



Kauno technologijos universitetas

Elektros ir elektronikos fakultetas

Paviršinių defektų aptikimo sistemos sukūrimas ir tyrimas

Baigiamasis magistro projektas

Vytautas Matulaitis

Projekto autorius

Doc. dr. Arūnas Lipnickas

Vadovas

Kaunas, 2019



Kauno technologijos universitetas

Elektros ir elektronikos fakultetas

Paviršinių defektų aptikimo sistemos sukūrimas ir tyrimas

Baigiamasis magistro projektas

Valdymo technologijos (6211EX014)

Vytautas Matulaitis

Projekto autorius

Doc. dr. Arūnas Lipnickas

Vadovas

Doc. dr. Vidas Raudonis

Recenzentas

Kaunas, 2019



Kauno technologijos universitetas

Elektros ir elektronikos fakultetas

Vytautas Matulaitis

Paviršinių defektų aptikimo sistemos sukūrimas ir tyrimas

Akademinio sąžiningumo deklaracija

Patvirtinu, kad mano, Vytauto Matulaičio, baigiamasis projektas tema „Paviršinių defektų aptikimo sistemos sukūrimas ir tyrimas“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

(vardą ir pavardę įrašyti ranka)

(parašas)

Matulaitis, Vytautas. Paviršinių defektų aptikimo sistemos sukūrimas ir tyrimas. Magistro baigiamasis projektas, vadovas doc. dr. Arūnas Lipnickas; Kauno technologijos universitetas, Elektros ir elektronikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): elektronikos inžinerija, inžinerijos mokslai.

Reikšminiai žodžiai: konvoliucinis neuroninis tinklas, vaizdų atpažinimas, defektų lokalizacija, segmentacija.

Kaunas, 2019. 45 p.

Santrauka

Šiame darbe yra tiriamos kelios skirtingos konvoliucinių neuroninių tinklų architektūros, bandant jas pritaikyti pramoninės gamybos kokybės užtikrinimui atpažįstant paviršinius defektus. Remiantis mokslinėje literatūroje pateiktais tiriamaisiais darbais su analogiška duomenų baze, pirmiausia yra sudaromas modelis ir ištiriamas jo pasiektų prognozių tikslumas, keičiant įėjimo atvaizdų dimensijas. antroje tyrimo dalyje yra bandoma pritaikyti konvoliucinių neuroninių tinklų modelius, kurie buvo jau apmokinti su ImageNet duomenų baze. Galiausiai paskutinėje tyrimo dalyje yra sudaromas defektų lokalizacijos funkcionalumas bei pritaikoma U-Net architektūra tiriamai duomenų bazei, siekiant išgauti defektų segmentacijos žemėlapius.

Modelių palyginimo metodiką apima keli skirtingi kriterijai įskaitant pasiektą tikslumą, apmokymo proceso sudėtingumą bei greitį ir defektų lokalizacijos arba segmentacijos funkcionalumą.

Tyrimo metu nustatyta, kad buvo lengviau apmokinti modelius, į kuriuos buvo importuoti ImageNet svoriai, nei sudaryti naujus modelius, siūlomus literatūros šaltiniuose, tačiau pasiūlyta defektų lokalizacijos funkcija bei apmokymo greitis nebuvo pakankamai geri lyginant su U-Net pasiektais segmentacijos rezultatais.

Matulaitis, Vytautas. Development and Analysis of Surface Defects Recognition System. Master's Final Degree Project, supervisor assoc. prof., Arūnas Lipnickas; Faculty of Electrical and Electronics Engineering, Kaunas University of Technology.

Study field and area (study field group): electronics engineering, engineering science.

Keywords: convolutional neural network, image recognition, defect localization, segmentation.

Kaunas, 2019. 45 pages.

Summary

In this work, several different architectures of convolutional neural networks are being investigated, trying to adapt them to quality assurance of industrial production by recognizing surface defects. Based on the research work presented in the sources with an analogous database, a model is first created and the accuracy of the results, obtained by changing the dimensions of the input images, is investigated. The second part of the study attempts to adapt the models of convolutional networks that have already been learned on the ImageNet database. Finally, in the last part of this research thesis, the defect localization functionality is created and the U-Net architecture is adapted to the database being studied in order to obtain defect segmentation maps.

The model comparison methodology includes several different criteria including the accuracy achieved, the complexity and speed of the training process, and ultimately the functionality for defect localization or segmentation.

The results of the study showed that it was easier to train models into which ImageNet weights were imported than to create new models as suggested in the related works, but the proposed defect localization function and training speed were not good enough compared to the segmentation results achieved by U-Net.

Turinys

Lentelių sąrašas	7
Paveikslų sąrašas	8
Santrumpų ir terminų sąrašas	9
Įvadas.....	10
1. Vaizdų atpažinimo algoritmai	11
1.1. Defektų aptikimas pramoninėje gamyboje	11
1.2. Klasikiniai požymių atpažinimo algoritmai	12
1.3. Vaizdų atpažinimo algoritmų palyginimas.....	15
1.4. Gilieji neuroniniai tinklai vaizdų atpažinimui.....	16
1.4.1. Prižiūrimasis mokymas	16
1.4.2. Bendroji konvoliucinio neuroninio tinklo struktūra	17
1.4.3. Objektų lokalizacija ir segmentacija	20
1.5. Susiję defektų aptikimo tiriamieji darbai	21
2. Tiriama duomenų bazė ir jos pritaikymo kriterijai.....	23
2.1. DAGM.....	23
2.2. Modelių sudarymo kriterijai DAGM duomenų bazei	24
3. Tyrimo metodika	25
3.1. Tiriamų modelių sudarymas	25
3.1.1. v1173 modelio architektūra ir pakeitimai.....	25
3.1.2. Apmokintų modelių architektūros ir pakeitimai.....	25
3.1.3. U-Net architektūra	27
3.2. Tiriamų modelių apmokymo metodika	27
3.2.1. v1173 apmokymas.....	27
3.2.2. MobileNet, DenseNet121 ir ResNet50 apmokymas	28
3.2.3. U-Net apmokymas.....	28
3.3. Tikslumo įvertinimo kriterijai	29
4. Tyrimo rezultatai.....	30
4.1. Rezultatų jautrumo nustatymas keičiant įėjimo atvaizdo dimensijas.....	30
4.2. MobileNet, DenseNet121 ir ResNet50 rezultatai	31
4.3. Visų paviršių klasių apmokinimas vienu metu.....	34
4.4. Požymių žemėlapiai.....	36
4.5. Defektų lokalizacija.....	37
4.6. Defektų segmentacija	40
Išvados	43
Literatūros sąrašas	44

Lentelių sąrašas

1 lentelė. DAGM atvaizdų paviršių klasių imčių pasiskirstymas	23
2 lentelė. Tiriamųjų iš anksto apmokintų modelių bendrieji parametrai [26].....	26
3 lentelė. v1173 tikslumo rezultatai pirmajai paviršių klasei.....	30
4 lentelė. <i>Keras</i> bibliotekos modelių tikslumo rezultatai.....	32
5 lentelė. Apmokintų modelių TPR ir TNR reikšmės.....	34
6 lentelė. Visų paviršių klasių apmokinimo vienu metu rezultatai.	34
7 lentelė. Visų paviršių klasių apmokinimo vienu metu rezultatai be ImageNet saugyklos svorių..	35
8 lentelė. Tiriamų modelių apmokinimo iteracijų trukmės.....	36
9 lentelė. U-Net pasiektų tikslumo rezultatų suvestinė.....	41

Paveikslų sąrašas

1 pav. Defekto pašalinimo išlaidų priklausomybė nuo gamybos ciklo etapo [3]	11
2 pav. Atraminių vektorių klasifikatoriaus [1].....	12
3 pav. Perceptrono matematinis modelis [5].....	13
4 pav. Sprendimų medžio algoritmo pavyzdys [1]	14
5 pav. k-NN algoritmo narių grupavimas [1].....	14
6 pav. Bajeso tinklo struktūra [1].....	15
7 pav. ImageNet tradicinių modelių tikslumo palyginimas su gilaus mokymosi modeliais [8].....	16
8 pav. Paklaidos funkcijos E vertės judėjimas greičiausio nusileidimo kryptimi [9].....	17
9 pav. Standartinė CNN architektūra [12].....	18
10 pav. Kairėje sigmoidinė, dešinėje ReLU aktyvacijos funkcijos [14].....	19
11 pav. Pilnai sujungto konvoliucinio neuroninio tinklo U-Net struktūra [20].....	21
12 pav. DAGM atvaizdų paviršių klasės [22].....	23
13 pav. v1173 architektūra [23]	25
14 pav. U-Net architektūra [20]	27
15 pav. v1173 priklausomybė nuo įvesties atvaizdo dimensijų.....	30
16 pav. v1173 apmokymo tikslumo ir sąnaudų kreivės.....	31
17 pav. Apmokintų modelių tikslumas skirtingoms paviršių klasėms.....	32
18 pav. Apmokintų modelių tikslumo ir sąnaudų kreivės kiekvienai klasei	33
19 pav. Susimaišymo matricos apmokinant visas paviršių klases su ImageNet saugyklos svoriais	34
20 pav. Susimaišymo matricos apmokinant visas paviršių klases be ImageNet saugyklos svorių...	35
21 pav. Visų paviršių klasių apmokymas be ImageNet saugyklos svorių.....	36
22 pav. Tiriamų modelių požymių žemėlapiai.....	37
23 pav. Ribojančios geometrijos projekcija ant originalaus atvaizdo.....	38
24 pav. DenseNet121 tiriamų klasių defektų ribojančios geometrijos	39
25 pav. U-Net tiriamų klasių defektų segmentacijos	40
26 pav. U-Net treniravimo ir validacijos imčių apmokymas	41

Santrumpų ir terminų sąrašas

Santrumpos:

Doc. – docentas;

CNN – konvoliucinis neuroninis tinklas (angl. *Convolutional Neural Network*);

k-NN – k artimiausių kaimynų (angl. *k-Nearest Neighbour*) klasifikatorius;

SVM – atraminių vektorių (angl. *Support Vector Machine*) klasifikatorius;

ReLU – Ištaisyta tiesinė (angl. *Rectified Linear*) aktyvacijos funkcija;

DAGM – Vokietijos požymių atpažinimo asociacija (vok. *Deutsche Arbeitsgemeinschaft für Mustererkennung*);

CSV – kableliu atskirtos vertės (angl. *Comma Separated Value*);

MSE – vidutinė kvadratinė paklaida (angl. *Mean Squared Error*);

BCE – binarinės kryžminės entropijos (angl. *Binary cross-entropy*) sąnaudų funkcija;

DSC – giliai atskiriama konvoliucija (angl. *Depthwise Separable Convolution*);

CCE – kategorinės kryžminės entropijos (angl. *Categorical Cross Entropy*) sąnaudų funkcija;

TPR – tikroji teigiama norma (angl. *True Positive Rate*);

TNR – tikroji neigiama norma (angl. *True Negative Rate*);

FCN – pilnai konvoliuciniai neuroniniai tinklai (angl. *Fully Convolutional Neural Networks*).

Įvadas

Pramoninėje gamyboje produktų kokybė turi svarbų vaidmenį ir nulemia pačios gamyklos efektyvumą, nes brokuoti produktai gali tik pakenkti pardavimams ir įmonės pajamoms bei pelningumui. Tai taip pat daro įtaką ir klientų požiūriui į įmonės patikimumą ilgalaikėje perspektyvoje. Dėl šių priežasčių kompiuterinės regos pritaikymas tampa vis patrauklesnė alternatyva kokybei užtikrinti, nes pakartotinai stebėti ir bandyti nustatyti ryšius tarp kelių skirtingų požymių yra darbas, kurį atlikdami žmonės yra dažnai linkę daryti klaidų bei įvertinant šiuolaikinės pramonės gamybos linijų greitį, žmonėms atlikti šį darbą būtų beveik neįmanoma. Tačiau gilų mokymąsi galima sėkmingai pritaikyti atlikti šias užduotis ir automatizuoti ne tik defektų aptikimo procesą, bet ir modelių sudarymą, priverčiant modelius pačius išmokti išskirti reikiamus požymius [1].

Darbo tikslas – sudaryti ir apmokinti skirtingų konfigūracijų konvoliucinių neuroninių tinklų modelius paviršiniams defektams atpažinti.

Darbo uždaviniai:

1. v1173 neuroninio tinklo architektūrą realizuoti *Python Keras* aplinkoje ir patikrinti jos prognozių tikslumo priklausomybę nuo įvesties atvaizdų dydžio;
2. parinkti 3 skirtingus neuroninių tinklų modelius ir pasinaudojant ImageNet *Keras* saugyklos modelių svoriais atlikti šių modelių žinių perkėlimo apmokinimą kiekvienai paviršių klasei;
3. parinktus modelius apmokinti atpažinti defektus visose tiriamose paviršių klasėse vienu metu;
4. realizuoti tiriamųjų modelių defektų aptikimą;
5. išgauti segmentacijos žemėlapius sudarant ir apmokinant U-Net architektūrą;
6. palyginti tiriamų modelių pasiektus rezultatus tarpusavyje bei su rezultatais pateiktais susijusiuose tiriamuosiuose darbuose.

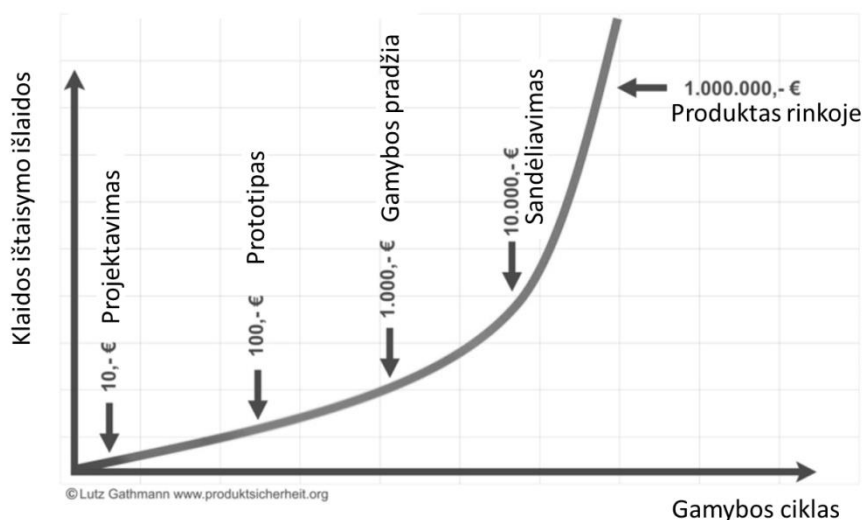
1. Vaizdų atpažinimo algoritmai

Šiame skyriuje yra aptariamas defektų aptikimo sistemų aktualumas, pritaikant mašininės regos sistemas, apžvelgiami keli skirtingi vaizdų atpažinimo algoritmai bei konkrečiau aprašoma konvoliucinių neuroninių tinklų struktūra.

1.1. Defektų aptikimas pramoninėje gamyboje

Išsivysčiusių šalių pramoninės gamybos linijos siekia gerokai padidinti produktyvumą, kad išliktų konkurencingi su besivystančių šalių gamintojais, kurie gali pagaminti analogiškus produktus mažesnėmis kainomis [2].

Bendruoju atveju galima teigti, kad kuo toliau produkto defektas yra nepastebimas, ypač vėlyvose gamybos etapuose, tuo didesnės yra klaidos ištaisymo išlaidos (1 pav.) [3].



1 pav. Defekto pašalinimo išlaidų priklausomybė nuo gamybos ciklo etapo [3]

Produktyvumas gali būti padidintas įvairiais būdais: mažinant sąnaudas, didinant žaliavų naudojimo efektyvumą arba gerinant produktų kokybę. Kokybės sąnaudos paprastai skirstomos į keturias kategorijas: vertinimo, prevencijos, vidinės gedimo ir išorės gedimo išlaidos. Šios kategorijos laikomos naudingomis organizacijoms, siekiančioms nustatyti savo kokybės užtikrinimo sąnaudas bei į kurią kategoriją sutelkti investicijas [2].

Kompiuterinės regos sistemos kokybės kontrolei gamybos aplinkoje yra plačiai pritaikomos, siekiant pašalinti žmonių padaromas klaidas viso proceso metu. Paprastai pramoninių dalių paviršius yra tikrinamas norint aptikti defektus, pvz., detalių įbrėžimai, lūžiai. Remiantis literatūroje pateiktomis apžvalgomis dažniausiai naudojami metodai industrinėje gamyboje defektams aptikti gali būti suskirstyti į 3 rūšis [4]: statistiniai, spektriniai ir modeliais paremti.

Gilaus mokymosi algoritmai iš esmės skiriasi nuo aukščiau išvardintų metodikų, nes jie atlieka automatizuotą reikšmingų požymių aptikimą, vadinasi paties modelio parametrų derinti nereikia

norint išgauti analogiškus rezultatus. Tačiau šie algoritmai reikalauja sąlyginai didesnės mokymosi atvaizdų duomenų bazės.

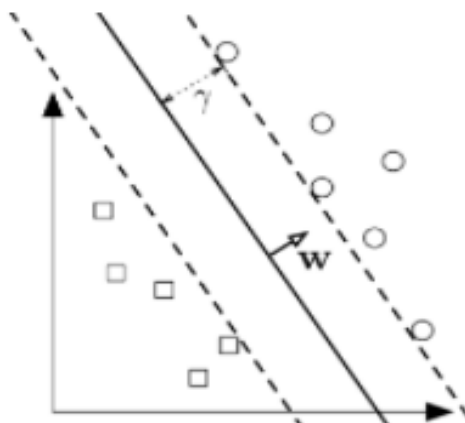
1.2. Klasikiniai požymių atpažinimo algoritmai

Kompiuterinės regos mokslo sfera apima vaizdų priėmimo, apdorojimo ir analizavimo metodus norint išgauti tam tikrą, reikalingą informaciją. Vaizdų atpažinimas aprašo tam tikrą vaizdo apdorojimo užduotį objektams lokalizuoti ir klasifikuoti, jis vaidina svarbų vaidmenį kompiuterinės regos srityje ir yra būtina dalis tam tikrose pramoninio valdymo procesuose bei autonomiškai judančiose sistemose [5]. Vaizdų atpažinimo sistema naudoja mokymo duomenų rinkinius, kuriuose yra atvaizdų su žinomais ir pažymėtais objektais, ir ištraukia skirtingų požymių informaciją pagal pasirinktą algoritmą. Iš duomenų rinkinio atvaizdų informacija yra paverčiama į matricas, kurias gali priimti ir apdoroti modelis. Tai gali būti informacija apie spalvas, kraštus ar geometrines formas [6]. Šis modelio paruošimo būdas priskiriamas prižiūrimo mokymo metodikai (angl. *Supervised Learning*). Atvirkščiai tokiam požiūriui taip pat taikomi ir neprižiūrimo modelių mokymo metodai (angl. *Unsupervised Learning*), kuomet sužymėti duomenys nėra prieinami ir tyrėjai tikisi, kad modelis pats pastebės svarbiausius bruožus, kurie ir darys įtaką klasifikavimo kokybei [7].

Šiame poskyryje bus apžvelgiami klasikiniai vaizdų atpažinimo metodai. Nors yra daugybė algoritmų ir jų atmainų, išrinkti metodai suskirstomi į kelis bazinius tipus: atraminiai vektoriai, perceptronai, loginiai ir statistiniai metodai.

Atraminų vektorių algoritmai

Atraminų vektorių klasifikatorius (angl. *Support Vector Machine*) – sukuria skiriamąjį paviršių daugiamatėje erdvėje (kernelio funkciją). Geras matavimo duomenų atributų atskyrimas pasiekiamas plokštuma, kuri turi didžiausią atstumą iki bet kurios klasės artimiausio mokymo duomenų taško. Kuo didesnis šis atstumas, tuo mažesnė klasifikatoriaus apibendrinamoji paklaida. Algoritmo atmainos taip pat naudoja netiesines plokštumas siekdamos gauti daugiau lankstumo atskiriant ribas duomenų rinkinyje [1].



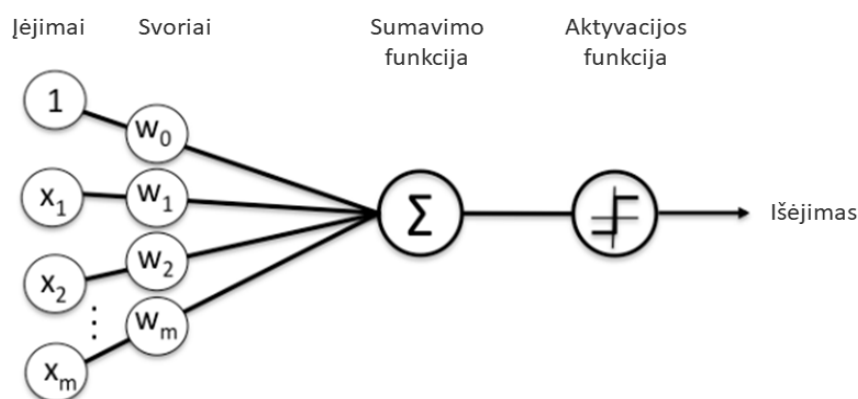
2 pav. Atraminų vektorių klasifikatoriaus [1]

Tam tikroms duomenų struktūroms neįmanoma surasti tinkančios plokštumos su pakankamai maža paklaida, nes pasitaiko ir klaidingai suklasifikuotų pavyzdžių. Ši problema sprendžiama įvedant

toleranciją vienetinėms klasifikacijos klaidoms matavimo duomenų rinkinyje. Taip pat reikia pabrėžti, kad šio algoritmo išėjimas yra binarinis ir sprendžiant uždavinius, kuriems reikia atskirti daugiau išėjimo klasių, būtina skaldyti šį algoritmą į kelias binarines dalis.

Perceptronu paremti algoritmai

Dirbtiniai neuroniniai tinklai (angl. *Artificial Neural Networks*) – šie modeliai yra įkvėpti biologinių neuronų ląstelių sistemų. Šio tipo modelis dažniausiai susideda iš kelių sluoksnių. Kiekvienas sluoksnis yra sudarytas iš tam tikro skaičiaus neuronų. Įvesties ir išvesties sluoksniai yra mažiausias sluoksnių kiekis, kurį gali turėti neuroninis tinklas. Visi viename sluoksnyje esantys neuronai yra sujungti su visais neuronaus aukštesniame sluoksnyje, taip potencialiai sukurdami didžiulį tinklą su milijonais parametru, kuriuos mokymo proceso metu reikia perskaičiuoti. Visos šios jungtys turi tam tikrą svorį, kuris atnaujinamas mokymosi fazėje.



3 pav. Perceptrono matematinis modelis [5]

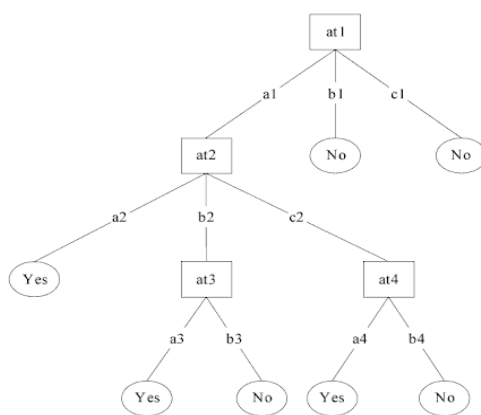
Jei įvesties signalų suma didesnė numatytai ribai, tuomet aktyvacijos funkcija perduoda išėjimo signalą [5]:

$$y = \sigma \left(b + \sum_{i=1}^n x_i \cdot w_i \right). \quad (1)$$

Yra daug skirtingų neuroninių tinklų architektūrų tipų. Reikia paminėti, kad skirtingoms klasifikacijos uždavimams naudojamos specifinės architektūros, kurios yra geriausiai optimizuotos tos rūšies problemoms spręsti.

Statistiniai ir loginiai algoritmai

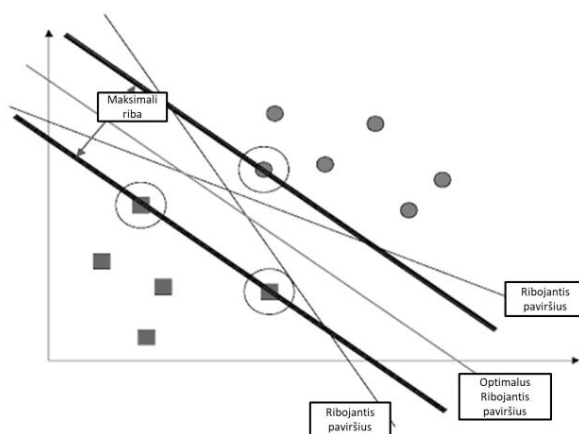
Sprendimų medžiai (angl. *Decision Trees*) – loginiai modeliai, kurie klasifikuoja požymius, rūšiuodami juos pagal ypatybių vertes. Kiekvienas sprendimų medžio mazgas nurodo egzistuojančią ypatybę, kuri turi būti klasifikuojama, o kiekviena šaka reiškia vertę, kurią gali priimti mazgas. Atvejai klasifikuojami pradedant šakniniu mazgu ir rūšiuojami pagal jų išmatuotų ypatybių vertes.



4 pav. Sprendimų medžio algoritmo pavyzdys [1]

Požymis, kuris geriausiai padalina mokymo duomenis, yra pagrindinis medžio mazgas. Nors yra keletas skirtingų būdų, kaip rasti funkciją, kuri geriausiai išskiria mokymo duomenis, tačiau literatūroje matoma, kad nėra universalus vieno geriausio metodo [1]. Vertingiausia sprendimų medžių savybė yra jų algoritmo suprantamumas, t. y. lengva suprasti, kodėl egzempliorius buvo priskirtas tam tikrai duomenų grupės klasei.

k-Artimiausių kaimynų klasifikatorius (angl. *k-Nearest Neighbour*) – Statistiniams algoritmams priskiriamas klasifikatorius, kuris pagrįstas principu, kad duomenų rinkinio egzemplioriai paprastai yra arti kitų panašių savybių turinčių matavimų duomenų taškų. Jei egzemplioriai yra pažymėti klasifikavimo etikete, tada neklasifikuoto egzemplioriaus etiketės vertę galima nustatyti stebint artimiausių kaimynų klasę. Šis algoritmas aptinka tiriamo taško – k artimiausius duomenų taškus (kaimynus) ir nustato jo klasę išskirdamas vienos arčiausiai ir dažniausiai pasitaikančios klasės etiketę [1].

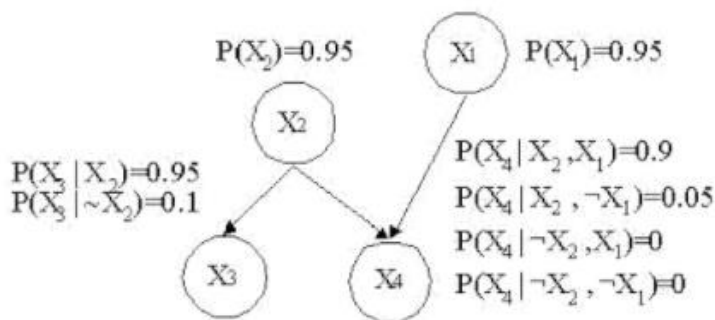


5 pav. k-NN algoritmo narių grupavimas [1]

Kadangi matavimų duomenys gali būti modeliuojami n -matėje erdvėje, absoliuti duomenų taškų šioje pozicijoje vieta nėra tokia reikšminga kaip santykinis atstumas tarp duomenų taškų. Santykinis atstumas iki kaimynų yra įvertinamas ir prognozė yra išduodama.

Bajeso tinklas (angl. *Bayesian Network*) – tai grafinis modelis skirtas parodyti tikimybės ryšius tarp kintamųjų rinkinio. Grafų struktūros mazgai atitinka ypatybių pasiskirstymą, t. y. ryšiai tarp mazgų daro įtaką atsitiktiniams elementams. Paprastai Bajeso tinklo apmokymo užduotis gali būti

suskirstyta į du etapus: pirmiausia nustatoma, tinkama tinklo struktūra, paskui apskaičiuojami jo parametrai.



6 pav. Bajeso tinklo struktūra [1]

Įdomiausia Bajeso tinklų savybė, lyginant su kitais algoritmais, yra galimybė susipažinti su iš anksto pateikta informacija apie tam tikrą problemą ir atsižvelgti į tai kuriant struktūrinius tarpusavio mazgų ryšius. Šių klasifikatorių problema yra tai, kad jie netinka duomenų rinkiniams turintiems daug skirtingų ypatybių, į kurias reikėtų atsižvelgti. Vadinasi, atsižvelgiant į laiką, labai didelio tinklo sukūrimas yra sunkiai įgyvendinamas [1].

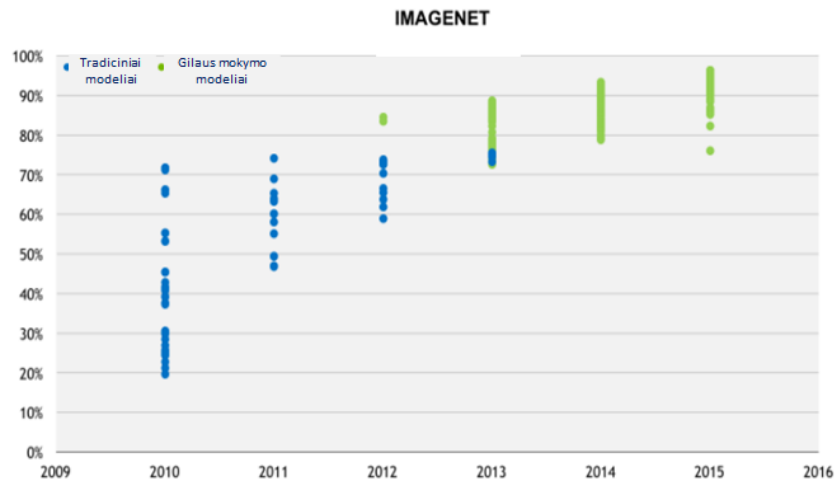
1.3. Vaizdų atpažinimo algoritmų palyginimas

Paprastai Atraminų vektorių klasifikatoriai ir neuroniniai tinklai, dirbdami su atvaizdais, kuriuose yra nepertraukiami, tolydūs požymiai, paprastai pasiekia geresnius rezultatus nei kiti algoritmai. Priešingai, loginiai algoritmai (pvz., sprendimų medžiai) paprastai veikia geriau, kuomet analizuojami atskiri ar kategoriški bruožai. Neuroniniams tinklams ir atraminų vektorių klasifikatoriams reikia didelės imties dydžio mokomųjų bandinių, kad būtų pasiektas maksimalus prognozavimo tikslumas, o Bajeso tinklams gali prireikti santykinai nedidelio duomenų rinkinio [1].

Taip pat nenuostabu, kad modelio apmokymo laikas skiriasi visiems algoritmams, priklausomai nuo pritaikymo užduoties pobūdžio ir apmokymo duomenų rinkinio. Pavyzdžiui k-artimiausio kaimyno algoritmas nereikalauja laiko, kad būtų apmokytas, nes mokymo duomenys turi būti atitinkamai saugomi, kad modelis galėtų juos pasiekti. Bajeso tinklams reikalingas palyginti trumpas apmokymo laikas, nes modeliui reikalingi mažesni apmokymo duomenų rinkiniai [1]. Šiuo atžvilgiu neuroniniai tinklai ir atraminų vektorių klasifikatoriai gali užimti daug laiko apmokymui, priklausomai nuo prieinamos kompiuterinės įrangos.

Loginiai algoritmai yra labai lengvai interpretuojami, o neuroniniai tinklai, SVM ir k-NN parodė, kad tam tikrais atvejais gali būti sunku suprasti šių modelių rezultatus, kurie savo ruožtu verčia modelius tapti mažiau patikimais [1].

Tiesiogiai įvertinti aptartus algoritmus nėra lengva, nes kiekvienas metodas prieina prie vaizdo atpažinimo problemos savaip ir tam tikrais atvejais patys algoritmai buvo sudaryti ne vienodoms užduotims atlikti. Kaip matoma 7 paveiksle iki šiol klasikiniai algoritmai sugebėjo parodyti apie 70% vaizdų atpažinimo rezultatus, tačiau išpopuliarėjus vaizdo plokščių utilizacijai šių algoritmų pritaikyme, buvo galima išplėsti juos ir pradėti eksperimentuoti su gilaus mokymosi (angl. *Deep Learning*) algoritmais.



7 pav. ImageNet tradicinių modelių tikslumo palyginimas su gilaus mokymosi modeliais [8]

Remiantis ImageNet duomenų bazės statistika matyti, kad gilaus mokymosi algoritmai pralenkė tradicinius algoritmus ir perkopė 90% tikslumo ribą [8]. Reikia pabrėžti, kad dėl koncepcijos populiarumo bei programavimo įrankių prieinamumo, perceptrono modeliu paremti neuroniniai tinklai šiuo metu yra pats populiariausias būdas atlikti klasifikavimo užduotis. Dėl to šis tiriamasis darbas labiau susikoncentruos į jų pritaikymą ir kiti skyriai apžvelgs giliau šio algoritmo tipo specifikas.

1.4. Gilieji neuroniniai tinklai vaizdų atpažinimui

Šiame poskyryje bus aptariama specifiniai giliųjų neuroninių tinklų modeliai skirti vaizdų atpažinimo, lokalizacijos bei segmentacijos uždaviniams spręsti.

1.4.1. Prižiūrimasis mokymas

Prižiūrimo mokymosi klasifikavimo sistema turi išmokti prognozuoti, t. y. atlikti pasvertus spėjimus, kai jai suteikiamas tam tikrų požymių vektorius. Bendroju atveju šio klasifikatoriaus tikslumą galima įvertinti palyginus modelio prognozę su žymės vektoriumi. Bet koks šių dviejų vektorių neatitikimas gali būti klaidos E (pvz. vidutinė kvadratinė paklaida) vertė sistemai modifikuoti, siekiant ateityje geriau atlikti prognozę. Šiai metodikai pritaikyti pirmiausia reikia surinkti tinkamų požymių vektorių duomenų bazę, kurią būtų galima naudoti kaip klasifikatoriaus įėjimą bei iš anksto žmogaus patikrintas ir paskirtas žymes. Šis paruošiamasis darbas daugeliu atvejų reikalauja daugiau laiko nei pačio klasifikatoriaus apmokymas [9].

Apie 1980 m., pastebėta, kad atgalinio sklaidimo (angl. *Backpropagation*) algoritmu apmokomi gilieji neuroniniai tinklai nėra lengvai įgyvendinami dėl didelio kiekio iteracinių skaičiavimų kuriuos reikia atlikti [10]. Remiantis [8] matoma, kad lengviau prieinama kompiuterinė galia viską pakeitė ir gilieji tinklai sugeba parodyti geresnius rezultatus vaizdų atpažinimo srityje.

Turint reikiamus duomenis modelio apmokymui, atgalinės sklaidos algoritmas, kurio pagalba iteraciniu būdu sureguliuojami dirbtinio neuroninio tinklo svoriai w , sumažina paklaidą tarp realių ir pageidaujamų rezultatų. Pagrindinis modelio apmokymo tikslas yra išsiaiškinti, kaip pakoreguoti svorius. Intuityviai žinoma, kad jei svorio pokytis padidins (sumažins) klaidą, tada norime

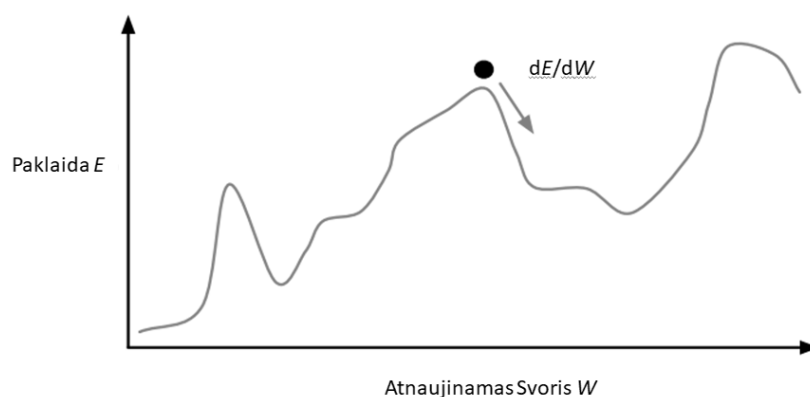
sumažinti (padidinti) šį svorį. Matematiškai tai reiškia, kad reikia atsižvelgti į klaidos dalinį diferencialą pagal svorį [9]:

$$\frac{\partial E}{\partial \omega_{ij}}, \quad (2)$$

kuris atspindi klaidos pasikeitimą, kai svoris pasikeičia. Apskaičiavus šią išvestinę, svoris atnaujinamas remiantis [9]:

$$\Delta \omega_{ij} = -\eta \frac{\partial E}{\partial \omega_{ij}}. \quad (3)$$

Dydis η , yra standartinis parametras neuroniniuose tinkluose ir dažnai vadinamas mokymosi greičiu ir tiesiog apibrėžia atstumą tarp gretimų taškų w ašyje. Pažangesniuose algoritmuose šis rodiklis gali palaipsniui mažėti mokymo fazės metu siekiant išvengti algoritmo užstrigimo lokaliuose paklaidos funkcijos minimumuose.



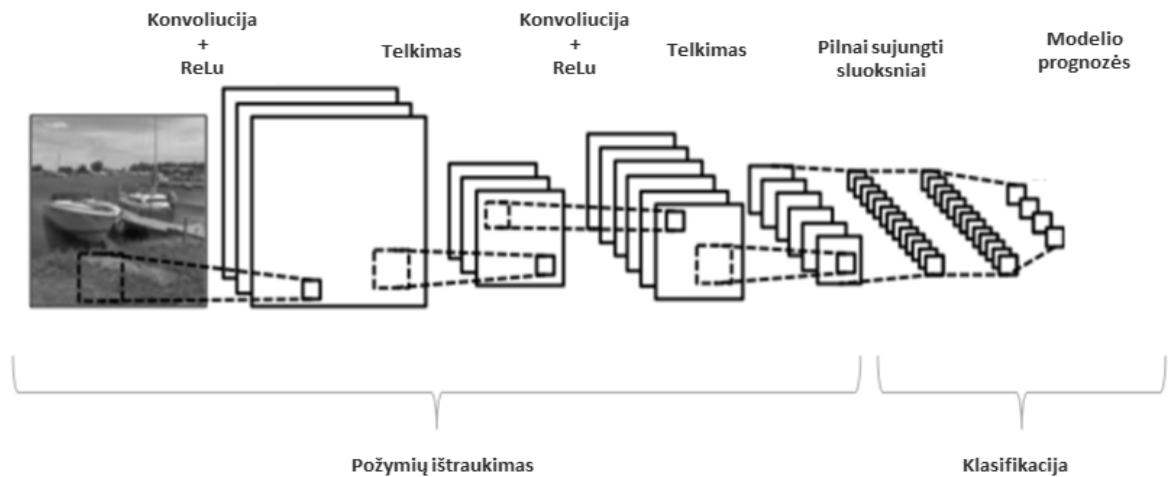
8 pav. Paklaidos funkcijos E vertės judėjimas greičiausio nusileidimo kryptimi [9]

Visi svoriai atnaujinami naudojant tą pačią formulę, vadinasi, kad paklaidos vertė judės greičiausio nusileidimo kryptimi išilgai sąnaudų funkcijos paviršiaus.

Dažniausiai paklaidos funkcija E yra neišgaubtas (angl. *Non-Convex*) daugiamatis paviršius, vadinasi universalus algoritmas skirtas surasti globaliam minimumui yra neįmanomas. Tai iš dalies reiškia, jog šios problemos sprendimų gali būti daug ir rezultatai priklausys nuo pasiektų lokalių minimumų. Dėl šios priežasties kiekvieną kartą inicializuojant atsitiktinius svorius ir iš naujo apmokinant modelį yra pasiekiami skirtingi rezultatai.

1.4.2. Bendroji konvoliucinio neuroninio tinklo struktūra

Skirtingi neuroninių tinklų tipai geriau susitvarko su tam tikrais uždaviniais. Vaizdų atpažinimo užduotims spręsti labiausiai paplitę ir dažniausiai naudojami konvoliuciniai (angl. *Convolutional*) neuroniniai tinklai, toliau – CNN [11]. Bendroji CNN architektūra pateikiama 9 paveiksle.



9 pav. Standartinė CNN architektūra [12]

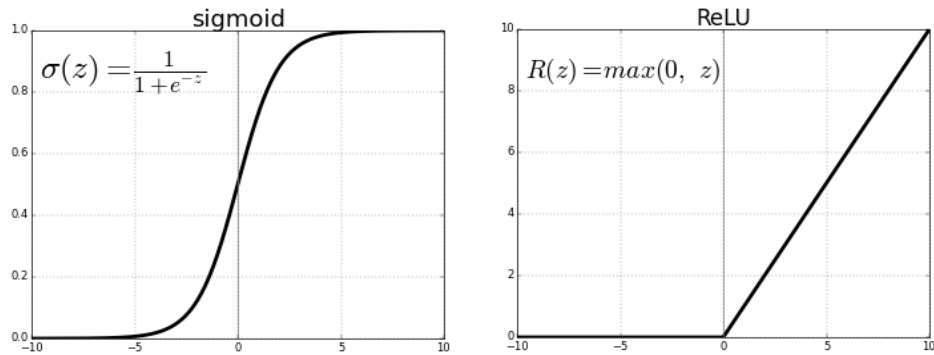
Tarp tradicinių neuroninių tinklų ir CNN architektūrų yra keli esminiai skirtumai. CNN architektūra gali būti skirstoma į 2 dalis pasikartojančių sluoksnių su tam tikromis informacijos apdorojimo operacijomis:

1) požymių ištraukimas

Įėjimo sluoksnis: priklausomai nuo naudojamos programinės įrangos, šis sluoksnis yra atsakingas už atvaizdo informacijos paėmimą iš sugeneruotų imčių ir jo pikselių informacijos laikymą vienos arba trijų dimensijų masyvuose, kol aukštesnio sluoksnio filtrai skenuoja šią informaciją.

Konvoliucija: turint omenyje, kad šiuo atveju modelio įvestis yra atvaizdas, modelis matys 3 pikselių intensyvumų reikšmių masyvus, kurių dydžiai priklausys nuo tiriamo atvaizdo skiriamosios gebos. Išskiriami 3 masyvai kiekvienai RGB spalvai, vadinasi kiekvienas masyvo elementas gali įgauti reikšmę intervale nuo 0 iki 255. Pati konvoliucijos operacija atliekama slenkant pasirinkto pikselių pločio ir aukščio filtrą per įvesties masyvus. Pats filtras yra taip pat masyvas, kurio elementams yra priskirti tam tikri svoriai. Slenkant šį filtrą per įvesties masyvą yra sudauginamos ir sudedamos filtro ir įvesties masyvo pikselių intensyvumo reikšmės. Šios operacijos išėjimas yra sveikasis skaičius kuris priskiriamas sekančio sluoksnio įvesties masyvui, vadinamam požymių žemėlapiu (angl. *Feature Map*) [13].

Aktyvacija: po konvoliucijos sluoksnių (vieno arba daugiau) yra taikomas aktyvacijos sluoksnis, kurio paskirtis yra įvesti netiesiškumą modeliui, kuris iki šiol iš esmės vykdė tik tiesines (sumavimo ir daugybos) operacijas. Buvo įprasta naudoti sigmoidines arba hiperbolines tangento funkcijas, tačiau empiriniai duomenys parodė, kad ištaisyta tiesinė (angl. *Rectified Linear*) toliau – ReLu, funkcija yra labiau tinkama naudoti konvoliucijos sluoksniams (10 pav.). Ji reikalauja mažesnės kompiuterinės galios ir dėl šios priežasties CNN apmokomas kur kas greičiau beveik neaukojant modelio tikslumo [12].



10 pav. Kairėje sigmoidinė, dešinėje ReLU aktyvacijos funkcijos [14]

Telkimas (angl. *Pooling*): Sekant aktyvacijos operaciją yra įprasta taikyti telkimo sluoksnį, nors ir yra keli skirtingi šios operacijos tipai, populiariausias yra maksimalus telkimas (angl. *Max Pooling*). Šio sluoksnio operacija yra savo ruožtu panaši į konvoliucijos, t. y. naudojamas filtras, kuris slenkamas per šio sluoksnio įvesties masyvą. Skirtumas yra tas, kad šiuo atveju filtras neturi jam priskirtų svorių, o filtro išėjimas yra didžiausias skenuojamo masyvo vietos pikselių intensyvumas [12].

Šio sluoksnio nauda grindžiama tuo, kad užfiksavus specifinį požymį jo tiksli vieta nėra tokia svarbi visame masyve, lyginant su jo pozicija kitų požymių pozicijų atžvilgiu. Ši operacija ženkliai sumažina kito konvoliucijos sluoksnio įvesties masyvų ilgį ir plotį, iš to seka, kad taip pat sumažėja ir svorių w skaičius, kuriuos reikės koreguoti apmokinant modelį.

Išmetimas (angl. *Dropout*): visi prijungti sluoksniai sudaro visumą atnaujinamų parametru, kurie gali apmokymo metu išsiugdyti tarpusavio priklausomybę, o tai gali apriboti kiekvieno neurono individualią dedamąją, siekiant išgauti tikslias prognozes. Tai bendroju atveju būtų vadinama prisitaikymu prie treniravimo duomenų. Išmetimas tai viena iš galimų šios problemos sprendimo išeičių, kurios metu parinktuose sluoksniuose išmetama tam tikra procentinė neuronų dalis p . Šiuo būdu išvengiama specifinių ir nereikalingų požymių išmokimo. Galima daryti 2 pagrindines išvadas naudojant šią metodiką [15]:

1. išmetimas priverčia modelį išmokti reikšmingus parametrus bei padaro jį patikimesniu;
2. taip pat modelio apmokymo procesas prailgėja beveik dvigubai;

Partijos normalizacija (angl. *Batch Normalization*): Siekiant padidinti neuroninio tinklo stabilumą, partijos normalizacija pakeičia ankstesnio sluoksnio išėjimo skalę, atimdama atvaizdų partijos vidurkį ir padalijant iš partijos standartinio nuokrypio. Sulyginant kiekvieno sluoksnio išėjimą gaunami 2 žymesni privalumai [16]:

1. Galima pradėti naudoti didesnes mokymosi spartas.
2. Sumažinamas prisitaikymas prie treniravimo duomenų.

2) klasifikacija.

CNN architektūros gale yra pilnai sujungtų neuronų sluoksniai (angl. *Fully Connected Layer*). Šios tinklo dalies paskirtis yra interpretuoti konvoliucijos, ReLU ir maksimalaus/vidutinio telkimo sluoksnių išskirtus požymius ir remiantis atitinkamais svoriais įvertinti, su kuriomis klasėmis tiriamas atvaizdas koreliuoja labiausiai. Šios dalies išėjimas ir pateikia prognozę N dydžio vektoriaus pavidalu [11].

Ši CNN dalis taip pat pasižymi išskirtine aktyvacijos funkcija. Iki šiol aptartos aktyvacijos funkcijos buvo pajėgios išvesti tik binarinį išėjimą, tačiau CNN klasifikatorius gali turėti daugiau nei 2 klases, kurias reikia atpažinti. Dėl to naudojama *Softmax* funkcija:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}. \quad (4)$$

Į šią funkciją įvedamas j dydžio vektorius z , kuris lemia CNN klasifikacijos rezultatus, o pati *Softmax* funkcija išveda tikimybinį pasiskirstymą, nurodanti, kokia yra tikimybė, kad kiekviena iš analizuojamų klasių buvo atpažinta [17]. Remiantis *Softmax* funkcijos rezultatais jau galima atmesti mažiau tikėtinas klases ir priimti prognozę.

CNN apmokymo proceso efektyvumas ir pačio modelio rezultatų kokybė labai priklauso nuo hiperparametrų pasirinkimo, todėl tai yra svarbi modelio sudarymo dalis. Šių parametrų parinkimas nėra intuityvus ir gilesnis tinklas ne visada užtikrins geresnius rezultatus. Toliau pateikiami keli parametrai, į kuriuos reikia atsižvelgti sudarant modelį [18]:

1. įėjimo atvaizdo dydis;
2. konvoliucijos filtro dydis;
3. požymių žemėlapių skaičius;
4. neuronų išmetimo intensyvumas;
5. apmokymo greitis;
6. sąnaudų ir optimizacijos funkcijos pasirinkimas.

Norint nustatyti parametrus, reikia stebėti ir mokymo paklaidos bei validacijos paklaidos kreives. Žinoma didžiąjai daliai klasifikacijos uždavinių galima remtis literatūroje siūlomomis modelių struktūromis bei joms rekomenduojamais parametrais.

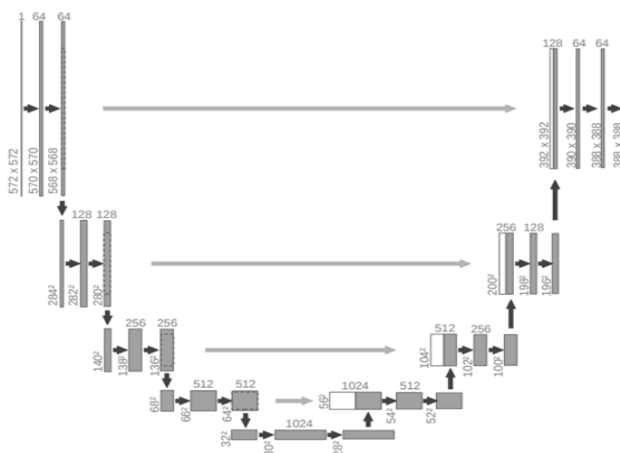
Taip pat, turint priėjimą prie didesnės kompiuterinės galios, galima ir automatizuoti modelio parametrų parinkimą. Tai gali būti apibūdinama kaip „brutalios jėgos“ paieška hiperparametrinėje erdvėje, kurioje atliekami įvairūs modelio deriniai, naudojant paklaidų duomenų rinkinį, iš kurio parenkamas modelis pasiekęs geriausius rezultatus.

1.4.3. Objektų lokalizacija ir segmentacija

Iki šiol šiame darbe buvo aptarti tik bendrieji konvoliucinių neuroninių tinklų veikimo principai, t.y. modelio išėjimas paprasčiausiu atveju yra užkoduotas vektorius, kuris *Softmax* funkcijos pagalba pateikia savo prognozę. Tačiau šis informacijos pateikimo būdas yra gan ribotas, nes jame nėra užfiksuota defekto padėtis ant tiriamo paviršiaus. Tai gali apsunkinti žmonių darbą nes neuroninių tinklų modelio rezultatus ne visada lengva interpretuoti. Dėl to būtų naudinga, kad modelis sugebėtų pateikti ne tik binarinį atsakymą į klausimą ar yra defektas tiriamame paviršiuje, bet taip pat ir paties defekto poziciją.

Norint išgauti šią papildomą informaciją reikia taip pat papildomos atvaizdų duomenų bazės, kuri koreliuotų su jau esama ir būtų sudaryta iš anotacinių atvaizdų, kuriuose būtų pažymėta reikiama objekto vietos informacija [19]. Tuomet CNN pasinaudodamas originaliu atvaizdu ir anotacija gali išmokti ir objekto poziciją originaliame atvaizde. Tada modelio išėjime gaunama informacija apie atskirus atvaizdo regionus braizant aplink juos ribojančias geometrines figūras.

Sekantis žingsnis, siekiant išgauti tikslesnę tiriamų objektų pozicijos informaciją, yra atvaizdo objektų segmentacija, kuomet modelis apmokinamas daryti kiekvieno pikselio prognozę, t.y. kiekvienam pikseliui priskiriama tam tikra klasė (11 pav.).



11 pav. Pilnai sujungto konvoliucinio neuroninio tinklo U-Net struktūra [20]

Tai yra atliekama pasitelkiant pilnai konvoliucinius neuroninius tinklus, toliau – FCN (angl. *Fully Convolutional Neural Networks*) [21]. Šio tipo modelių pagrindinės sudedamosios dalys yra konvoliucijos ir telkimo operacijos bei de-konvoliucijos operacija, kuri yra atsakinga už atvaizdo išplėtimą, siekiant modelio išėjime gauti analogiško įėjimo atvaizdo dydžiui segmentacijos žemėlapi.

1.5. Susiję defektų aptikimo tiriamieji darbai

Jau buvo atliktos kelios studijos, susijusios su paviršinių defektų aptikimu, tačiau ne visi autorių pasiūlyti sprendimai buvo sudaryti remiantis analogiškais kriterijais. Vienas tokių pavyzdžių yra pateikiamas literatūroje [23], kurioje autoriai siūlo kompaktišką CNN architektūrą, turinčią 2 išėjimus, t.y. neuroną pateikiantį defekto tikimybę ant tiriamo paviršiaus ir segmentacijos žemėlapi. Šiuo atveju autoriai apmokino CNN atpažinti defektus naudojant tik po vieną paviršių klasę.

Kitu atveju literatūroje [24] autoriai taip pat pasiūlė savo CNN architektūrą, kuri veikė 2 klasifikacijos etapais, pirma modelis atpažįsta, kuri paviršių klasė (viena iš pirmų 6) buvo užfiksuota vaizdo stebėjimo sistemos, antra modelis atpažįsta ar tiriamame atvaizde yra defektas. Šiuo atžvilgiu autorių darbas yra unikalus, nes jiems pavyko įrodyti, kad galima apmokinti savo pasiūlytą CNN architektūrą atpažinti ne tik defektus bet ir pačias paviršių klases tuo pačiu metu, tačiau jie nepateikė jokio defektų lokalizacijos sprendimo.

Dar vienas bandymas pasiūlyti savo architektūrą yra pateikiamas literatūroje [25]. Šiuo atveju tinklas yra apmokinamas įvedant 3 atvaizdus į modelį vienu metu, t. y. 2 atvaizdai priklauso vienai klasei, o trečias ne. Taip tikimasi, kad tinklui pavyks išmokti reikšmingus požymius siekiant atpažinti paviršinius defektus. Tačiau iš kelių pasiūlytų tinklų variacijų geriausias pasiektas tikslumas buvo 83% visoms 10 paviršių klasių. Segmentacijos žemėlapių pavyzdžiai taip pat parodė, kad defektai buvo sunkiai išskiriami ir kartais sumaišomi su gerais paviršiu priklausančiais pikseliais.

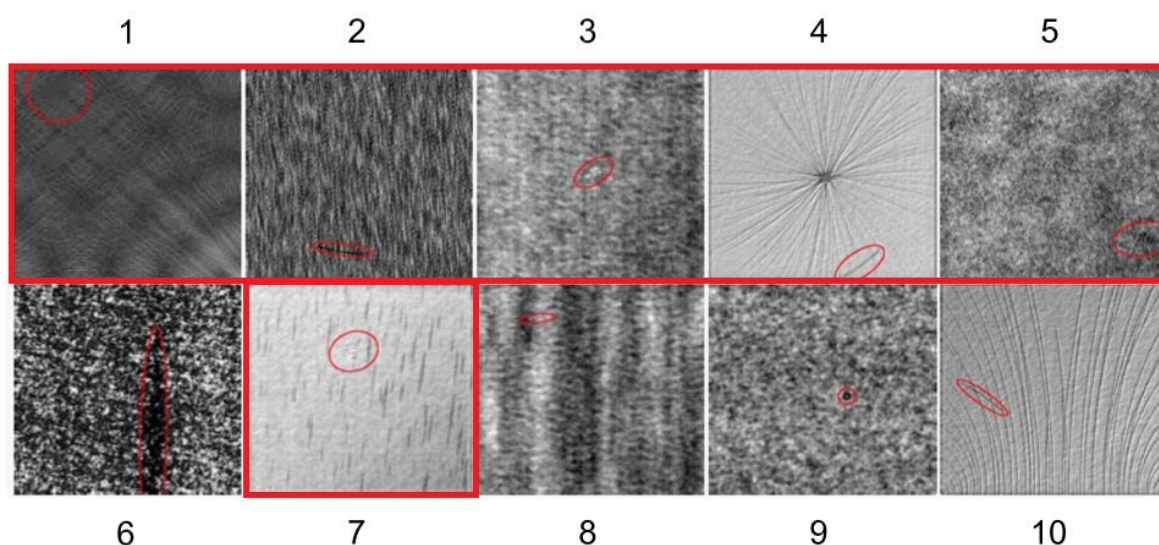
Bendroju atveju visi aptarti literatūros šaltinių modeliai turi savų privalumų ir trūkumų, tačiau papildoma problema iškyla bandant atkurti rezultatus dėl naudotų programinių paketų. Dažniausiai autoriai neskelbia kokią programinę įrangą naudojo sudarant savo modelius kaip pavyzdžiui literatūroje [23] pateikiamas modelis buvo sudarytas su *Theano* programiniu paketu, kuris jau yra nebepalaikomas ir retai naudojamas. Savaiame suprantama, kad bandant atkartoti autorių pasiektus rezultatus su kitais programiniais paketais, gali kilti nesutapimų.

2. Tiriama duomenų bazė ir jos pritaikymo kriterijai

Šiame skyriuje yra aptariami specifiniai atvaizdai, kurie bus naudojami defektų aptikimui šio darbo metu, bei užsibrėžiami tiriamų modelių sudarymo ir veikimo kriterijai.

2.1. DAGM

Tyrimui yra pasitelkiama Vokietijos požymių atpažinimo asociacijos (vok. *Deutsche Arbeitsgemeinschaft für Mustererkennung*, toliau – DAGM) [22] duomenų bazė, kuri yra sudaryta iš kompiuteriu sugeneruotų 512x512x1 dydžio atvaizdų. Visa imtis sudaryta iš 10 paviršių klasių su skirtingais pasikartojančiais požymiais, kurie turėtų būti panašūs į pramonėje naudojamų tikrų medžiagų paviršius. Kiekviena paviršių klasė turi tik vienos rūšies defektą.



12 pav. DAGM atvaizdų paviršių klasės [22]

Kadangi keli paviršiai yra tarpusavyje panašūs, šiam tyrimui išrenkamos 6 paviršių klasės (12 pav, apibrauktos). Atvaizdai su defektu yra augmentuojami, nes originaliai imtyje yra neproporcingai daugiau atvaizdų be defekto. Pritaikomos 90, 180, 270 laipsnių rotacijos, vertikalūs bei horizontalūs veidrodiniai atspindžiai bei rotacijų ir atspindžių kombinacijos. Galiausiai imtis paskirstoma atitinkamai: 70 % treniravimui, 20 % testavimui, 10 % validavimui. Atvaizdams su defektu priskiriama žymė – pozityvus, be defekto – negatyvus. Galutinis kiekvienos klasės atvaizdų pasiskirstymas pateikiamas 1 lentelėje.

1 lentelė. DAGM atvaizdų paviršių klasių imčių pasiskirstymas

Paviršiaus klasė	Treniravimo imtis		Validacijos imtis		Testavimo imtis	
	P	N	P	N	P	N
1	850	850	100	100	250	250
2	850	850	100	100	250	250
3	850	850	100	100	250	250
4	850	850	100	100	250	250
5	850	850	100	100	250	250
7	1680	1680	240	240	480	480
Visos klasės	5100	5100	600	600	1500	1500

Išskiriamas vienas tyrimo variantas, kuomet visos klasės naudojamos modelių apmokymui vienu metu. Kadangi ne visos klasės turi po tiek pat atvaizdų, šiuo atveju 7 paviršių klasė yra sumažinama iki 2400 atvaizdų, siekiant išlaikyti pusiausvyrą tarp visų paviršių klasių.

2.2. Modelių sudarymo kriterijai DAGM duomenų bazei

Pramoninės gamybos linijose kokybės užtikrinimui pritaikomo vaizdų atpažinimo modelio paskirtis nėra tokia pati kaip kitos jau ankstesniuose skyriuose aptartos architektūros. Šiuo atveju uždavinys nereikalauja atpažinti 1000 ar daugiau skirtingų objektų klasių kaip ImageNet konkurse. Labiau šioje srityje yra pabrėžiamas įmanomas išgauti tikslumas apmokinant modelį iš sąlyginai nedidelės duomenų imties bei modelio greitimeika. Dėl to šiame darbe tiriamas modelis turėtų atitikti tam tikrus iš anksto apibrėžtus kriterijus [23]:

1. Modelis turi būti kompaktiškas, kad galėtų būti apmokintas iš sąlyginai nedidelės duomenų bazės ir galėtų būti palaikyta pakankama greitimeika.
2. Būti atsparus trikdžiams, kuomet modelis nesunkiai būtų permokomas atlikti prognozes analogiškiems uždaviniams, neprisitaikant prie duomenų.
3. Būti suprantamas, pateikiami rezultatai turi būtų aiškūs, vadinasi taip pat turi būti pateikiama defektų lokalizacija, kad vartotojas sugebėtų suprasti modelio priimamus sprendimus.

3. Tyrimo metodika

Tyrimo metodika susideda iš 3 pagrindinių dalių, t. y. v1173 modelio apmokymo, keičiant modelio įvesties atvaizdo dimensijas, jau apmokintų tinklų pritaikymas tiriamai problemai ir U-Net apmokymo. *CSV* ir *Matplotlib* programiniai paketai vėliau naudojami rezultatų duomenų analizei.

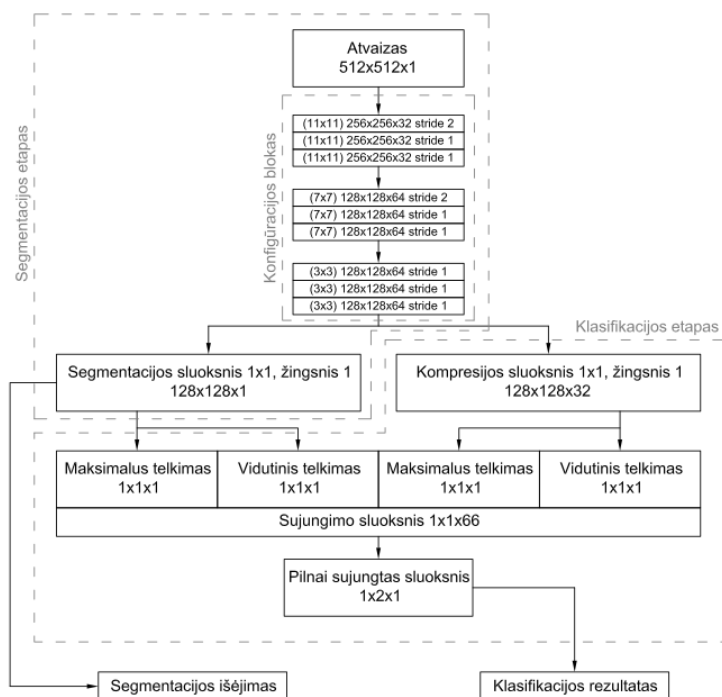
3.1. Tiriamų modelių sudarymas

Modeliai sudaromi naudojant *Python* programavimo kalbos *Keras* ir *Tensorflow* programinių funkcijų bibliotekas.

3.1.1. v1173 modelio architektūra ir pakeitimai

Šioje tyrimo dalyje bus bandoma atkurti analogišką tinklą ir rezultatus pateiktus [23]. v1173 architektūra, kaip yra aptarta ankstesniuose skyriuose, susideda iš požymių ištraukimo ir klasifikacijos dalių. Klasifikacijos dalis susideda iš 3 blokų po 3 konvoliucijos sluoksnius, kiekviename bloke mažinant filtro dydį. Klasifikacijos dalis apima 4 lygiagrečiai sujungtus maksimalaus ir vidutinio telkimo sluoksnius, po kurių seka sprendimą priimančias pilnai sujungtas sluoksnis, susidedantis iš 1 neurono.

Visi sluoksniai modelyje naudoja ReLu aktyvacijos funkciją išskyrus segmentacijos ir pilnai sujungtus sluoksnius, kuriuose pritaikoma sigmoidinė aktyvacijos funkcija.



13 pav. v1173 architektūra [23]

Esamam modeliui yra atliekamas vienas pakeitimas, t. y. maksimalaus ir vidutinio telkimo sluoksnių filtrai yra padidinami iki 2x2.

3.1.2. Apmokintų modelių architektūros ir pakeitimai

Šioje tyrimo dalyje bus bandoma utilizuoti standartinius *Keras* bibliotekoje esančius modelius, kurie jau yra apmokinti su ImageNet duomenų baze. Daroma prielaida, kad šiuos modelius bus

lengviau apmokinti tiriamoms atvaizdų klasėms, nes konvoliucijos sluoksniai jau bus išmokyti atpažinti tam tikrus požymius ir nereikės inicializuoti atsitiktinių svorių.

Kaip jau užsiminta ankstesniame skyriuje, vienas iš kriterijų yra modelio kompaktiškumas, arba sluoksnių kiekis. Tai nulemia ne tik duomenų imties mažas dydis, bet ir sąlyginai ribotų parametrų vaizdo plokštė, žinant, kad 4GM vRAM operatyviosios atminties nėra pakankama apmokinti didelius modelius kaip pvz. VGG16. Dėl to iš *Keras* jau apmokintų modelių bibliotekos išrenkami mažiausiai parametrų turintys modeliai. Parinkti modeliai pateikiami 2 lentelėje.

2 lentelė. Tiriamųjų iš anksto apmokintų modelių bendrieji parametrai [26]

Modelis	Dydis diske	Top-1 Tikslumas	Top-5 Tikslumas	Parametrų skaičius
MobileNet	16 MB	0,704	0,894	4,253,864
DenseNet121	33 MB	0,750	0,923	8,062,504
ResNet50	99 MB	0,749	0,921	25,636,712

Kaip matoma 2 lentelėje, parinktos CNN architektūros yra sąlyginai nedidelės, tačiau nepaisant to jos parodė rezultatus nenusileidžiančius ir didesniems modeliams, kaip AlexNet bei VGG16. Top-1 tikslumas nurodo modelio prognozių tikslumą atspėjant reikiamą klasę, kai Top-5 tikslumas numato, kad teisinga modelio prognozė pateko tarp 5 klasių, kurias modelis pateikė kaip labiausiai tikėtinas.

Kadangi kiekvieno modelio autoriai savaip pasiūlo priesti prie vaizdų atpažinimo problemos sprendimo, toliau bus trumpai aptariami kiekvienas iš pasirinktų iš anksto apmokintų modelių struktūrų.

MobileNet – pagrindinis pateikiamas šios modelio struktūros tikslas yra, ne taip kaip kiti giliojo mokymo modeliai, atkreipti dėmesį į modelio našumą atsižvelgiant į jo dydį ir greitaveiką. Šiame modelyje tradiciniai konvoliucijos sluoksniai pakeičiami giliai atskiriamos konvoliucijos (angl. *Depthwise Separable Convolution*) toliau – DSC, sluoksniais [27]. Dėl DSC, *Keras* aplinkoje padaugėja sluoksnių, tačiau kiekvieno atliekama operacija yra paprastesnė. Iš viso modelis sudarytas iš 88 sluoksnių [26].

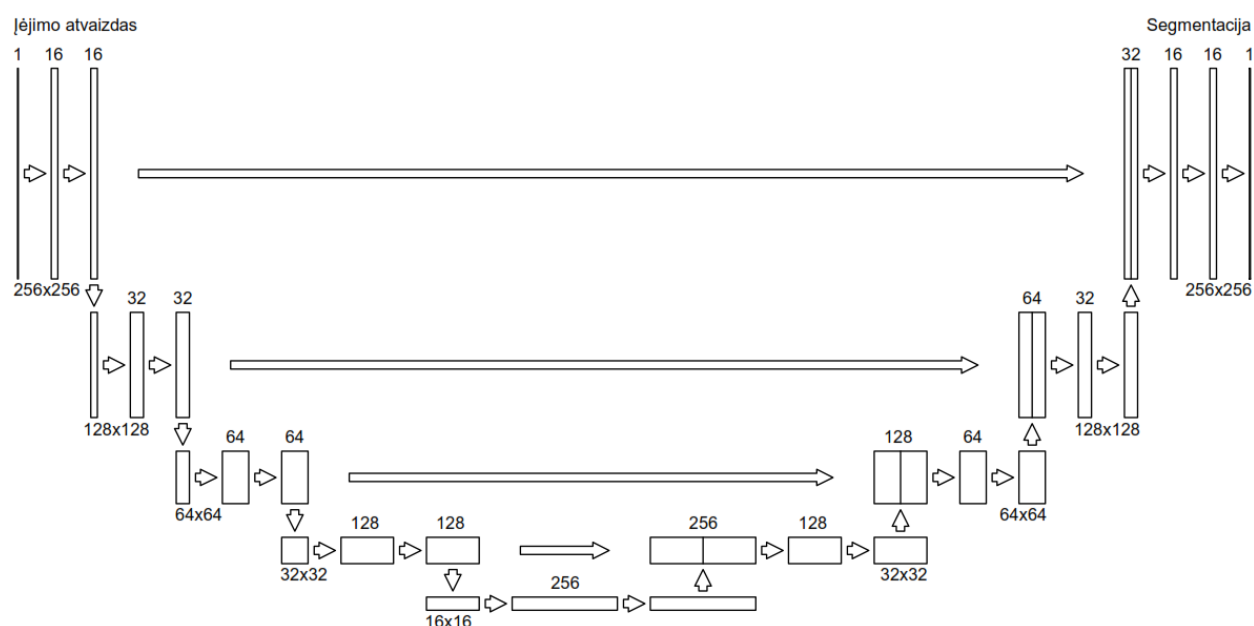
DenseNet – šio modelio autoriai savo siūloma architektūra bando išspręsti išnykstančio gradiento problemą pritaikydami kelis tankiai sujungtus konvoliucijos sluoksnius. Šiuose sluoksniuose išlaikomas pastovus požymių žemėlapių dydis bei kiekvienas konvoliucijos sluoksnis yra dalinai sujungtas su visais ankstesniais sluoksniais. Pilna modelio struktūra pateikiama [28], o *Keras* aplinkoje importuotas modelis (DenseNet121) yra sudarytas iš 121 sluoksnio [26].

ResNet – Analogiškai kaip ir DenseNet, šio modelio autoriai pritaiko tankiai sujungtus konvoliucijos sluoksnius, tačiau šiuo atveju modelio pagrindą sudaro 2 konvoliucijos sluoksniai, kuomet pirmojo įėjimas taip pat perduodamas ir susumuojamas su antrojo išėjimu [29]. Šiame tyrime naudojama ResNet50 variacija, kuri *Keras* aplinkoje turi 168 sluoksnius [26].

Norint importuoti ImageNet svorius, reikia išsaugoti originalų modelio įėjimo sluoksnį. Vadinasi visų trijų modelių atveju atvaizdo įėjimas bus 224x224 pikselių. Visiems modeliams pritaikomas analogiškos modifikacijos, t. y. išlaikomi visi sluoksniai išskyrus pasukinį pilnai sujungtą, jis pritaikomas 2 išėjimo klasėms su Softmax aktyvacijos funkcija. Žinoma įvykdžius šiuos pakeitimus kiekviename modelyje sumažėja parametrų, nes ImageNet duomenų imčiai priklauso 1000 išėjimo klasių.

3.1.3. U-Net architektūra

U-Net modelis sudaromas remiantis pilnai konvoliucinių neuroninių tinklų struktūra [20]. Kaip ir ankstesniais tyrimo atvejais įėjimo atvaizdo dydis išlaikomas panašaus dydžio – 256x256 pikselių. Pirmą atliekamos 4x2 atvaizdo konvoliucijos bei maksimalus telkimas, kol pasiekiamas tam tikras atvaizdo požymių žemėlapių dydis bei skaičius, tuomet atliekamos 4 de-konvoliucijos išplečiant atvaizdo požymių žemėlapius. Visuose sluoksniuose naudojamas 3x3 pikselių dydžio filtras konvoliucijai ir 2x2 pikselių dydžio filtras telkimui bei išplėtimui. Šiems sluoksniams naudojamos ReLu aktyvacijos funkcijos. Tyrimo naudota U-Net modelio struktūra pateikiama 14 paveiksle.



14 pav. U-Net architektūra [20]

Paskutinis sluoksnis yra tiesiog 1x1 pikselio dydžio konvoliucijos operacija, naudojant sigmoidinę aktyvacijos funkciją, kurios išėjime yra tik 1 juodai baltas požymių žemėlapis su reikiama defekto segmentacija.

3.2. Tiriamų modelių apmokymo metodika

Apmokymo proceso skaičiavimai yra atliekami su NVIDIA GeForce GTX 1050 Ti vaizdo plokšte, turinčia 4GB GDDR5 vRAM operatyviosios atminties. CUDA versija – 9,0, cuDNN – 7,2.

3.2.1. v1173 apmokymas

Pradžioje modelio svoriai yra inicializuojami su normaliu pasiskirstymu aplink 0. Toliau modelio apmokymas vykdomas 2 etapais. Pirmojo etapo metu klasifikacijos dalies svoriai yra užšaldomi, o apmokoma tik segmentacijos dalis. Taip yra atliekama 50 iteracijų. Antrojo etapo metu užšaldomi segmentacijos dalies svoriai ir apmokoma tik klasifikacijos dalis. Šiuo atveju yra atliekama 25 iteracijos. Ši metodika yra naudojama siekiant, kad klasifikacijos dalis pradėtų būti mokinama tik tada kai konvoliucijos sluoksniai jau bus išmokę ištraukti reikšmingus požymius iš atvaizdo.

Kaip patariama [23] pirmojo apmokymo etapo metu naudojama vidutinės kvadratinės paklaidos (angl. *Mean-Squared Error*) toliau – MSE, sąnaudų funkcija:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - y_i^p)^2, \quad (5)$$

čia: n – mėginių skaičius imtyje, y_i – esama tikroji vertė, y_i^p – modelio prognozuojama vertė.

Antrojo apmokymo etapo metu naudojama binarinės kryžinės entropijos (angl. *binary cross-entropy*) toliau – BCE, sąnaudų funkcija:

$$BCE = -\frac{1}{n} \sum_{i=1}^n [y_i \log(y_i^p) + (1 - y_i) \log(1 - y_i^p)]. \quad (6)$$

Abiejų etapų atveju naudojama Adadelta optimizacijos funkcija. Taip pat, norint užtikrinti rezultatų atkartojimą yra įtvirtinami atsitiktinių skaičių generatoriai, kad kiekvieną kartą iš naudo pradedant apmokinimą būtų inicializuojamos tokios pačios svorių reikšmės.

Šiuo atveju šis modelis pagal aptartą tvarką bus apmokinamas kelis kartus, keičiant įėjimo atvaizdo dimensijas ir bus tikrinama, kada prarandama defekto raiška ir kokie pasiekiami klasifikacijos rezultatai. Atvaizdo dimensijos bus keičiamos nuo 512x512 mažinant per pusę iki 32x32. Kadangi šio darbo tikslas nėra atkartoti [23] rezultatus, skaičiavimai bus atliekami tik su pirma DAGM paviršių klase.

3.2.2. MobileNet, DenseNet121 ir ResNet50 apmokymas

Kadangi šiuo atveju yra naudojami ImageNet svoriai, gauti iš oficialios *Keras* bibliotekos, kaip atspirties taškas priimama, kad modelis jau yra išmokęs ištraukti tam tikrus požymius ir dėl to apmokymas nebevykdomas 2 etapais. Žinoma taip pat reikia paminėti, kad nors ir modelis kai kuriuos požymius jau yra išmokintas ištraukti nereiškia, kad tai pravers tiriamajai duomenų bazei, nes nieko panašaus į tiriamus atvaizdus ImageNet konkurse nėra. Dėl to apmokymo metu neužšaldomi jokie sluoksniai ir svoriai gali būti atnaujinami visame modelyje. Pirma apmokoma kiekviena paviršiaus klasė atskirai, tada visos kartu, importuojant ImageNet svorius, ir galiausiai modeliai apmokinami neimportuojant ImageNet svorių su standartiniu pasiskirstymu aplink 0. Kiekvienu atveju yra atliekama 30 iteracijų išskyrus, kai neimportuojami ImageNet svoriai, tuomet yra atliekama 60 iteracijų.

Visiems modeliams naudojama kategorinė kryžminė entropijos (angl. *Categorical cross entropy*) toliau – CCE, sąnaudų funkcija [30]:

$$CCE = -\sum_{i=1}^n y_i \log(y_i^p). \quad (7)$$

Taip pat su sąnaudų funkcija naudojama Adam optimizacijos funkcija. Dėl vaizdo plokštės atminties apribojimų atvaizdai apmokymo metu į modelį įkeliami partijomis MobileNet – 10, DenseNet121 – 5, ResNet50 – 2.

3.2.3. U-Net apmokymas

U-Net sudaromas kaip naujas modelis, vadinasi svoriai inicializuojami su standartiniu pasiskirstymu aplink 0. Atvaizdai neaugmentuojami (tiriami tik originalūs 150 atvaizdų su defektu),

nes literatūroje [20] nurodyta, kad ši architektūra yra pajėgi būti apmokyta iš nedidelės duomenų imties. Modelio tikslumui įvertinti naudojamos atvaizdų su defektu anotacijos, kuriose juodame fone baltu ovalu pažymėti defektai. Kadangi į pažymėtą ovalą įeina ne tik pačio defekto pikseliai bet ir jį supantys gero paviršiaus pikseliai numatoma, kad tai turės įtakos modelio apmokymo procesui ir modelio tikslumo rezultatams.

Apmokymas vyksta su BCE sąnaudų funkcija ir Adam optimizacijos funkcija. Kiekvienos paviršių klasės apmokymas trunka 100 iteracijų, apribojant atvaizdų partijos dydį iki 5. Dėl nedidelio apmokymo duomenų kiekio testavimo imtis yra nesudaroma, o galutinis modelio tikslumas bus įvertinamas atsižvelgiant į validacijos imties (10 atsitiktinai parenkamų atvaizdų iš originalių 150) tikslumą.

3.3. Tikslumo įvertinimo kriterijai

Įvertinimo kriterijai parenkami analogiškai, kurie buvo naudojami 2.2 poskyryje aptartuose tiriamuosiuose darbuose, t. y. atvejams, kuomet apmokinami modeliai su 1 paviršių klase įvertinamos tikrosios teigiamos vertės (angl. *True Positive Rate*), toliau – TPR bei tikrosios neigiamos vertės (angl. *True Negative Rate*), toliau – TNR, parametrai [31].

$$TPR = \frac{TP}{P}, \quad (8)$$

čia: TP – teisingai atpažintos teigiamos reikšmės, P – visos imtyje esančios teigiamos reikšmės.

$$TNR = \frac{TN}{N}, \quad (9)$$

čia: TN – teisingai atpažintos neigiamos reikšmės, P – visos imtyje esančios neigiamos reikšmės.

Atvejams, kuomet apmokomos visos klasės sudaromos susimaišymo matricos. Abiejų atvejų metu taip pat bus pateikiamos absoliučios tikslumo reikšmės. Tuomet bus galima tiesiogiai palyginti pasiektus tiriamųjų modelių tikslumo rezultatus su pateiktais literatūroje.

4. Tyrimo rezultatai

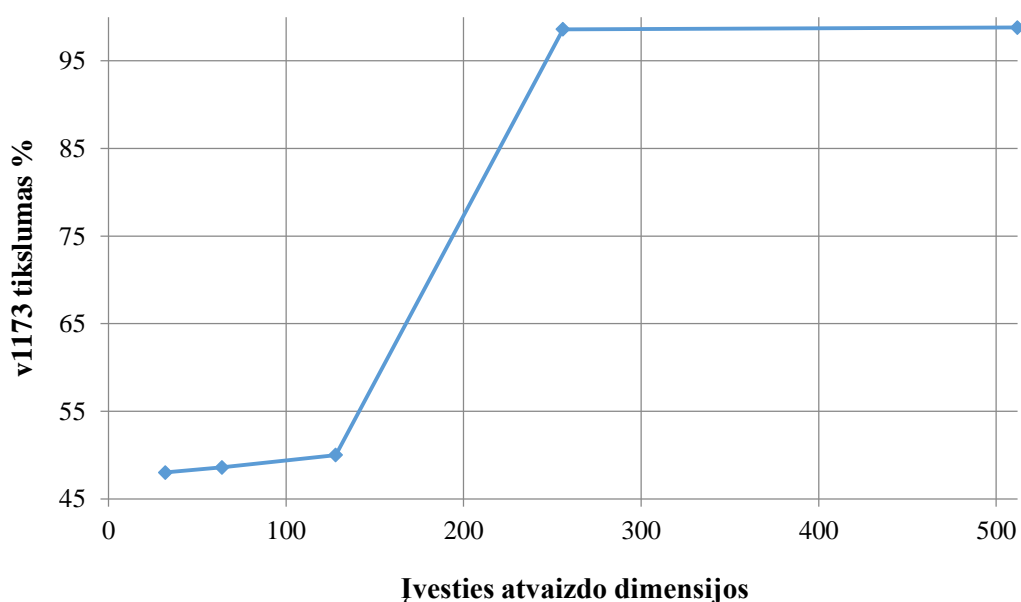
Šiame skyriuje yra pateikiami gauti modelių tikslumo rezultatai, apmokymo metu gautos sąnaudų funkcijų ir tikslumo kreivės bei defektų lokalizacijos ir segmentacijos rezultatai pasiekti su pavyzdiniais defektiniais atvaizdais.

4.1. Rezultatų jautrumo nustatymas keičiant įėjimo atvaizdo dimensijas

Atvaizdo dimensijos buvo keičiamos nuo 512x512 pikselių, mažinant per pusę iki 32x32 pikselių. Iš viso modelis apmokinamas 5 kartus su pirmąja paviršių klase. Pasiekti rezultatai pateikiami 3 lentelėje.

3 lentelė. v1173 tikslumo rezultatai pirmajai paviršių klasei

Dimensijos	32	64	128	256	512
Tikslumas, %	48	48,6	50	98,6	98,8

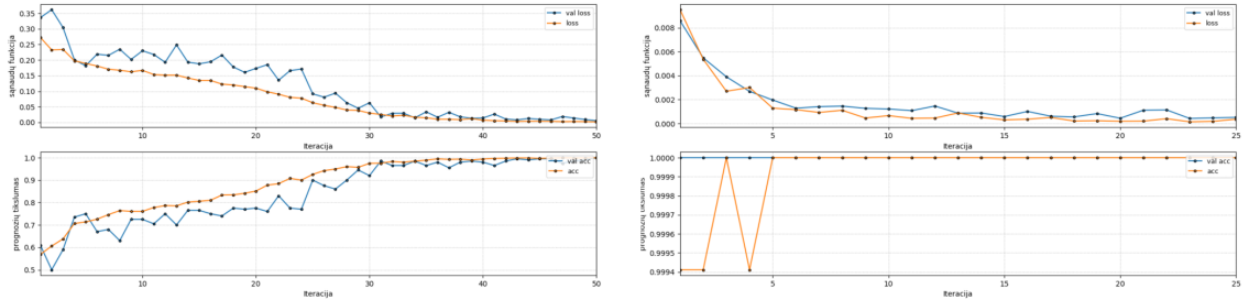


15 pav. v1173 priklausomybė nuo įvesties atvaizdo dimensijų

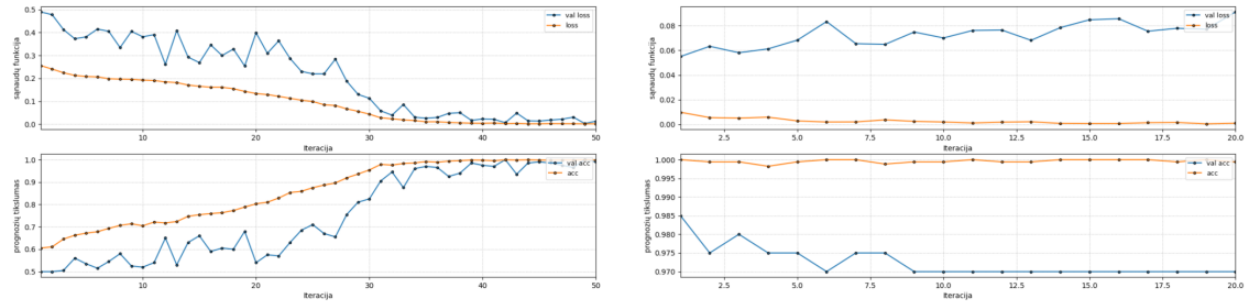
Remiantis 3 lentelės duomenimis ir 15 paveikslu matoma, kad kaip ir tikėtasi mažinant įėjimo atvaizdo rezoliuciją taip pat mažėja ir išgaunamas modelio tikslumas. Defekto raiška prarandama 200-300 pikselių riboje. Toliau pateikiamos tikslumo ir sąnaudų funkcijų reikšmės apmokymo proceso metu.

16 paveiksle galima įžvelgti analogišką tendenciją, kuri matėsi ir 15 paveiksle, t. y. mažinant įėjimo atvaizdo rezoliuciją apmokymo procesas tampa vis labiau chaotiškas. 512 ir 256 rezoliucijų atveju matoma, kad kreivės konverguoja panašiai – 30 iteracijos metu, bet aiškiai matosi, kad validacijos imties rezultatai vis labiau atsilieka nuo treniravimo rezultatų. Vadinasi įžvelgiamas vis didesnis modelio prisitaikymas prie treniravimo imties duomenų.

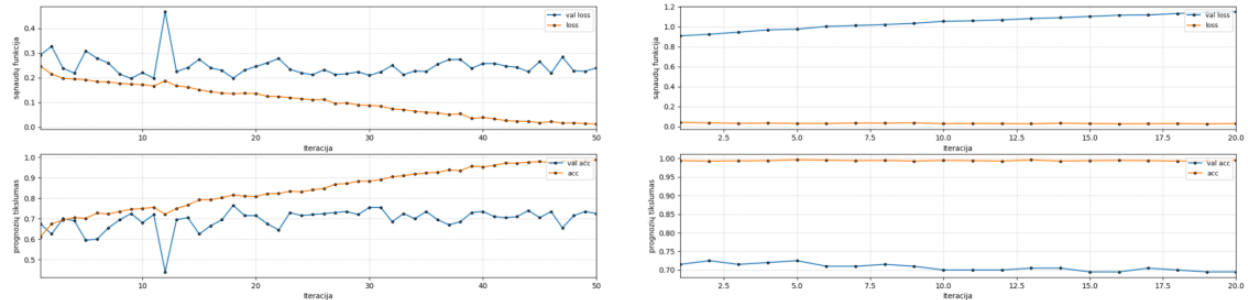
512x512



256x256



128x128



16 pav. v1173 apmokimo tikslumo ir sąnaudų kreivės

Galiausiai iš šių rezultatų galima spręsti, kad galima panaudoti jau apmokintus modelius, kurių įėjimas yra fiksuotas 200-300 pikselių intervale, siekiant išgauti geresnius rezultatus.

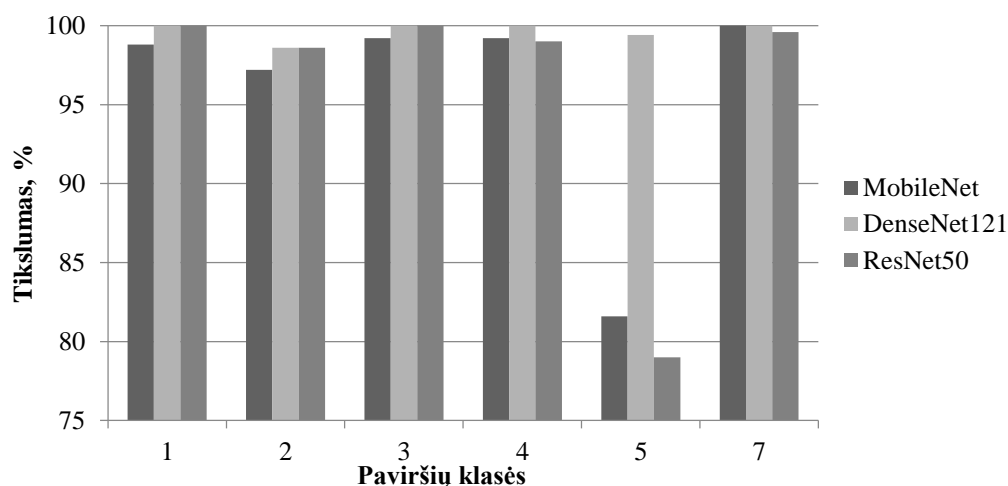
4.2. MobileNet, DenseNet121 ir ResNet50 rezultatai

Šiuo atveju atrinkti tyrimui jau apmokinti modeliai ImageNet konkursui. Esami svoriai importuojami į modelį ir jis yra toliau apmokinamas su tiriamąja atvaizdų imtimi. Pirmą kiekvienas modelis buvo apmokintas su kiekviena paviršių klase atskirai, o vėliau su visomis paviršių klasėmis vienu metu. Taip pat atliktas apmokinimas su visomis paviršių klasėmis ir neimportuojant ImageNet svorių, norint palyginti skirtumą tarp gautų rezultatų. Pasiękti rezultatai pateikiami 4 lentelėje ir palyginami su gautaisiais literatūroje [23].

4 lentelė. Keras bibliotekos modelių tikslumo rezultatai

Paviršius	Tikslumas, %			
	MobileNet	DenseNet121	ResNet50	v1173 [23]
1	98,8	100	100	99,4
2	97,2	98,6	98,6	99,9
3	99,2	100	100	98,2
4	99,2	100	99	99,2
5	81,6	99,4	79	100
7	100	100	99,6	100
Vidurkis	96,0	99,7	96,0	99,4

Iš 4 lentelės matoma, kad bendras vidurkis tarp visų paviršių klasių didžiausias yra DenseNet121, nepaisant to, kad ResNet50 turi 3 kartus daugiau parametų. Analogiškos išvalgos galioja ir atvejams kuomet visos paviršių klasės apmokinamos vienu metu. Norint geriau matyti skirtumą tarp rezultatų remiantis 4 lentelės duomenimis sudaroma stulpelinė diagrama (17 pav.).

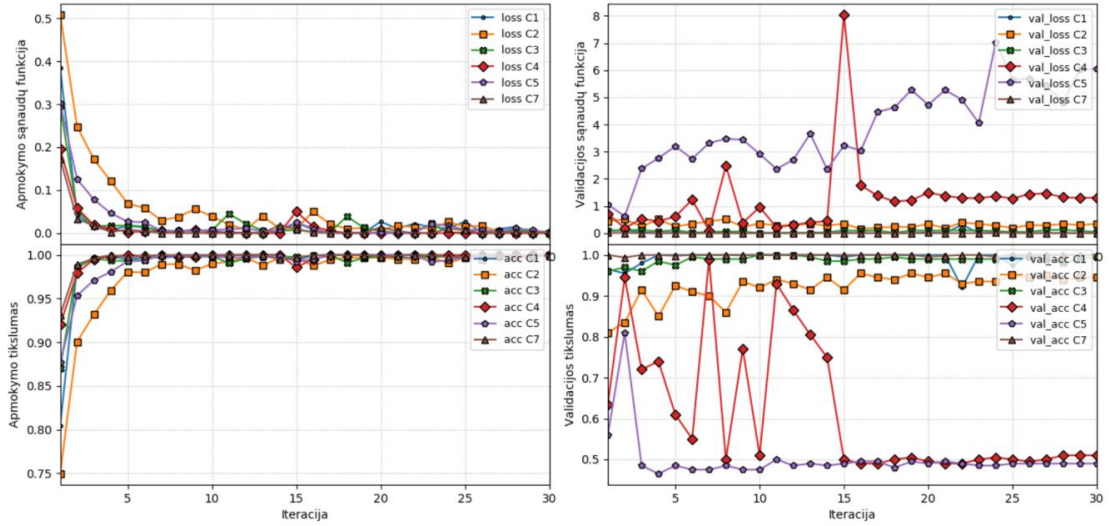


17 pav. Apmokintų modelių tikslumas skirtingoms paviršių klasėms

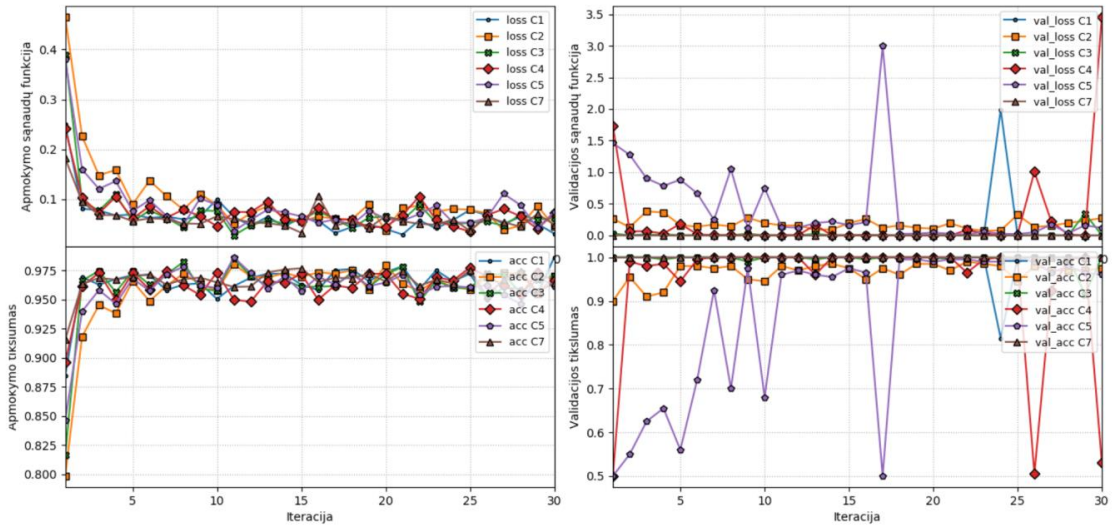
17 paveikslas geriau parodo skirtumą tarp individualiai apmokintų paviršių klasių. Visumoje matoma, kad modeliai pasiekia ne tik gerus, bet ir pastovius rezultatus visiems paviršiams, išskyrus 5 klasei, tačiau net kuomet MobileNet ir ResNet50 rezultatai smuko iki apytiksliai 80%, DenseNet121 pasiekė 99,4%. Šie rezultatai taip pat atsispindi ir sąnaudų funkcijos bei tikslumo kreivėse, kurios buvo išmatuotos apmokymo metu ir yra pateiktos 18 paveiksle.

Iš duomenų, surinktų apmokymo metu matoma, kad treniravimo imtį visi modeliai sugeba priversti pasiekti arti 100% tikslumą, tačiau validacijos kreivės yra kur kas labiau chaotiškos. Tik DenseNet121 atveju validacijos kreivės parodo laikinus nukrypimus, o MobileNet ir ResNet50 tik kelias paviršių klases sugeba priversti arti 100% tikslumo ribos. Taip pat nenuostabu, kad penktoji paviršių klasė rodo didžiausią nestabilumą apmokymo metu validacijos imtyje.

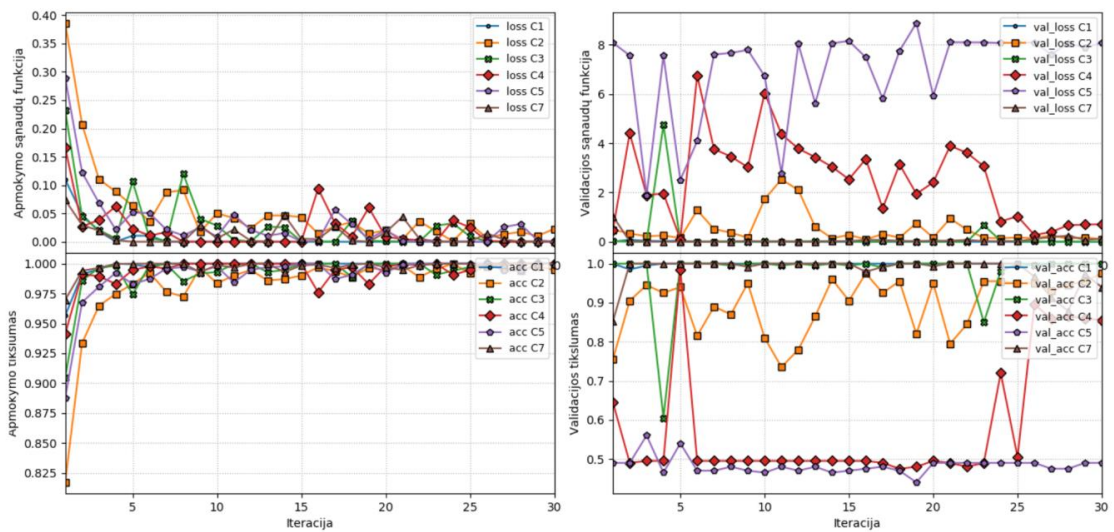
Mobilenet



DenseNet121



ResNet50



18 pav. Apmokintų modelių tikslumo ir sąnaudų kreivės kiekvienai klasei

Siekiant geriau suprasti gautus rezultatus taip pat yra įvertinami TPR ir TNR parametrai.

5 lentelė. Apmokintų modelių TPR ir TNR reikšmės

Paviršius	MobileNet		DenseNet121		ResNet50		v1173 [23]	
	TPR	TNR	TPR	TNR	TPR	TNR	TPR	TNR
1	97,6	100	100	100	100	100	100	98,8
2	100	94,4	99,6	97,6	100	97,2	100	99,8
3	100	98,4	100	100	100	100	100	96,3
4	99,6	98,8	100	100	98	100	99,8	99,8
5	98,8	64,4	100	98,8	68	88,8	100	100
7	100	100	100	100	99,2	100	100	100
Parametru skaičius	3,230,914		7,039,554		23,591,810		1,133,732	

TPR ir TNR parametrai parodo tai, ko neparodo bendri tikslumo rodikliai, t. y. modelio kiekvienos išėjimo klasės prognozes. Iš 5 lentelės atrodo, kad nors ir MobileNet bei ResNet50 pasiekė vidutiniškai lygius rezultatus – 96% tikslumą, abu modeliai padarė klaidas skirtingose vietose. Matoma, kad MobileNet modelis iš viso 129 kartus Neigiamą išėjimo klasės atvaizdą palaikė teigiamu, o ResNet50 89 kartus teigiamos išėjimo klasės atvaizdą palaikė neigiamu.

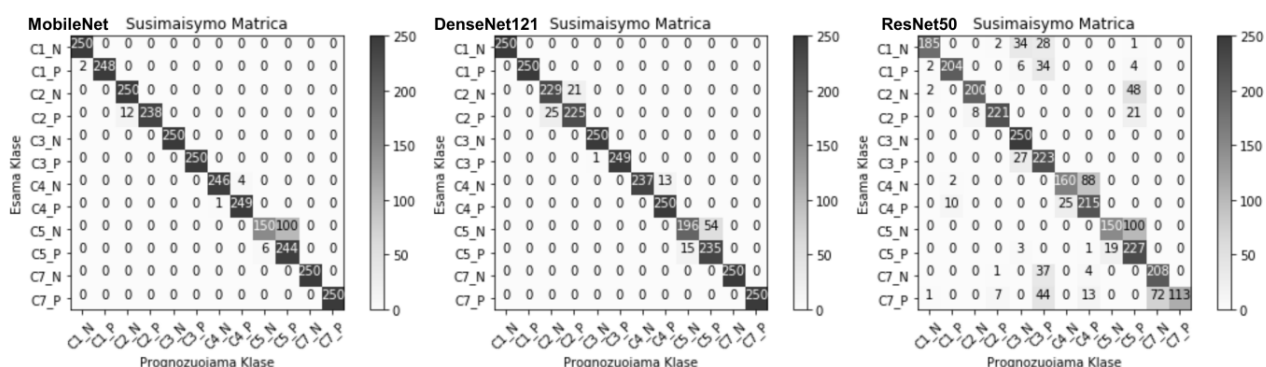
4.3. Visų paviršių klasių apmokinimas vienu metu

Šiuo atveju modelis apmokinamas pažinti visas tiriamas paviršių klases vienu metu, t. y. modelis turės atskirti ne tik ar defektas yra ant paviršiaus, bet taip pat ir kuriai paviršių klasei tiriamas atvaizdas priklauso. Apmokymas vykdomas importuojant ImageNet svorius.

6 lentelė. Visų paviršių klasių apmokinimo vienu metu rezultatai.

Modelis	MobileNet	DenseNet121	ResNet50
Tikslumas, %	95,8	95,7	78,5

Kadangi apmokinant visas paviršių klases vienu metu gaunamos iš viso 12 modelio išėjimo klasių, rezultatų interpretacijai sudaromos susimaišymo matricos.



19 pav. Susimaišymo matricos apmokinant visas paviršių klases su ImageNet saugyklos svoriais

Tolimesnes įžvalgas galima daryti remiantis 19 paveikslu. Kartu MobileNet ir DenseNet121 modeliai yra kelis kartus paprastesni nei ResNet50, tačiau parodo geresnius rezultatus apmokinant visas paviršių klases vienu metu. Šiuo atveju MobileNet ir DenseNet121 pasirodė panašiai, rezultatai tarp jų skiriasi tik per vieną dešimtąją procento. Taip pat matoma iš šių modelių rezultatų,

kad testuojant modeliai gali atskirti ne tik ar defektas yra ant paviršiaus bet ir pilnai atskirti koks tai paviršius yra, t. y. paviršių klasės tarpusavyje visiškai nesimaišo. To negalima pasakyti apie ResNet50, kuris pasiekė tik 75,8% prognozių tikslumą. iš rezultatų priimama, kad šis modelis netinkamas daugelio išėjimo klasių apmokymui.

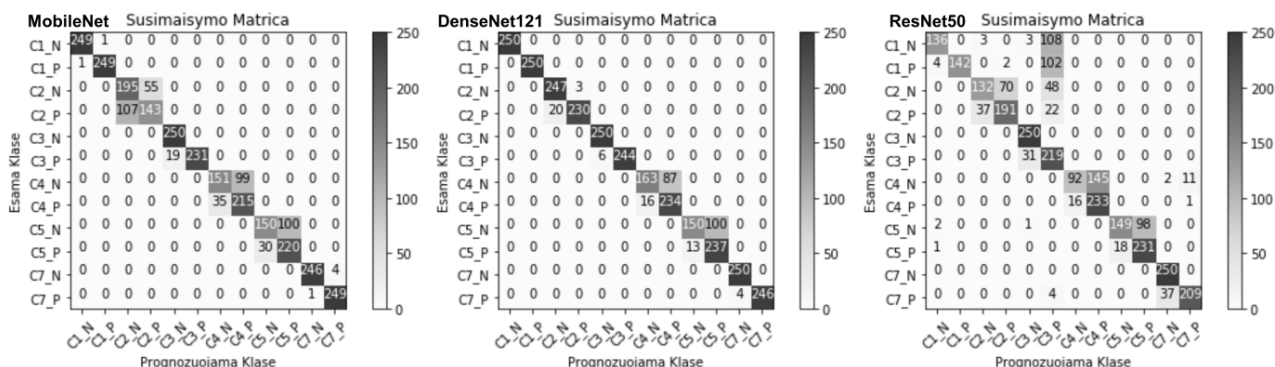
Kaip ir apmokinant modelius su viena paviršiaus klase, matoma kad modeliai turėjo daugiausiai problemų klasifikuojant 5 paviršiaus klase. Be to kelios klasifikacijos klaidos pastebimos ir su 2 paviršiaus klase, tačiau jos yra nežymios.

Toliau atliekamas panašus tyrimas, t.y. naudojamos visos paviršių klasės, bet ImageNet svoriai neimportuojami. Sąnaudų ir optimizacijos funkcijos paliekamos tokios pačios kaip ir ankstesniuose tyrimuose, tačiau šį kartą atliekama 60 iteracijų. Pasiekti rezultatai pateikiami 7 lentelėje.

7 lentelė. Visų paviršių klasių apmokinimo vienu metu rezultatai be ImageNet saugyklos svorių

Modelis	MobileNet	DenseNet121	ResNet50
Tikslumas, %	84,9	91,7	74,5

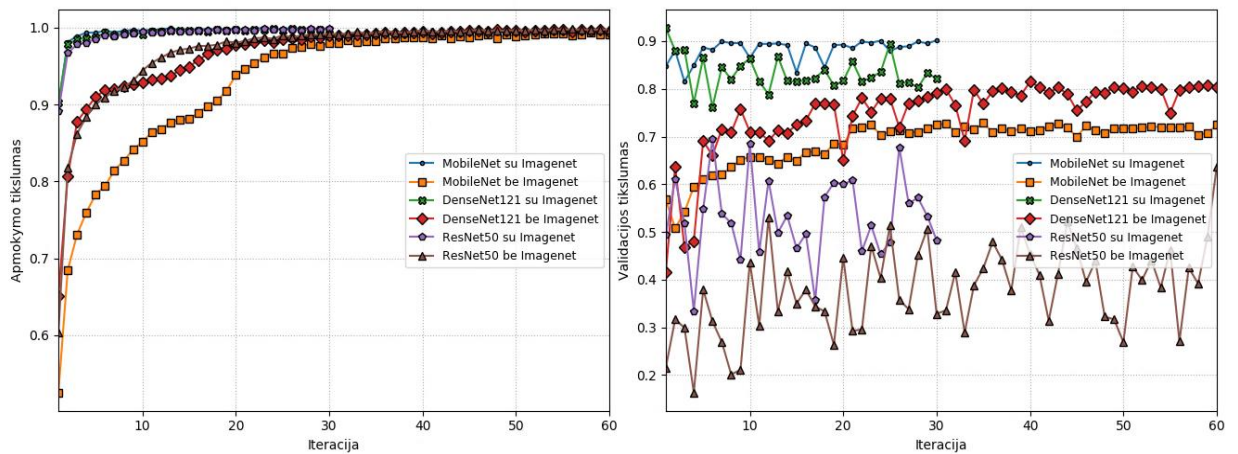
Po 60 iteracijų matoma, kad modeliams nepavyko pasiekti anksčiau parodytų rezultatų. Tačiau DenseNet121 parodė arčiausiai ankstesniems rezultatams pasiektą tikslumą, t.y. tik 4% mažiau nei importuojant ImageNet svorius. 20 paveiksle pateikiamos šių rezultatų susimaišymo matricos.



20 pav. Susimaišymo matricos apmokinant visas paviršių klases be ImageNet saugyklos svorių

Šiuo atveju matoma panaši situacija kaip ir importavus ImageNet svorius. MobileNet ir DenseNet121 vis dar sugeba puikiai atskirti skirtingas paviršių klases, tačiau dabar matoma daugiau neteisingų prognozių klasifikuojant defektus. Išryškėja modelių problemos klasifikuojant antrąją ir trečiąją paviršių klases. ResNet50 kaip ir ankstesniame tyrime nesugeba atskirti ne tik paviršių klasių tarpusavyje bet ir suklasifikuoti defektų.

Nenuostabu, kad neimportuojant ImageNet svorių apmokinimas turėtų užtrukti ilgiau, kaip tik tai ir pastebima 21 paveiksle. Treniravimo imties kreivėje matoma, kad neimportuojant ImageNet svorių, tikslumas priartėja prie 99-100% ribos tik pasiekus 30 iteraciją. Atmetant ResNet50 duomenis validacijos imties kreivės rodo, kad importavus ImageNet svorius 80-90% tikslumas pasiekiamas jau nuo pirmosios iteracijos, kuomet neimportuojant šių svorių tikslumas nusistovi 70-80% ribose viso mokymo metu.



21 pav. Visų paviršių klasių apmokinimas be ImageNet saugyklos svorių

Modelių apmokymo tikslumo kreivės parodo ne visą informaciją apie modelio veikimą, taip pat reikia atsižvelgti į modelio sudėtingumą, dėl šios priežasties taip pat yra įvertinamas modelių apmokymo greitis, tyrimo metu išmatuoti rezultatai pateikiami 8 lentelėje.

8 lentelė. Tiriamų modelių apmokymo iteracijų trukmės

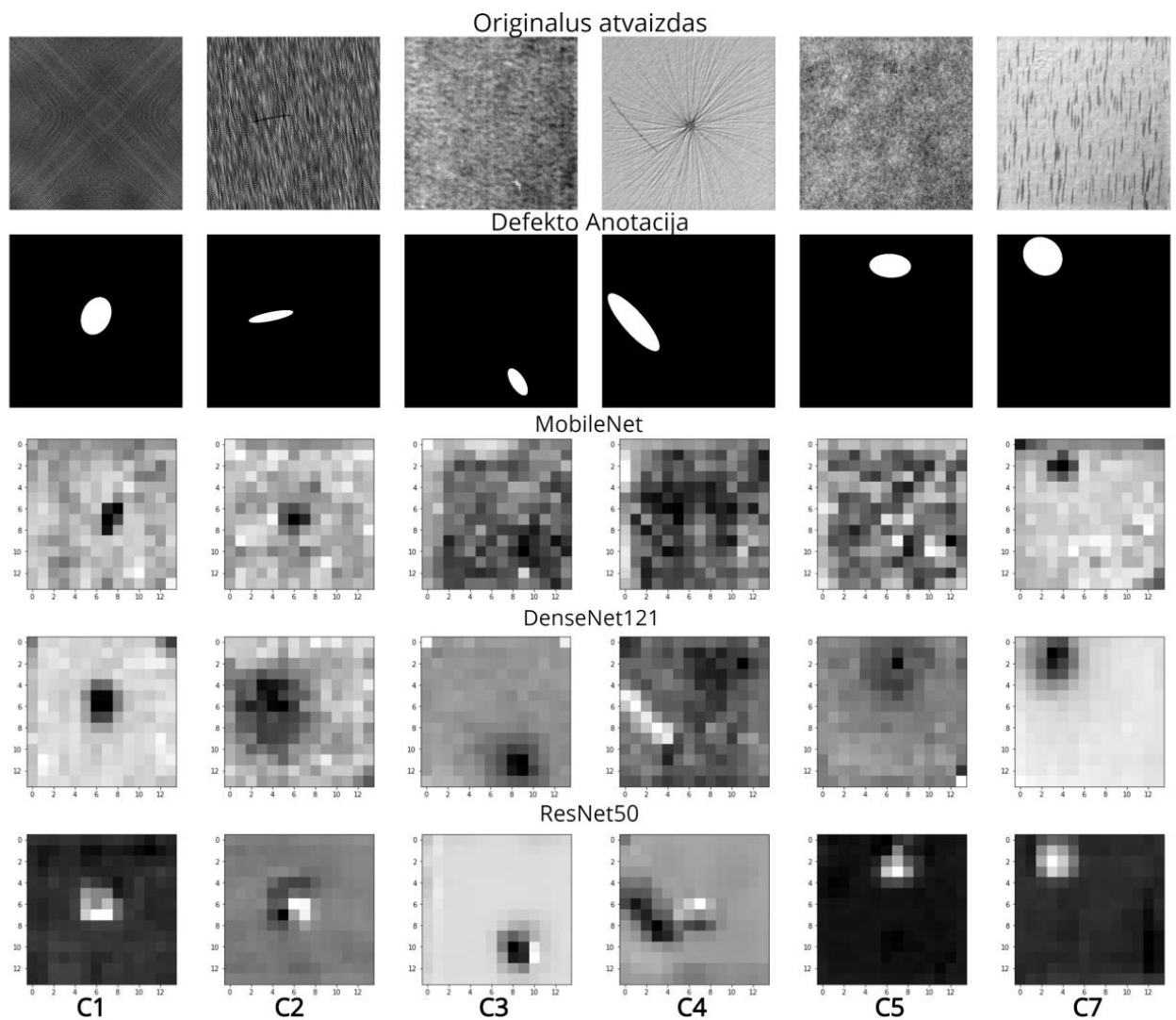
	Vidutinė iteracijos trukmė, s			
	MobileNet	DenseNet121	ResNet50	v1173
Klasės 1-5	42	211	325	270

Kaip matoma 8 lentelėje, kiekvieno modelio apmokymo trukmė taip pat priklauso nuo iteracijos trukmės, kuri tiesiogiai priklauso nuo parametrų skaičiaus modelyje. Didėjant modelio sudėtingumui taip pat ir ilgėja modelio apmokymo procesas, dėl tuo nenuostabu, kad MobileNet vienos iteracijos trukmė yra mažesnė negu DenseNet121 ir ResNet50. Tačiau v1173 apmokymas užtruko ilgiau ne tik dėl didesnio reikalingo iteracijų skaičiaus bet taip pat ir dėl iteracijų apdorojimo trukmės, kuri buvo mažesnė tik lyginant su ResNet50.

4.4. Požymių žemėlapiai

Remiantis jau ankstesniuose skyriuose aptartais apmokintais modeliais matoma, kad pakankamas tikslumas gali būti pasiekiamas, norint juos pritaikyti pramonėje. Žinoma taip pat reikia pabrėžti, kad galutinis modelio klasifikacijos rezultatas nėra pakankama informacija, norint gamybos linijos operatoriui pašalinti brokuotą detalę. Vienas iš išvardintų kriterijų šio darbo pradžioje apibrėžiantis neuroninių tinklų taikymą industrinės gamybos reikmėms specifikavo, kad modelis turi būti suprantamas ir jo rezultatai nesunkiai interpretuojami. Dėl šios priežasties defektų lokalizacijos funkcija taip pat turi būti integruota į modelį.

Vienas iš paprasčiausių šios funkcijos įgyvendinimo būdų yra panaudoti jau pačiame modelyje esančią informaciją. Norint tai atlikti, yra išrenkamas vienas atvaizdas iš kiekvienos tiriamos paviršių klasės ir įvedamas į jau apmokytus modelius. Modeliui jau įvykdžius klasifikaciją ištraukiamas prieš paskutinio konvoliucijos sluoksnio 14x14 pikselių dydžio požymių žemėlapis (22 pav.).



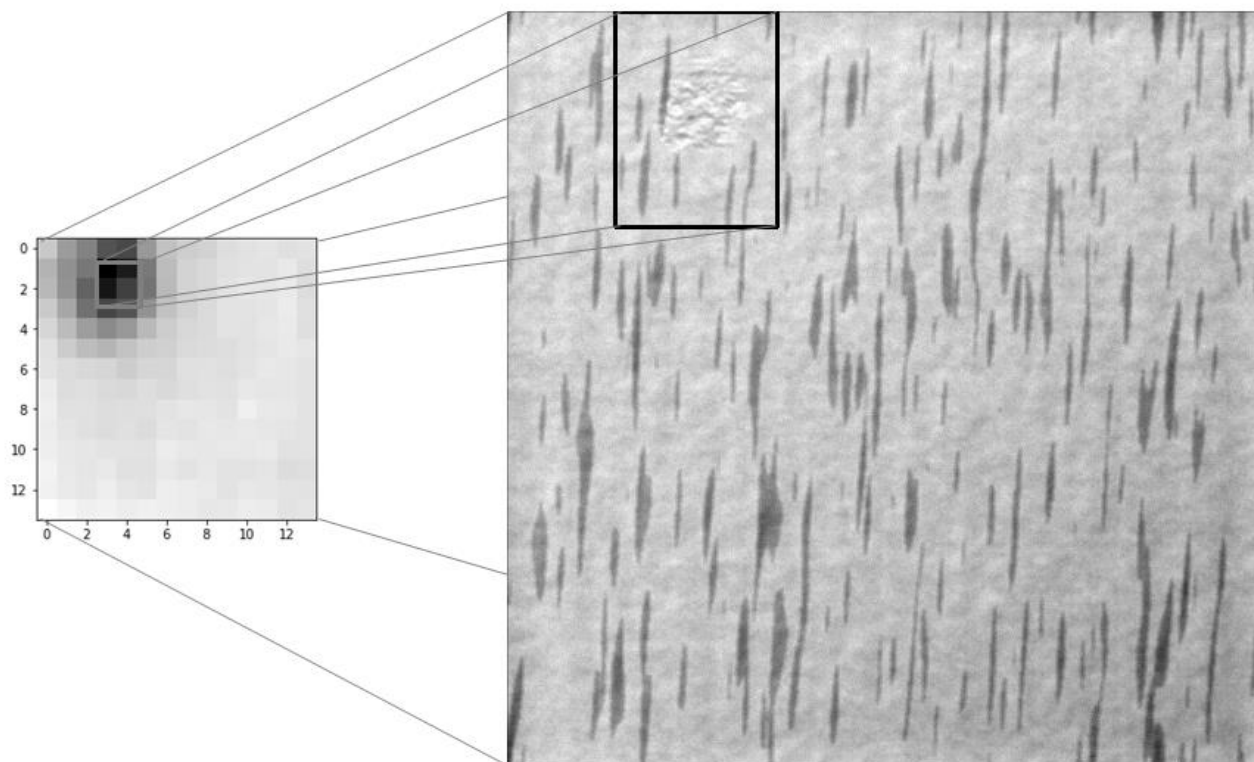
22 pav. Tiriamų modelių požymių žemėlapiai

Iš 22 paveiksle matomų rezultatų galima daryti išvadą, kad modeliai iš tiesų išmoko reikiamus požymius. Defekto anotacijos atvaizdai stipriai koreliuoja su išgautais požymių žemėlapiams, ypač DenseNet121 ir ResNet50 modelių atveju. Įdomu pastebėti tai, kad defektai tiriamuose modeliuose išskiriami su skirtingais pikselių intensyvumais, ypač ResNet50 atveju matoma, kad defektai išskiriami balta spalva, kuomet kiti 2 modeliai juos išskyrė juoda bei tamsiai pilka spalvomis. Išskirtinis atvejis yra DenseNet121 ketvirtoji paviršių klasė išmokusi išskirti defektą taip pat balta spalva.

4.5. Defektų lokalizacija

Kaip matoma iš 22 paveikslo, tiriamų modelių paskutinių sluoksnių požymių žemėlapiai stipriai koreliuoja su originalaus atvaizdo defektu. Tuo galima pasinaudoti, sudarant apribojančią geometrinę figūrą aplink patį defektą norint išskirti reikiamą regioną.

Tai galima atlikti nustačius tam tikrą svertinę vertę, kuri palyginama su kiekvienu požymio žemėlapio pikselio verte. Jeigu tiriamasis pikselis yra aukščiau nustatytos vertės jo koordinatės yra įrašomos. Tokiu būdu galima nustatyti ribojančios geometrijos viršutinę kairiąją ribą bei apatinę dešiniąją ribą. Žinant šias ribas galima jas proporcingai perskaičiuoti originaliam atvaizdo dydžiui – 512x512 ir suprojektuoti ribojančią geometrinę figūrą ant modelio įėjimo atvaizdo (23 pav.).



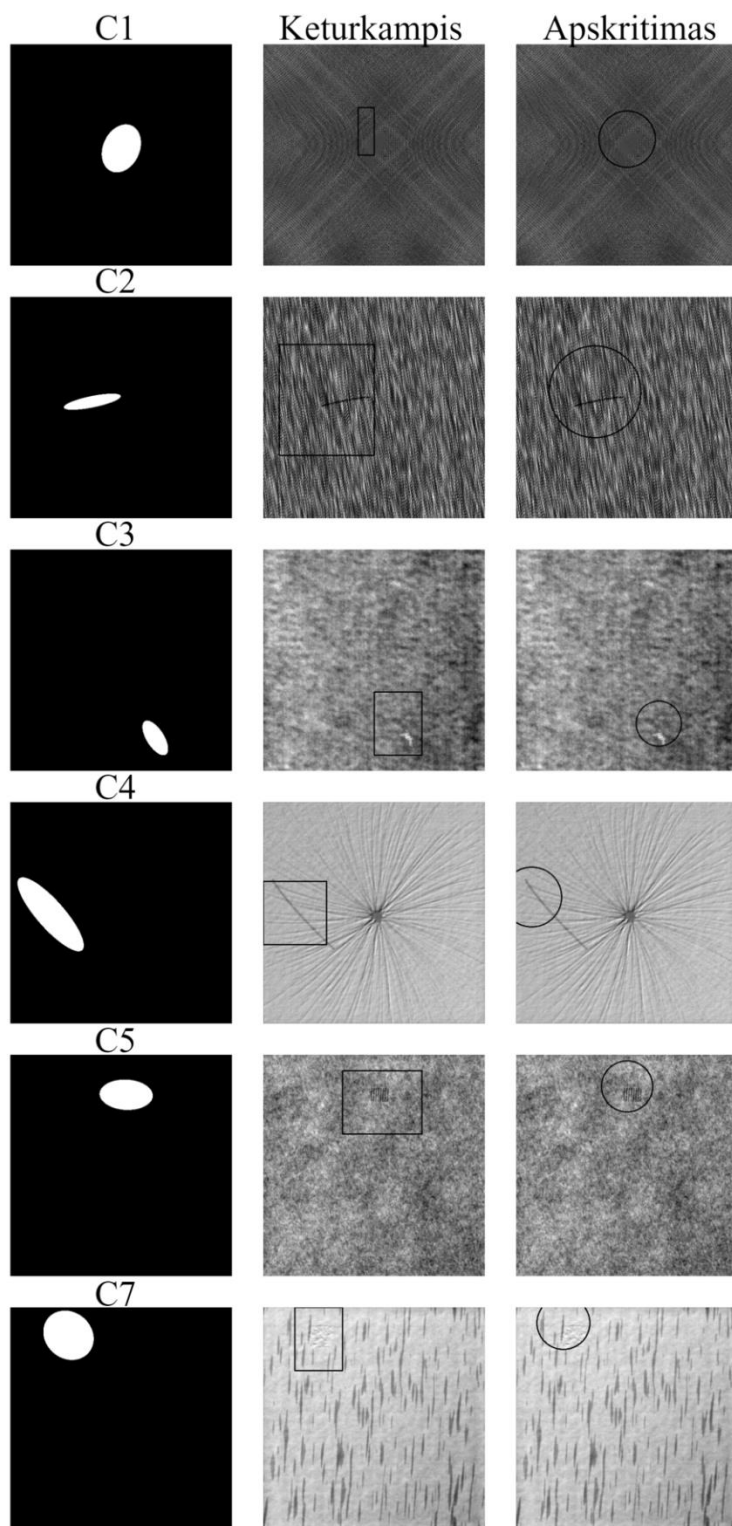
23 pav. Ribojančios geometrijos projekcija ant originalaus atvaizdo

Ištraukus požymių žemėlapius ribojančių geometrinių figūrų atvaizdavimui naudojama OpenCV bibliotekos `rect()` ir `circle()` funkcijos, kurios priima kaip argumentus jau žinomus dydžius – viršutinio kairiojo ir apatinio dešiniojo kampų koordinates.

Kaip pavyzdys parenkamas DenseNet121 modelis ir pasinaudojant ankstesniuose skyriuose gautais kiekvienos tiramos paviršių klasės apmokinto modelio svoriais gaunamos ribojančios geometrinės figūros, siekiant lokalizuoti paviršiaus defektą, gauti rezultatai pateikiami 24 paveiksle.

išskiriamos 2 geometrinės figūros – keturkampio ir apskritimo. Galima pastebėti, kad defektams panašiams į ilgos formos įbrėžimą geriau tinka keturkampio formos geometrinės figūros, nes apgaubia didesnę defekto dalis, tačiau mažesniems defektams labiau tinka apskritimo forma, kuri yra panašesnė į pateiktą defekto anotaciją ir apgaubia mažiau nereikalingų pikselių.

Iš 24 paveikslo taip pat galima daryti išvadą, kad ribojančių geometrinių figūrų tikslumas apgaubiant defektą yra sąlygojamas mažos raiškos požymių žemėlapių ir didesnis tikslumas būtų išgaunamas tik derinant svertines vertes ir užregistruojant defektų pikselius požymių žemėlapiuose. Tačiau tai apsunkintų ir prailgintų modelio sudarymo procesą, kuomet pasitelkiant gilų mokymą siekiama visą šį procesą automatizuoti.

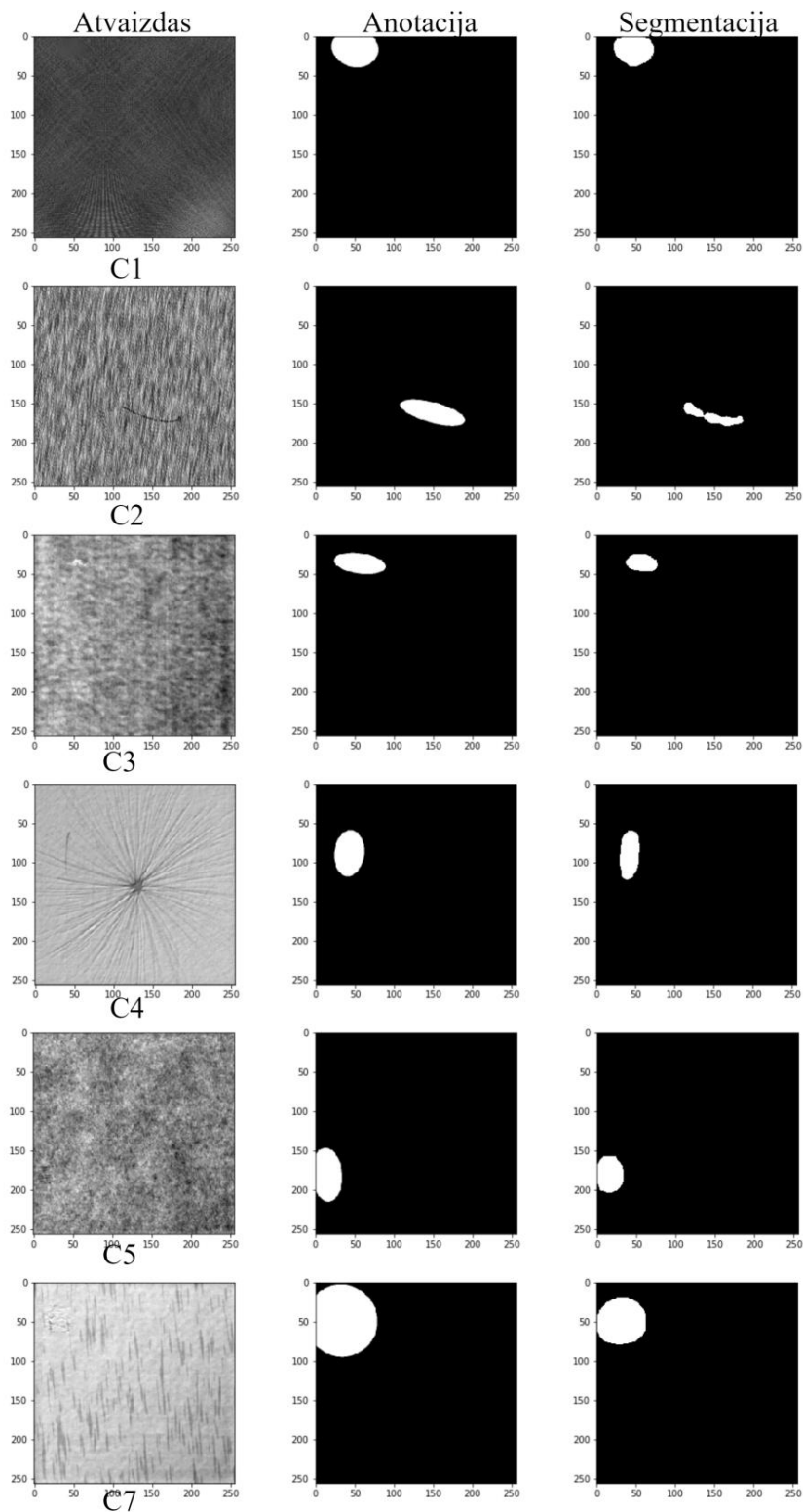


24 pav. DenseNet121 tiriamų klasių defektų ribojančios geometrijos

Nors šis defektų lokalizacijos būdas yra gan paprastas, tačiau jis nereikalauja papildomų skaičiavimų, nes iš modelio ištraukiama jau apdorota informacija ir su ja atliekamos tik tiesinės operacijos.

4.6. Defektų segmentacija

Pilnai konvoliucinių neuroninių tinklų tikslumas paprastai yra įvertinamas remiantis anotaciniais atvaizdais, palyginant kaip tiksliai tinklui pavyko išskirti tiriamus objektų pikselius. Tačiau iš rezultatų pateiktų 24 paveiksle matoma, kad U-Net po 100 iteracijų daugeliu atvejų pavyko pasiekti geresnius segmentacijos rezultatus nei išskirta anotacijose.



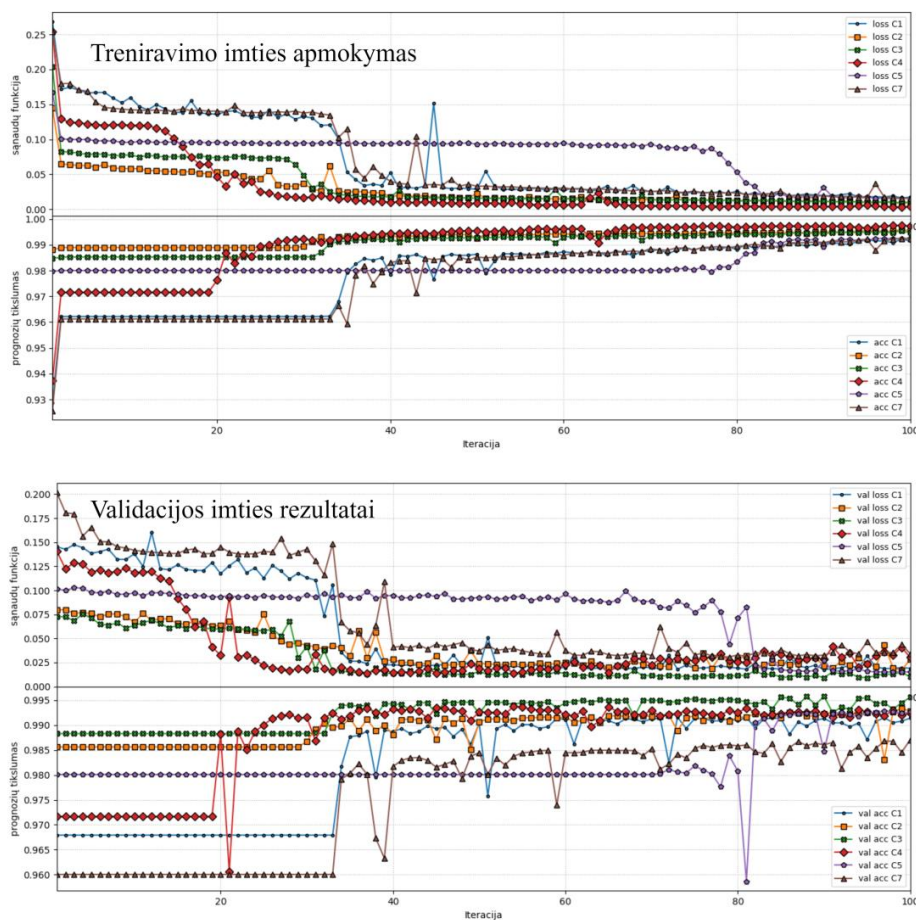
25 pav. U-Net tiriamų klasių defektų segmentacijos

Remiantis 25 paveikslu matoma, kad beveik kiekvieno tiriamo paviršiaus atveju galima pastebėti, kad U-Net pavyko išskirti defektą, apimant mažiau pikselių be defekto nei pažymėta anotacijose. U-Net pasiekti rezultatai pateikiami 9 lentelėje.

9 lentelė. U-Net pasiektų tikslumo rezultatų suvestinė

Paviršius	Treniravimo tikslumas, %	Validacijos tikslumas, %	Vidutinė iteracijos trukmė, s
1	99,15	99,14	4
2	99,55	99,28	4
3	99,47	99,56	4
4	99,73	99,24	4
5	99,25	99,26	4
7	99,23	98,70	9
Vidurkis	99,40	99,20	–

Iš 9 lentelės matoma, kad U-Net pasiekti rezultatai yra panašūs į DenseNet121 bei literatūroje [23] pateiktais v1173 rezultatus. Nepastebėta jokio tikslumo sumažėjimo tiriant 5 paviršių klasę kaip su MobileNet bei ResNet50 modeliais. Vis dėlto reikia pabrėžti, kad U-Net buvo apmokintas neaugmentuojant duomenų ir vidutinė vienos iteracijos trukmė buvo nuo 10 iki 70 kartų trumpesnė nei bandant apmokinti anksčiau aptartus modelius. Iš šių įžvalgų matoma, kad U-Net susidorojo su tirama duomenų baze efektyviausiai it tuo pačiu metu sugebėjo pateikti tiksliausius defektų lokalizacijos rezultatus.



26 pav. U-Net treniravimo ir validacijos imčių apmokymas

Remiantis 26 paveikslu, galima teigti, kad kiekvienos paviršiaus klasės treniravimo ir validacijos kreivės stipriai koreliuoja. Iš to daroma išvada, kad tinklas neprisitaikė prie duomenų nors iš šiuo atveju apmokymo duomenų imtis buvo 16 kartų mažesnė negu apmokinant anksčiau aptartus CNN modelius.

Įdomu pabrėžti, kad nors ir kiekvienai paviršių klasei apmokinti buvo skirta 100 iteracijų, aiškiai matoma, jog atvirkščiai negu ankstesnių CNN modelių tyrimų metu pateiktų 18 paveiksle, kiekvienos klasės atveju modelio tikslumas konvergavo skirtingu metu. Taip pat reikia pastebėti, kad kaip ir su anksčiau tirtais CNN modeliais U-Net taip pat patyrė problemų apmokinant 5 paviršių klasę, nes ji konvergavo, lyginant su kitomis paviršių klasėmis, paskutinė, tik praėjus 80 iteracijų. Apžvelgtoje literatūroje pateiktuose rezultatuose bei autorių komentaruose nebuvo pastebėta analogiškų problemų apmokinant 5 paviršių klasę.

Išvados

1. Ištyrus modelio v1173 atpažinimo kokybę su DAGM duomenų baze nustatyta, kad modelio tikslumas mažai kito sumažinus duomenų vaizdą du kartus, o suspaudus daugiau kartų – atpažinimo tikslumas krito iki ~50%.
2. Pritaikius 3 standartinius modelius iš *Keras* bibliotekos su ImageNet saugyklos svoriais tiriamai duomenų bazei, nustatyta, kad DenseNet121 atveju gauti rezultatai pasiekė 99,7% tikslumą, kai MobileNet bei ResNet50 pasirodė analogiškai – 96%. Iš to seka, kad tik DenseNet121 sugebėjo pasiekti 0,3% geresnius rezultatus nei v1173 [23].
3. Apmokinus modelius su visomis tiriamomis paviršių klasėmis vienu metu, buvo išmatuota, kad jie gali atskirti defektus visose tiriamose paviršių klasėse, Mobile Net ir DenseNet121 atvejais visiškai nesumaišant skirtingų paviršių klasių požymių tarpusavyje. Importavus ImageNet svorius modelių apmokinimas užtruko apytiksliai 30 iteracijų trumpiau nei leidžiant modeliams inicializuoti atsitiktinius svorius.
4. Remiantis priešpaskutiniiais 14x14 pikselių dydžio požymių žemėlapiais pavyko realizuoti defektų lokalizaciją, tačiau nustatyta, kad ribojančios geometrinės figūros apėmė daugumą defektus supančių gerų pikselių o C4 atveju ribojančioms geometrinėms figūroms nepavyko apgaubti viso defekto.
5. Apmokinus U-Net su 150 originalių neaugmentuotų atvaizdų bei inicializuojant atsitiktinius svorius nustatyta, kad pasiektas tikslumas yra vidutiniškai 99,2% ir nesunkiai lyginasi su DenseNet121 bei literatūroje [23] pateiktais v1173 rezultatais. Iš išgautų segmentacijos žemėlapių galima teigti, kad ši architektūra išmoko geriau išskirti pikselius su defektu negu nurodyta atvaizduose su anotacijomis.
6. Tyrimo metu nustatyta, kad naudojant žinių perkėlimą galima apmokinti CNN modelius greičiau, t. y. naudojant 30 iteracijų, negu siūloma susijusiuose tiriamuosiuose darbuose. Tačiau U-Net pateikė analogiškus prognozių tikslumo rezultatus inicializuojant atsitiktinius svorius ir panaudojant tik 150 atvaizdų apmokinimui. Dėl to U-Net apmokavimo procesas užtruko trumpiau nei anksčiau ištirtų modelių.

Literatūros sąrašas

- [1] Maglogiannis, I., Karpouzis, K., Wallace, B.A., Soldatos, J. Emerging Artificial Intelligence Applications in Computer Engineering. IOS Press, 2007m., 3-25 p., Pasiukiama: https://books.google.lt/books?hl=lt&lr=&id=vLiTXDHR_sYC&oi=fnd&pg=PA3&dq=machine+learning+algorithm+comparison&ots=CY1AuwXCmp&sig=FgN1OgX3enZC15fCU9-SIrrw-50&redir_esc=y#v=onepage&q=machine%20learning%20algorithm%20comparison&f=false
- [2] ElMekkawy T.Y., Hachowski P.A., Strong D., Mann D.D. Defect Analysis for Quality and Productivity Improvements in a Manufacturing System. Canadian Biosystems Engineering. Volume 48, 2006m. Pasiukiama: <http://www.csbe-scgab.ca/docs/journal/48/c0522.pdf>
- [3] Rule of Ten, die Fehlerkosten Zehnerregel. birželio 19 2017m., pasiekiama: <https://produktsicherheit.blogspot.com/2017/01/rule-of-ten-die-fehlerkosten-zehnerregel.html>
- [4] Patil M., Verma S., Wakode J. A Review of Fabric Defect Detection Techniques. International Research Journal of Engineering and Technology. Volume: 04, Issue 09, 2017m. pasiekiama: <https://pdfs.semanticscholar.org/cc72/a45d389a41d463035534b2b622f8d4e87378.pdf>
- [5] Achatz S. State of the Art Object Recognition Techniques. TUM Scientific Seminar. 2016m., 13-18 p., Pasiukiama: https://www.nst.ei.tum.de/fileadmin/w00bqs/www/publications/as/SS2016_AS_ObjectRecogn.pdf
- [6] Dutt V., Chaudhry V., Khan I. Pattern Recognition: an Overview. American Journal of Intelligent Systems. 2012 m., 23-27 p., Pasiukiama: <http://article.sapub.org/pdf/10.5923.j.ajis.20120201.04.pdf>
- [7] Ayodele T. O. Types of Machine Learning Algorithms. INTECH, 2010m., 19-27 p. Pasiukiama: <https://cdn.intechopen.com/pdfs-wm/10694.pdf>
- [8] Barker J. Deep Learning on GPUs. Nvidia, 2016m., Pasiukiama: <http://on-demand.gputechconf.com/gtc/2015/webinar/deep-learning-course/intro-to-deep-learning.pdf>
- [9] Supervised Learning in Neural Networks, 2010m., 1-3 p. Pasiukiama: <https://www.idi.ntnu.no/emner/it3105/materials/neural/learn-sup.pdf>
- [10] Schmidhuber J. Deep learning in Neural Networks: an Overview. Neural Networks 61, 2014m., 86-104 p. Pasiukiama: https://ac.els-cdn.com/S0893608014002135/1-s2.0-S0893608014002135-main.pdf?tid=f01a12e4-5abe-487d-b789-2d22b22cab9&acdnat=1520609776_db1acb5d895ca0c41f209f98c26d5307
- [11] Cohen N., Sharir O., Shaha A. On the Expressive Power of Deep Learning: A Tensor Analysis. JMLR: Workshop and Conference Proceedings vol 49. 2016m., 1-31 p. Pasiukiama: <http://proceedings.mlr.press/v49/cohen16.pdf>
- [12] An Intuitive explanation of Convolutional Neural Networks. 2016m., Pasiukiama: <https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>
- [13] LeCun Y., Haffner P., Bottou L., Bengio Y. Object Recognition With Gradient Based Learning. Pasiukiama: <http://yann.lecun.com/exdb/publis/pdf/lecun-99.pdf> (žiūrėta 2018 06 01)
- [14] Sharma S. Activation Functions: Neural Networks. 2017m., Pasiukiama: <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6> (žiūrėta 2018 05 25)
- [15] Budhiraja A. Dropout in (Deep) Machine Learning. 2016m., Pasiukiama: <https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5>
- [16] Doukkali F. Batch Normalization in Neural Networks. 2017m., Pasiukiama: <https://towardsdatascience.com/batch-normalization-in-neural-networks-1ac91516821c>
- [17] Yang J. ReLu and Softmax activation functions. 2017 m., Pasiukiama: <https://github.com/Kulbear/deep-learning-nano-foundation/wiki/ReLU-and-Softmax-Activation-Functions> (žiūrėta 2018 01 15)
- [18] Varma S., Das S. Deep Learning. 2018m., pasiekiama: <https://srdas.github.io/DLBook/> (žiūrėta 2018 04 15)

- [19] Szegedy Ch., Toshev A., Erhan D. Deep Neural Networks for Object Detection. Pasiukiama: <https://papers.nips.cc/paper/5207-deep-neural-networks-for-object-detection.pdf> (žiūrėta 2019 04 18)
- [20] Ronneberger O., Fischer P., Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation. 18 03 2015m. Pasiukiama: <https://arxiv.org/pdf/1505.04597.pdf>
- [21] Long J., Shelhamer E., Darrell T. Fully Convolutional Neural Networks for Semantic Segmentation. Pasiukiama: https://people.eecs.berkeley.edu/~jonlong/long_shelhamer_fcn.pdf (žiūrėta 2019 04 18)
- [22] Weakly Supervised Learning for Industrial Optical Inspection, Competition at DAGM 2007 symposium, pasikiama: <https://hci.iwr.uni-heidelberg.de/node/3616>
- [23] Rački D., Tomažević D., Skočaj D. The effect of different CNN configurations on textured-surface defect segmentation and detection performance. 23rd Computer Vision Winter Workshop, vasario 5-7, 2018 m. pasikiama: <http://cmp.felk.cvut.cz/cvww2018/papers/26.pdf>
- [24] Wang T., Chen Y., Qiao M., Snoussi H. A fast and robust convolutional neural network-based defect detection model in product quality control. International Journal for Advanced Manufacturing Technolgy, rugpjūčio 15, 2017m. Pasiukiama: <https://link.springer.com/article/10.1007/s00170-017-0882-0>
- [25] Staar B., Lütjen M., Freitag M. Anomaly detection with convolutional neural networks for industrial surface inspection. 12th CIRP Conference in Intelligent Computation in Manufacturing Engineering, sausio 9, 2019m. Pasiukiama: <https://www.sciencedirect.com/science/article/pii/S2212827119302409>
- [26] Available models, Models for image classification with weights trained on ImageNet, pasikiama: <https://keras.io/applications/>
- [27] Howard A.G., Zhu M., Chen B., Kalenichenko D. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. Google Inc., April 17, 2017 m. pasikiama: <https://arxiv.org/pdf/1704.04861.pdf>
- [28] Huang G., Liu Z., Maaten L., Weinberger K.Q. Densely Connected Convolutional Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), liepos 21-26 2017m. pasikiama: http://openaccess.thecvf.com/content_cvpr_2017/papers/Huang_Densely_Connected_Convolutional_CVPR_2017_paper.pdf
- [29] He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition. Microsoft Research, gruodžio 10, 2015 m. pasikiama: <https://arxiv.org/pdf/1512.03385.pdf>
- [30] Koidl K., Loss Functions in Classification Tasks. School of Computer Science and Statistic Trinity College Dublin, ADAPT Research Centre. Pasiukiama: <https://www.scss.tcd.ie/~koidlk/cs4062/Loss-Functions.pdf>
- [31] Tamim H., Receiver Operating Characteristic (ROC) Curve. Department of Internal Medicine, American University of Beirut. Pasiukiama: <https://website.aub.edu.lb/sharp/Documents/ROC.pdf>