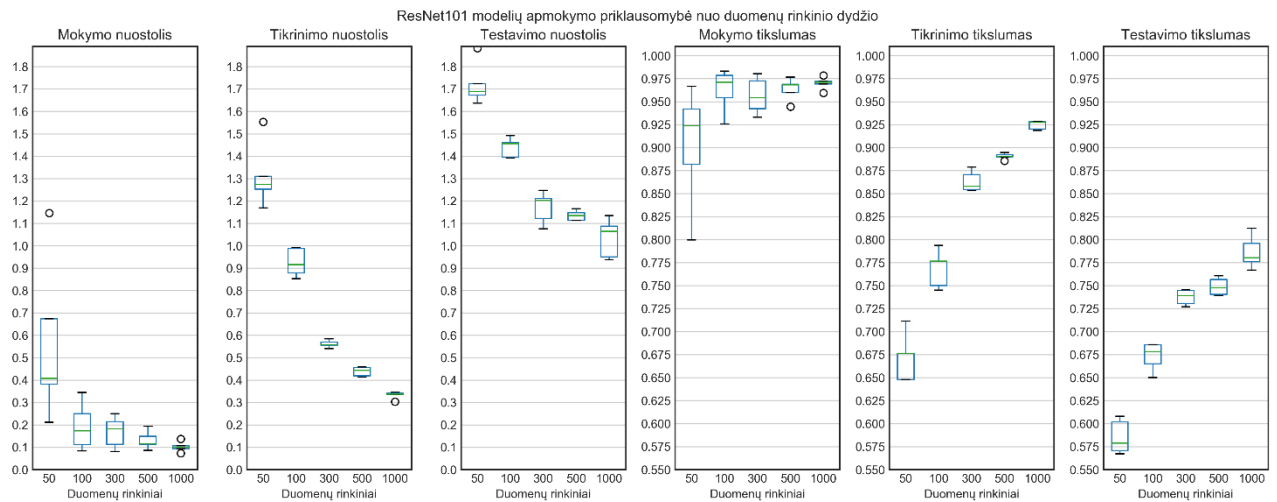
Lentelė 6 ResNet50 modelių rezultatai

Klasė turi pavyzdžių	Mokymo nuostolis	Mokymo tikslumas	Tikrinimo nuostolis	Tikrinimo tikslumas	Testavimo nuostolis	Testavimo Tikslumas	Vaizdo dydis	Bandymas
50	0,4952	0,9209	1,3805	0,6602	1,5166	0,5914	32	1
	0,3653	0,9343	1,2556	0,6901	1,588	0,5948		2
	0,3705	0,9384	1,3128	0,6831	1,5264	0,6011		3
	0,1031	0,9838	1,0229	0,7113	1,6739	0,6096		4
	0,4343	0,9311	1,2537	0,6831	1,5846	0,5985		5
100	0,3188	0,939	1,0397	0,7209	1,2985	0,6585		1
	0,3947	0,919	1,1368	0,6893	1,2843	0,6674		2
	0,0963	0,9805	1,03	0,7379	1,3341	0,6851		3
	0,0911	0,9849	1,1173	0,7306	1,4041	0,6748		4
	0,0661	0,9902	1,164	0,7344	1,3461	0,6903		5
300	0,139	0,962	0,5883	0,8465	0,9987	0,7603		1
	0,1091	0,9716	0,6106	0,8535	0,9579	0,7644		2
	0,2109	0,9411	0,6356	0,8405	0,9078	0,7662		3
	0,0671	0,983	0,5852	0,8606	1,0083	0,7737		4
	0,0899	0,9742	0,6088	0,8546	0,9079	0,7696		5
500	0,1365	0,9589	0,4665	0,8733	0,9068	0,7807		1
	0,1005	0,9715	0,467	0,876	0,8315	0,8055		2
	0,1646	0,9531	0,4824	0,8659	0,7824	0,7977		3
	0,153	0,9551	0,4495	0,8698	0,848	0,7925		4
	0,121	0,9666	0,4923	0,8659	0,8653	0,7896		5
1000	0,0483	0,986	0,3162	0,9232	0,7528	0,8307		1
	0,04	0,9878	0,3208	0,9274	0,693	0,8377		2
	0,0617	0,9812	0,3101	0,9232	0,7173	0,8351		3
	0,0384	0,988	0,309	0,9268	0,6946	0,8448		4
	0,0723	0,9774	0,3093	0,9209	0,6826	0,8307		5

#### 4.1.2.2. ResNet101

Apmokius ResNet101 iš gautų rezultatų pateiktų 14 paveiksle, galima pastebėti, kad didinant kiekvienos klasės pavyzdžių kiekį tinklo gebėjimas įsisavinti informaciją didėja. Su 50 kiekvienos klasės pavyzdžių apmokyto modelio tikslumas su testavimo duomenimis svyravo apie 57,5%, kai su 1000 – svyravo apie 78%.

Atsižvelgiant į kiekvieno modelio duomenis, pateiktus lentelėje 7, galima teigti, kad modeliai, kurie buvo apmokyti su 1000 kiekvienos klasės pavyzdžių buvo tiksliausi. Jų mokymo periodu atliktų tikrinimų tikslumai svyravo apie 92%, o testavimo tikslumas vidutiniškai siekė 78%. Penktu bandymu mokytas tinklas (paryškinta eilutė) pasiekė geriausius rezultatus: tikrinimo tikslumas siekė 92,9%, testavimo metu tikslumas siekė 81,3%.



14 pav. ResNet101 modelių priklausomybė nuo duomenų rinkinio dydžio

Lentelė 7 ResNet101 modelių rezultatai

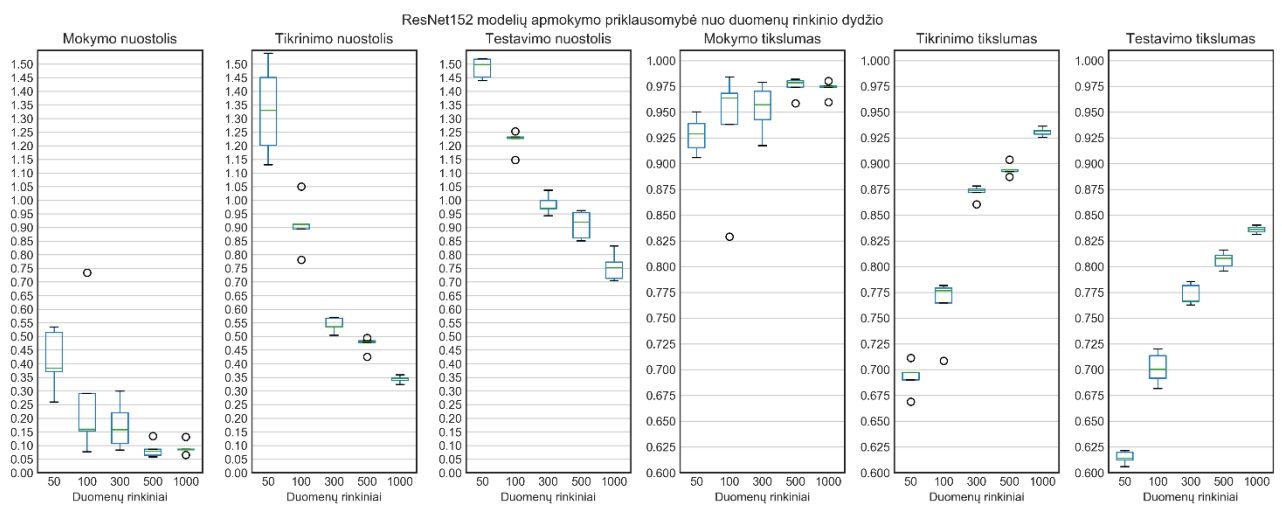
Klasė turi pavyzdžių	Mokymo nuostolis	Mokymo tikslumas	Tikrinimo nuostolis	Tikrinimo tikslumas	Testavimo nuostolis	Testavimo Tikslumas	Vaizdo dydis	Bandymas
50	1,1465	0,7996	1,5538	0,6479	1,6894	0,5788	32	1
	0,381	0,9421	1,1686	0,7113	1,8825	0,567		2
	0,674	0,8819	1,2529	0,6761	1,7237	0,5707		3
	0,2118	0,9668	1,2744	0,6761	1,6369	0,6081		4
	0,4077	0,9236	1,3095	0,6479	1,6729	0,6018		5
100	0,0835	0,9833	0,8781	0,7937	1,3945	0,6855	32	1
	0,2492	0,9542	0,9152	0,75	1,3924	0,6651		2
	0,1731	0,971	0,9928	0,7451	1,456	0,6781		3
	0,3448	0,9257	0,8533	0,7767	1,4917	0,6499		4
	0,1108	0,9785	0,9876	0,7767	1,4597	0,6859		5

300	0,2495	0,9331	0,5554	0,8532	1,2026	0,727	1
	0,113	0,9726	0,5697	0,8706	1,1216	0,7448	2
	0,2134	0,9424	0,5405	0,8579	1,0762	0,7392	3
	0,1817	0,9542	0,5843	0,8546	1,2114	0,7303	4
	0,0812	0,9803	0,5565	0,8787	1,2474	0,7455	5
500	0,0854	0,9765	0,4182	0,89	1,1648	0,7566	1
	0,1928	0,9444	0,4587	0,8857	1,1126	0,7403	2
	0,1156	0,9682	0,4432	0,8904	1,1134	0,7477	3
	0,1483	0,9599	0,4563	0,8923	1,147	0,7392	4
	0,1111	0,9689	0,4136	0,8946	1,1348	0,7607	5
1000	0,1062	0,9692	0,3458	0,9186	0,9506	0,7959	1
	0,1371	0,9595	0,342	0,9203	1,1344	0,767	2
	0,0931	0,9721	0,3374	0,9275	1,0866	0,7803	3
	0,0994	0,9711	0,304	0,9287	1,0635	0,7759	4
	0,0732	0,9786	0,336	0,9285	0,9379	0,8125	5

#### 4.1.2.3. ResNet152

Apmokius ResNet101 iš gautų rezultatų pateiktų 15 paveiksle, galima pastebėti, kad didinant kiekvienos klasės pavyzdžių kiekį tinklo gebėjimas įsisavinti informaciją didėja. Su 50 kiekvienos klasės pavyzdžių apmokyto modelio tikslumas su testavimo duomenimis svyravo apie 61,5%, kai su 1000 – svyravo apie 83%.

Atsižvelgiant į kiekvieno modelio duomenis, pateiktus lentelėje 8, galima teigti, kad modeliai, kurie buvo apmokyti su 1000 kiekvienos klasės pavyzdžių buvo tiksliausi. Jų mokymo periodu atliktų tikrinimų tikslumai svyravo apie 92,5%, o testavimo tikslumas vidutiniškai siekė 83%. Penktu bandymu mokytas tinklas (paryškinta eilutė) pasiekė geriausius rezultatus: tikrinimo tikslumas siekė 93,2%, testavimo metu tikslumas siekė 84%.



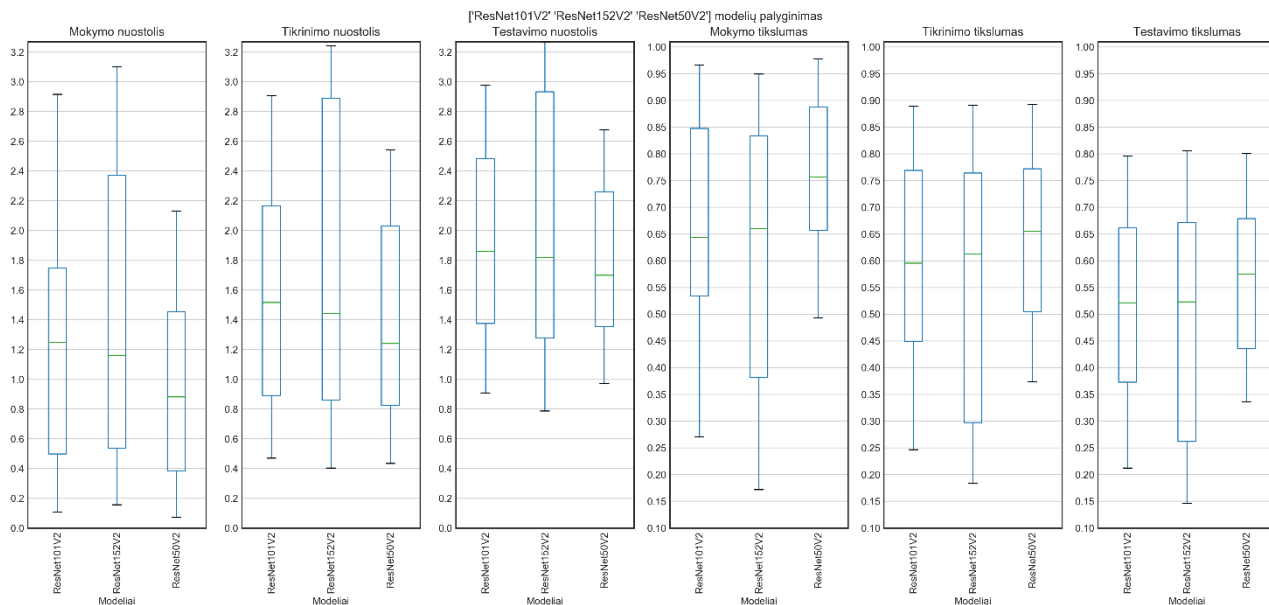
15 pav. ResNet152 modelių priklausomybė nuo duomenų rinkinio dydžio

Lentelė 8 ResNet152 modelių rezultatai

Klasė turi pavyzdžių	Mokymo nuostolis	Mokymo tikslumas	Tikrinimo nuostolis	Tikrinimo tikslumas	Testavimo nuostolis	Testavimo Tikslumas	Vaizdo dydis	Bandymas
50	0,5144	0,9156	1,1298	0,6901	1,4977	0,6059	32	1
	0,3704	0,939	1,3296	0,6972	1,4528	0,6214		2
	0,3834	0,9288	1,5393	0,6972	1,5173	0,6122		3
	0,2595	0,9503	1,4502	0,7113	1,5192	0,614		4
	0,5346	0,9057	1,2016	0,669	1,4382	0,6196		5
100	0,1534	0,9683	0,8942	0,7767	1,2318	0,7003		1
	0,1594	0,9637	0,7808	0,7791	1,1479	0,7137		2
	0,0764	0,9843	0,912	0,7816	1,2263	0,7199		3
	0,2904	0,9381	0,9111	0,7646	1,2525	0,6918		4
	0,7342	0,8292	1,0497	0,7087	1,229	0,6818		5
300	0,158	0,9571	0,5355	0,8753	0,9677	0,7818		1
	0,1075	0,9702	0,5345	0,8783	0,9985	0,7666		2
	0,2195	0,9428	0,5044	0,872	1,037	0,7629		3
	0,0834	0,9789	0,5672	0,874	0,9714	0,7855		4
	0,2989	0,9174	0,5699	0,8606	0,9434	0,7659		5
500	0,0576	0,982	0,4254	0,904	0,9616	0,8162		1
	0,0866	0,9743	0,4947	0,8939	0,861	0,8081		2
	0,1343	0,9587	0,48	0,8872	0,8507	0,8007		3
	0,0645	0,9806	0,4838	0,8923	0,9544	0,7959		4
	0,0788	0,9787	0,4775	0,8942	0,92	0,8111		5
1000	0,0646	0,9804	0,3595	0,9365	0,8317	0,8362		1
	0,0838	0,9749	0,3481	0,9285	0,7051	0,8381		2
	0,1311	0,9596	0,3378	0,9256	0,7517	0,8314		3
	0,0862	0,9753	0,3235	0,9313	0,7729	0,834		4
	0,0838	0,9743	0,3455	0,9323	0,7129	0,8403		5

#### 4.1.3. ResNetV2

Atsižvelgiant į apmokytus ResNetV2 modelius, pateiktus 16 paveiksle, galima pastebėti, kad visi modelių nuostoliai yra gan plačiose imtyse, ko nebuvo prieš tai mokytuose tinkluose. Tikslumai ResNet101V2 ir ResNet152V2 mokymo, tikrinimo ir testavimo atvejais buvo panašūs, ResNet201V2 turėjo geriausius tikslumus lyginant su kitais tinklais, neatsižvelgiant į duomenų rinkinio dydžius naudotus tinklo apmokymams.

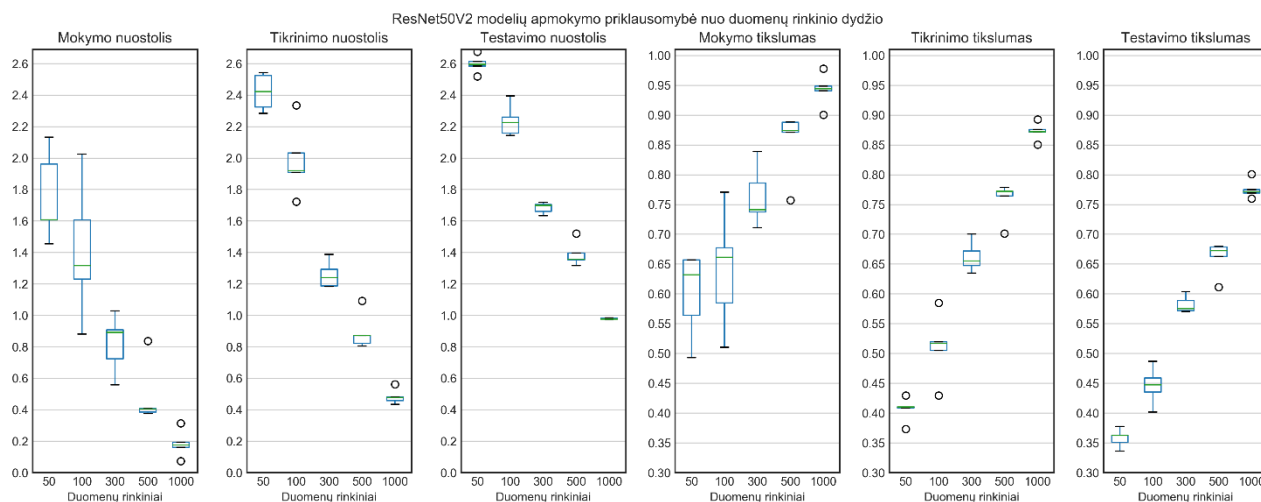


16 pav. ResNet101V2, ResNet152V2, ResNet50V2 modelių palyginimas

#### 4.1.3.1. ResNet50V2

Apmokius ResNet50V2 iš gautų rezultatų pateiktų 17 paveiksle, galima pastebėti, kad didinant kiekvienos klasės pavyzdžių kiekį tinklo gebėjimas įsisavinti informaciją didėja. Su 50 kiekvienos klasės pavyzdžių apmokyto modelio tikslumas su testavimo duomenimis svyravo apie 35,5%, kai su 1000 – svyravo apie 77%.

Atsižvelgiant į kiekvieno modelio duomenis, pateiktus lentelėje 9, galima teigti, kad modeliai, kurie buvo apmokyti su 1000 kiekvienos klasės pavyzdžių buvo tiksliausi. Jų mokymo periodu atliktų tikrinimų tikslumai svyravo apie 87%, o testavimo tikslumas vidutiniškai siekė 77%. Ketvirtu bandymu mokytas tinklas (paryškinta eilutė) pasiekė geriausius rezultatus: tikrinimo tikslumas siekė 89,3%, testavimo metu tikslumas siekė 80%.



17 pav. ResNet50V2 modelių priklausomybė nuo duomenų rinkinio dydžio

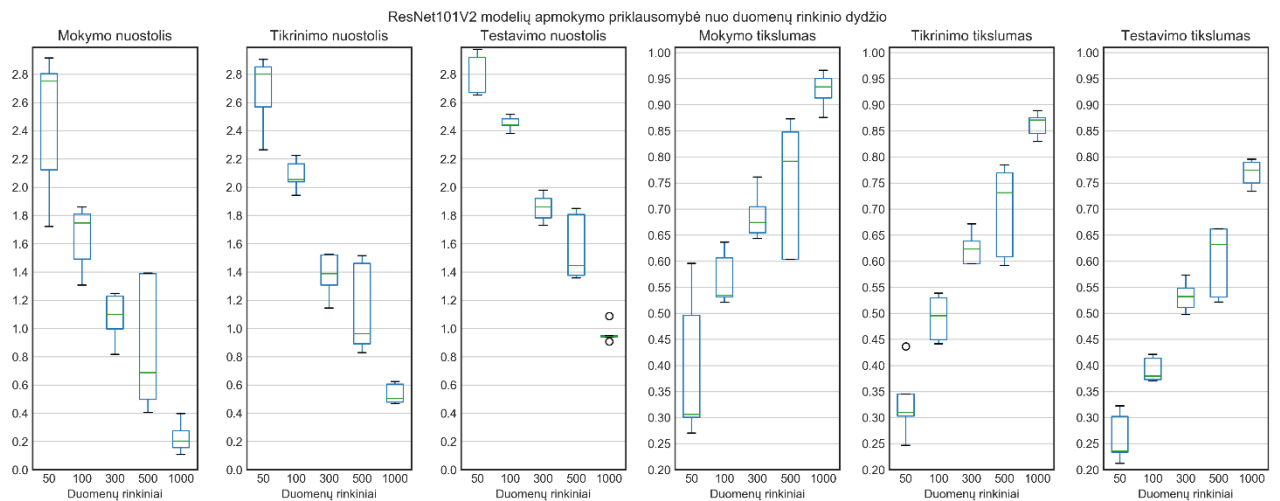
Lentelė 9 ResNet50V2 modelių rezultatai

Klasė turi pavyzdžių	Mokymo nuostolis	Mokymo tikslumas	Tikrinimo nuostolis	Tikrinimo tikslumas	Testavimo nuostolis	Testavimo Tikslumas	Vaizdo dydis	Bandymas
50	1,9613	0,5641	2,422	0,3732	2,6134	0,3503	32	1
	1,6069	0,6318	2,323	0,4102	2,5178	0,3625		2
	2,1315	0,4929	2,5431	0,4085	2,6758	0,3362		3
	1,4539	0,6571	2,2844	0,4102	2,5847	0,3774		4
	1,6065	0,6571	2,5236	0,4296	2,595	0,3625		5
100	1,2305	0,6772	1,9078	0,517	2,1585	0,4588		1
	0,881	0,7704	1,7219	0,585	2,1426	0,4866		2
	1,6056	0,5849	2,0317	0,5049	2,2254	0,4351		3
	1,3152	0,6612	1,9189	0,5194	2,2606	0,447		4
	2,0239	0,5104	2,334	0,4296	2,3954	0,4014		5
300	0,5572	0,8389	1,1831	0,7004	1,6605	0,6033		1
	0,7234	0,786	1,1867	0,6716	1,6328	0,5888		2
	0,8911	0,7374	1,2399	0,6548	1,7032	0,5751		3
	0,9073	0,7409	1,2918	0,6475	1,7193	0,5718		4
	1,0275	0,7107	1,3861	0,6347	1,6986	0,5703		5
500	0,4053	0,874	0,8704	0,7644	1,3529	0,6722		1
	0,3757	0,8877	0,8717	0,7784	1,3156	0,6788		2
	0,3846	0,8882	0,8228	0,7718	1,3557	0,6796		3
	0,406	0,871	0,8049	0,7718	1,3964	0,6625		4
	0,8365	0,7568	1,0919	0,701	1,5201	0,6111		5
1000	0,1746	0,9443	0,477	0,8716	0,9837	0,7692		1
	0,1612	0,9482	0,4567	0,8756	0,9757	0,7748		2
	0,1932	0,9406	0,4806	0,8718	0,971	0,7714		3
	0,0728	0,9776	0,434	0,8926	0,9754	0,8007		4
	0,314	0,9004	0,5611	0,8505	0,9814	0,7599		5

#### 4.1.3.2. ResNet101V2

Apmokius ResNet101V2 iš gautų rezultatų pateiktų 18 paveiksle, galima pastebėti, kad didinant kiekvienos klasės pavyzdžių kiekį tinklo gebėjimas įsisavinti informaciją didėja. Su 50 kiekvienos klasės pavyzdžių apmokyto modelio tikslumas su testavimo duomenimis svyravo apie 25%, kai su 1000 – svyravo apie 78%.

Atsižvelgiant į kiekvieno modelio duomenis, pateiktus lentelėje 10, galima teigti, kad modeliai, kurie buvo apmokyti su 1000 kiekvienos klasės pavyzdžių buvo tiksliausi. Jų mokymo periodu atliktų tikrinimų tikslumai svyravo apie 85%, o testavimo tikslumas vidutiniškai siekė 78%. Ketvirtu bandymu mokytas tinklas (paryškinta eilutė) pasiekė geriausius rezultatus: tikrinimo tikslumas siekė 88,8%, testavimo metu tikslumas siekė 79,5%.



18 pav. ResNet101V2 modelių priklausomybė nuo duomenų rinkinio dydžio

Lentelė 10 ResNet101V2 modelių rezultatai

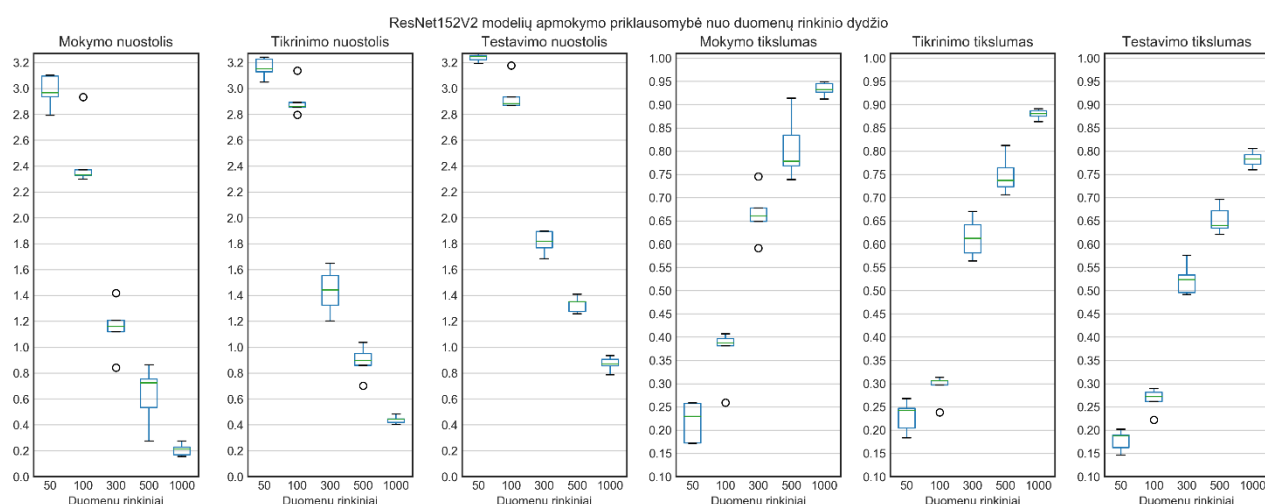
Klasė turi pavyzdžių	Mokymo nuostolis	Mokymo tikslumas	Tikrinimo nuostolis	Tikrinimo tikslumas	Testavimo nuostolis	Testavimo Tikslumas	Vaizdo dydis	Bandymas
50	2,7517	0,3061	2,8015	0,3028	2,9186	0,2337	32	1
	1,7222	0,5959	2,2652	0,4366	2,6691	0,3222		2
	2,8039	0,3004	2,8527	0,3099	2,9195	0,2355		3
	2,1224	0,4959	2,5686	0,3451	2,6516	0,3018		4
	2,9147	0,27	2,9058	0,2465	2,9755	0,2122		5
100	1,8089	0,5316	2,0545	0,4951	2,4348	0,3796		1
	1,3069	0,6362	1,9421	0,5388	2,441	0,4214		2
	1,86	0,5342	2,165	0,4417	2,5159	0,3703		3
	1,4911	0,6059	2,0382	0,5291	2,3799	0,4137		4
	1,746	0,5208	2,2243	0,449	2,483	0,3729		5
300	1,2269	0,6542	1,5201	0,5952	1,9214	0,5111		1
	0,9957	0,7043	1,3065	0,6387	1,7817	0,5477		2
	1,2481	0,6434	1,5262	0,5952	1,9787	0,4974		3
	0,8158	0,7616	1,144	0,6716	1,7295	0,5733		4
	1,099	0,6743	1,3886	0,6233	1,86	0,5318		5
500	0,4052	0,8728	0,8288	0,7846	1,3749	0,6614		1
	0,6861	0,7917	0,9622	0,7313	1,4447	0,6318		2
	1,3883	0,6035	1,5159	0,5921	1,8502	0,5214		3
	1,3914	0,6032	1,4606	0,6085	1,8072	0,5307		4
	0,4981	0,8474	0,8904	0,7691	1,3563	0,6622		5
1000	0,2033	0,9343	0,5041	0,8706	0,9406	0,7748		1
	0,2752	0,9128	0,6055	0,8448	0,9443	0,7499		2
	0,1563	0,95	0,4775	0,8752	0,9073	0,7892		3
	0,1066	0,9662	0,4689	0,8888	0,95	0,7955		4
	0,3961	0,8758	0,6246	0,8293	1,0886	0,7337		5



### 4.1.3.3. ResNet152V2

Apmokius ResNet152V2 iš gautų rezultatų pateiktų 19 paveiksle, galima pastebėti, kad didinant kiekvienos klasės pavyzdžių kiekį tinklo gebėjimas įsisavinti informaciją didėja. Su 50 kiekvienos klasės pavyzdžių apmokyto modelio tikslumas su testavimo duomenimis svyravo apie 17%, kai su 1000 – svyravo apie 79%.

Atsižvelgiant į kiekvieno modelio duomenis, pateiktus lentelėje 11, galima teigti, kad modeliai, kurie buvo apmokyti su 1000 kiekvienos klasės pavyzdžių buvo tiksliausi. Jų mokymo periodu atliktų tikrinimų tikslumai svyravo apie 88%, o testavimo tikslumas vidutiniškai siekė 79%. Trečiu bandymu mokytas tinklas (paryškinta eilutė) pasiekė geriausius rezultatus: tikrinimo tikslumas siekė 89,1%, testavimo metu tikslumas siekė 79,5%.



19 pav. ResNet152V2 modelių priklausomybė nuo duomenų rinkinio dydžio

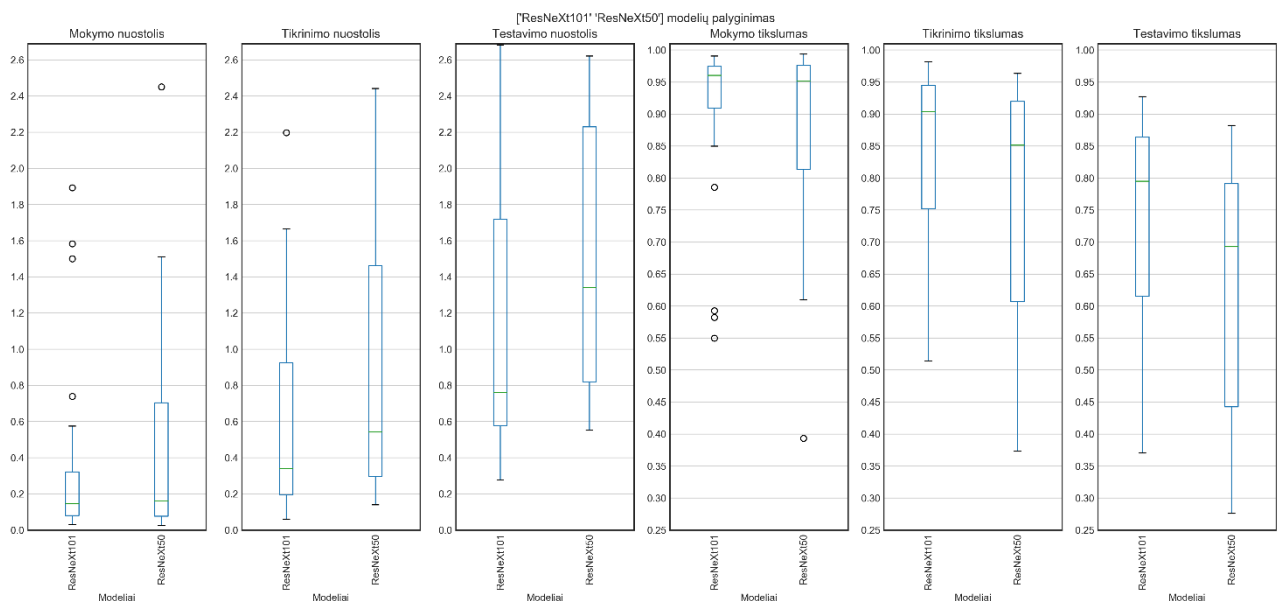
Lentelė 11 ResNet152V2 modelių rezultatai

Klasė turi pavyzdžių	Mokymo nuostolis	Mokymo tikslumas	Tikrinimo nuostolis	Tikrinimo tikslumas	Testavimo nuostolis	Testavimo Tikslumas	Vaizdo dydis	Bandymas
50	3,0951	0,1717	3,2259	0,2042	3,265	0,1625	32	1
	2,9668	0,2294	3,1505	0,2465	3,2204	0,1874		2
	2,792	0,2584	3,0488	0,2422	3,1919	0,2018		3
	3,1016	0,1724	3,2409	0,1831	3,2545	0,1462		4
	2,9353	0,2575	3,1272	0,2676	3,2449	0,1892		5
100	2,2987	0,4069	2,7964	0,3058	2,8825	0,2718	32	1
	2,328	0,3965	2,8527	0,3058	2,8671	0,2892		2
	2,9339	0,2593	3,1378	0,2379	3,1776	0,2222		3
	2,33	0,3872	2,8597	0,3131	2,8682	0,2814		4
	2,3711	0,3812	2,8901	0,2969	2,9327	0,2618		5

300	1,1195	0,6779	1,3245	0,6414	1,768	0,5337	1
	0,841	0,7452	1,2029	0,6702	1,6836	0,5762	2
	1,4176	0,5915	1,6489	0,5637	1,8957	0,4914	3
	1,2064	0,6488	1,5537	0,5811	1,894	0,4959	4
	1,1607	0,6602	1,4416	0,6126	1,8187	0,5233	5
500	0,7522	0,7683	0,9504	0,7236	1,3515	0,6403	1
	0,5339	0,834	0,8595	0,764	1,2771	0,6714	2
	0,8639	0,7387	1,0372	0,7061	1,4096	0,6211	3
	0,2749	0,9138	0,7027	0,8118	1,2588	0,6962	4
	0,7237	0,778	0,8971	0,7368	1,3516	0,6344	5
1000	0,156	0,9491	0,4184	0,8864	0,7858	0,8055	1
	0,2123	0,9322	0,4447	0,8748	0,8556	0,7829	2
	0,1673	0,9455	0,4032	0,8913	0,871	0,7925	3
	0,2738	0,9119	0,4837	0,8631	0,9057	0,7596	4
	0,2288	0,9267	0,4431	0,8803	0,9349	0,7718	5

#### 4.1.4. ResNeXt

Nors ResNeXt101 turėjo keletą reikšmių, kurios nepateko į mokymo imtį, matoma 20 paveiksle. Tikrinimo ir testavimo tikslumų maksimumai abiejų imčių panašūs, nors ResNeXt50 minimumo reikšmė tikrinimo tikslume labiau nutolusi.

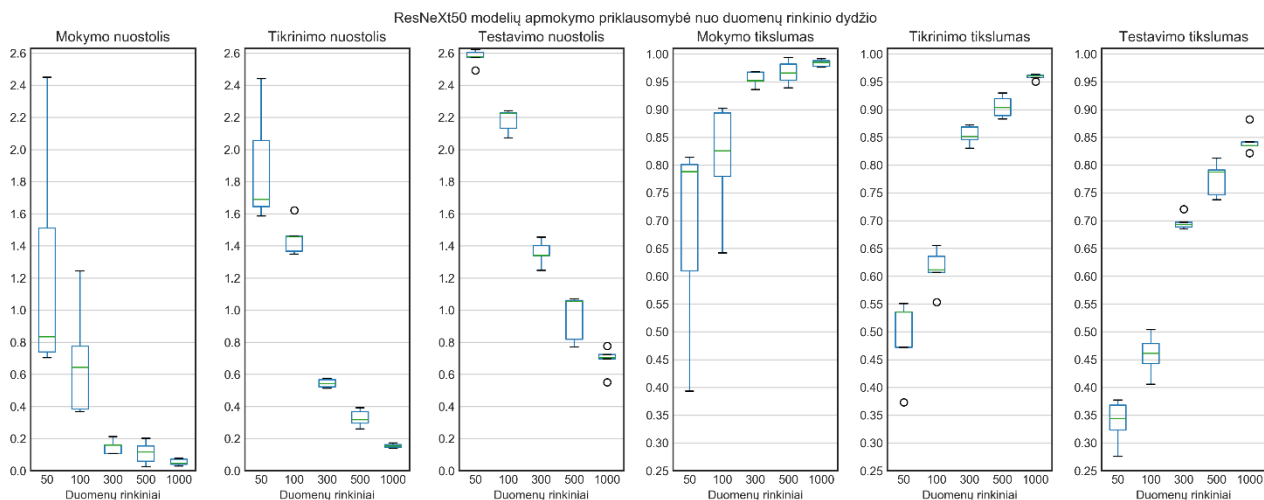


20 pav. ResNeXt101 ir ResNeXt50 modelių palyginimas

##### 4.1.4.1. ResNeXt50

Apmokius ResNeXt50 iš gautų rezultatų pateiktą 21 paveiksle, galima pastebėti, kad didinant kiekvienos klasės pavyzdžių kiekį tinklo gebėjimas įsisavinti informaciją didėja. Su 50 kiekvienos klasės pavyzdžių apmokyto modelio tikslumas su testavimo duomenimis svyravo apie 35%, kai su 1000 – svyravo apie 84%.

Atsižvelgiant į kiekvieno modelio duomenis, pateiktus lentelėje 12, galima teigti, kad modeliai, kurie buvo apmokyti su 1000 kiekvienos klasės pavyzdžių buvo tiksliausi. Jų mokymo periodu atliktų tikrinimų tikslumai svyravo apie 96%, o testavimo tikslumas vidutiniškai siekė 84%. Trečiu bandymu mokytas tinklas (paryškinta eilutė) pasiekė geriausius rezultatus: tikrinimo tikslumas siekė 96,3%, testavimo metu tikslumas siekė 88,2%.



21 pav. ResNeXt50 priklausomybė nuo duomenų rinkinio dydžio

Lentelė 12 ResNeXt50 modelių rezultatai

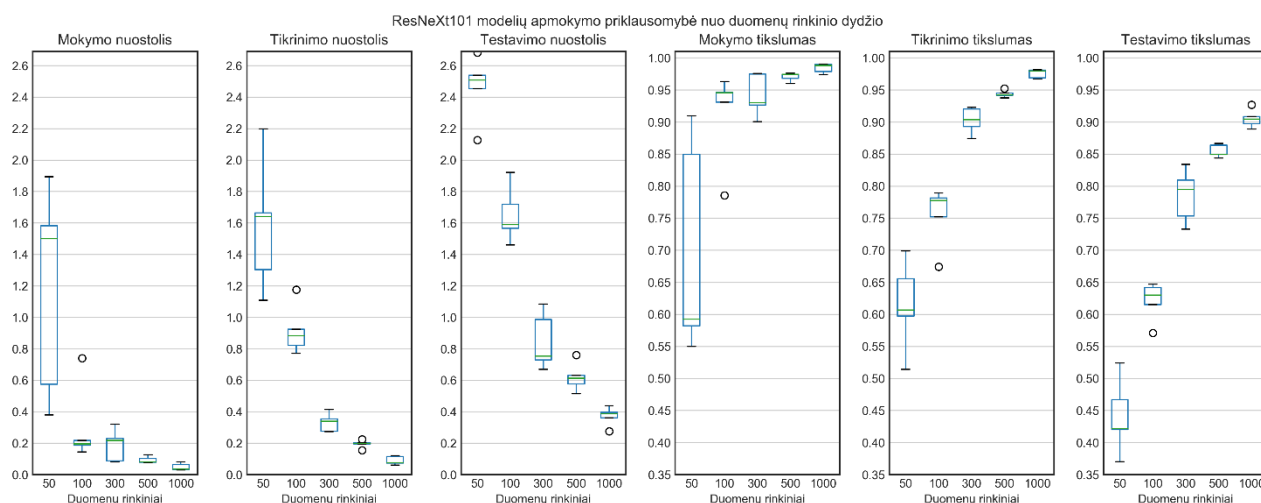
Klasė turi pavyzdžių	Mokymo nuostolis	Mokymo tikslumas	Tikrinimo nuostolis	Tikrinimo tikslumas	Testavimo nuostolis	Testavimo Tikslumas	Vaizdo dydis	Bandymas
50	1,5104	0,6093	2,0567	0,4718	2,5791	0,3233	32	1
	0,7393	0,801	1,5861	0,5352	2,6049	0,3437		2
	0,7039	0,8136	1,6448	0,5352	2,5731	0,3774		3
	2,4501	0,3934	2,4413	0,3732	2,6212	0,2759		4
	0,8338	0,7881	1,6886	0,5508	2,492	0,3677		5
100	0,644	0,8256	1,4618	0,6113	2,1325	0,4611		1
	0,3839	0,8938	1,3488	0,6553	2,2292	0,4788		2
	1,2438	0,6418	1,6218	0,5534	2,241	0,4055		3
	0,3666	0,9023	1,4566	0,6359	2,0721	0,504		4
	0,7748	0,7799	1,3671	0,6068	2,2256	0,4425		5
300	0,1072	0,9674	0,5222	0,8686	1,4541	0,6888		1
	0,2115	0,9356	0,5672	0,8304	1,4016	0,6851		2
	0,1067	0,9676	0,5129	0,872	1,2478	0,7207		3
	0,1586	0,9516	0,5764	0,8458	1,3375	0,6929		4
	0,1596	0,9514	0,5411	0,8512	1,3408	0,697		5
500	0,2015	0,9389	0,3913	0,883	1,0551	0,7377		1
	0,0246	0,9933	0,2957	0,9296	1,0606	0,7877		2
	0,1158	0,9655	0,318	0,9032	0,8185	0,7911		3
	0,1531	0,9526	0,3665	0,8888	1,0683	0,7462		4
	0,0587	0,9817	0,2603	0,9195	0,7704	0,8122		5

1000	0,0391	0,9881	0,1538	0,9617	0,7768	0,8348		1
	0,0711	0,9775	0,1616	0,9577	0,6959	0,8351		2
	0,0286	0,9913	0,1456	0,9634	0,5514	0,8822		3
	0,0758	0,9758	0,1721	0,9503	0,7235	0,8214		4
	0,0457	0,9849	0,1395	0,9607	0,7046	0,8418		5

#### 4.1.4.2. ResNeXt101

Apmokius ResNeXt101 iš gautų rezultatų pateiktų 22 paveiksle, galima pastebėti, kad didinant kiekvienos klasės pavyzdžių kiekį tinklo gebėjimas įsisavinti informaciją didėja. Su 50 kiekvienos klasės pavyzdžių apmokyto modelio tikslumas su testavimo duomenimis svyravo apie 45%, kai su 1000 – svyravo apie 90%.

Atsižvelgiant į kiekvieno modelio duomenis, pateiktus lentelėje 13, galima teigti, kad modeliai, kurie buvo apmokyti su 1000 kiekvienos klasės pavyzdžių buvo tiksliausi. Jų mokymo periodu atliktų tikrinimų tikslumai svyravo apie 97%, o testavimo tikslumas vidutiniškai siekė 90%. Ketvirtu bandymu mokytas tinklas (paryškinta eilutė) pasiekė geriausius rezultatus: tikrinimo tikslumas siekė 98,1%, testavimo metu tikslumas siekė 90,4%.



22 pav. ResNeXt101 modelių priklausomybė nuo duomenų rinkinio dydžio

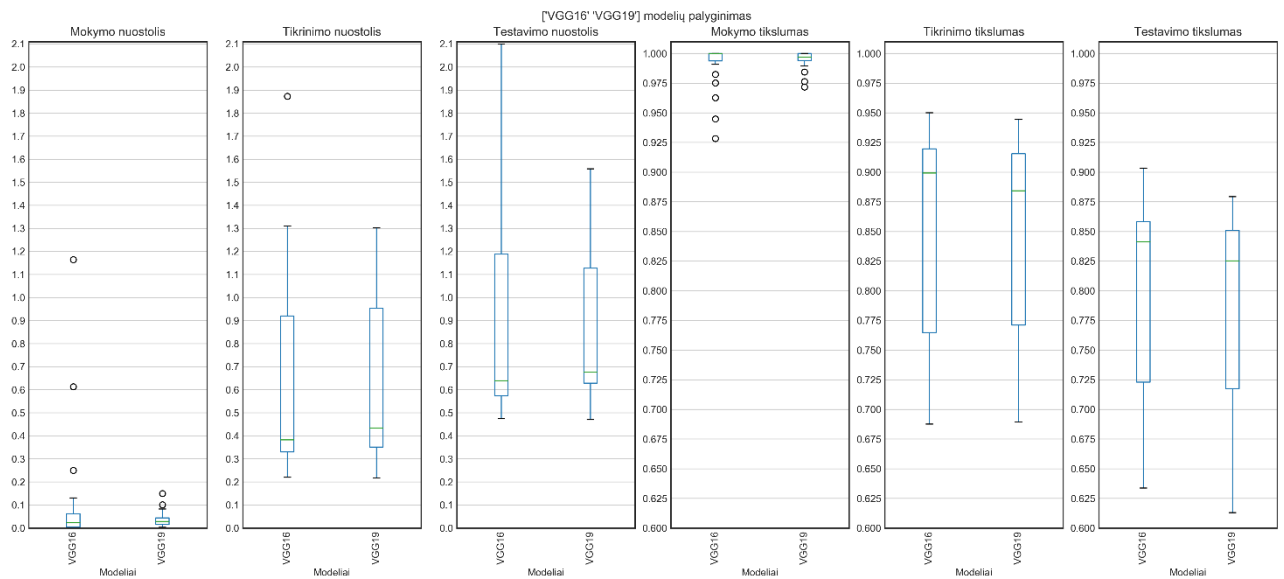
Lentelė 13 ResNeXt101 modelių rezultatai

Klasė turi pavyzdžių	Mokymo nuostolis	Mokymo tikslumas	Tikrinimo nuostolis	Tikrinimo tikslumas	Testavimo nuostolis	Testavimo Tikslumas	Vaizdo dydis	Bandymas
50	1,8934	0,5498	2,1975	0,5141	2,6827	0,3703	32	1
	0,5745	0,8496	1,3039	0,6553	2,5392	0,4666		2
	0,3793	0,9092	1,1075	0,699	2,1267	0,524		3
	1,4994	0,5924	1,6645	0,6068	2,4533	0,4214		4
	1,5824	0,582	1,6395	0,5971	2,5082	0,4203		5

100	0,2173	0,931	0,8225	0,7815	1,5659	0,63	1
	0,7393	0,7855	1,1748	0,6744	1,9208	0,5711	2
	0,1445	0,9629	0,7703	0,7891	1,589	0,6414	3
	0,1956	0,9454	0,9234	0,7521	1,719	0,6151	4
	0,1854	0,9465	0,8819	0,7773	1,4605	0,647	5
300	0,2174	0,9301	0,339	0,9036	0,7283	0,7948	1
	0,2293	0,9265	0,354	0,8927	0,9852	0,7533	2
	0,0868	0,9746	0,2733	0,9203	0,7529	0,8092	3
	0,0801	0,9756	0,2751	0,9229	0,6687	0,8337	4
	0,3211	0,9005	0,4133	0,874	1,0837	0,7329	5
500	0,0778	0,9742	0,1554	0,9522	0,5763	0,8637	1
	0,0769	0,9758	0,199	0,945	0,6308	0,8496	2
	0,0797	0,9741	0,2024	0,942	0,7596	0,8496	3
	0,1266	0,9601	0,1944	0,9374	0,5132	0,8666	4
	0,1029	0,968	0,2243	0,9416	0,6124	0,8437	5
1000	0,0294	0,9905	0,0743	0,9816	0,3602	0,9085	1
	0,0798	0,9737	0,1141	0,9687	0,3946	0,8974	2
	0,0302	0,9901	0,073	0,9798	0,2759	0,927	3
	0,0349	0,9881	0,0598	0,9811	0,3857	0,9044	4
	0,0641	0,9788	0,1199	0,967	0,4372	0,8888	5

#### 4.1.5. VGG

Lyginant VGG16 ir VGG19 rezultatus, pateiktus 23 paveiksle, galima pastebėti, kad mokymo tikslumai buvo panašūs, nors VGG16 turėjo keletą verčių už imties esančių toliau nei likę. Tikrinimo tikslumas abiejų tinklų yra labai panašūs, testavimo tikslumas VGG16 yra truputį geresnis nei VGG19.

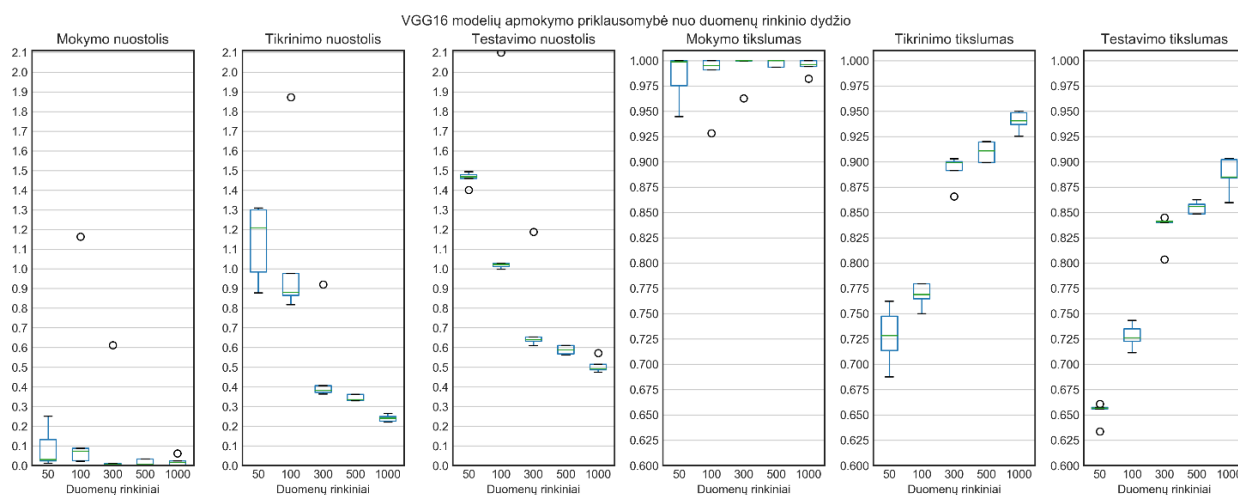


23 pav. VGG16 ir VGG19 modelių palyginimas

### 4.1.5.1. VGG16

Apmokius VGG16 iš gautų rezultatų pateiktų 24 paveiksle, galima pastebėti, kad didinant kiekvienos klasės pavyzdžių kiekį tinklo gebėjimas įsisavinti informaciją didėja. Su 50 kiekvienos klasės pavyzdžių apmokyto modelio tikslumas su testavimo duomenimis svyravo apie 66%, kai su 1000 – svyravo apie 88%.

Atsižvelgiant į kiekvieno modelio duomenis, pateiktus lentelėje 14, galima teigti, kad modeliai, kurie buvo apmokyti su 1000 kiekvienos klasės pavyzdžių buvo tiksliausi. Jų mokymo periodu atliktų tikrinimų tikslumai svyravo apie 93,5%, o testavimo tikslumas vidutiniškai siekė 88%. Trečiu bandymu mokytas tinklas (paryškinta eilutė) pasiekė geriausius rezultatus: tikrinimo tikslumas siekė tik 94,9%, nors antru atveju tikslumas siekė 95, tačiau trečio bandymo testavimo metu tikslumas siekė 90,3% lyginant su antro atveju 90,2%.



24 pav. VGG16 modelių priklausomybė nuo duomenų rinkinio dydžio

Lentelė 14 VGG16 modelių rezultatai

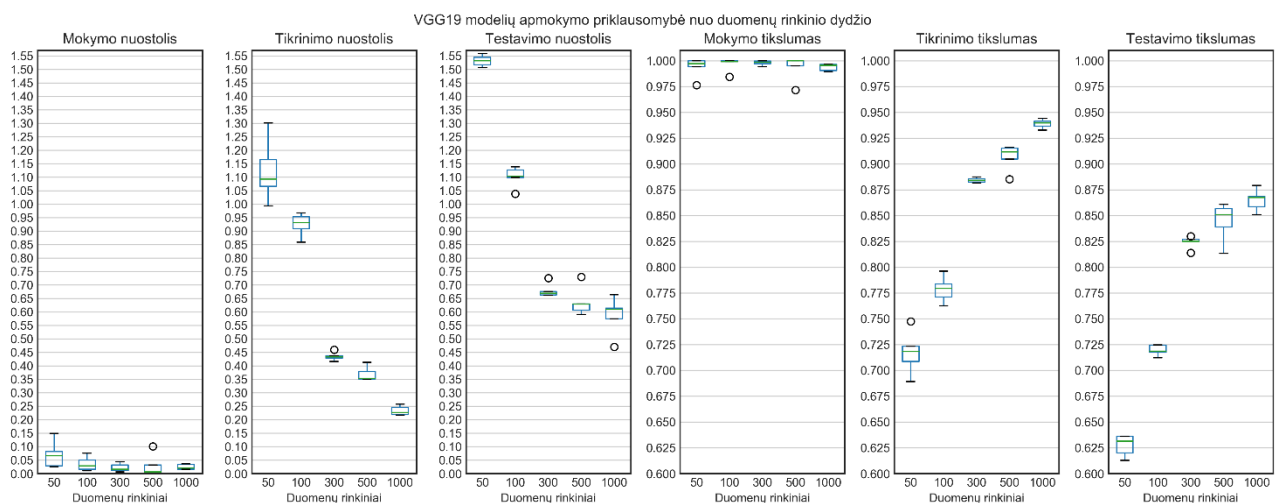
Klasė turi pavyzdžių	Mokymo nuostolis	Mokymo tikslumas	Tikrinimo nuostolis	Tikrinimo tikslumas	Testavimo nuostolis	Testavimo Tikslumas	Vaizdo dydis	Bandymas
50	0,0305	0,999	1,3	0,7136	1,4685	0,6574	32	1
	0,2497	0,9448	1,3099	0,6875	1,4934	0,6337		2
	0,0226	1	0,9832	0,7621	1,4597	0,6607		3
	0,131	0,9752	1,2073	0,7282	1,4016	0,6559		4
	0,0115	1	0,8784	0,7476	1,4783	0,6566		5
100	1,1643	0,9283	1,8726	0,75	2,0993	0,7114	32	1
	0,0875	0,9908	0,976	0,7689	1,0284	0,7229		2
	0,0722	0,995	0,8182	0,7647	1,0113	0,7262		3
	0,024	1	0,8806	0,7794	0,9976	0,7433		4
	0,0217	1	0,8649	0,7794	1,022	0,7351		5
300	0,005	1	0,3629	0,8991	0,6091	0,8451		1

	0,6125	0,9627	0,9201	0,8657	1,1876	0,8037	2
	0,0093	0,9998	0,3712	0,9004	0,6317	0,8411	3
	0,0036	1	0,3821	0,903	0,653	0,8399	4
	0,0042	1	0,4059	0,8914	0,6389	0,8414	5
500	0,0021	1	0,3349	0,92	0,5882	0,8559	1
	0,0329	0,9936	0,3621	0,8991	0,6103	0,8485	2
	0,0329	0,9936	0,3621	0,8991	0,6103	0,8485	3
	0,0032	1	0,3296	0,9196	0,5617	0,8625	4
	0,0048	1	0,3288	0,9108	0,5669	0,8581	5
1000	0,0223	0,9943	0,2487	0,937	0,4863	0,8851	1
	2,28E-04	1	0,2259	0,9501	0,492	0,9022	2
	7,09E-04	1	0,2195	0,9487	0,4741	0,9033	3
	0,0614	0,9823	0,2644	0,9254	0,5722	0,8596	4
	0,0159	0,996	0,2406	0,9406	0,5143	0,884	5

#### 4.1.5.2. VGG19

Apmokius VGG19 iš gautų rezultatų pateiktų 25 paveiksle, galima pastebėti, kad didinant kiekvienos klasės pavyzdžių kiekį tinklo gebėjimas įsisavinti informaciją didėja. Su 50 kiekvienos klasės pavyzdžių apmokyto modelio tikslumas su testavimo duomenimis svyravo apie 66%, kai su 1000 – svyravo apie 88%.

Atsižvelgiant į kiekvieno modelio duomenis, pateiktus lentelėje 15, galima teigti, kad modeliai, kurie buvo apmokyti su 1000 kiekvienos klasės pavyzdžių buvo tiksliausi. Jų mokymo periodu atliktų tikrinimų tikslumai svyravo apie 93,5%, o testavimo tikslumas vidutiniškai siekė 88%. Ketvirtu bandymu mokytas tinklas (paryškinta eilutė) pasiekė geriausius rezultatus: tikrinimo tikslumas siekė tik 93,3% lyginant su trečio bandymo 93,67, tačiau ketvirto bandymo testavimo metu tikslumas siekė 90,3% lyginant su trečio 85,9%.



25 pav. VGG19 modelių priklausomybė nuo duomenų rinkinio dydžio

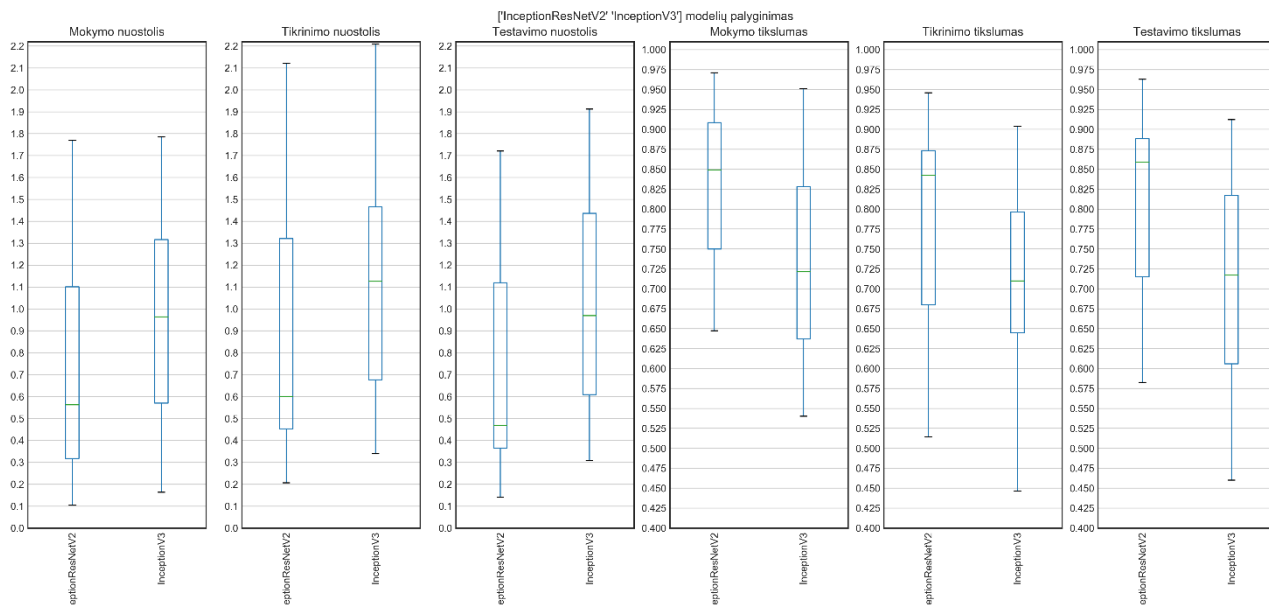
Lentelė 15 VGG19 modelių rezultatai

Klasė turi pavyzdžių	Mokymo nuostolis	Mokymo tikslumas	Tikrinimo nuostolis	Tikrinimo tikslumas	Testavimo nuostolis	Testavimo Tikslumas	Vaizdo dydis	Bandymas
50	0,0256	1	1,1645	0,7233	1,5582	0,6129	32	1
	0,1491	0,9763	1,3014	0,6893	1,5317	0,62		2
	0,0824	0,9941	1,0658	0,7184	1,5064	0,6314		3
	0,0289	1	0,9938	0,7476	1,5165	0,6362		4
	0,0656	0,9971	1,092	0,7087	1,5446	0,6362		5
100	0,0155	1	0,9533	0,7836	1,1032	0,7174		1
	0,011	1	0,858	0,7794	1,0992	0,7248		2
	0,0286	0,9991	0,9081	0,7962	1,038	0,7244		3
	0,0754	0,9844	0,9305	0,7626	1,1266	0,7122		4
	0,0499	0,9991	0,9673	0,771	1,1378	0,7185		5
300	0,0158	0,9985	0,4289	0,8843	0,7254	0,814		1
	0,0323	0,9971	0,4164	0,8875	0,6611	0,8251		2
	0,0439	0,9941	0,4373	0,8856	0,6628	0,8251		3
	0,0065	1	0,4595	0,8824	0,6758	0,827		4
	0,0121	0,9998	0,4334	0,8817	0,6691	0,8299		5
500	0,1007	0,9717	0,4122	0,8854	0,7297	0,8133		1
	0,0059	0,9999	0,3505	0,9116	0,5904	0,8607		2
	0,0025	1	0,3513	0,9154	0,6058	0,8507		3
	0,0316	0,995	0,3536	0,9048	0,6294	0,8388		4
	0,0025	1	0,3788	0,9162	0,6285	0,8566		5
1000	0,0159	0,9964	0,2578	0,9398	0,6134	0,8674		1
	1,63E-02	0,9967	0,2171	0,9415	0,61	0,8685		2
	3,41E-02	0,9906	0,2462	0,9367	0,5749	0,8585		3
	0,0361	0,9894	0,227	0,9329	0,4706	0,8792		4
	0,0202	0,9955	0,2194	0,9443	0,6638	0,8507		5

#### 4.1.6. Inception

Lyginant Inception modelius, pateiktus 26 paveiksle, galima pastebėti, kad InceptionResNetV2 tikslumai yra didesni ir InceptionV3 imtis turi mažesnius tikslumus.



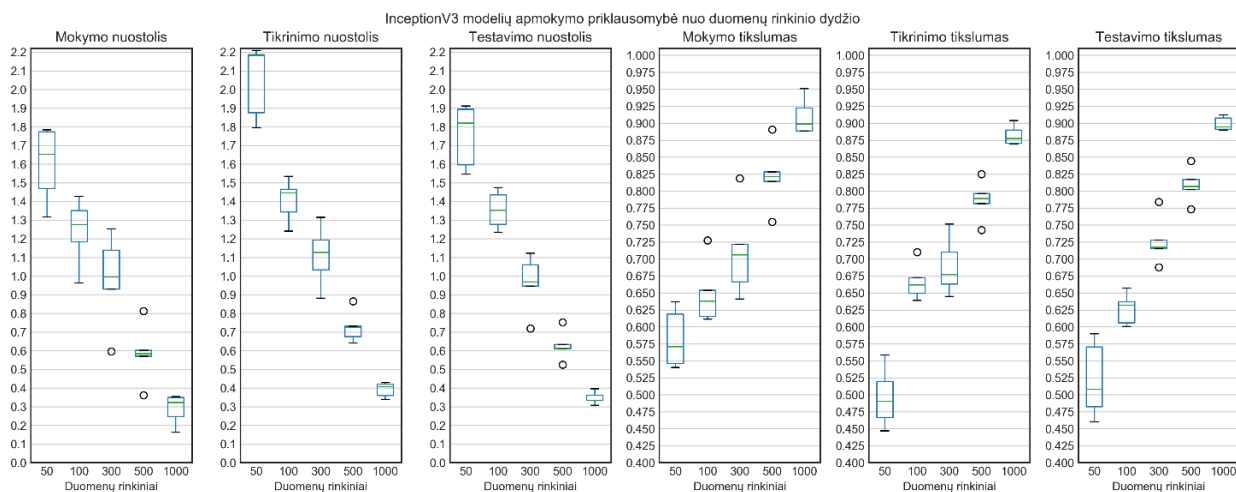


26 pav. InceptionResNetV2 ir InceptionV3 modelių palyginimas

#### 4.1.6.1. InceptionV3

Apmokius InceptionV3 iš gautų rezultatų pateiktų 27 paveiksle, galima pastebėti, kad didinant kiekvienos klasės pavyzdžių kiekį tinklo gebėjimas įsisavinti informaciją didėja. Su 50 kiekvienos klasės pavyzdžių apmokyto modelio tikslumas su testavimo duomenimis svyravo apie 51%, kai su 1000 – svyravo apie 90%.

Atsižvelgiant į kiekvieno modelio duomenis, pateiktus lentelėje 16, galima teigti, kad modeliai, kurie buvo apmokyti su 1000 kiekvienos klasės pavyzdžių buvo tiksliausi. Jų mokymo periodu atliktų tikrinimų tikslumai svyravo apie 88%, o testavimo tikslumas vidutiniškai siekė 90%. Ketvirtu bandymu mokytas tinklas (paryškinta eilutė) pasiekė geriausius rezultatus: tikrinimo tikslumas siekė 90,3%, testavimo metu tikslumas siekė 91,2%. Šiuo atveju tai pirmasis tinklas, kurio testavimo tikslumas yra geresnis nei tikrinimo tikslumas.



27 pav. InceptionV3 modelių priklausomybė nuo duomenų rinkinio dydžio

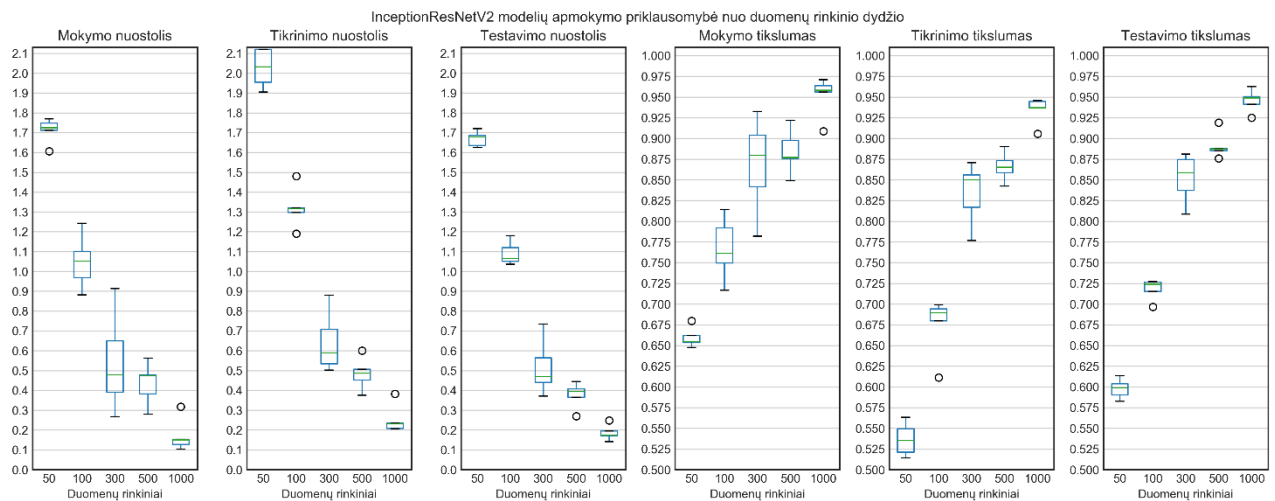
Lentelė 16 InceptionV3 modelių rezultatai

Klasė turi pavyzdžių	Mokymo nuostolis	Mokymo tikslumas	Tikrinimo nuostolis	Tikrinimo tikslumas	Testavimo nuostolis	Testavimo Tikslumas	Vaizdo dydis	Bandymas
50	1,3164	0,637	1,7938	0,5194	1,5463	0,5896	75	1
	1,7842	0,5401	2,1863	0,4466	1,911	0,46		2
	1,7733	0,5459	2,2095	0,4903	1,8933	0,4822		3
	1,6534	0,5707	2,1816	0,466	1,8195	0,5077		4
	1,4698	0,6186	1,8761	0,5583	1,5956	0,5703		5
100	0,9631	0,7274	1,2406	0,7101	1,235	0,657		1
	1,1841	0,6537	1,4461	0,6618	1,3527	0,6318		2
	1,4273	0,6111	1,465	0,6492	1,4355	0,6059		3
	1,2751	0,6376	1,3428	0,6723	1,2766	0,637		4
	1,3499	0,6154	1,5339	0,6387	1,4734	0,6003		5
300	0,5961	0,8186	0,8811	0,7513	0,7188	0,784		1
	0,9942	0,706	1,1261	0,6769	0,9692	0,7148		2
	0,9302	0,7214	1,0337	0,7098	0,9454	0,7277		3
	1,1381	0,6663	1,192	0,6629	1,0598	0,717		4
	1,2523	0,6409	1,3146	0,6448	1,1227	0,6877		5
500	0,3611	0,8904	0,6403	0,8247	0,525	0,8444		1
	0,8122	0,7545	0,8655	0,7426	0,7514	0,7733		2
	0,602	0,8141	0,7311	0,7889	0,6126	0,8066		3
	0,5841	0,8213	0,7243	0,7815	0,6325	0,8022		4
	0,57	0,8281	0,676	0,7963	0,6092	0,817		5
1000	0,3485	0,8886	0,4285	0,8703	0,3608	0,8911		1
	0,3224	0,8987	0,4081	0,8775	0,3318	0,8944		2
	0,2468	0,9224	0,3601	0,8898	0,3336	0,9074		3
	0,1626	0,9508	0,3385	0,9036	0,3069	0,9122		4
	0,3529	0,8886	0,4192	0,8689	0,3952	0,8896		5

#### 4.1.6.1. InceptionResNetV2

Apmokius InceptionResNetV2 iš gautų rezultatų pateiktų 28 paveiksle, galima pastebėti, kad didinant kiekvienos klasės pavyzdžių kiekį tinklo gebėjimas įsisavinti informaciją didėja. Su 50 kiekvienos klasės pavyzdžių apmokyto modelio tikslumas su testavimo duomenimis svyravo apie 60%, kai su 1000 – svyravo apie 95%.

Atsižvelgiant į kiekvieno modelio duomenis, pateiktus lentelėje 17, galima teigti, kad modeliai, kurie buvo apmokyti su 1000 kiekvienos klasės pavyzdžių buvo tiksliausi. Jų mokymo periodu atliktų tikrinimų tikslumai svyravo apie 93%, o testavimo tikslumas vidutiniškai siekė 95%. Penktu bandymu mokytas tinklas (paryškinta eilutė) pasiekė geriausius rezultatus: tikrinimo tikslumas siekė 94,6%, testavimo metu tikslumas siekė 96,3%.



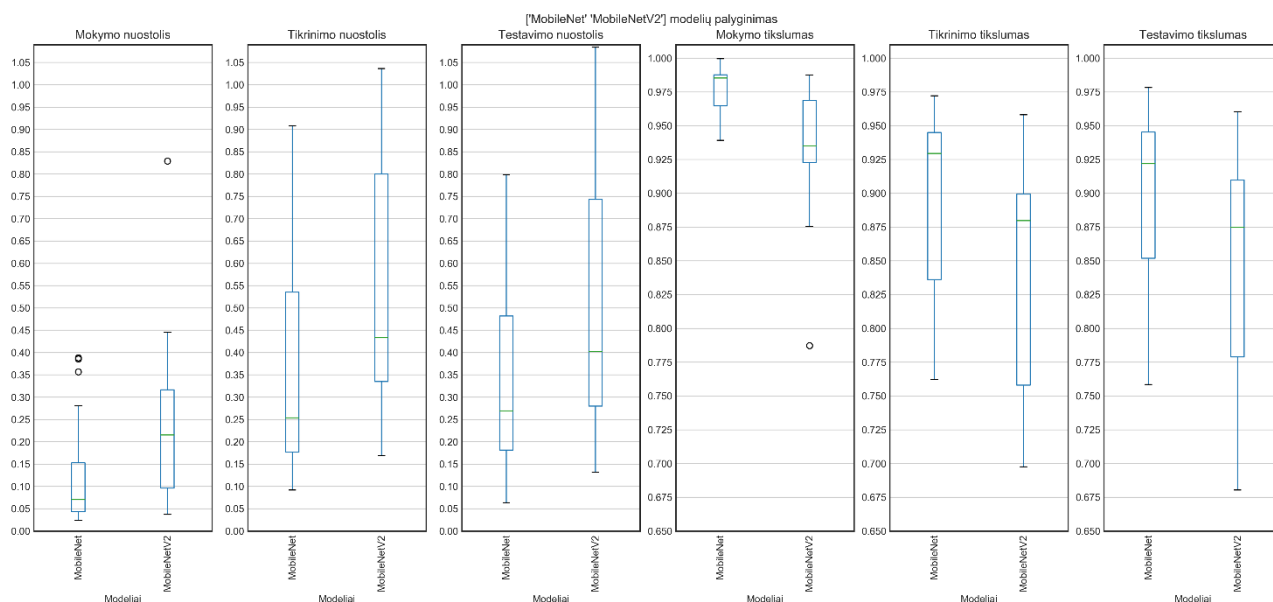
28 pav. InceptionResNetV2 modelių priklausomybė nuo duomenų rinkinio dydžio

Lentelė 17 InceptionResNetV2 modelių rezultatai

Klasė turi pavyzdžių	Mokymo nuostolis	Mokymo tikslumas	Tikrinimo nuostolis	Tikrinimo tikslumas	Testavimo nuostolis	Testavimo Tikslumas	Vaizdo dydis	Bandymas
50	1,606	0,6798	1,905	0,5211	1,6244	0,6037	75	1
	1,7121	0,662	2,1205	0,5352	1,6856	0,5988		2
	1,7248	0,6543	2,0323	0,5634	1,6351	0,6133		3
	1,7691	0,6475	2,1193	0,5141	1,7204	0,5825		4
	1,7473	0,6535	1,9543	0,5493	1,6784	0,5903		5
100	1,2423	0,7166	1,4805	0,6113	1,1808	0,6966		1
	0,8812	0,814	1,3155	0,6796	1,0363	0,7262		2
	1,052	0,7611	1,2977	0,6942	1,1197	0,7151		3
	0,9683	0,792	1,1902	0,699	1,0656	0,727		4
	1,1007	0,7496	1,321	0,6893	1,0507	0,724		5
300	0,6491	0,8419	0,709	0,817	0,5641	0,8374		1
	0,9132	0,782	0,8797	0,7768	0,7347	0,8085		2
	0,267	0,9325	0,5024	0,8706	0,3702	0,8811		3
	0,477	0,8798	0,5887	0,8499	0,4684	0,8588		4
	0,3912	0,9038	0,5343	0,8559	0,4399	0,8744		5
500	0,4778	0,8754	0,5068	0,8585	0,3941	0,8851		1
	0,3826	0,8978	0,4518	0,8733	0,365	0,8874		2
	0,4727	0,8768	0,487	0,8651	0,4063	0,8881		3
	0,5623	0,849	0,6002	0,8425	0,445	0,8759		4
	0,2794	0,922	0,3751	0,89	0,2702	0,9192		5
1000	0,1266	0,9637	0,2066	0,9446	0,1739	0,9507		1
	0,1534	0,956	0,234	0,9366	0,1723	0,9488		2
	0,1499	0,9582	0,231	0,937	0,1955	0,9414		3
	0,3175	0,9086	0,3827	0,9055	0,2486	0,9251		4
	0,104	0,971	0,2076	0,9458	0,1412	0,9625		5

### 4.1.7. MobileNet

Peržvelgus gautus MobileNet rezultatus, pateikiamus 29 paveiksle, galima pastebėti, kad abu tinklai gerai išmoksta, kadangi mokymosi tikslumai MobileNet viršija 94%, o MobileNetV2 svyruoja apie 93%. Tačiau tikrinimo ir testavimo tikslumai yra labai plačiose imtyse: nuo 75% iki 97% - MobileNet ir nuo 70% iki 96% - MobileNetV2.

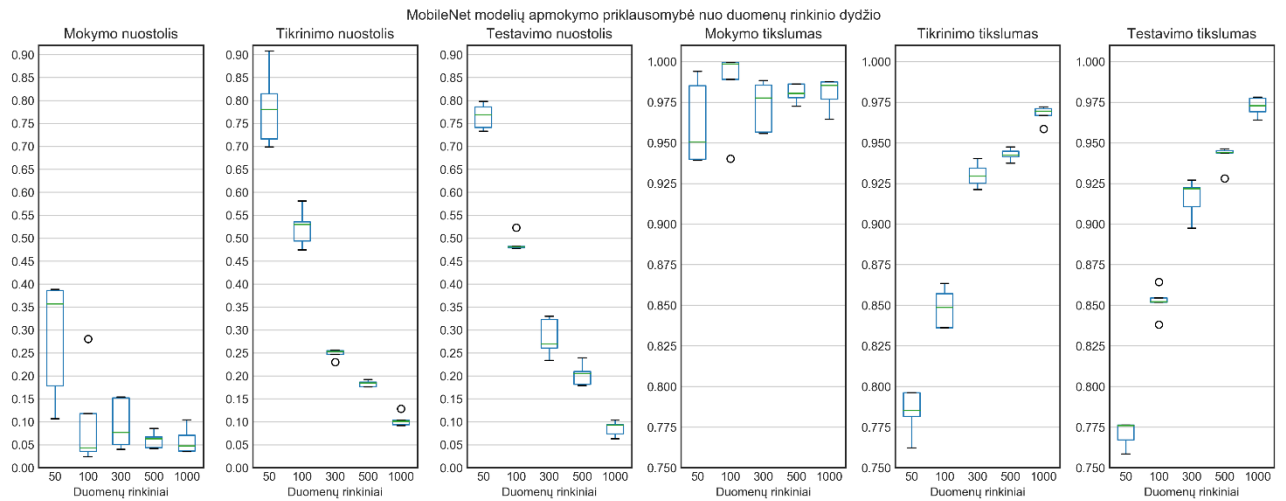


29 pav. MobileNetV2 ir MobileNet modelių palyginimas

#### 4.1.7.1. MobileNet

Apmokius MobileNet iš gautų rezultatų pateiktą 30 paveiksle, galima pastebėti, kad didinant kiekvienos klasės pavyzdžių kiekį tinklo gebėjimas įsisavinti informaciją didėja. Su 50 kiekvienos klasės pavyzdžių apmokyto modelio tikslumas su testavimo duomenimis svyravo apie 77,5%, kai su 1000 – svyravo apie 97,5%.

Atsižvelgiant į kiekvieno modelio duomenis, pateiktus lentelėje 18, galima teigti, kad modeliai, kurie buvo apmokyti su 1000 kiekvienos klasės pavyzdžių buvo tiksliausi. Jų mokymo periodu atliktų tikrinimų tikslumai svyravo apie 96%, o testavimo tikslumas vidutiniškai siekė 97,5%. Ketvirtu bandymu mokytas tinklas (paryškinta eilutė) pasiekė geriausius rezultatus: tikrinimo tikslumas siekė 97,2%, testavimo metu tikslumas siekė 97,8%.



30 pav. MobileNet modelių priklausomybė nuo duomenų rinkinio dydžio

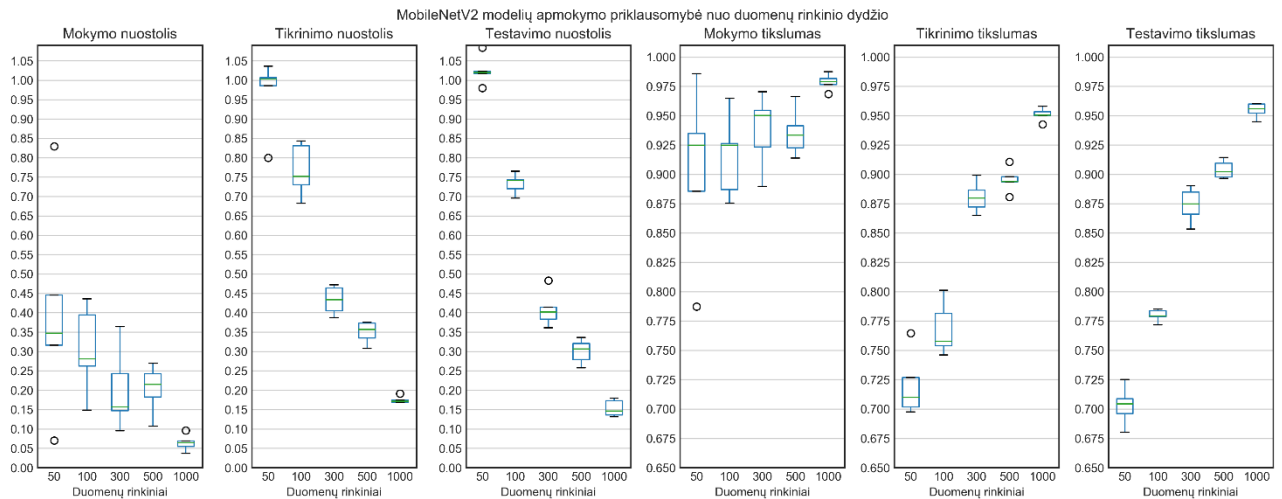
Lentelė 18 MobileNet modelių rezultatai

Klasė turi pavyzdžių	Mokymo nuostolis	Mokymo tikslumas	Tikrinimo nuostolis	Tikrinimo tikslumas	Testavimo nuostolis	Testavimo Tikslumas	Vaizdo dydis	Bandymas
50	0,3568	0,9505	0,9073	0,7621	0,798	0,7585	128	1
	0,3857	0,9392	0,7802	0,7816	0,7863	0,767		2
	0,1063	0,994	0,6988	0,7852	0,7329	0,7762		3
	0,1783	0,9852	0,7164	0,7961	0,7412	0,7762		4
	0,3884	0,9399	0,8149	0,7961	0,7684	0,7759		5
100	0,1181	0,989	0,5353	0,8487	0,481	0,8518		1
	0,2806	0,9403	0,5812	0,8361	0,5228	0,8381		2
	0,0237	0,9995	0,4742	0,8634	0,479	0,8644		3
	0,0356	0,9986	0,5298	0,8361	0,4773	0,8522		4
	0,0432	0,9995	0,4937	0,8571	0,4817	0,8544		5
300	0,0398	0,9884	0,2298	0,9402	0,3231	0,9107		1
	0,0503	0,9855	0,2551	0,9295	0,2607	0,9222		2
	0,1523	0,9567	0,2469	0,9251	0,269	0,9218		3
	0,0767	0,9776	0,2533	0,9345	0,2337	0,927		4
	0,1539	0,9558	0,2561	0,9213	0,3298	0,8974		5
500	0,0627	0,9804	0,1756	0,9449	0,1815	0,9451		1
	0,0438	0,9862	0,1764	0,9475	0,2097	0,9462		2
	0,0851	0,9725	0,1844	0,9415	0,1786	0,944		3
	0,0667	0,9778	0,1862	0,9374	0,2395	0,9281		4
	0,042	0,9862	0,1922	0,9423	0,2048	0,9437		5
1000	0,0703	0,9768	0,1033	0,9668	0,0936	0,9692		1
	0,0471	0,9854	0,1009	0,9694	0,0632	0,9774		2
	0,1041	0,9646	0,1282	0,9585	0,1035	0,964		3
	0,0346	0,9877	0,0938	0,9722	0,0735	0,9781		4
	0,036	0,9875	0,0916	0,9711	0,0931	0,9729		5

#### 4.1.7.2. MobileNetV2

Apmokius MobileNetV2 iš gautų rezultatų pateiktų 31 paveiksle, galima pastebėti, kad didinant kiekvienos klasės pavyzdžių kiekį tinklo gebėjimas įsisavinti informaciją didėja. Su 50 kiekvienos klasės pavyzdžių apmokyto modelio tikslumas su testavimo duomenimis svyravo apie 71%, kai su 1000 – svyravo apie 95,5%.

Atsižvelgiant į kiekvieno modelio duomenis, pateiktus lentelėje 19, galima teigti, kad modeliai, kurie buvo apmokyti su 1000 kiekvienos klasės pavyzdžių buvo tiksliausi. Jų mokymo periodu atliktų tikrinimų tikslumai svyravo apie 95%, o testavimo tikslumas vidutiniškai siekė 95,5%. Pirmu bandymu mokytas tinklas (paryškinta eilutė) pasiekė geriausius rezultatus: tikrinimo tikslumas siekė 95,8%, testavimo metu tikslumas siekė 96%.



31 pav. MobileNetV2 modelių priklausomybė nuo duomenų rinkinio dydžio

Lentelė 19 MobileNetV2 modelių rezultatai

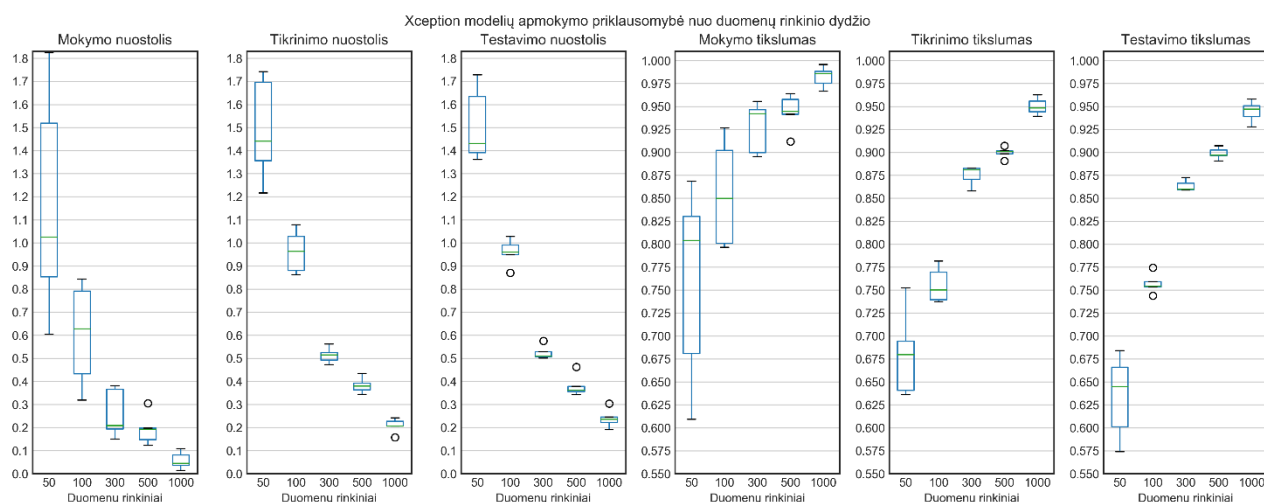
Klasė turi pavyzdžių	Mokymo nuostolis	Mokymo tikslumas	Tikrinimo nuostolis	Tikrinimo tikslumas	Testavimo nuostolis	Testavimo Tikslumas	Vaizdo dydis	Bandymas
50	0,3164	0,935	0,9859	0,7101	0,9802	0,7044	128	1
	0,07	0,9858	0,7995	0,7647	1,0229	0,7251		2
	0,3469	0,9249	1,003	0,7269	1,0174	0,7088		3
	0,8291	0,7872	1,0361	0,6975	1,0838	0,6803		4
	0,4452	0,8857	1,0066	0,7017	1,0197	0,6962		5
100	0,3946	0,8871	0,7516	0,7539	0,7201	0,7837		1
	0,4356	0,8755	0,8308	0,7461	0,7417	0,7718		2
	0,2618	0,9263	0,7301	0,7815	0,6957	0,7851		3
	0,2809	0,9249	0,8431	0,7579	0,7431	0,7788		4
	0,1486	0,965	0,6824	0,8012	0,7647	0,7792		5
300	0,2427	0,9234	0,472	0,8652	0,4146	0,8662	1	
	0,1566	0,9503	0,4332	0,8797	0,402	0,8748	2	

	0,3638	0,8897	0,4638	0,8722	0,4829	0,8533	3
	0,147	0,9547	0,4044	0,8866	0,3831	0,8848	4
	0,0951	0,9706	0,3867	0,8992	0,3613	0,8903	5
500	0,1072	0,9663	0,3077	0,9108	0,258	0,9144	1
	0,2694	0,9139	0,3727	0,8808	0,3357	0,8977	2
	0,1823	0,9414	0,357	0,8936	0,2795	0,9096	3
	0,2421	0,9227	0,3348	0,8981	0,3062	0,9022	4
	0,2151	0,9334	0,3749	0,8939	0,3206	0,8966	5
1000	0,0372	0,9875	0,1688	0,9581	0,1361	0,9603	1
	0,0546	0,9816	0,1685	0,9534	0,1318	0,9599	2
	0,0643	0,979	0,1712	0,9504	0,1731	0,9522	3
	0,0687	0,9767	0,1744	0,9501	0,1464	0,9559	4
	0,0959	0,9686	0,1913	0,9426	0,1791	0,9448	5

#### 4.1.8. Xception

Apmokius Xception iš gautų rezultatų pateiktų 31 paveiksle, galima pastebėti, kad didinant kiekvienos klasės pavyzdžių kiekį tinklo gebėjimas įsisavinti informaciją didėja. Su 50 kiekvienos klasės pavyzdžių apmokyto modelio tikslumas su testavimo duomenimis svyravo apie 65%, kai su 1000 – svyravo apie 95%.

Atsižvelgiant į kiekvieno modelio duomenis, pateiktus lentelėje 20, galima teigti, kad modeliai, kurie buvo apmokyti su 1000 kiekvienos klasės pavyzdžių buvo tiksliausi. Jų mokymo periodu atliktų tikrinimų tikslumai svyravo apie 95%, o testavimo tikslumas vidutiniškai siekė 95%. Ketvirtu bandymu mokytas tinklas (paryškinta eilutė) pasiekė geriausius rezultatus: tikrinimo tikslumas siekė tik 95,6%, nors penkto bandymo tikrinimo tikslumas siekė 96,3%, tačiau ketvirto bandymo testavimo tikslumas siekė 95,8% lyginant su penkto 95%.



32 pav. Xception modelių priklausomybė nuo duomenų rinkinio dydžio

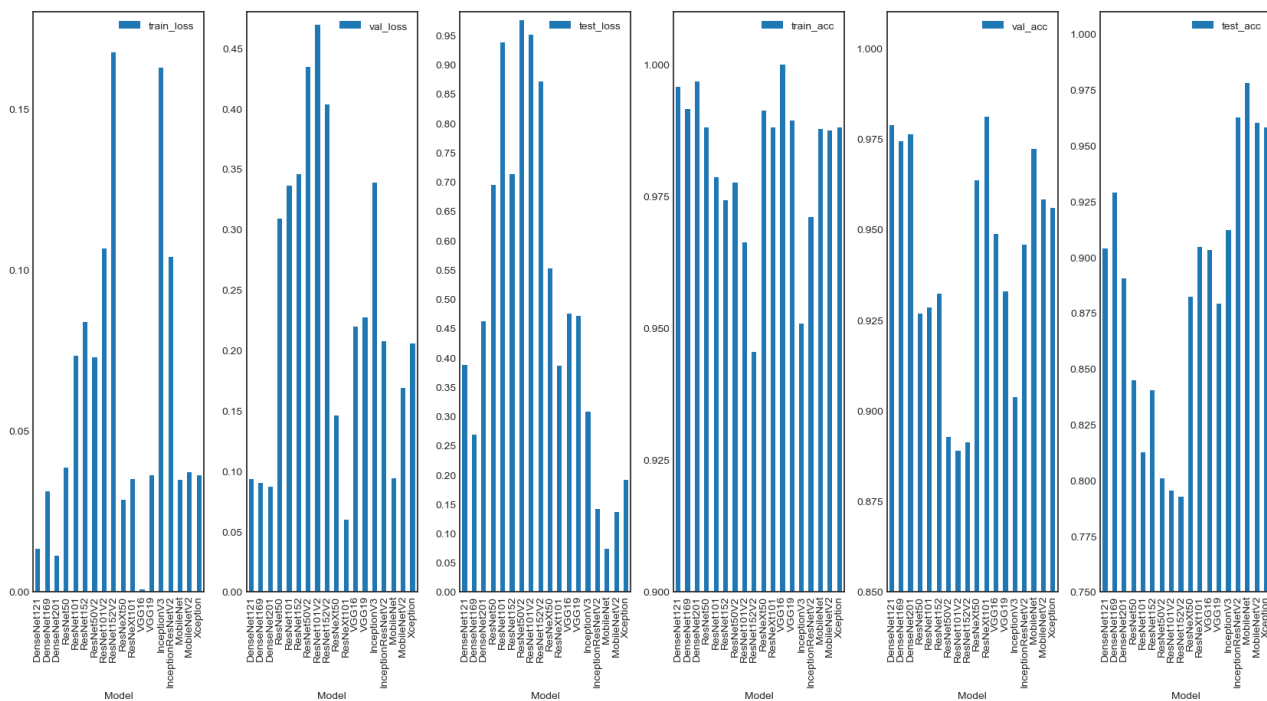
Lentelė 20 Xception modelių rezultatai

Klasė turi pavyzdžių	Mokymo nuostolis	Mokymo tikslumas	Tikrinimo nuostolis	Tikrinimo tikslumas	Testavimo nuostolis	Testavimo Tikslumas	Vaizdo dydis	Bandymas
50	1,5182	0,681	1,6963	0,6408	1,6335	0,6011	71	1
	0,6038	0,8686	1,2164	0,7524	1,3615	0,684		2
	1,025	0,8039	1,3561	0,6942	1,4303	0,6448		3
	0,8533	0,8301	1,4409	0,6796	1,3905	0,6659		4
	1,8238	0,6091	1,7423	0,6359	1,7283	0,574		5
100	0,843	0,7966	1,0276	0,75	0,9597	0,7533		1
	0,3186	0,9265	0,8615	0,7815	0,8706	0,7744		2
	0,7911	0,801	1,0782	0,7395	0,9907	0,754		3
	0,6277	0,8499	0,9628	0,7374	1,028	0,7437		4
	0,4333	0,9021	0,8798	0,7695	0,9482	0,7592		5
300	0,2083	0,9422	0,4921	0,883	0,5064	0,8599		1
	0,3663	0,8996	0,5246	0,8708	0,5101	0,8588		2
	0,1496	0,9556	0,5151	0,8824	0,5287	0,8725		3
	0,1935	0,9465	0,4725	0,8811	0,5004	0,8662		4
	0,3806	0,8954	0,5616	0,858	0,5749	0,8592		5
500	0,1226	0,964	0,3636	0,9071	0,342	0,907		1
	0,197	0,9415	0,3795	0,8983	0,3782	0,8966		2
	0,1479	0,9576	0,344	0,9017	0,3561	0,9025		3
	0,1923	0,9447	0,3923	0,901	0,3622	0,897		4
	0,3051	0,9117	0,4343	0,8907	0,4624	0,8907		5
1000	0,0807	0,9753	0,2271	0,9443	0,2352	0,9388		1
	1,08E-01	0,9668	0,2416	0,9391	0,3042	0,9277		2
	4,50E-02	0,9857	0,2058	0,9483	0,2225	0,947		3
	0,0362	0,9881	0,2053	0,9558	0,1911	0,9581		4
	0,0136	0,9956	0,1578	0,9627	0,2454	0,9507		5

#### 4.2. Bendra konvoliucinių neuroninių tinklų rezultatų analizė

Atsižvelgiant į kiekvienos architektūros geriausio modelio gautus rezultatus, pateiktus lentelėje 21 ir 33 paveiksle, geriausi rezultatai buvo MobileNet architektūrų, blogiausius tikslumus pateikė ResNetV2 architektūros. Originalios ResNet architektūros parodė geresnius rezultatus, mokymo tikslumas buvo panašus, siekė 97%, tikrinimo tikslumas svyravo apie 93%, o testavimo tikslumai svyravo apie 83%. Kitos architektūros pateikė vidutiniškus rezultatus, pvz.: DenseNet pateikė labai aukštus mokymosi tikslumus, svyruojančius apie 99,5%, jų tikrinimo tikslumai buvo irgi ganėtinai aukšti apie 97%, tačiau testavimo tikslumai vidutiniškai tesiekė 92%. VGG architektūros mokymo metu turėjo labai gerus tikslumus siekiančius 98% ir 95%, tačiau testavimo duomenims tikslumas siekė tik 90% ir 88%.



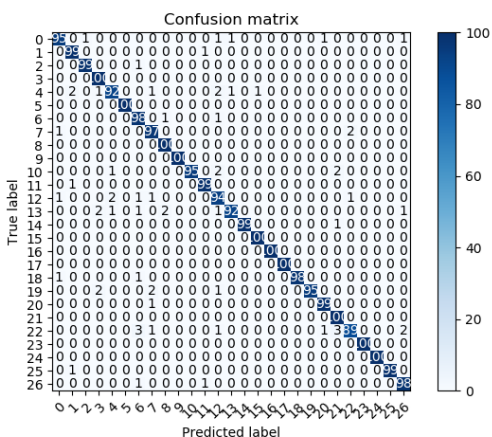


33 pav. Geriausių modelių palyginimas

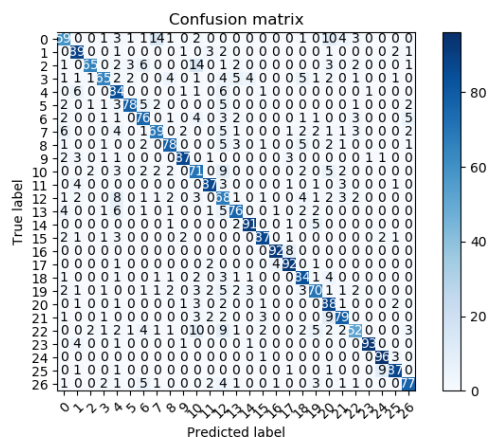
Lentelė 21 Visų tinklų, geriausiai apmokytų, modelių palyginimas

Modelis	Mokymo nuostolis	Mokymo tikslumas	Tikrinimo nuostolis	Tikrinimo tikslumas	Testavimo nuostolis	Testavimo tikslumas	Vaizdo dydis
DenseNet121	0,0134	0,9957	0,0934	0,9786	0,3864	0,904	32
DenseNet169	0,0311	0,9915	0,0901	0,9742	0,268	0,9288	32
DenseNet201	0,0111	0,9967	0,0866	0,9761	0,4611	0,8903	32
ResNet50	0,0384	0,988	0,309	0,9268	0,6946	0,8448	32
ResNet101	0,0732	0,9786	0,336	0,9285	0,9379	0,8125	32
ResNet152	0,0838	0,9743	0,3455	0,9323	0,7129	0,8403	32
ResNet50V2	0,0728	0,9776	0,434	0,8926	0,9754	0,8007	32
ResNet101V2	0,1066	0,9662	0,4689	0,8888	0,95	0,7955	32
ResNet152V2	0,1673	0,9455	0,4032	0,8913	0,871	0,7925	32
ResNeXt50	0,0286	0,9913	0,1456	0,9634	0,5514	0,8822	32
ResNeXt101	0,0349	0,9881	0,0598	0,9811	0,3857	0,9044	32
VGG16	7,09E-04	1	0,2195	0,9487	0,4741	0,9033	32
VGG19	0,0361	0,9894	0,227	0,9329	0,4706	0,8792	32
InceptionV3	0,1626	0,9508	0,3385	0,9036	0,3069	0,9122	75
InceptionResNetV2	0,104	0,971	0,2076	0,9458	0,1412	0,9625	75
MobileNet	0,0346	0,9877	0,0938	0,9722	0,0735	0,9781	128
MobileNetV2	0,0372	0,9875	0,1688	0,9581	0,1361	0,9603	128
Xception	0,0362	0,9881	0,2053	0,9558	0,1911	0,9581	71

Geriausius rezultatus šiuo atveju pateikė MobileNet tinklas, kurios mokymo tikslumas siekė 98%, tikrinimo ir testavimo siekė 97%. Šis tinklas buvo vienas iš penkių, kurių testavimo tikslumai buvo geresni nei tikrinimo, tai lemia gerą naujai apmokyto tinklo praktinį pritaikymą, kadangi jis gerai apibendrina dar nematytus duomenis, taip pat jis turi žemą nuostolio vertę, kuri lemia gerą klasių atpažinimą, kaip galima pastebėti maišos matricoje pateiktoje 34 paveiksle, visos klasės turi tik keletą neatpažintų paveikslėlių. Prasčiausi buvo ResNet152V2 rezultatai, nors mokymo metu pasiekiamas aukštas beveik 96% siekiantis tikslumas, tikrinimo metu gaunamas 89% tikslumas, tačiau testavimo tikslumas nesiekia nei 80%. Maišos matricoje, pateiktoje 35 paveiksle galima pastebėti, kad tinklas dažnai neatpažįsta klasių ir jas interpretuoja kaip kitas.

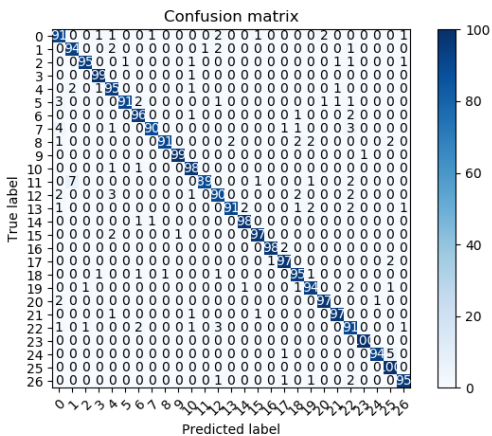


34 pav. MobileNet maišos matrica

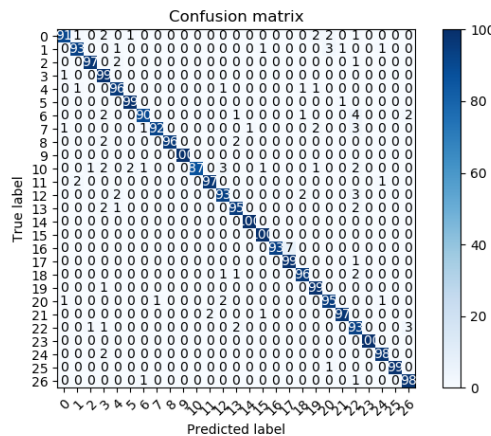


35 pav. ResNet152V2 maišos matrica

Kaip alternatyvas MobileNet galima paminėti Xception, maišos matrica pateikta 36 paveiksle, ir InceptionResNetV2, maišos matrica pateikta 37 paveiksle. Jų visų mokymo tikslumas svyravo netoli 98%, tikrinimo tikslumas siekė 95% vidutiniškai, o testavimo tikslumas buvo šiek tiek didesnis lyginant su tikrinimo tikslumu. Klasių spėjime jos pateikia panašius, galbūt net geresnius rezultatus nei MobileNet.



36 pav. Xception maišos matrica



37 pav. InceptionResNetV2 maišos matrica

## 5. Išvados

Iš šiame darbe ištirtų aštuoniolikos konvoliucinių neuroninių tinklų architektūrų, dauguma pateikė vidutiniškus rezultatus. Šiame tyrime geriausia architektūra žiedadulkių klasifikavimui yra MobileNet su aukščiausiu tikslumu identifikuojant žiedadulkių klases. Kaip alternatyvias architektūras galima taikyti MobileNetV2, Xception ir InceptionResNetV2. Atsižvelgiant į tai, galima teigti, kad tinklų apmokymas su didesniais paveikslėliais yra efektyvesnis, taip lengviau randama bei įsisavina požymių informacija, lyginant su tinklais, kurie apmokomi su mažesnių dimensijų paveikslėliais.

Atsakant į klausimą, kokį duomenų rinkinio dydį reikėtų naudoti: visos architektūros pateikė geriausius rezultatus panaudojus didžiausią duomenų rinkinį, kurį sudarė po 1000 kiekvienos žiedadulkės rūšių pavyzdžių. Skirtingos tinklų architektūros apsimokydavo skirtingai su duomenų rinkiniais, vienos architektūros pasiekdavo tam tikrą tikslumą su mažesniais duomenų rinkiniais, kitoms prireikdavo didesnių pasiekti tokį pat tikslumą.

Šio tyrimo rezultatai parodė, kad MobileNet yra tinkamiausias tinklas žiedadulkių rūšių atpažinimui, tačiau tyrimas buvo atliktas su nevisomis konvoliucinių neuroninių tinklų architektūromis, pvz.: NASNet as SENet. Taip pat reikėtų paminėti, kad tinklų mokyme buvo pasitelktas mokėjimo perkėlimas, kas gan smarkiai padėjo tinklams apibendrinti žiedadulkių požymius, įdomu ar tinklai taip pat gerai apsimokyti juos apmokant su tais pačiais duomenų rinkiniais be pradinių svorių. Galiausiai šiame tyrime pastebėtas momentas, kad žiedadulkės vaizdo dydis lemia tinklų apmokymą, kelia klausimą ar tinklus apmokius su didesniais paveikslėliais rezultatai liktų tokie pat ar pagerėtų.

## LITERATŪRA

- Agarap, A. F. (2019 m. Sausio 7 d.). Deep Learning using Rectified Linear Units (ReLU). Nuskaityta iš <https://arxiv.org/pdf/1803.08375.pdf>
- Buda, M., Maki, A., & Mazurowski, M. A. (2018 m. Spalis 13 d.). A systematic study of the class imbalance problem in convolutional neural networks. Paimta 2019 m. Balandis 19 d. iš <https://arxiv.org/pdf/1710.05381.pdf>
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002 m. Birželis 2 d.). SMOTE: Synthetic Minority Over-sampling Technique. Paimta 2019 m. Balandis 19 d. iš <https://arxiv.org/pdf/1106.1813.pdf>
- Google. (2019 m. Sausis 28 d.). *Google*. Nuskaityta iš Advanced Guide to Inception v3 on Cloud TPU: <https://cloud.google.com/tpu/docs/inception-v3-advanced>
- GradientBased Learning Applied to Document Recognition. (1998 m. Lapkritis).
- Halbritter, H., Ulrich, S., Grímsson, F., Weber, M., Zetter, R., Hesse, M., . . . Frosch-Radivo, A. (2018). *Illustrated Pollen Terminology Second Edition*. Springer One.
- He, K., Zhang, X., Ren, S., & Sun, J. (2015 m. Gruodis 10 d.). Deep Residual Learning for Image Recognition. Nuskaityta iš <https://arxiv.org/abs/1512.03385>
- Huang, G., Liu, Z., & Maaten, L. v. (2018 m. Sausis 28 d.). Densely Connected Convolutional Networks. Nuskaityta iš <https://arxiv.org/abs/1608.06993>
- Yuheng, S., & Hao, Y. (2017 m. Liepa 7 d.). *Image Segmentation Algorithms Overview*. Paimta 2019 m. Balandis 20 d. iš <https://arxiv.org/https://arxiv.org/ftp/arxiv/papers/1707/1707.02051.pdf>
- Johnson, J., & Karpathy, A. (2015). *Image Classification*. Paimta 2019 m. Balandis 9 d. iš CS231n: Convolutional Neural Networks for Visual Recognition: <http://cs231n.github.io/>
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012 m. Sausis). ImageNet Classification with Deep Convolutional.
- Liu, D., Soran, B., Petrie, G., & Shapiro, L. (2019 m. Balandis 10 d.). A Review of Computer Vision. Nuskaityta iš <https://courses.cs.washington.edu>
- Rosebrock, A. (2017 m. Kovas 20 d.). *pyimagesearch*. Nuskaityta iš ImageNet: VGGNet, ResNet, Inception, and Xception with Keras: <https://www.pyimagesearch.com/2017/03/20/imagenet-vggnet-resnet-inception-xception-keras/>
- Simonyan, K., & Zisserman, A. (2015 m. Balandis 10 d.). VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION. Nuskaityta iš <https://arxiv.org/pdf/1409.1556.pdf>
- SkyMind.ai. (2019 m. Balandis 17 d.). *Datasets and Machine Learning*. Nuskaityta iš SkyMind: <https://skymind.ai/wiki/datasets-ml>

- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., . . . Rabinovich, A. (2014 m. Rugsėjis 17 d.). Going deeper with convolutions. Nuskaityta iš <https://arxiv.org/pdf/1409.4842.pdf>
- Szegedy, C., Vanhoucke, V., Ioffe, S., & Shlens, J. (2015 m. Gruodis 2 d.). Rethinking the Inception Architecture for Computer Vision.
- Szeliski, R. (2010). Computer Vision: Algorithms and Applications. Esantis R. Szeliski, *Computer Vision: Algorithms and Applications* (p. 3-5). Springer. Paimta 2019 m. Balandis 9 d. iš <http://szeliski.org/Book/>
- Zeiler, M. D., & Fergus, R. (2013 m. Sausis 16 d.). Stochastic Pooling for Regularization of Deep Convolutional Neural Networks. Nuskaityta iš <https://arxiv.org/pdf/1301.3557.pdf>