



Kauno technologijos universitetas

Elektros ir elektronikos fakultetas

GAN tipo tinklo galimybių tyrimas paviršiaus defektų aptikimui

Baigiamasis magistro projektas

Julius Liatukis

Projekto autorius

Doc. Arūnas Lipnickas

Vadovas

Kaunas, 2020



Kauno technologijos universitetas

Elektros ir elektronikos fakultetas

GAN tipo tinklo galimybių tyrimas paviršiaus defektų aptikimui

Baigiamasis magistro projektas

Valdymo technologijos (6211EX014)

Julius Liatukis

Projekto autorius

Doc. Arūnas Lipnickas

Vadovas

Lekt. Vygandas Vaitkus

Recenzentas

Kaunas, 2020



Kauno technologijos universitetas

Elektros ir elektronikos fakultetas

Julius Liatukis

GAN tipo tinklo galimybių tyrimas paviršiaus defektų aptikimui

Akademinio sąžiningumo deklaracija

Patvirtinu, kad mano, Juliaus Liatukio, baigiamasis projektas tema „GAN tipo tinklo galimybių tyrimas paviršiaus defektų aptikimui“ 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)

Liatukis Julius. GAN tipo tinklo galimybių tyrimas paviršiaus defektų aptikimui. Magistro baigiamasis projektas / vadovas doc. Arūnas Lipnickas; Kauno technologijos universitetas, Elektros ir elektronikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Elektronikos inžinerija, Inžinerijos mokslai.

Reikšminiai žodžiai: GAN, diskriminatorius, DCGAN, defektas, atpažinimas, neuroninis tinklas

Kaunas, 2020. 48 p.

Santrauka

Darbe analizuojami metodai skirti paviršiaus defektui atpažinti. Kadangi šiuo metu žmogiškųjų išteklių nepakanka, jais nepasitikima ir ilguoju laikotarpiu šis sprendimas neatsiperka, šiuo metu taikomos automatizuotos sistemos, kurios paviršiaus defektą atpažįsta per itin trumpą laiką, kuris kartais nesiekia nei sekundės.

Automatizuotos sistemos dažniausiai realizuotos neuroninio tinklo pagrindu. Tai yra dirbtinio intelekto sistemos, neuroninio ir giliojo neuroninio tinklo sistemos. Šiame darbe apžvelgiamas gilioju konvoliuciniu neuroniniu tinklu paremtas GAN tinklas, kuris dar vadinamas DCGAN, ir jo galimybės aptikti paviršiuje esantį defektą. Šiai užduočiai atlikti naudojama DAGM2007 duomenų bazė, kuri turi 10 skirtingų duomenų rinkinių, kurios yra paskirstytos į mokymo ir testavimo duomenis. Taip pat ši duomenų bazė turi pažymėtas defekto vietas, kurios naudojamos rezultatų įvertinimui.

Darbe naudojamas diskriminatoriaus modelis, kuris klasifikuoja segmentuotą paviršiaus plotą, jį įvardijant geras ar turintis defektą. Diskriminatoriaus modelis parinktas kiekvienam iš DAGM2007 duomenų rinkinių individualiai. Parinktas diskriminatoriaus modelis nuo pradinės jo struktūros yra tobulinamas ir pritaikomas būtent tam duomenų rinkiniui, kuriam jis yra priskirtas. Diskriminatoriaus modelis vertinamas pagal taiklumą, kuris nusako, kaip gerai diskriminatoriaus modelis geba atskirti paviršiaus defektą nuo paviršiaus, kuriame defekto nėra.

Po diskriminatoriaus modelio tobulinimo ir pritaikymo individualiam duomenų rinkiniui buvo gautas 98,96 % taiklumas identifikuojant defektus antrame duomenų rinkinyje. Žemiausias taiklumas – 90,31 %, buvo gautas identifikuojant defektus esančius ketvirtame duomenų rinkinyje.

Liatukis Julius. Research of GAN network applicability for surface defect detection. Master's Final Degree Project / supervisor doc. Arūnas Lipnickas; Faculty of Electrical and Electronics, Kaunas University of Technology.

Study field and area (study field group): Electronics Engineering, Engineering Sciences.

Keywords: GAN, discriminator, DCGAN, defect, recognition, neural network.

Kaunas, 2020. 48 p.

Summary

The methods analyzed in the work are designed to identify the surface defect. As human resources are currently insufficient, unreliable and do not pay off in the long run, automated systems are currently in place that detect a surface defect in a very short time, sometimes less than a second.

Automated systems are usually implemented on the basis of neural network. These are artificial intelligence systems, neural and deep neural network systems. This paper reviews a GAN network based on a deep convolutional neural network, also called a DCGAN, and its ability to detect a surface defect. This task uses the DAGM2007 database, which has 10 different data sets that are divided into training and testing data. This database also has ground true images that are used to evaluate the results.

The paper uses a discriminator model, which classifies the segmented surface area, and label it as good and as defective otherwise. The discriminator model was selected for each of the DAGM2007 datasets individually. The chosen discriminator model is modified from its initial structure and adapted to the particular data set to which it is assigned. The discriminator model is evaluated by accuracy, which describes how well the discriminator model is able to distinguish a surface defect from a surface which have no defect.

After modifications of the discriminator model and application to the individual data set, 98.96% accuracy was obtained in identifying defects in the second data set. The lowest accuracy, 90.31%, was obtained by identifying defects in the fourth data set.

Turinys

Lentelių sąrašas	7
Paveikslų sąrašas	8
Santrumpų ir terminų sąrašas	9
Įvadas.....	10
1. Apžvalginė dalis.....	11
1.1. Paviršiaus defekto atpažinimas.....	11
1.2. Neuroninis tinklas.....	13
1.2.1. Neuroninio tinklo veikimas	13
1.2.2. Neuroninio tinklo apmokymas	15
1.2.3. Neuroninio tinklo taikymas	16
1.3. Generatyvinis priešininkų tinklas (GAN).....	17
1.3.1. GAN tinklo veikimas.....	19
1.3.2. GAN tipo tinklo pritaikymas paviršiaus defekto aptikimui	22
2. Metodinė dalis.....	24
2.1. GAN tinklo modelis	24
2.2. Modelio realizavimas	27
3. Rezultatai.....	39
Išvados	46
Literatūros sąrašas	47

Lentelių sąrašas

1 lentelė. Papildomos duomenų rinkinių informacijos tvarka teksto faile.....	26
2 lentelė. Naudojamų programų ir bibliotekų sąrašas	27
3 lentelė. Po segmentavimo gaunamų nuotraukų skaičius.....	30
4 lentelė. Modelio atsakymo vertinimas	36
5 lentelė. Dalis bandomų modelio konfigūracijų. Įtrauktos geriausius rezultatus parodžiusios konfigūracijos	39
6 lentelė. Bandymų metu nustatytos geriausios kombinacijos diskriminatoriaus modeliai kiekvienam duomenų rinkiniui	42
7 lentelė. Tinklo modelis po modifikacijos.....	43
8 lentelė. Po atliktų modifikacijų gautų diskriminatoriaus modelių rezultatai	43
9 lentelė. Generatoriaus modelis skirtas antram duomenų rinkinio paviršiaus segmentams generuoti	44
10 lentelė. Gautų diskriminatoriaus modelių parametrų palyginimas su kitų tyrėjų giliuoju regresiniu neuroniniu tinklu pagrįsto modeliu	45

Paveikslų sąrašas

1 pav. Skirtingų defektų tipai medvilnės pramonėje [1]	11
2 pav. Išmaniojo telefono stiklo defektas kairėje ir dešinėje sistemos aptiktas defektas [3]	12
3 pav. Neuroninio tinklo sandara	13
4 pav. Neurono vertės apskaičiavimas	14
5 pav. Aktyvacijos funkcija	14
6 pav. Vieno svorio pakeitimo įtaka tolimesniems dviems sluoksniams.....	15
7 pav. Pavedimų skaičiaus prognozavimo pavyzdys nuo asmens duomenų	16
8 pav. Standartinis bendras GAN tinklo modelis.....	17
9 pav. GAN tinklo vystymosi kryptys	18
10 pav. GAN tema atliktų tyrimų skaičius per metus	19
11 pav. GAN tinklo generatoriaus modelis.....	19
12 pav. GAN tinklo diskriminatoriaus modelis	20
13 pav. Generatoriaus ir diskriminatoriaus mokymo modelis	21
14 pav. Generatoriaus modelio mokymas, kai turimas optimalus diskriminatorius	21
15 pav. Konvoliucinio tinklo modelis	24
16 pav. Duomenų rinkinių pavyzdžiai	25
17 pav. Supaprastintas GAN tinklo modelis	26
18 pav. Pagrindinis programos algoritmas.....	27
19 pav. Konvoliucija su 3x3 filtro branduolio dydžiu. Viršuje su padidiniu, apačioje matomas išėjimo matricos sumažėjimas	28
20 pav. Modelio apmokymo algoritmas	29
21 pav. Duomenų paruošimo mokymui algoritmas	30
22 pav. Defekto aptikimas su 8 x 8 segmentavimu viršuje, 32 x 32 segmentavimu viduryje ir 64 x 64 segmentavimu apačioje.....	31
23 pav. Defektuoto paveiklo defekto segmentų išsaugojimo algoritmas	32
24 pav. Defekto priskyrimo vaizdui algoritmas	33
25 pav. Naujų defekto pavyzdžių sukūrimas, juos pasukant kampu.....	34
26 pav. Modelio testavimo algoritmas	34
27 pav. Duomenų paruošimo testavimui algoritmas.....	35
28 pav. Defekto priskyrimo vaizdui mokymo etape algoritmas	36
29 pav. Defektuoto paviršiaus vaizdo segmentacija, diskriminatoriaus apdorojimas ir gautų segmentų įvertinimo rekonstrukcija.....	38
30 pav. Nepavykusių modelių išėjimo pavyzdžiai.....	40
31 pav. Atrinktų diskriminatoriaus modelių išėjimai kiekvienam duomenų rinkiniui	41
32 pav. Tyrimo metu gautų diskriminatoriaus modelių mokymo procesas	42
33 pav. Tyrimo metu gautų diskriminatoriaus modelių mokymo procesas	44
34 pav. Generatoriaus modelio mokymo funkcija	45

Santrumpų ir terminų sąrašas

Santrumpos:

GAN – generatyvinis priešininkų tinklas

DCGAN – gilusis konvoliucinis generatyvinis priešininkų tinklas

MM – mašininis mokymasis

CNN – konvoliucinis neuroninis tinklas

Terminai:

Konvoliucinis neuroninis tinklas – dirbtinio neuroninio tinklo rūšis, kuri dažniausiai taikoma kompiuterinės regos srityje: vaizdų atpažinimui, objektų identifikavimui ir radimui vaizduose.

Įvadas

Pastaraisiais metais buvo padaryta didelė pažanga naudojant neuroninius tinklus (angl. *neural networks* - NN), kurią lėmė giliojo mokymosi pasisėkimai. Vienas esminių reikalavimų geram neuroninio tinklo algoritmui yra jo taisyklės – galimybė atpažinti įėjimo duomenis, kurie yra pakankamai artimi apmokymo duomenims. Tačiau, dažniausiai nėra apibrėžiama, kaip DI turi veikti esant kitokio tipo įėjimo duomenims, kurie iš esmės skiriasi nuo duomenų, kurias šis tinklas buvo prieš tai apmokytas. Tokiu atveju, algoritmas klaidingai identifikuoja naujus įėjimo duomenis ir priskiria juos vienai iš klasių, kuriomis jis buvo apmokytas.

Generatyviniai priešininkų tinklai (angl. *Generative Adversarial Networks* – GAN) yra gerai žinomas algoritmo realizavimas. Šis karkasas buvo pasiūlytas kaip generacinio modeliavimo metodas ir daugiausiai naudojamas natūralių vaizdų realistiškiems pavyzdžiams kurti. Tai yra generatyvinio modeliavimo metodas, kurio metu du modeliai, kurie atitinkamai yra vadinami generatoriumi ir diskriminatoriumi, yra treniruojami tuo pačiu metu. Diskriminatoriaus tikslas yra klasifikuoti įėjimo duomenis ir nurodyti ar tie duomenys neatitinka pradinių duomenų rinkinio (yra suklasoti, padirbti, modifikuoti), kuriuo jis buvo apmokytas, ar atitinka pradinį duomenų rinkinį (tikri, tenkinantys užsibrėžtą kriterijų). Generatorius veikia priešingai. Jo tikslas yra sudaryti tokius išėjimo duomenis savo išėjime, kurie diskriminatoriui atrodytų kaip atitinkantys pradinį duomenų rinkinį.

Naujumo, arba kitaip, defekto aptikimas (angl. *novelty detection*) gali būti apibūdinamas kaip naujų duomenų atpažinimas, kurie kažkuriuo tai aspektu skiriasi nuo duomenų, kuriais tinklas buvo apmokytas. Su naujumo aptikimo problema susiduriama daugelyje sričių. Tai labai panašus uždavinys, kaip broko ar defekto atpažinimas, kur reikia identifiukuoti pažeistus, defektuotus pavyzdžius duomenų rinkiniuose. Naujumo aptikimas yra rimta problema, kuri sulaukia palyginti mažai dėmesio diskusijose, apie mašininio mokymosi (angl. *machine learning* - ML) problemas. Tačiau naujumo aptikimas turėtų būti vienas pagrindinių elementų kiekvienoje šių dienų MM sistemoje.

GAN tinklai randa vis daugiau savo pritaikymo sferų, kadangi jiems reikia mažiau pradinių duomenų apmokymui, palyginus su DI paremtomis sistemomis. Šio darbo tikslas yra nustatyti GAN tinklo galimybes atpažinti paviršiaus defektą esant ribotam kiekiui įėjimo duomenų ir palyginti gautus rezultatus su neuroniniu tinklu realizuotomis sistemomis.

Šio darbo tikslas: ištirti GAN tipo tinklo galimybes paviršiaus defektų aptikimui.

Tikslui pasiekti keliami uždaviniai:

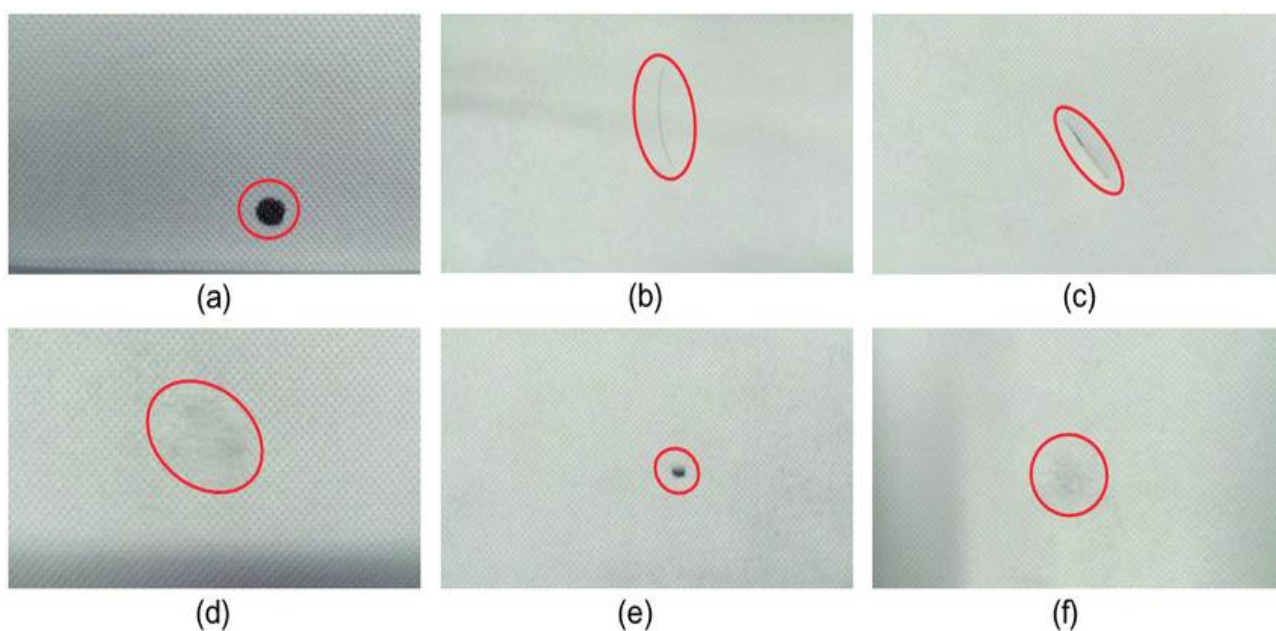
1. Apžvelgti esamas paviršiaus defekto nustatymo priemones;
2. Išanalizuoti ir realizuoti GAN tinklą;
3. Nustatyti tinkamiausią GAN tinklo realizaciją individualiam paviršiui;
4. Ištirti GAN tinklo modelių veiksnumą individualiems paviršiams;
5. Palyginti GAN tinklo modelius su kitais realizuotais modeliais;

1. Apžvalginė dalis

1.1. Paviršiaus defekto atpažinimas

Įvairiose pramonės srityse, tiek produkcijos gaminimo metu, tiek jos eksploataciniu laikotarpiu yra reikalingas kokybės patikrinimas dėl galimai atsiradusių paviršiaus defektų. Šių defektų būna įvairaus dydžio, pobūdžio, struktūros ar atsiradimo kilmės (žr. 1 pav.). Nors šiais laikais yra realizuotų automatizuotų sprendimų, kai kuriose pramonės srityse vis dar yra naudojami žmogiškieji išteklių ir pasitikima jų vertinimu, kuris ne visada atspindi tikrąjį defekto dydį ir svarbą. Dažniausiai toks sprendimas priimamas dėl mažesnių kaštų trumpuoju laikotarpiu. Tačiau dėl žmogaus vertinimo kyla keletas problemų:

- Patikimumo problema;
- Netikslus atsakymo pateikimas;
- Didelės laiko sąnaudos;
- Išaugę patikrinimo kaštai;
- Žmogiškųjų resursų trūkumas;



1 pav. Skirtingų defektų tipai medvilnės pramonėje [1]

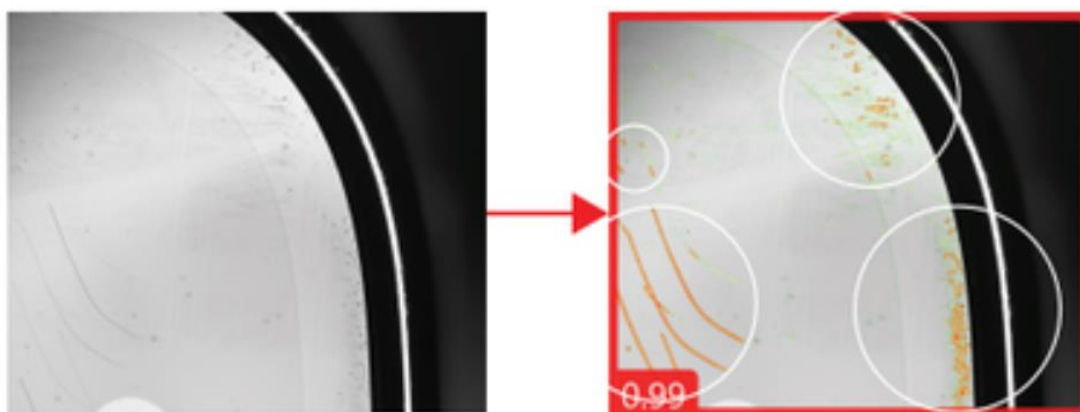
Patikimumo problema atsiranda tuomet, kai žmogus bando surasti defektą tam tikrame gaminyje ar dangoje, tačiau kai kurie defektai gali būti labai maži ir nepastebimi žmogaus akiai. Reikia nepamiršti, kad žmogaus sugebėjimą susikaupti įtakoja nuovargis, darbo valandų skaičius, emocijos, amžius, gyvenimo būdas ir kiti veiksniai, todėl tikimybė nepastebėti esančio defekto dar daugiau išauga. Taip pat, žmogus negali visada tiksliai įvertinti, kokio masto defektas buvo aptiktas. Gaunamas tik labai grubus, apytikslis įvertinimas, kuris geriausiu atveju gali atspindėti balus nuo 1 iki 10, tačiau sprendimas taip pat ne visada yra objektyvus ir tas pats defektas skirtingam vertinančiajam gali atrodyti skirtingos svarbos ir dydžio. Netgi tas pats vertinantysis, tą patį defektą vertindamas keletą kartų, jį gali įvertinti skirtingai, kartais net ir nepastebėti prieš tai paties identifikuoto paviršiaus defekto.

Didelės laiko sąnaudos siejamos su tuo, kad norint kokybiškai ir nuosekliai viską patikrinti žmogus sugaišta daug laiko, lyginant su automatinėmis sistemomis. Automatinės sistemos patikrą gali atlikti per kelias sekundes dalis, o esant nedideliame tikrinamos produkcijos paviršiaus plotui, tikrinti ir po kelias dešimtis gaminių vienu metu. Esant didelėms laiko sąnaudoms išauga ir sunaudojami kaštai (patalpų nuoma, darbuotojų algos, kiti susiję mokesčiai). Reikia paminėti, kad automatizuotos sistemos visada yra pigesnės (išlaikymo atžvilgiu, bei ilguoju laikotarpiu) ir našesnės už žmogaus dirbamą darbą. Aišku, yra veiklos sričių kur žmogaus indėlis yra nepakeičiamas (bet koks tiesioginis bendravimas su žmogumi, psichologo profesija ir kt. sritys). Šiose srityse reikalinga nuojauta, nuovokumas, analizavimas iš pirmo žvilgsnio nesusijusių reiškinių ar įvykių. Automatinės sistemos kol kas šių užduočių negali suprasti ir išspręsti.

Taip pat, kai kuriais atvejais pastebimas kvalifikuotų darbuotojų trūkumas, konkrečioms tekstilės, betono, kelio įtrūkimų ir kitų defektų nustatymui. Kiekvieną iš naujų darbuotojų reikia apmokyti, skirti laiko, darbo pradžioje stebėti kaip darbuotojas susitvarko su jam paskirtomis užduotimis ir papildomai patikrinti atliktų darbų rezultatą. Kadangi dažnu atveju kokybės inspektoriaus pareigos nėra gerai apmokamos, egzistuoja natūrali darbuotojų kaita, o kiekvienam naujai atėjusiam darbuotojui reikalingi anksčiau minėti apmokymai. Jie padidina įmonės kaštus, bei sumažina produkcijos patikimumą, kadangi dirbantys žmonės neturi ilgametės patirties konkrečioje srityje, o patikimų specialistų išlaikymas darbo vietoje ir motyvuojančio atlyginimo skyrimas įmonei taip pat mažina pelną.

Dėl šių priežasčių yra kuriamos automatinės sistemos, kurios yra specializuotos tam tikrų defektų identifikavimui konkrečiose srityse (žr. 2 pav.). Aptikti defektai yra kaupiami duomenų bazėje ir analizuojami. Analizės metu identifikuojamos problemišiausios gamybos vietos, dėl kurių atsiranda gamybinis brokas. Žinant šias vietas galima atnaujinti, pagerinti, įvesti tam tikras korekcijas, kurios pagerintų bendrą produkcijos kokybę.

Geriausiai žinomi automatinių defektų aptikimo sprendimai yra paremti neuroninių tinklų pagrindu veikiančiomis AI sistemomis. Šiuo metu populiarėja ir GAN tinklų panaudojimas vaizdų atpažinimo ir generavimo srityse. Abu minėti metodai turi savo stipriųjų ir silpnųjų pusių apmokymo etape, apmokymui reikalingo minimalaus duomenų kiekio dydyje, tikslumo atžvilgiu, bei taikymo srityse. Nepaisant esamų skirtumų metodai ganėtinai panašūs ir gali atlikti artimas vienas kitam užduotis su panašiu patikimumo lygiu.



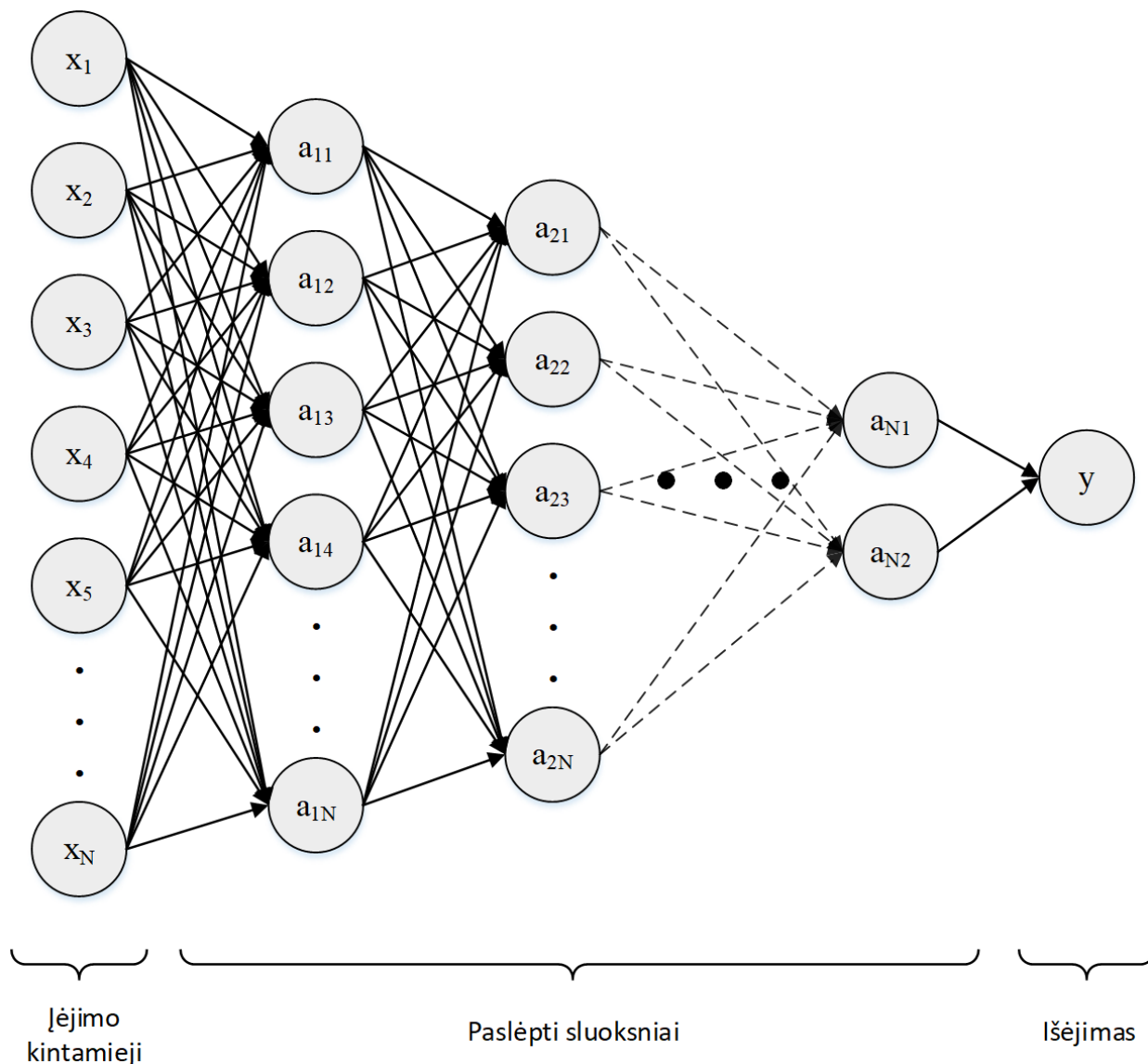
2 pav. Išmaniojo telefono stiklo defektas kairėje ir dešinėje sistemos aptiktas defektas [3]

1.2. Neuroninis tinklas

1.2.1. Neuroninio tinklo veikimas

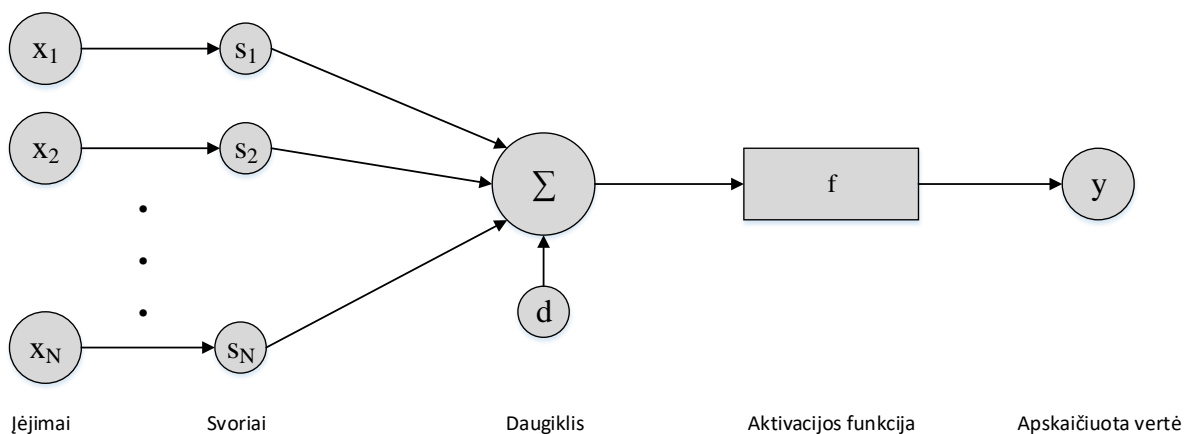
Sudėtingas neuroninis tinklas dažniausiai susideda iš trijų esminių dalių: įėjimo sluoksnio, paslėptų sluoksnių ir išėjimo sluoksnio. Visi šie sluoksniai sudaryti iš neuronų kurie sąveikauja tarpusavyje. Paslėptieji sluoksniai (angl. *hidden layers*), kurie dar vadinami giliaisiais neuroniniais tinklais (angl. *deep neural networks*) atlieka pagrindinius skaičiavimus. Juose taip pat vyksta gilusis mokymas (angl. *deep learning*).

Įėjimo x ir išėjimo y parametrai turi po vieną sluoksnį (žr. 3 pav.). Sluoksnyje neuronų skaičius nėra ribojamas. Paslėptieji sluoksniai yra tarp išėjimo ir įėjimo sluoksnių, jų skaičius nėra ribojamas, kaip ir neuronų skaičius tuose sluoksniuose. Įprastai neuronų skaičius kiekviename sluoksnyje, žvelgiant nuo įėjimo sluoksnio link išėjimo sluoksnio, mažėja. Kiekvienas iš neuronų sąveikauja tik su neuronais, kurie yra prieš jį ir po jo esančiuose sluoksniuose. Prieš neuroną esantis sluoksnis naudojamas paties neurono vertei apskaičiuoti, kuri bus naudojama apskaičiuoti sekancio sluoksnio neuronų vertėms.



3 pav. Neuroninio tinklo sandara

Neurono vertė apskaičiuojama atliekant 4 veiksmus (žr. 4 pav.). Visų pirma įėjimai, kurie yra paimami iš prieš tai buvusio sluoksnio, yra padauginami iš jų svorių, kurie kiekvienam neuronų sujungimui yra skirtingi. Svoris nusako ryšį tarp dviejų neuronų. Mokymosi proceso metu svoriai yra koreguojami. Po to gautos reikšmės yra sudedamos. Sudėjus visas reikšmes, kartais yra naudojamas daugiklis tai vertei koreguoti. Jis yra parenkamas dar prieš pradedant apmokymo procesą.



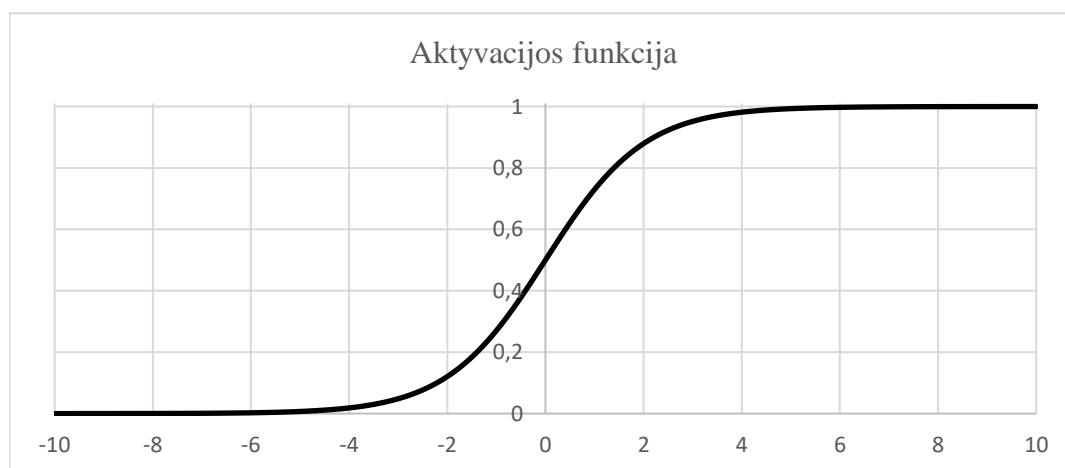
4 pav. Neurono vertės apskaičiavimas

Toliau yra naudojama aktyvacijos funkcija (žr. 5 pav.). Ši funkcija yra apskaičiuojama pagal žemiau pateiktą formulę:

$$y(\text{sum}) = \frac{1}{1+e^{\text{sum}}} \quad (1)$$

Ši funkcija yra naudojama paversti prieš tai apskaičiuotą reikšmę į 1 arba 0. Naudojamos ir kitokios funkcijos, su kitokiomis ribomis nei pateikta (1) formulė, tačiau naudojimo tikslas išlieka toks – apriboti reikšmę.

Kai sluoksnyje visi neuronai atlieka veiksmus ir įgyja atitinkamas reikšmes, neuroninis tinklas pereina į sekantį sluoksnį. Perėjus visus sluoksnius gaunamas išėjimo parametras ar parametų rinkinys.

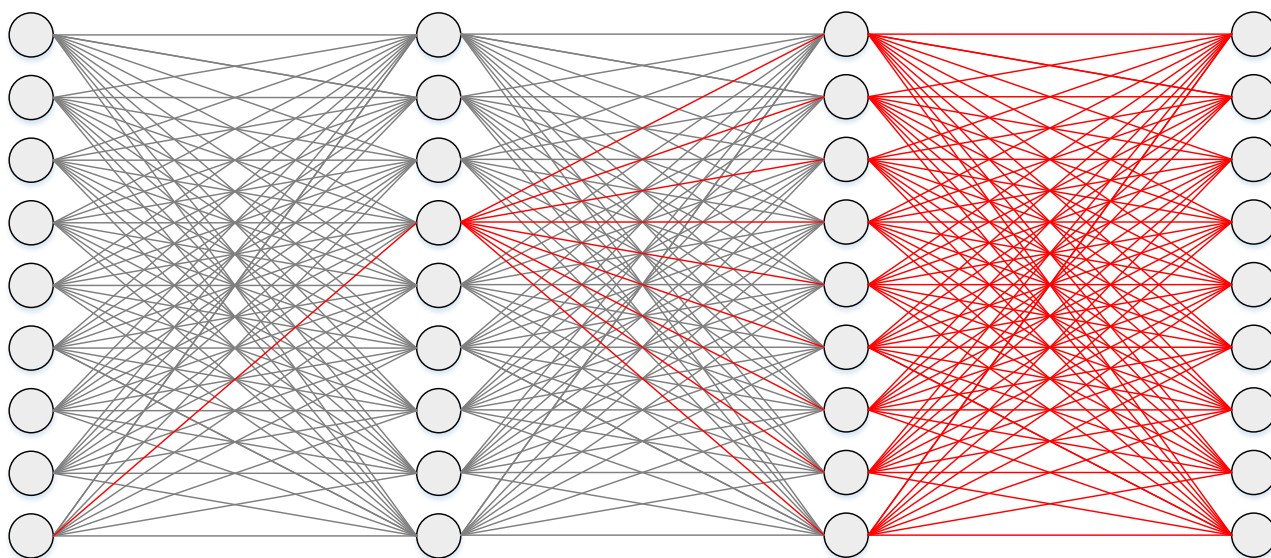


5 pav. Aktyvacijos funkcija

1.2.2. Neuroninio tinklo apmokymas

Neuroniniui tinklui apmokyti reikia didelio kiekio atrinktų duomenų (duomenų rinkinio). Šie duomenys turi turėti etiketes, kurios nusakytų ar atitinkamas pavyzdys duomenų rinkinyje yra teisingas ar ne. Apmokymo metu į įėjimą paduodami duomenų rinkiniai, kuriuose yra teisingai ir neteisingai pavyzdžiai. Tai gali būti nuotrauka, žmogaus savybių aprašymas, supančios aplinkos kintamieji, kai kuriais atvejais, viskas kartu. Išėjime stebimas atsakymas, kuris arba tenkina užsiduotas sąlygas arba ne, pagal jam iš anksto priskirtą etiketę. Atėmus iš viso duomenų rinkinio neteisingai įvertintus pavyzdžius, gaunamas neuroninio tinklo tikslumas. Jei gautas tikslumas netenkina, yra koreguojami neuroninio tinklo svoriai tarp skirtingų neuronų.

Neuroninio tinklo svoriai paslėptuose sluoksniuose yra glaudžiai susiję vienas su kitu. Svorio keitimas viename sluoksnyje turės įtakos ne tik tai konkrečiai neuronų poros jungčiai, tačiau ir sekančiuose sluoksniuose esančioms neuronų jungtims (žr. 6 pav.). Svorio keitimas pradiniuose sluoksniuose turi didesnę įtaką bendram neuroninio tinklo išėjimui. Atitinkamai svorio keitimas galiniuose neuroninio tinklo sluoksniuose turės mažesnę įtaką išėjimui. Dėl šios priežasties neuroninio tinklo svoriai dažniausiai keičiami keli iškarto, ieškant geriausios svorių kombinacijos. Esant mažam neuroniniam tinklui, kuris sudarytas iš 20-50 neuronų, galima išbandyti kiekvieną svorių kombinaciją ir pasirenkant geriausią. Reikia paminėti, kad kiekvienas svoris atitinkamai gali įgyti neribotą skaičių reikšmių, tačiau įprastai 100 skirtingų svorio reikšmių pakanka. Šiais laikais naudojami neuroniniai tinklai susideda iš kelių dešimčių tūkstančių ar net milijonų neuronų. Skirtingų svorių kombinacijų skaičius auga eksponentiškai, todėl visų jų išbandyti neužtenka laiko ir skaičiavimo galios net ir galingiausiems superkompiuteriams.



6 pav. Vieno svorio pakeitimo įtaka tolimesniems dviems sluoksniams

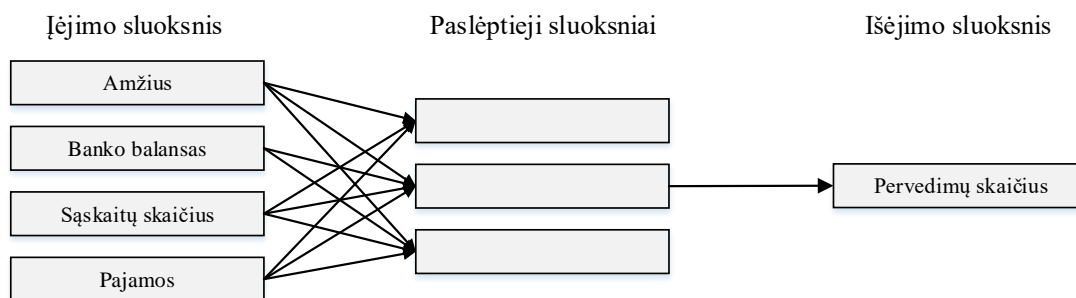
Gavus norimą neuroninio tinklo veikimo tikslumą, galima naudoti kitą duomenų rinkinį ir stebėti ar apmokytas tinklas išlaiko prieš tai buvusį tikslumą. Neuroniniui tinklui apmokyti gali prireikti kelių tūkstančių pavyzdžių duomenų rinkinyje, o kai kuriais atvejais norint pasiekti itin tikslų neuroninį tinklą, gali prireikti ir kelių milijonų pavyzdžių, kuriais tinklas turi būti apmokytas prieš pradėdamas jį naudoti.

1.2.3. Neuroninio tinklo taikymas

Neuroninis tinklas plačiai taikomas raidžių, skaičių ir ženklų atpažinimui. Šis tinklas gali vienu metu priimti didelį kiekį kintamųjų ir juos apdoroti paslėptuose sluoksniuose ne linijinėmis priklausomybėmis vienas nuo kito. Tai leidžia atpažinti net ir neryškiai užrašytus ar pateiktus ženklus su itin aukšta tikimybe, kuri kartais siekia net 99 %. Neuroninis tinklas visų pirma pateiktą vaizdą paverčia nespaltotu paveikslu. Tam yra naudojamas slenkstinis įvertis, kuris paverčia paveikslą į dvejetainį. Tada gautam vaizdui atliekamas ryšio testas, kuris nurodo ar paveiksle yra vienas ar daugiau simbolių. Visi aptikti simboliai padalijami į atskirus papildomus vaizdus, kurie toliau bus apdorojami. Kadangi aptikti simboliai nėra vienodo dydžio, jie konvertuojami į iš anksto nustatyto dydžio paveikslus. Toliau paveikslas perduodamas apdorojimui ir išėjime gaunamos tikimybės, su atpažintais simboliais. Neuroninis tinklas, dėl tikslaus ranka rašytų ženklų atpažinimo, plačiai taikomas saugumo srityse, kur reikalinga aptikti suklastotus dokumentus ir parašus. Lygiai taip pat veikia ir kalbos, žodžių, gestų ir veidų atpažinimas. Modeliai skiriasi tik apmokymui skirtų duomenų rinkiniu ir tų duomenų formatu.

Automobilinė pramonė jau senai įžvelgė neuroninio tinklo ir dirbtinio intelekto potencialą. Šiuo metu matome autonomines mašinas, kurios vis dar yra tobulinamos dėl taikomų itin aukštų reikalavimų, bet poros metų bėgyje jau bus prieinamos įsigyti ir galės pakeisti krovininių transporto priemonių vairuotojus ar taksi įmonėse dirbančius žmones. Autonominės mašinos surenka duomenis iš įvairaus tipo sensorių ir atpažįsta kelio nelygumus, posūkius, kitas mašinas, kelio ženklus, šviesoforo signalus, kelio dangą ir daug kitų aplinkos parametrų esant skirtingoms oro sąlygoms ar skirtingu paros metu. Visi šie duomenys yra apdorojami ir perduodami neuroninio tinklo pagrindu veikiančiam dirbtiniam intelektui, kuris sprendžia kokiu kampu susukti vairą posūkio metu, kada didinti greitį ir kokiu atveju stabdyti. Dirbtinis intelektas įvertina ir kokia galima didžiausia stabdymo jėga pagal esamą kelio dangą, atpažįsta pėsčiuosius, kitas stabdančias mašinas ir su visa turima informacija priima įvairius sprendimus, kuriuos įprastu atveju turi atlikti vairuotojas.

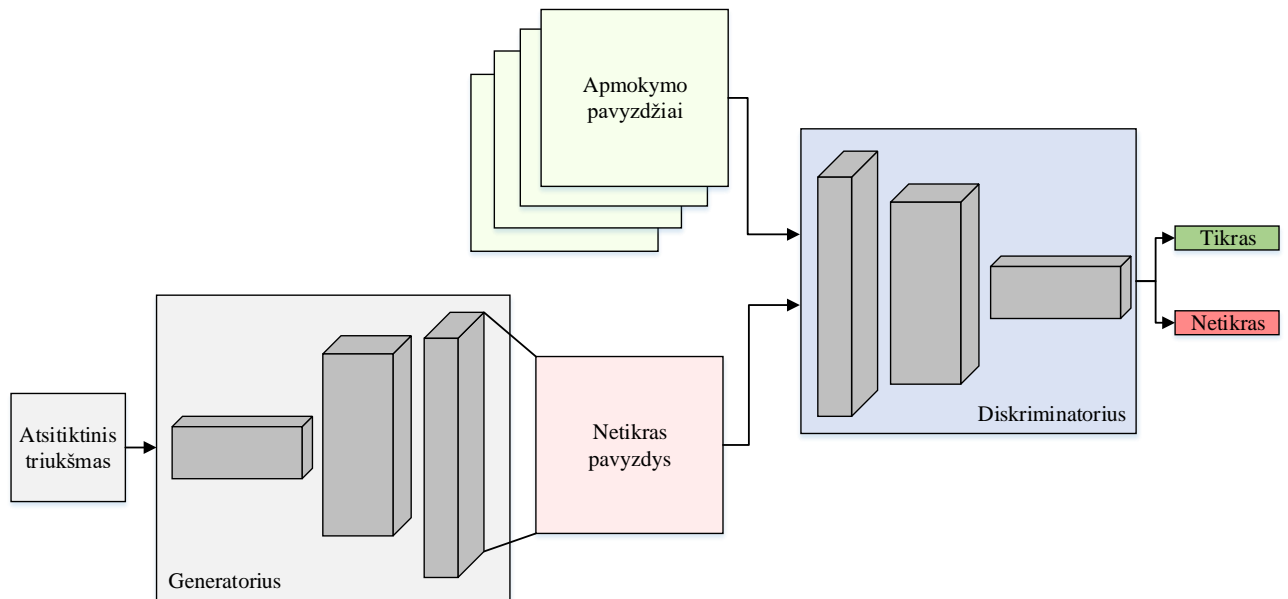
Kita sritis, kurioje yra naudojamas neuroninis tinklas yra prognozavimas. Neuroninis tinklas naudojamas pardavimų, orų, akcijų rinkų, ekonomikos ir dar daug kitų nuolat besikeičiančių sričių prognozavimui. Klasikiniai metodai skirti nuolatiniam prognozavimui laike, tokie kaip Box-Jenkins, ARMA arba ARIMA turi nežymų linijinį ryšį tarp įėjimų ir išėjimų. Neuroninis tinklas šio ryšio neturi. Dažniausiai prognozavimo problema yra sudėtinga, pavyzdžiui prognozuoti pardavimų skaičius, kai jie priklauso nuo daugybės kitų faktorių, iš kurių dalis yra žinoma, o kita dalis ne. Bankai tam tikrus neuroninio tinklo modelius naudoja pervedimų skaičiaus nustatymui (žr. 7 pav.), mokymo tikrinimui.



7 pav. Pervedimų skaičiaus prognozavimo pavyzdys nuo asmens duomenų

1.3. Generatyvinis priešininkų tinklas (GAN)

Generatyvinis priešininkų tinklas yra pakankamai neseniai pasiūlytas metodas generatyviam modeliavimui. Pagrindinė GAN idėja yra turėti dvi konkuruojančias diferencijuojamas funkcijas, paprastai įgyvendinamas kaip neuroninio tinklo modeliai. Vienas modelis, kuris vadinamas generatoriumi $G(z; \theta^G)$, nusako triukšmo pavyzdį z , paimtą iš tam tikro ankstesnio pasiskirstymo $p(z)$ ir aprašo sugeneruotus pavyzdžius $x = G(z; \theta^G)$. x turi būti panašus į tikrąją imtį, paimtą iš originalaus duomenų pasiskirstymo $p_{duom}(x)$.



8 pav. Standartinis bendras GAN tinklo modelis

Kito modelio, vadinamo diskriminatoriumi, tikslas $D(x; \theta^D)$ yra teisingai atskirti generatoriaus sugeneruotus pavyzdžius nuo mokymo duomenų, iš kurių imami pavyzdžiai $p_{duom}(x)$. Tai yra dviejų modelių tarpusavio apjungimas juos supriešinant su sprendimu Nash pusiausvyroje. Ši pusiausvyra pasirenkama individualiai, tačiau dažniausiai naudojama 50 % pusiausvyra, kurioje abu modeliai klysta vienodai kartu.

Deja, nėra uždaros formos sprendimo tokioms problemoms spręsti. Todėl sprendimas randamas naudojant iteracinį gradientu pagrįstą generatoriaus ir diskriminacinių funkcijų optimizavimą. Diskriminatorius optimizuojamas maksimaliai padidinant:

$$\max_{\theta^D} E_{x \sim p_{duom}(x)} [\log D(x; \theta^D)] + E_{z \sim p(z)} [\log(1 - D(G(z; \theta^G); \theta^D))] \quad (2)$$

Generatorius optimizuojamas sumažinant:

$$\min_{\theta^G} E_{z \sim p(z)} [\log(1 - D(G(z; \theta^G); \theta^G))] \quad (3)$$

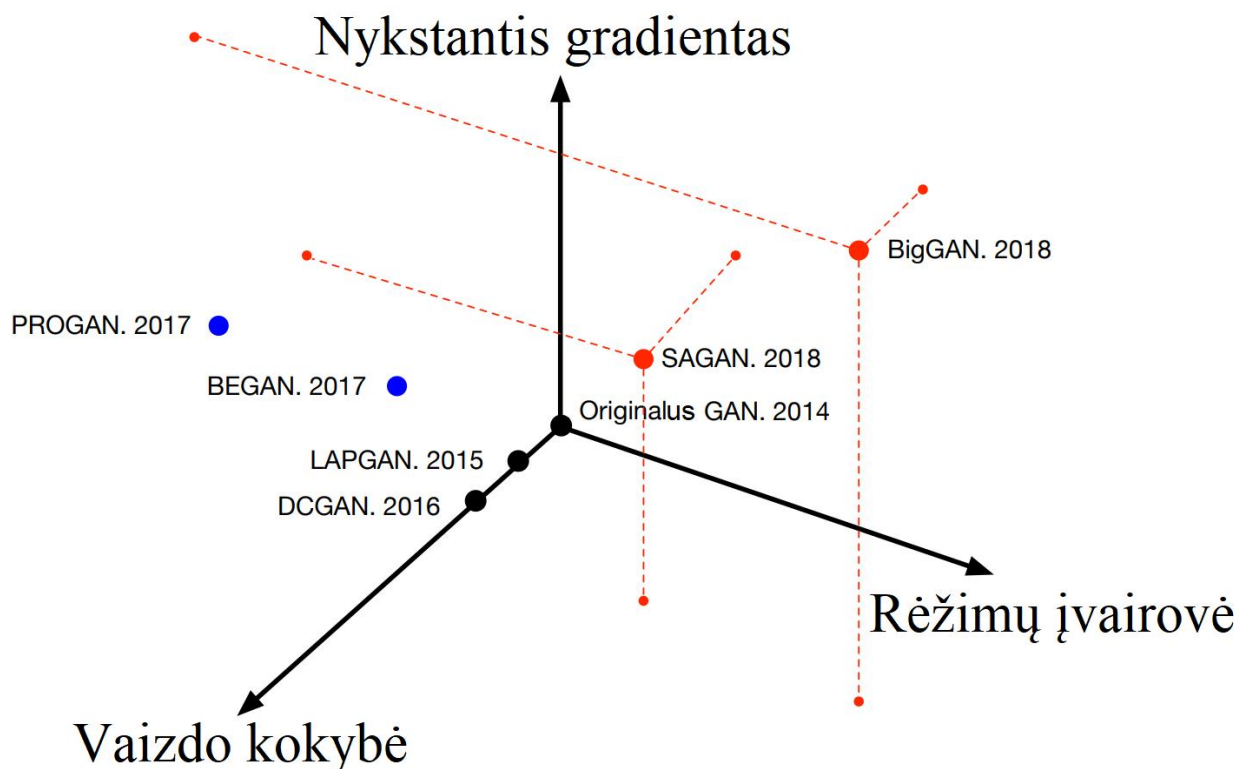
Fiksuotam generatoriui $G(z)$ galima analitiškai apskaičiuoti optimalią diskriminatoriaus formą:

$$D_G^*(x) = \frac{p_{duom}(x)}{p_{duom}(x) + p_g(x)} \quad (4)$$

Nuo to laiko, kai „Goodfellow“ ir kt. pristatė generatyvinius priešinių tinklus 2014 metais, literatūroje paskelbti keli GAN variantai, kurie sudaryti iš generatoriaus ir diskriminatoriaus. GAN tinklas buvo pritaikytas įvairioms užduotims, tokioms kaip: tikroviškas vaizdų generavimas (Radford, 2015 m.), teksto į atvaizdą generavimas (Reed ir kt., 2016 m.), paveikslo dažymas (Pathak ir kt., 2016 m.), vaizdo generavimas panaudojus kitą vaizdą (Isola ir kt., 2016 m.), vaizdo įrašų generavimas (Vondrickui ir kt., 2016 m.), itin aukšta skiriamoji geba (Ledig ir kt., 2017 m.) ir dar daugiau. Beveik visose šiose programose bandymo metu naudojamas tik generatorius G, o diskriminatorius D yra treniruojamas generatoriaus mokymui ir bandymo metu yra išmetamas.

Atskirais atvejais yra taikomos kitokios GAN tinklo atmainos. Xuanqing Liu ir Cho-Jui Hsieh savo darbe teigia, kad Rob-GAN: generatorius, diskriminatorius ir priešpriešinį modelis aprašo unikalų priešpriešinį tinklą. Tinklas buvo realizuotas pridėjus generatorių prie priešpriešinio mokymo procedūros ir pridėjus priešpriešinę ataką generatoriaus mokymo metu. Po tinklo apmokymo buvo gautas tvirtesnis diskriminatorius ir geresnis generatorius, o mokymo procesas vyko greičiau nei standartiniame modelyje.

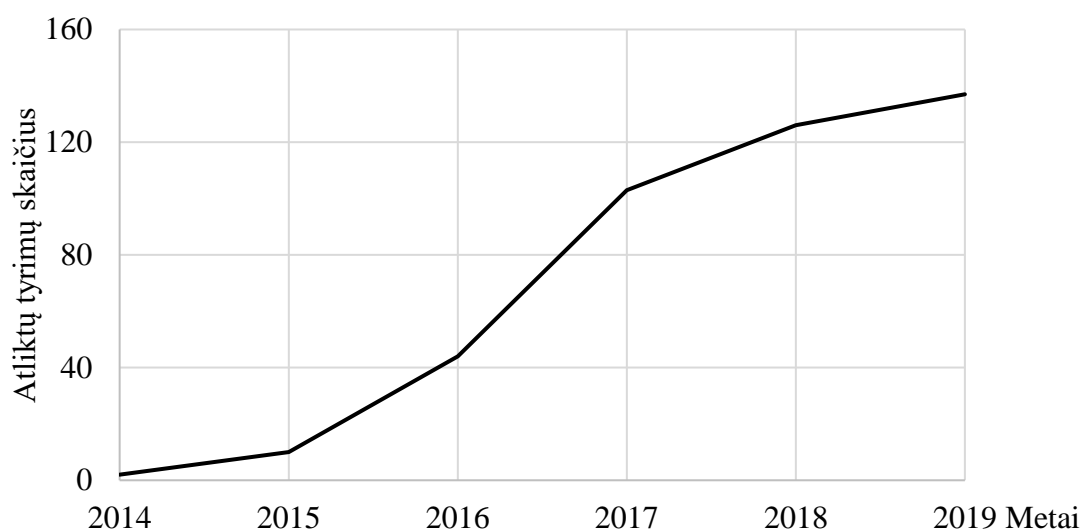
Pasak mokslininkų Zhengvei Vang, Kui She ir Tomo E. Vard, kiekviena iš GAN tinklo versijų stengiasi pagerinti charakteristikas trimis aspektais: vaizdo kokybė, režimų įvairovė ir nykstančiu gradientu [4] (žr. 9 pav.). Kaip matoma iš pateikto GAN tinklo vystymosi grafiko, visos vėliau pasiūlytos versijos (po originalaus GAN tinklo atsiradimo) pagerino vaizdo kokybę. SGAN buvo sukurtas pagerinti kelių klasių mokymąsi GAN tinkluose, tam kad generuojama didesnė įvairovė vaizdų. BigGAN privalumai yra tie patys kaip SGAN, bei geresnė tų vaizdų kokybė. Reikia paminėti, kad BigGAN ir PROGAN gali generuoti didelės rezoliucijos vaizdus.



9 pav. GAN tinklo vystymosi kryptys

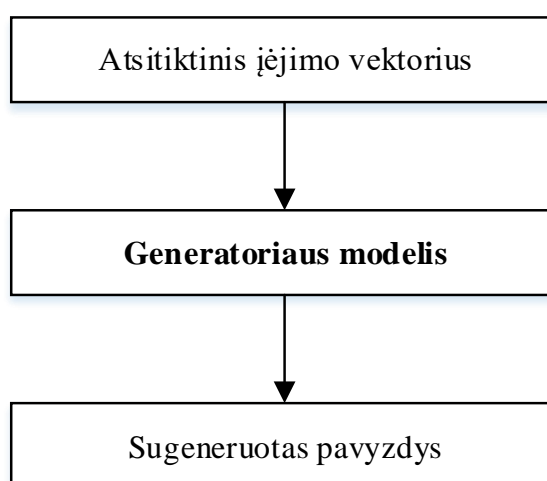
1.3.1. GAN tinklo veikimas

Nors kiekvienais metais didėja mokslinių darbų, kuriuose tyrinėjami GAN tinklai (žr. 10 pav.), tačiau dažniausiai GAN tinklą sudarantys du mažesni modeliai lieka pagrindiniu karkasu. Tai yra generatorius modelis, kuris skirtas generuoti naujiems pavyzdžiams ir diskriminatoriaus modelis, kuris nusprendžia ar pavyzdys yra tikras (paimtas iš originalaus duomenų rinkinio) ar sumodeliuotas generatoriaus. GAN tinklas yra naudingas atvejais, kai apmokymo duomenų skaičius yra labai ribotas, pavyzdžiui žmonių sergančių vėžine liga ir sveikų žmonių skaičius.



10 pav. GAN tema atliktų tyrimų skaičius per metus

Generatoriaus modelis paima fiksuoto ilgio atsitiktinį vektorių kaip įėjimą ir generuoja pavyzdį (žr. 11 pav.). Vektorius yra generuojamas iš Gauso triukšmo, jis yra naudojamas kaip pamatas generavimo procesui. Po treniravimo, taškai šioje kelių matmenų vektorinėje erdvėje atitiks taškus tikruosiuose pavyzdžiuose, sudarant suglaudintą duomenų paskirstymo atvaizdą.



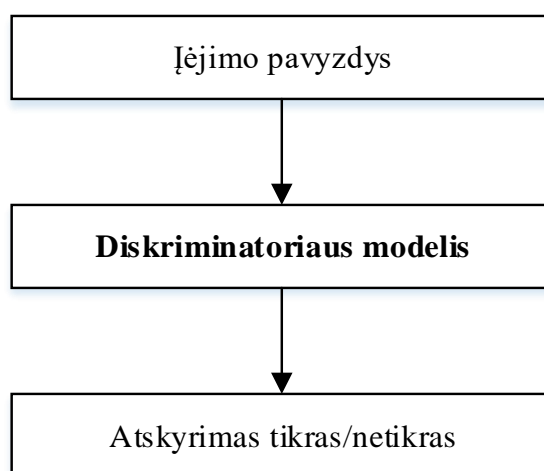
11 pav. GAN tinklo generatoriaus modelis

Ši vektorių erdvė yra vadinama latentine erdve arba vektorine erdve, kurią sudaro latentiniai kintamieji. Latentiniai kintamieji arba paslėpti kintamieji yra tie kintamieji, kurie yra svarbūs domenui, bet nėra tiesiogiai stebimi.

Latentiniai kintamieji arba latentinė erdvė dažnai yra apibrėžiami kaip duomenų paskirstymo projekcija arba glaudinimas. Latentinė erdvė pateikia suglaudintų arba aukšto lygio stebimų neapdorotų duomenų, tokių kaip įvesties duomenų paskirstymas, sąvokas. GAN atveju generatoriaus modelis atitinka reikšmes pasirinktos latentinės erdvės taškuose. Nauji taškai, surinkti iš latentinės erdvės, gali būti pateikti generatoriaus modeliui kaip įvestis ir naudojami generuoti naujus ir skirtingus išvesties pavyzdžius.

Diskriminatoriaus modelis (žr. 12 pav.) paima pavyzdį įėjime (realų arba sugeneruotą) ir nusprendžia ar duotas pavyzdys tikras ar netikras (sugeneruotas). Tikras pavyzdys paimamas iš apmokymo duomenų rinkinio $p_{duom}(x)$, o sugeneruoti pavyzdžiai iš generatoriaus modelio išėjimo $x = G(z; \theta^G)$.

Diskriminatorius yra gerai žinomas ir ištirtas klasifikavimo modelis. Po treniravimo proceso, diskriminatoriaus arba generatoriaus dažnai būna atsisakoma, priklauso nuo tuo metu sprendžiamos užduoties.



12 pav. GAN tinklo diskriminatoriaus modelis

Generatyvinis modeliavimas yra neprižiūrima mokymosi problema, tačiau GAN turi intelektualią architektūros savybę, kai generatyvinio modelio mokymas įvardijamas kaip prižiūrima mokymosi problema.

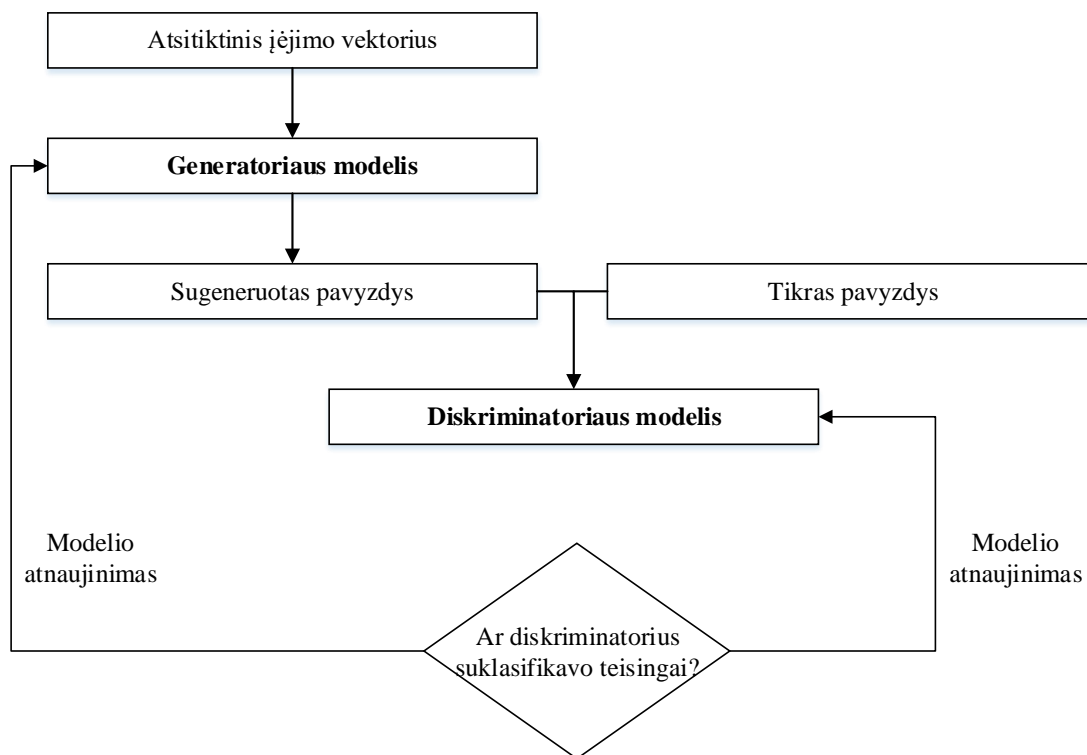
Abu modeliai, generatorius ir diskriminatorius yra mokomi kartu. Generatorius sugeneruoja pavyzdžių paketą, kuris kartu su realiais pavyzdžiais yra pateikiamas į diskriminatoriaus įėjimą, kuris juos klasifikuoja ar jie tikri ar suklastoti.

Tada diskriminatorius atnaujinamas, kad būtų galima geriau atskirti tikrus ir suklastotus pavyzdžius kitame etape. Svarbu, kad generatorius būtų atnaujinamas atsižvelgiant į tai, ar gerai, ar ne, sukurti pavyzdžiai apgavo diskriminatorių. Lygiai tokiu pačiu principu yra atnaujinamas ir diskriminatoriaus modelis, jeigu jis nesugebėjo teisingai atsakyti, ar pavyzdys tikras ar ne.

Šiuo atveju nulis reiškia, kad diskriminatorius sėkmingai identifikavo tikrus ir suklastotus pavyzdžius, tokiu atveju nereikia keisti jo modelio parametrų, tuo tarpu generatorius modelio parametrai yra pakeičiami. Analogiškai vienetas reiškia, kad generatorius sėkmingai sumodeliavo

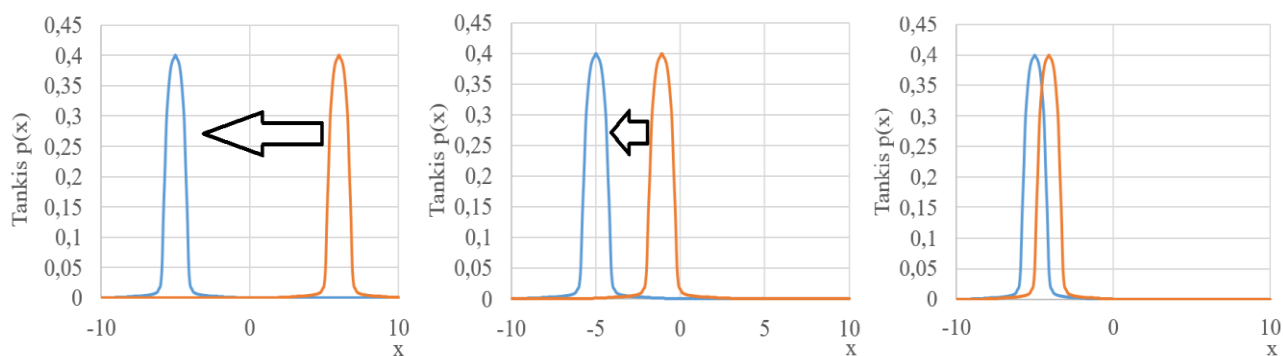
suklastotus pavyzdžius, tokiu atveju nereikia keisti jo modelio parametru, tuo tarpu diskriminatoriaus modelio parametrai yra pakeičiami.

Tokio mokymo ir parametru gerinimo eigoje (žr. 13 pav.), generatorius generuoja nepriekaištingas replikas išėjime kiekvieną kartą, o diskriminatorius negali pasakyti skirtumo ir kiekvienu atveju prognozuoja „nežinanti“ (pvz., 50% tikra ir netikra). Tai tik idealizuoto atvejo pavyzdys, realybėje nereikia patekti į šią ribą norint turėti naudingą generatoriaus modelį.



13 pav. Generatoriaus ir diskriminatoriaus mokymo modelis

Praktikoje dažniausiai modeliai mokomi po vieną. Mokomas arba diskriminatorius arba generatorius. Pavyzdžiui, jeigu turimas optimalus diskriminatorius arba generatorius, tuomet jo modelis nėra atnaujinamas. Esant optimaliam diskriminatoriui, apmokomas tik generatorius. Tokiu atveju jo tankio pasiskirstymas artėja prie diskriminatoriaus pasiskirstymo (žr. 14 pav.), o diskriminatoriaus pasiskirstymo funkcija lieka nepakitusi. Esant optimaliam generatoriui, keičiama tik diskriminatoriaus tankio pasiskirstymo funkcija, kuri kaip ir prieš tai aprašytu analogišku atveju, artėja prie generatoriaus pasiskirstymo funkcijos.



14 pav. Generatoriaus modelio mokymas, kai turimas optimalus diskriminatorius

1.3.2. GAN tipo tinklo pritaikymas paviršiaus defekto aptikimui

Defekto (naujumo) aptikimo užduotyje numatoma, kad diskriminatorius turi būti išbandytas su duomenimis, kurie nėra apmokymo duomenų aibėje $p_{duom}(x)$, arba kitaip tariant, su naujais duomenimis iš $p_{nauj}(x)$ aibės, kurie nepriklauso nei vienai iš K klasių, kurioms diskriminatoriaus klasifikatorius įprastai turėtų priskirti tikrus įėjimo duomenis. Jeigu aibės $p_{duom}(x)$, ir $p_{nauj}(x)$ yra iš anksto žinomos, tuomet apmokymo ir naujų duomenų kiekis optimaliam defekto aptikimo koeficientui α gali būti pasiektas pereinant ribą $(p_{duom}(x))/(p_{nauj}(x))$ ties tam tikra tenkinama verte.

Vienas iš standartinių defekto nustatymo būdų yra įvertinti apmokymo duomenų tankio nustatytą lygį $p_{duom}(x) > \alpha$, ir parinkti taškus už nustatyto lygo, kurie bus laikomi defekto aptikimu. Nustatytas tankio lygio įvertinimas yra padaromas esant prielaidai, kad defektai yra vienodai išsisklaidę nuo $p_{duom}(x)$. Tačiau, šie metodai yra sunkiai įgyvendinami, nes jiems reikia įvertinti platų, turintį daug dimensijų, apmokymo duomenų kiekį.

Blanchard ir kt. (2010) aptarė dalinai prižiūrimo naujumo aptikimo (angl. *Semi-Supervised Novelty Detection - SSND*) problemą, kai papildomai be apmokymo duomenų rinkinio yra prieinamas ir nepažymėtas duomenų rinkinys, sudarytas iš susimaišiusių apmokymo ir naujų duomenų pavyzdžių. Autoriai parodė, kad tokiu atveju yra galimybė pasiekti optimalų (su užsibrėžtu klaida/tiesa lygiu) defekto detektorius. Jeigu turimas priėjimas prie apmokymo duomenų $p_{duom}(x)$ ir susimaišiusių duomenų $\pi \cdot p_{nauj}(x) + (1 - \pi) \cdot p_{duom}(x)$, tuomet:

$$\frac{\pi \cdot p_{nauj}(x) + (1 - \pi) \cdot p_{duom}(x)}{p_{duom}(x)} = \frac{\pi \cdot p_{nauj}(x)}{p_{duom}(x)} + (1 - \pi) \quad (5)$$

, iš ko susidaro tinkamai išmatuojamas naujumo detektorius. Tačiau, SSND atveju, defekto aptikimas gali sumažėti dėl prižiūrimo klasifikavimo metodo. Tai priešingai tankio numatymo problemai, kai nėra turima naujų duomenų. Tai pat, problema nustatyti tinkamą patikimumo lygį apmokomiems duomenims, išlieka.

GAN tinklas buvo sukurtas tam kad išvengti daugelio trūkumų, kuriuos turi kiti generatyviniai modeliai:

- GAN tinklai gali generuoti pavyzdžius lygiagrečiai, vietoj to, kad naudotų laiką, proporcingą x matmeniui;
- GAN tinklo apmokymui nereikia daug realių pavyzdžių, kadangi generatorius modeliuoja labai tikroviškus pavyzdžius;
- Generatoriaus funkcijos dizainas turi labai mažai apribojimų. Tai yra privalumas palyginti su Boltzmano mašinomis, kurių tikimybės pasiskirstymas leidžia atpažinti Markovo grandinės pavyzdžius, ir palyginti su netiesine ICA, kuriai generatorius turi būti apverčiamas, o latentinio kodo z matmenys turi būti tokie patys kaip mėginių x ;
- Nereikia Markovo grandinių. Tai yra pranašumas, palyginti su „Boltzmann“ mašinomis ir GSN;
- Nereikia jokių variantų ribojimų, o konkrečios modelių šeimos, naudojamos GAN sistemoje, jau yra žinomos kaip universalios aproksimacijos, taigi GAN jau yra žinomas kaip asimptotiškai nuoseklus;

- Manoma, kad modeliuojant GAN tinklais gaunami geresni pavyzdžiai nei modeliuojant kitais metodais;

Tačiau GAN tinklai turi ir trūkumą: tam kad juos treniruoti, reikia rasti Nash pusiausią tarp diskriminatoriaus ir generatoriaus, kas yra didesnis iššūkis lyginant su objektyvios funkcijos optimizavimu.

Generatyviniai modeliai yra naudojami įvairiose srityse:

- Treniravimas ir pavyzdžių ėmimas iš generatyvinių modelių yra tinkams tikrinti galimybei atvaizduoti ir manipuluoti daug dimensijų tikimybių turintiems karkasams;
- Generatyviniai modeliai gali būti apmokyti su duomenimis kurie turi trūkstamos informacijos ir tuomet numatyti, kad trūkta informacijos įėjime;
- Suteikia galimybę mašininiam mokymuisi operuoti su keletu išėjimų.
- Labai aukštos rezoliucijos vaizdams generuoti. Nors aukštos rezoliucijos vaizdo galimybių iš vieno žemos rezoliucijos vaizdo yra ypatingai daug, parenkamas labiausiai atitinkantis tikimybių pasiskirstymą vaizdas;
- Sukurti interaktyvias programas, kurios padėtų vartotojui sukurti tikroviškus vaizdus, kurie atitiktų eskizą;

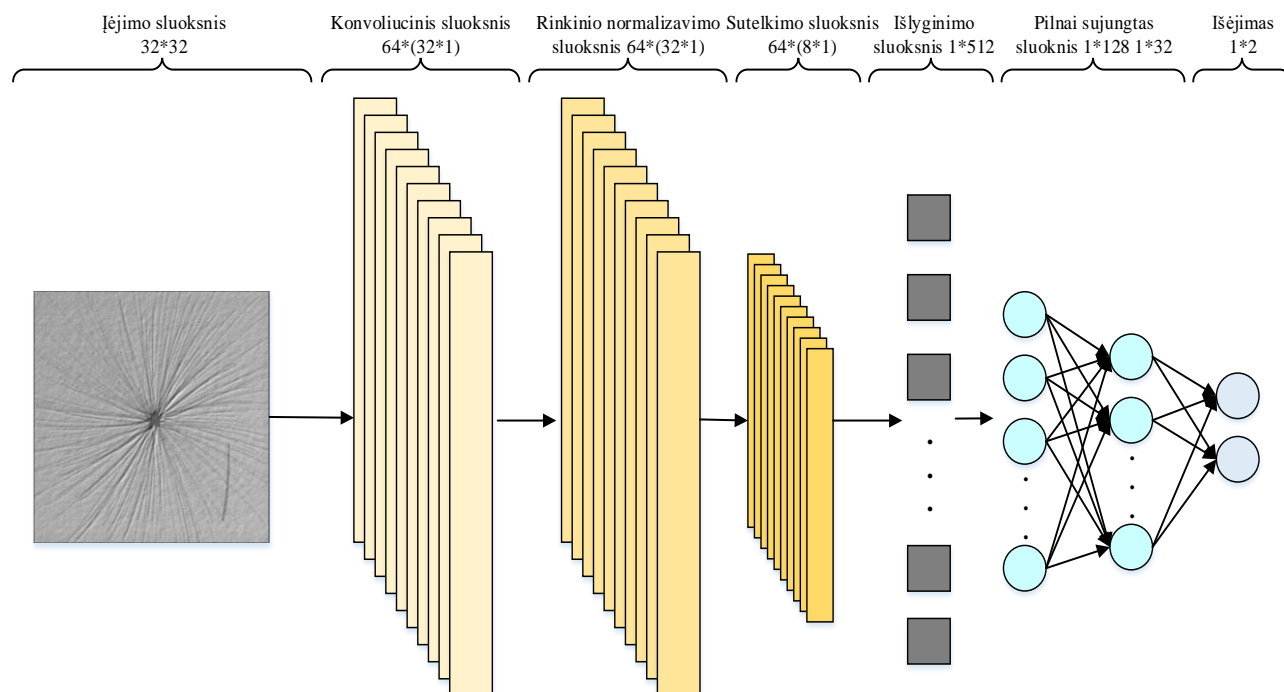
2. Metodinė dalis

2.1. GAN tinklo modelis

Gan tinklo realizavimui naudojamas gilusis konvoliucinis generatyvinis priešpriešinys tinklas – DCGAN (angl. *deep convolutional generative adversarial network*). Ši tinklo atmaina pasirinkta dėl gebėjimo tiksliau ir patikimiau apdoroti paveikslus ir vaizdo medžiagą [10],[11],[12] lyginant su standartiniu GAN tinklu, taip pat dėl gebėjimo optimaliau dirbti su didesniu sluoksnių skaičiumi, kas įtakoja bendrą modelio veikimo spartą.

Konvoliucinį tinklą sudaro įėjimo sluoksnis, kuriame yra paduodamas norimas apdoroti vaizdas. Po to seka konvoliucinis sluoksnis, kuriame naudojami pasirinkti filtrai. Dažniausiai po konvoliucinio sluoksnio naudojamas rinkinio normalizavimo sluoksnis (angl. *batch normalization layer*), kuris padidina modelio stabilumą. Po normalizavimo, naudojamas sutelkimo sluoksnis (angl. *pooling layer*), kuris suspaudžia turimus duomenis, juos sutraukia. Pastarieji 3 sluoksniai gali būti kartojami, taip padidinant modelio sluoksnių skaičių, tuo pačiu ir to modelio sudėtingumą. Esant skirtingoms konfigūracijoms gaunami skirtingi modeliai ir išėjimo rezultatai.

Prieš pereinant į pilnai sujungtus sluoksnius, būtina naudoti išlyginimo sluoksnį, po kurio duomenys tampa vienmačiai. Turint duomenis tokioje konfigūracijoje, jie perduodami į pilnai sujungtus sluoksnius (angl. *dense layers*). Šių sluoksnių skaičius taip pat nėra apibrėžtas. Skirtingas jų skaičius ir konfigūracija, sukuria skirtingai veikiančią modelį. Bendra konvoliucinio tinklo modelio sandara pateikta žemiau.

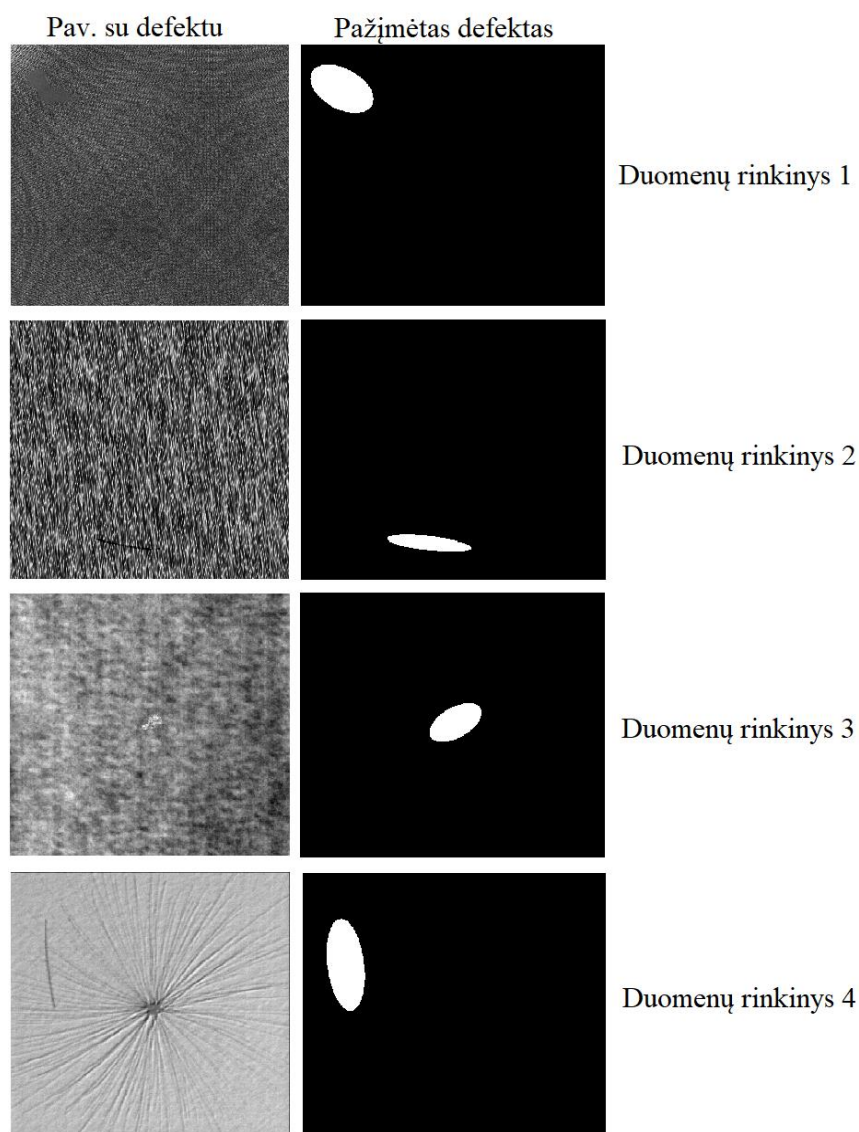


15 pav. Konvoliucinio tinklo modelis

Mokymui naudojama atvira duomenų bazė – DAGM2007 [13]. Duomenų bazė sudaryta iš 10 skirtingų duomenų rinkinių. Šie duomenų rinkiniai yra dirbtinai sugeneruoti, tačiau labai artimi realioms pavyzdžiams, su kuriais susiduriama. Pirmieji 6 iš 10 duomenų rinkinių yra pažymėti kaip

apmokymo duomenų rinkiniai, jie naudojami modelio tobulinimui. Likę 4 duomenų rinkiniai yra nurodyti kaip testavimo rinkiniai, jie gali būti naudojami modelio charakteristikų vertinimui esant naujam paviršiaus defekto pobūdžiui. Kiekvienas iš 10 duomenų rinkinių yra sudarytas iš:

- 1000 nedefektuotų ir 150 defektuotų vaizdų, kurie laikomi išsaugoti 8-bit PNG formatu;
- Kiekvienas iš duomenų rinkinių yra sugeneruotas naudojant vis kitokią tekstūrą ir defekto modelį (žr. 16 pav.);
- Nedefektuoti vaizdai rodo fono tekstūrą be defektų, o defektuoti vaizdai turi vieną fono tekstūros trūkumą, kuris yra pažymėtas;
- Visi duomenų rinkiniai yra atsitiktinai padalinti į apmokymo ir testavimo smulkesnius rinkinius, kurie yra vienodo dydžio;
- Defektuoti vaizdai papildomai yra pateikti pilku 8-bit vaizdu, PNG formatu. Reikšmės 0 ir 255 atitinkamai nurodo foną ir defektinę sritį;



16 pav. Duomenų rinkinių pavyzdžiai

Kaip matyti iš pateiktų duomenų rinkinių pavyzdžių, jie nėra spalvoti ir turi tik vienos spalvos ryškumo kitimą. Kadangi paveikslai aprašomi viena spalva, jie turi tik vieną kintamąjį papildomoje dimensijoje, kuris nusako spalvos ryškumą. Tuo tarpu jeigu būtų naudojami spalvoti paveikslai, tinklas turėtų papildomą 3 kintamųjų dimensiją.

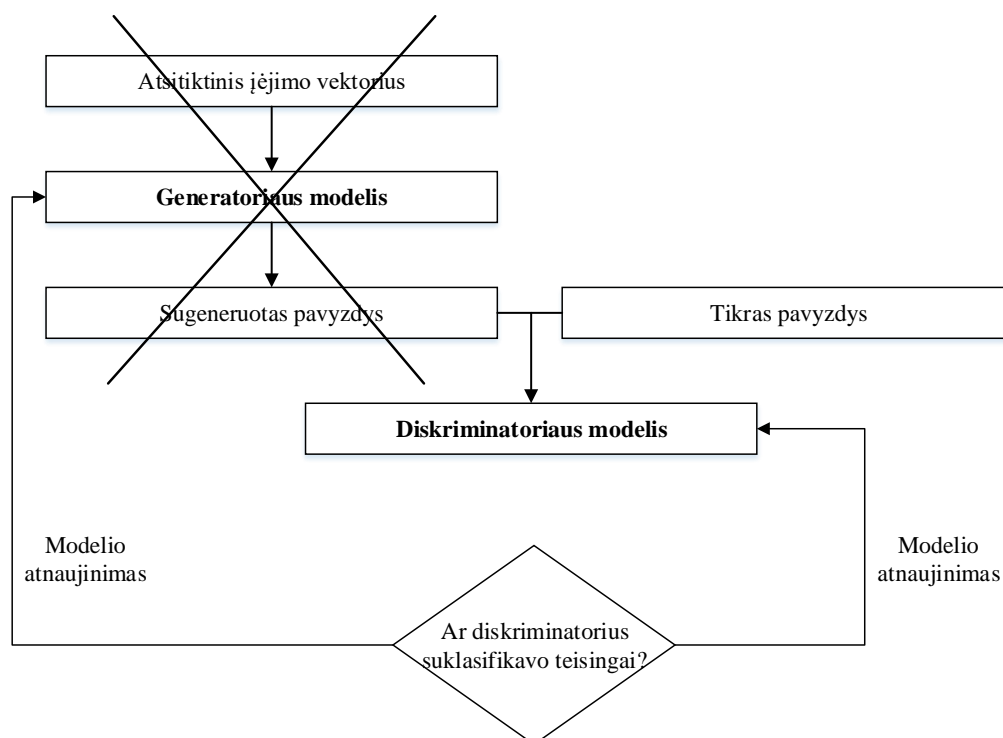
Visa papildoma informacija apie kiekvieną duomenų rinkinį yra sudėta į papildomą teksto failą ASCII formatu. Informacija yra pateikiama tokia struktūra:

1 lentelė. Papildomos duomenų rinkinių informacijos tvarka teksto faile

Elemento nr. 1 ID	„0“ jei defektuotas, „1“ jei nedefektuotas	Neapdoroto atvaizdo 1 failo pavadinimo numeris	0	Jei atvaizdas 1 defektuotas, tai defekto failo pavadinimas, „0“ kitu atveju
Elemento nr. 2 ID	„0“ jei defektuotas, „2“ jei nedefektuotas	Neapdoroto atvaizdo 2 failo pavadinimo numeris	0	Jei atvaizdas 2 defektuotas, tai defekto failo pavadinimas, „0“ kitu atveju
...
Elemento nr. N ID	„0“ jei defektuotas, „N“ jei nedefektuotas	Neapdoroto atvaizdo N failo pavadinimo numeris	0	Jei atvaizdas N defektuotas, tai defekto failo pavadinimas, „0“ kitu atveju

Duomenys iš papildomo failo naudojami kuriant diskriminatoriaus modelį. Paduodant paveikslą įėjime iš karto yra žinoma ar paveiksle yra defektas. Jei jo nėra, tai visi aptikti defektai laikomi diskriminatoriaus modelio klaida. Kitu atveju, esant defektui, papildomai yra naudojamas to defekto tikrosios vietos (angl. *ground true*) paveikslas. Jis papildomai apdorojamas ir sulyginamas su diskriminatoriaus išvesties duomenimis.

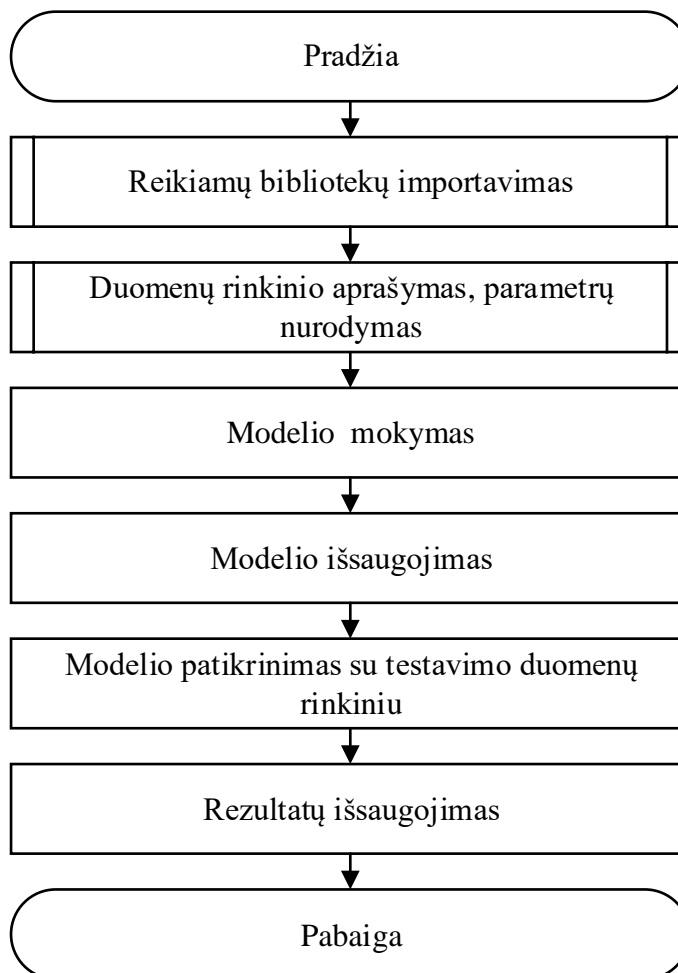
Kadangi naudojamoje DAGM2007 duomenų bazėje yra pakankamai duomenų apmokyti GAN tinką, generatoriaus modelis kol kas yra nenaudojamas. Tokiu atveju turimas tinklas pasidaro ne toks sudėtingas, lieka tik diskriminatoriaus modelis (žr. 17 pav.).



17 pav. Supaprastintas GAN tinklo modelis

2.2. Modelio realizavimas

Programos pagrindinis algoritmas pateikiamas 18 pav. Programos pradžioje importuojamos visos bibliotekos, kurios bus reikalingos stabiliai veikti programai. Naudojamos programos, bibliotekos ir kitos priklausomybės pateikiamos 2 lentelėje.



18 pav. Pagrindinis programos algoritmas

2 lentelė. Naudojamų programų ir bibliotekų sąrašas

Eil. nr.	Pavadinimas	Versija
1	Python	3.7.6
2	TensorFlow	2.0.0a0
3	Numpy	1.16.4
4	Keras	2.3.1
5	TensorBoard	1.6.0
6	Pip	20.1
7	Visual Studio	16.5.3

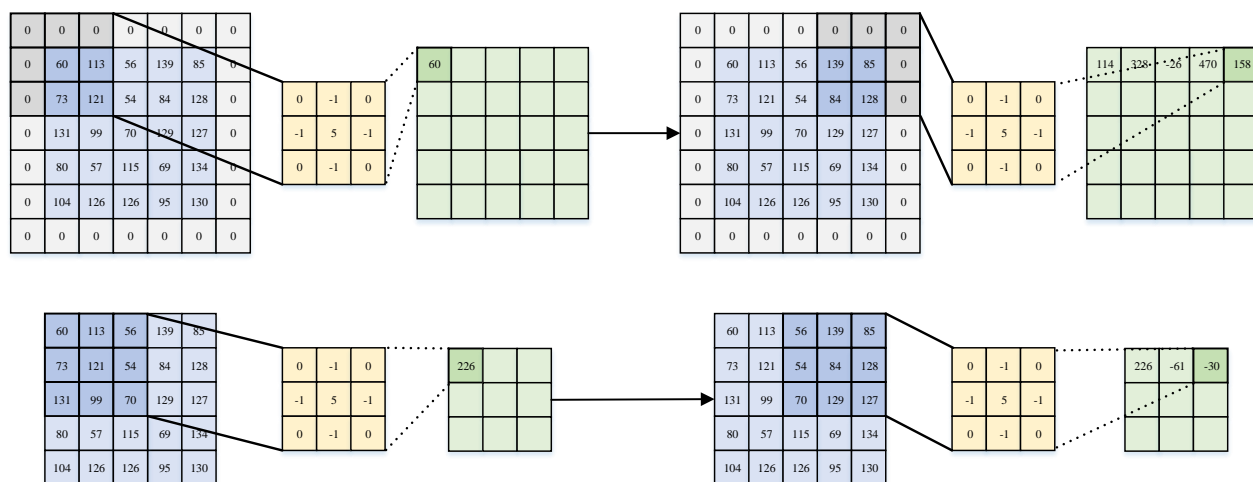
Duomenų rinkinys darbo metu buvo aprašomas skirtingais metodais ir parametrais, kurie nurodo segmentuojamo paveikslėlio segmento dydį, konvoliucijų, sutelkimo ir pilnai sujungtų sluoksnių

skaičių, filtro branduolio dydį ir kitus parametrus. Toliau vykdomas duomenų mokymas, į kurį įeina duomenų paruošimas, jų pažymėjimas. Po šio proceso gaunamas galutinis diskriminatoriaus modelis, kuris yra išsaugomas. Po to vykdomas modelio patikrinimas su duomenimis, kuriais diskriminatoriaus modelis nebuvo prieš tai mokomas. Gauti rezultatai išsaugojami tolesniam tyrimui ir palyginimui tarp skirtingų modelių.

Duomenų rinkinys aprašomas, kaip nespaltvoti vaizdai, kurie yra 512 x 512 pikselių raiškos. Kiekvienam tinklo apmokymui naudojami 575 paveikslai, iš kurių 14 – 16 % turi defektus. Nurodomi kiekvienos klasės vaizdų numeriai, nuo kurio prasideda klasės numeracija aplanke. Šis numeris bus naudojamas vaizdų įkėlimui į pačią programą. Taip pat nurodoma vieta, kur kiekvienas duomenų rinkinys ir jam priskirti aplanakai yra patalpinti. Nurodoma, kuri dalis iš turimų mokymo duomenų bus naudojama mokymosi patikrinimui, modelio mokymosi metu.

Duomenų aprašymo stadijoje yra nurodoma, kokio dydžio segmentais vaizdas bus segmentuojamas (plačiau apie segmentaciją duomenų paruošimo etape). Taip pat nurodomas standartinis 3 x 3, 5 x 5 arba 7 x 7 filtro branduolio dydis. Esant mažiems, iki 128 x 128 taškų vaizdams, dažniausiai parenkamas 3 x 3 filtro branduolio dydis. Esant didesniems vaizdams, galima naudoti ir didesnio dydžio 5 x 5 arba 7 x 7 filtro branduolį, tačiau didesni nei 7 x 7 filtro branduolio dydžiai praktikoje retai sutinkami.

Aprašant pradinius duomenis, taip pat nurodoma, koks bus naudojamas konvoliucinio filtro dydis ir ar bus naudojamas padidinimas (pridedami papildomi 0 aplink įėjimo parametrus), tam kad nebūtų sumažinamos dimensijos konvoliucijos metu (žr. 19 pav.).



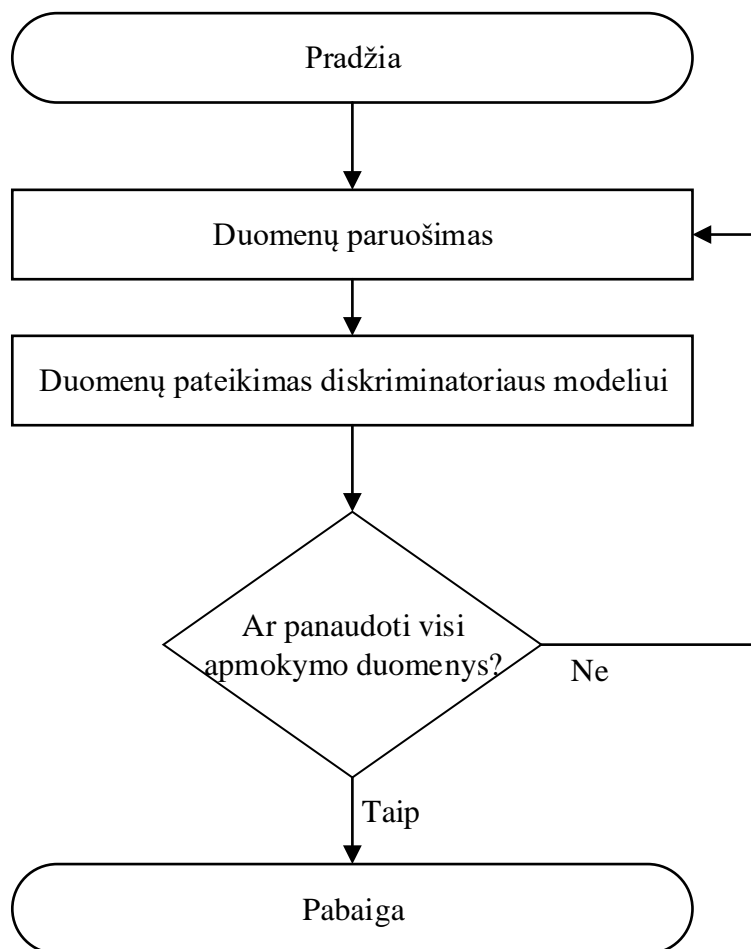
19 pav. Konvoliucija su 3x3 filtro branduolio dydžiu. Viršuje su padidiniu, apačioje matomas išėjimo matricos sumažėjimas

Išėjimo matricos dydis apskaičiuojamas pagal formulę:

$$x = \frac{W-K+2 \cdot P}{S} + 1 \quad (6)$$

, kur x – išėjimo matricos forma, W – įėjimo matricos forma, K – filtro branduolio dydis, P – padidinimo parametras (1 arba 0), S – žingsnis. Žingsnis naudojamas sumažinti išėjimo matricą ir atlieka funkciją, panašią kaip ir algoritmas maksimalus sutelkimas. Kiekvienu kuriamo modelio atveju, eksperimentiškai galima nustatyti, kuri iš šių dviejų algoritmų tinka labiau.

Aprašius duomenų rinkinį ir nurodžius pagrindinius parametrus pereinama prie tinklo apmokymo. Apmokymo svarbiausia dalis – teisingas duomenų paruošimas (apie pilną duomenų paruošimo realizaciją, žiūrėti duomenų paruošimo algoritmo aiškinime). Tik paruoštus duomenis galima perduoti į diskriminatoriaus modelio apmokymo funkciją (žr. 20 pav.).

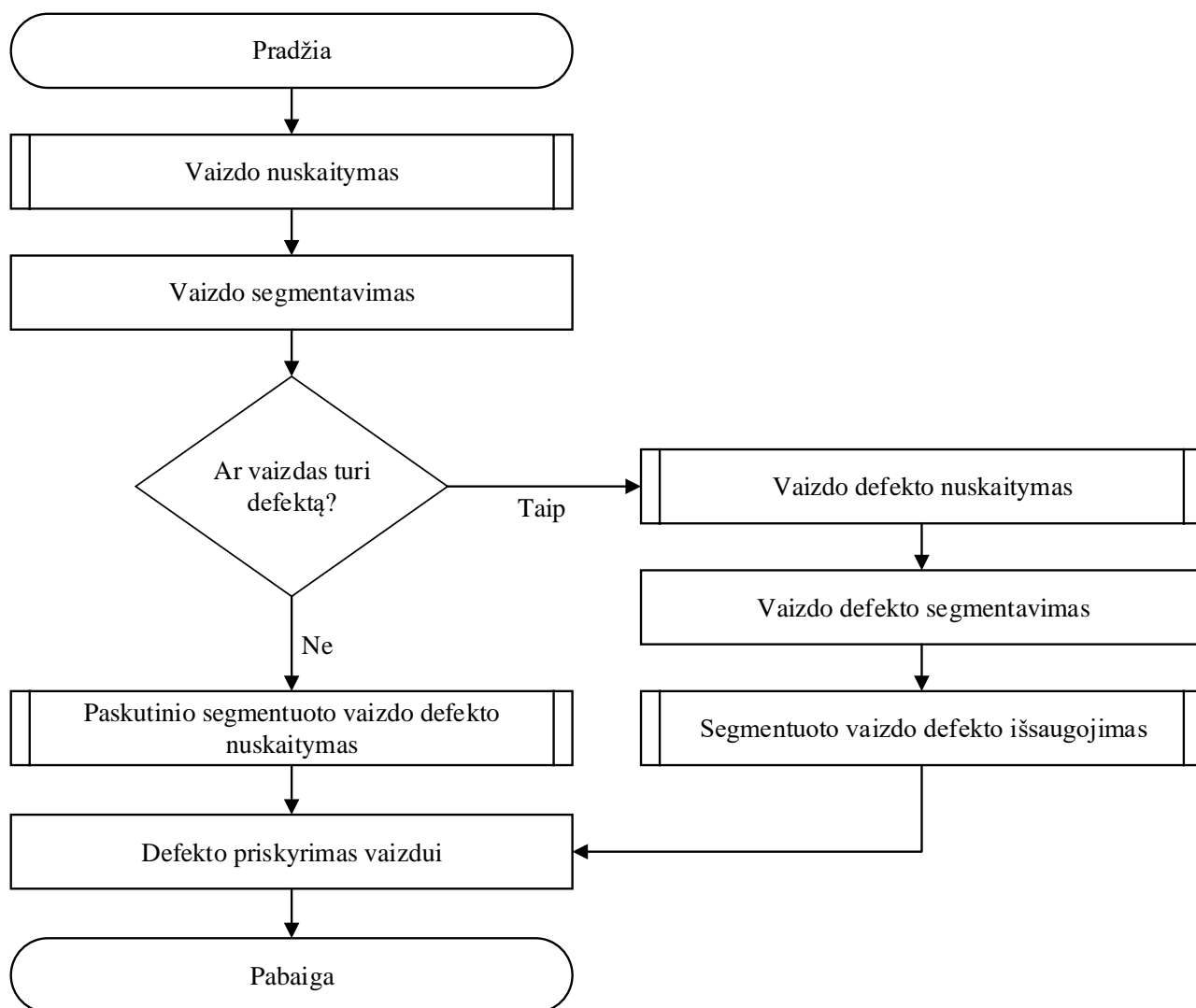


20 pav. Modelio apmokymo algoritmas

Apmokymas vyksta su kiekvienu paruoštu duomenų rinkiniu, kuriuo apmokomas diskriminatorius. Iš viso į diskriminatorių paduodama 575 duomenų rinkiniai (šiuo atveju), kurie yra sugrupuoti ir pažymėti – geras ar blogas pavyzdys. Kiekvienas duomenų rinkinys sudarytas iš 4 dimensijų: vaizdo plotis, vaizdo aukštis, dimensija nurodanti ar vaizdas yra spalvotas (n spalvoto vaizdo dimensija turi 1 narį, o spalvoto vaizdo 3 narius) ir dimensija nurodanti vaizdo klasifikaciją. Skirtingai nuo klasifikuojamų elementų, modelio išėjime galima turėti atsakymą taip arba ne, o jei bandoma atpažinti skaičius nuo 0 iki 9, išėjime galime turėti ir reikšmes atitinkančias tuos skaičius. Tos reikšmės atitinkamai turi būti nurodytos klasifikavimo dimensijoje ir pateikiamos modelio apmokymo etape. Kol visi duomenų rinkiniai nėra panaudoti, teigiama, kad diskriminatoriaus modelis vis dar yra mokomas.

Mokymo pradžioje visada gaunamas mažas tikslumas ir diskriminatorius dažnai klysta. Po 50 ir daugiau apmokymo duomenų rinkinių, dažniausiai mokomo modelio charakteristikos pradeda nusistovėti. Pradžioje esantis tikslumas nuo 20 – 30 % pakyla iki 90 % ir daugiau, priklausomai nuo to, kaip gerai pavyko apmokyti modelį. Tuo pačiu metu, modelio klaidingi spėjimai mažėja atvirkščiai tikslumo kreivei.

Duomenų paruošimo etape visų pirma yra nuskaitymas vaizdas (žr. 21 pav.), pagal prieš tai nurodytą jo būvimo vietą ir pavadinimą. Šiam veiksmui atlikti yra naudojama „Visual Studio“ cv2.imread(vieta) funkcija.



21 pav. Duomenų paruošimo mokymui algoritmas

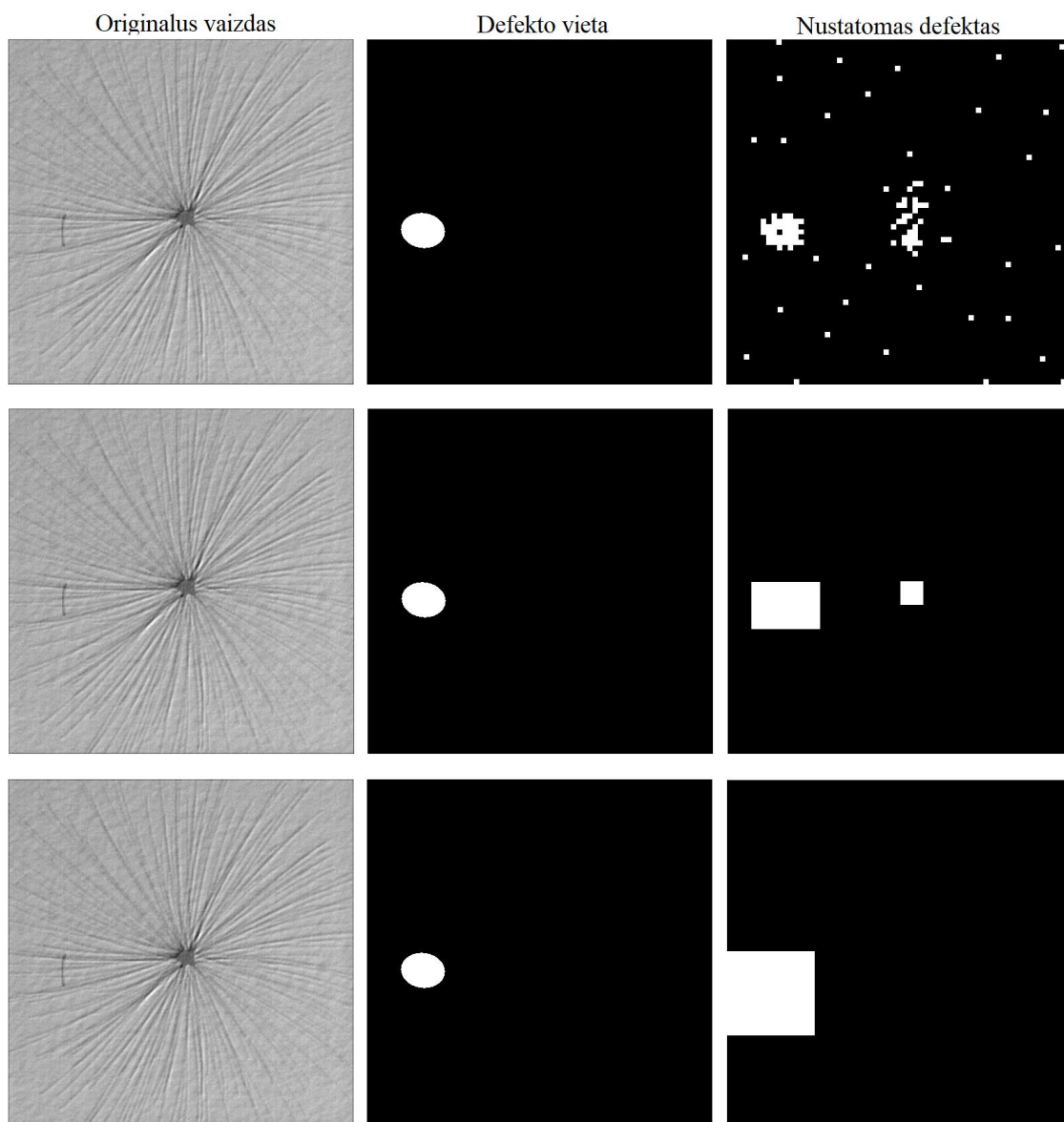
Kadangi reikalinga aptikti konkrečią defekto vietą, o ne tik faktą ar vaizdas turi defektą, pradinės nuotraukos yra padalijamos į daug mažų segmentų [16]. Žemiau pateiktoje lentelėje pateikiamas gaunamas mažesnių nuotraukų skaičius, priklausomai nuo to, koks pasirinkamas segmentavimo dydis.

3 lentelė. Po segmentavimo gaunamų nuotraukų skaičius

Eil. nr.	Nuotraukos dydis (taškais)	Segmento dydis (taškais)	Po segmentavimo gaunamų nuotraukų skaičius (vnt.)
1	512 x 512	8 x 8	4096
2	512 x 512	16 x 16	1024
3	512 x 512	32 x 32	256
4	512 x 512	64 x 64	64
5	512 x 512	128 x 128	16

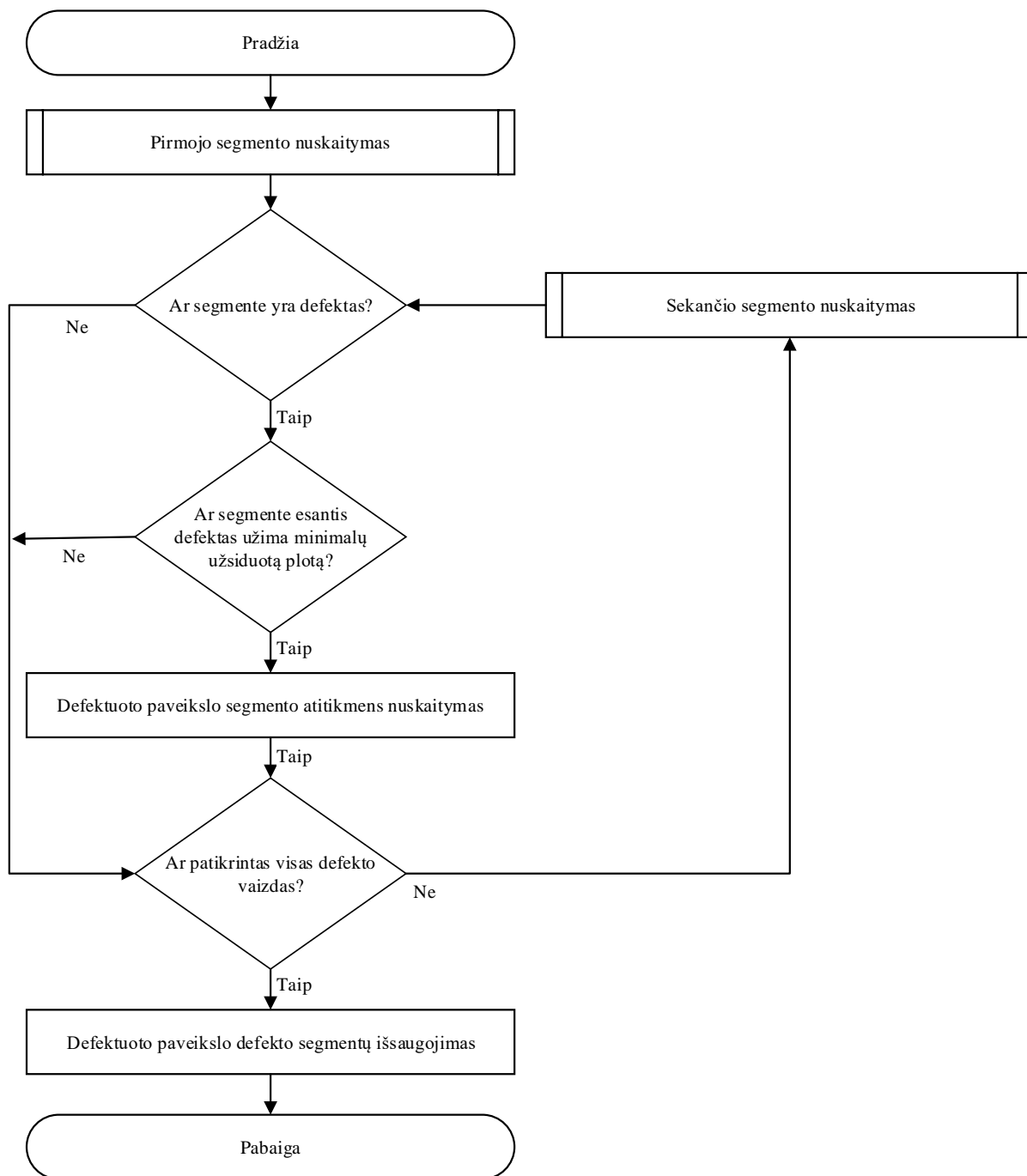
Po tokio segmentavimo ir modelio apmokymo šiomis nuotraukomis, bus gaunamas diskriminatorius, kuris galės klasifikuoti tokius pačius segmentus kaip ir apmokymo stadijoje, kaip turinčius defektą arba ne. Padavus visą vaizdą po vieną segmentą, nustačius ar tas segmentas yra defektuotas ar ne, bus gaunama apytikrė defekto būvimo vieta.

Esant skirtingiems segmentavimo dydžiams susiduriama su skirtingomis problemomis. Jeigu naudojamas 8 x 8 segmentavimas, gaunamas didelis duomenų rinkinio skaičius ir modelis gali būti geriau apmokomas. Tačiau kyla problemų su triukšmu ir neteisingai atpažintais segmentais, kadangi paimamas mažas vaizdo plotas gali būti atpažintas kaip defektas (žr. 22 pav., viršuje). Kitu atveju, jeigu parenkamas per didelis segmentavimas, defekto vieta gali būti labai grubiai nustatyta ir neatspindėti tikrojo defekto dydžio (žr. 22 pav., apačioje). Dėl minėtų problemų reikia tinkamai parinkti segmentavimo dydį, pagal siekiamą gauti rezultatą. Dažniausiai siekiamas optimalus variantas tarp triukšmo ir perdėtos defekto atpažinimo apimties.



22 pav. Defekto aptikimas su 8 x 8 segmentavimu viršuje, 32 x 32 segmentavimu viduryje ir 64 x 64 segmentavimu apačioje

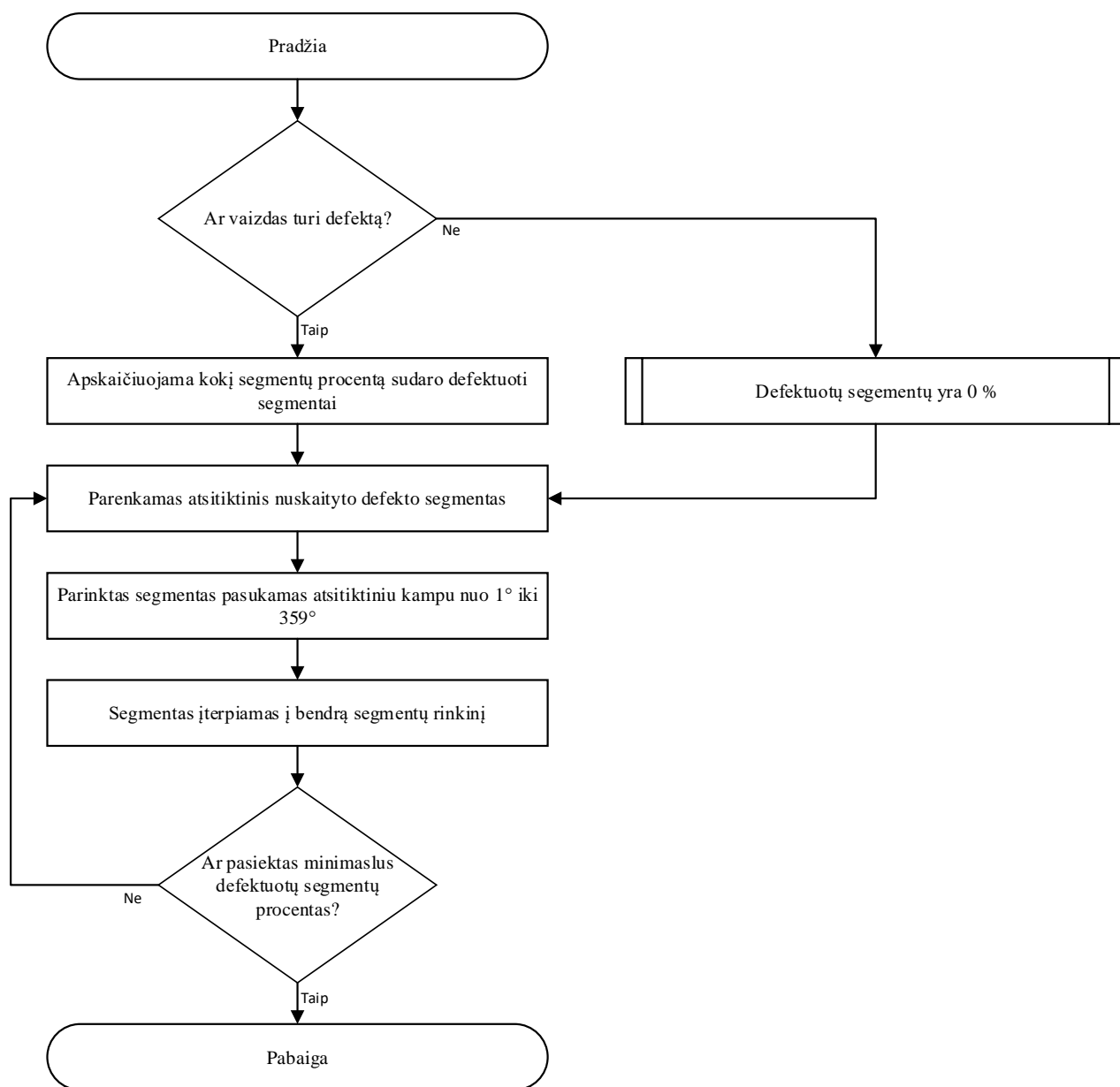
DAGM2007 duomenų rinkinys taip pat turi paveikslus su tikraisiais defektais. Jei su paviršiaus vaizdu susieto defekto vaizde pažymėtas defektas, tai tikrajame vaizde ta pati vieta laikoma turinti defektą. Tokiu atveju, defekto vaizdas yra segmentuojamas tokiais pačiais segmentais kaip ir vaizdas turintis tą defektą. Nustačius, kad segmente yra defektas, segmento defekto dydis yra palyginamas su užsiduota verte (jei neviršija 20 %, laikoma, kad defekto segmente nėra), galutiniam turinčio defektą segmento įvertinimui. Jei defektas segmente nustatytas, paimama to segmento atitiktis originalaus turinčio defektą paviršiaus vaizde. Patikrinus pilną defekto vaizdą ir surinkus visus aptiktus defekto atitikmenis originalaus turinčio defektą paviršiaus vaizde, surinkti segmentai išsaugojami (žr. 23 pav.).



23 pav. Defektuoto paveikslo defekto segmentų išsaugojimo algoritmas

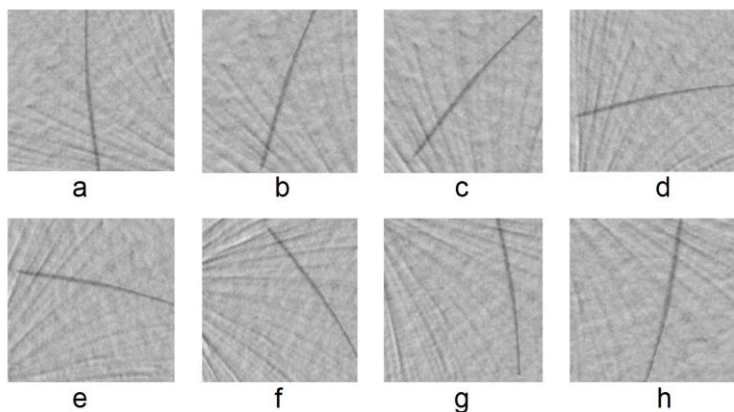
Mokymo eigoje geriausia turėti apie 50 – 70 % duomenų, kurie būtų teisingi ir likusius duomenis su defektais, tokiu atveju diskriminatoriaus modelis apmokomas optimaliausiai ir gali panašiu procentu atskirti defektuotus ir gerus duomenis. DAGM2007 duomenų rinkinį sudaro tik 14 – 16 % vaizdų, kurie turi defektą. Kadangi vaizduose užimamų defektų plotas yra santykinai mažas, tai bendras turinčio defektą paviršiaus plotas kartais nesiekia net 3 % visų vaizdų ploto bendrai paėmus.

Dėl šios priežasties reikalinga sukurti papildomų turinčio defektą paviršiaus pavyzdžių ir dirbtinai išplėsti apmokymui naudojamą duomenų kiekį, jį papildant modifikuotais turinčio defektą paviršiaus pavyzdžiais (žr. 24 Pav.). Tinklo apmokymas su papildomais identiškais turinčiais defektą segmentais, kurie buvo aptikti anksčiau, nesuteiktų mokomam modeliui jokios papildomos informacijos. Dėl šios priežasties, kiekvienas iš atsitiktinai parinktų defekto segmentų yra pasukamas atsitiktinai parinktu kampu, nuo 1° iki 359°. Gauti modifikuoti segmentai įterpiami į bendrą segmentuotų vaizdų duomenų rinkinį atsitiktinėse vietose.



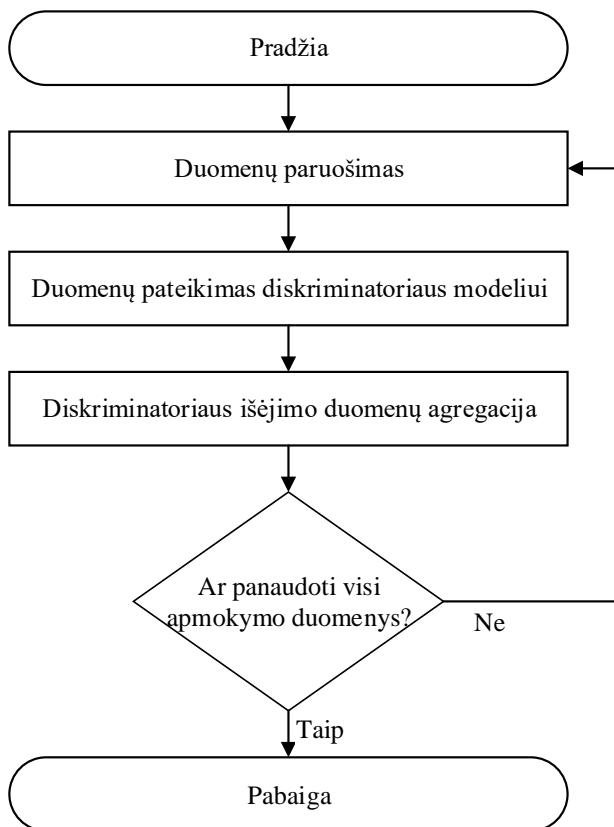
24 pav. Defekto priskyrimo vaizdai algoritmas

Kiekvienas defektuotas segmentas kuris yra pasuktas kampu apmokomam modeliui suteikia papildomos informacijos, kaip tas defektas atrodytu kitoje vietoje ar paverstas šonu. Tokiu atveju modelis yra apmokomas atpažinti įvairiomis kryptimis atsiradusį defektą (žr. pav. 25.). Sugeneravus ir priskyrus pakankamą kiekį defektų segmentuotų vaizdų duomenų rinkiniui, rinkinys yra perduodamas diskriminatoriaus apmokymui. Kai diskriminatoriui perduodami kiekvienos nuotraukos segmentuoti duomenų rinkiniai, mokymas laikomas baigtu ir gautas modelis išsaugomas. Išsaugotas modelis vėliau gali būti naudojamas palyginimui tarp skirtingai realizuotų diskriminatoriaus modelių, jų palyginimui.



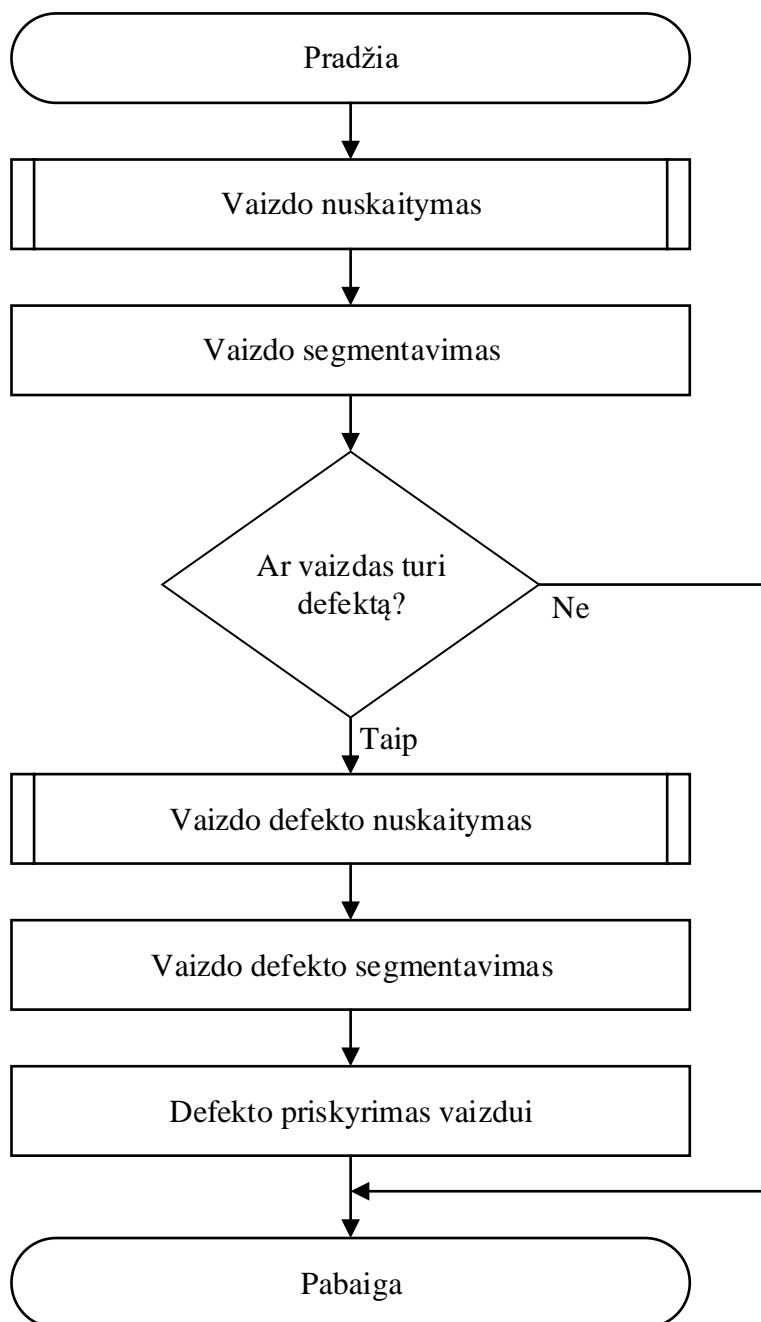
25 pav. Naujų defekto pavyzdžių sukūrimas, juos pasukant kampu

Po išsaugojimo diskriminatoriaus modelis yra testuojamas (žr. 26 pav.).



26 pav. Modelio testavimo algoritmas

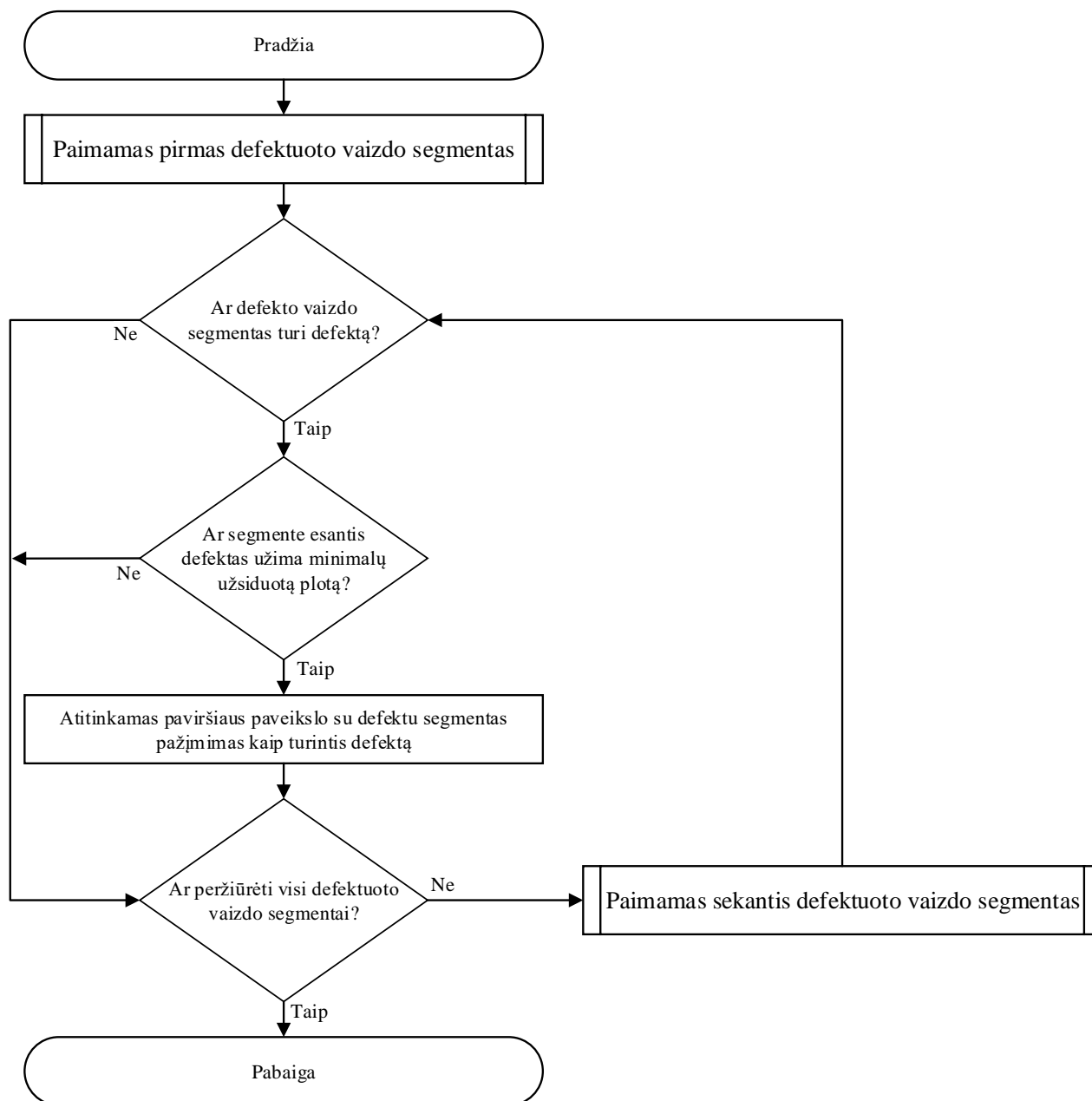
Modelio testavimas realizuotas labai panašiais žingsniais kaip ir jo mokymas, tačiau yra esminių skirtumų kiekviename realizacijos etape. Visų pirma, testavimo etape reikalingas duomenų paruošimas (žr. 27 pav.).



27 pav. Duomenų paruošimo testavimui algoritmas

Vaizdo paruošimo testavimui etape, kaip ir vaizdo paruošimo apmokymo etape, taip pat yra daromas vaizdo segmentavimas. Jei vaizdas neturi defekto, kiekvienas iš segmentų pažymimas kaip geras. Kitu atveju yra vykdomas vaizdo defekto nuskaitymas ir priskyrimas individualiam segmentui.

Šis priskyrimo algoritmas skiriasi nuo algoritmo, kuris yra naudojamas modelio apmokymo etape. Šiuo atveju, turinčiais defektą segmentais pažymimi tik tie vaizdai, kurie iš tikrųjų turi defektą, jų dirbtinai nedidinat (žr. 28 Pav.). Modeliui bus paduodamas segmentas, su defekto žyma, tačiau pažymė bus naudojama tik modelio atsakymui įvertinti.



28 pav. Defekto priskyrimo vaizdui mokymo etape algoritmas

Paruošti duomenys toliau po vieną segmentą pateikiami diskriminatoriaus modeliui, kuris turi atpažinti ar šis segmentas yra su defektu ar ne. Pagal tai, kaip diskriminatorius įvertina segmentą ir priklausomai nuo to, koks tas segmentas buvo, diskriminatoriaus atsakymas skirstomas į 4 grupes (žr. 4 lentelę). Išanalizavus visus testavimo duomenis, galima apskaičiuoti modelio tikslumą ir kitus modeliui būdingus parametrus.

4 lentelė. Modelio atsakymo vertinimas

Eil. nr.	Segmento defektas	Modelio įvertinimas	Atsakymo vertinimas
1	Ne	Ne	FN (angl. <i>false positive</i>)
2	Ne	Taip	FP (angl. <i>false positive</i>)
3	Taip	Ne	TN (angl. <i>false negative</i>)
4	Taip	Taip	TP (angl. <i>true positive</i>)

Turint šiuos parametrus visam testavimo duomenų rinkiniui, galima apskaičiuoti pagrindinius sukurto modelio kokybės įvertinimo parametrus.

$$\alpha = \frac{FP}{FP+TN} \quad (7)$$

α – klaidingai teigiamas rodiklis. Tai pirmo tipo paklaida, kuri nurodo neigiamų atvejų dalį, neteisingai nustatytą kaip teigiami atvejai.

$$\beta = \frac{TP}{TP+FN} \quad (8)$$

β – klaidingai neigiamas rodiklis arba atšaukimo statistika (angl. *recal*). Tai antro tipo paklaida, kuri yra apibrėžiama kaip galimybė atkurti elementą iš daiktų (paveikslų) sąrašo, kuris iš pradžių buvo pateiktas ankstesniame mokymosi etape, pateikus tinkamą nurodymą arba spontaniškai [15].

$$P = 1 - \beta \quad (9)$$

P – jautrumas. Dažniausiai jautrumas reiškia galimybę aptikti efektą ar reikšmę tam tikrame duomenų rinkinyje.

$$LR+ = \frac{P}{\alpha} \quad (10)$$

$LR+$ – teigiamas tikimybės koeficientas (angl. *likelihood ratio positive*). Šis koeficientas nusako kelios TP reikšmės atitinka vieną FP reikšmę.

$$LR- = \frac{\beta}{1-\alpha} \quad (11)$$

$LR-$ – neigiamas tikimybės koeficientas (angl. *likelihood ratio negative*).

$$Prec = \frac{TP}{TP+FP} \quad (12)$$

$Prec$ – tikslumas. Tikslumas įvertina koks tikslus yra modelis numatant teigiamai pažymėtus pavyzdžius. Tikslumas atsako į klausimą, kiek kartų modelis buvo teisingas, kiek dažnai jis buvo teisingas.

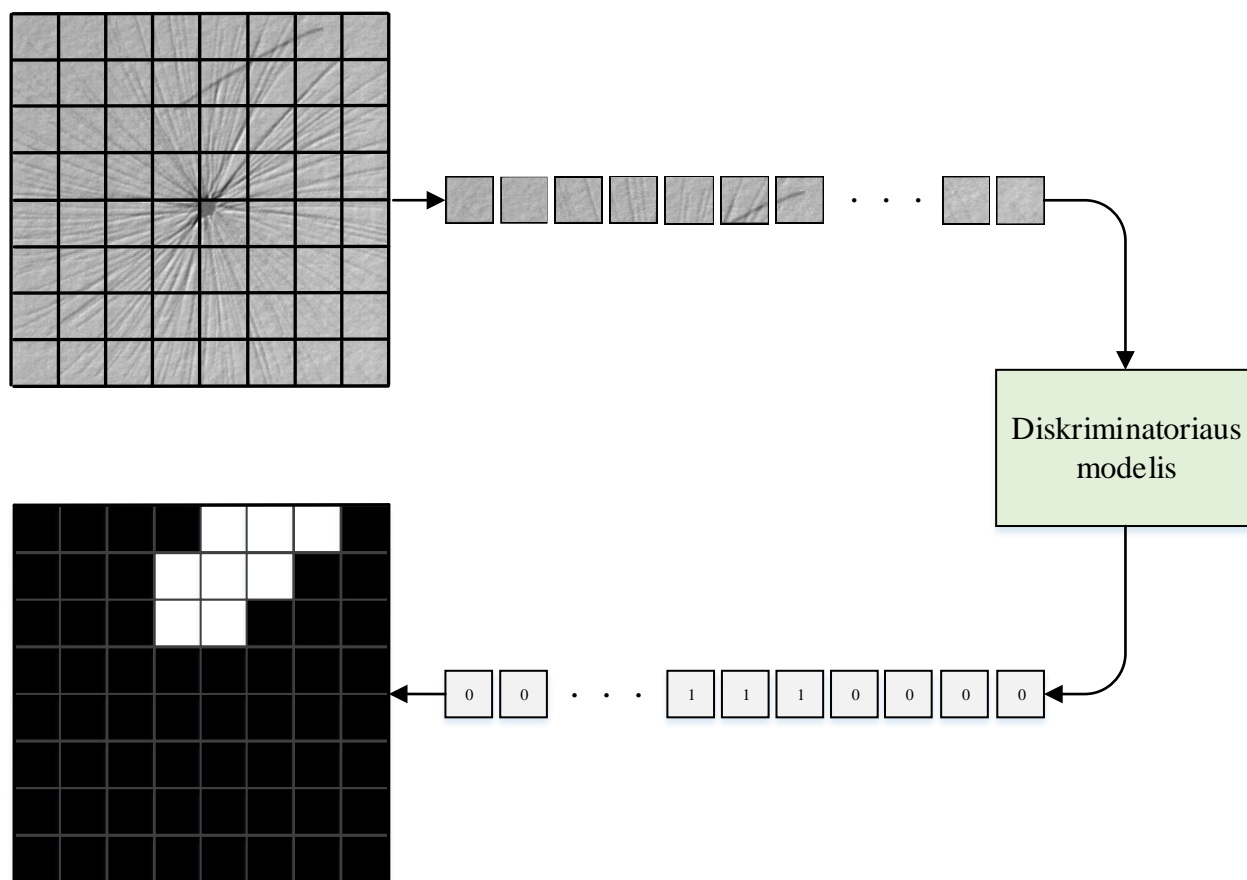
$$F_1 = 2 \cdot \frac{\beta \cdot Prec}{\beta + Prec} \quad (13)$$

F_1 – balas nusakantis tikslumo ir atšaukimo svertinį vidurkį. Šis balas atsižvelgia tiek į klaidingus teigiamus, tiek į melagingus negatyvus. Intuityviai tai nėra taip lengva suprasti kaip tikslumą, tačiau F_1 paprastai yra naudingesnis nei tikslumas, ypač jei klasių pasiskirstymas yra netolygus. Tikslumas geriausiai veikia, jei klaidingų teigiamų ir klaidingų negatyvų vertė yra panaši. Jei melagingų teigiamų ir melagingų negatyvų vertė labai skiriasi, geriau žiūrėti į tikslumą ir atšaukimo statistiką.

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \quad (14)$$

Acc – taiklumas. Taiklumas yra labiausiai intuityvus atlikimo matas ir tai yra teisingai numatytų stebėjimų santykis su visais stebėjimais. Taiklumas gerai nusako sistemos veikimą tik tada, kai turimi duomenų rinkiniai yra simetriniai, kuriuose klaidingų teigiamų ir klaidingų neigiamų reikšmės yra beveik vienodos.

Kai visi vaizdo segmentai yra įvertinami diskriminatoriaus, vaizdas yra rekonstruojamas į pradinį dydį, vizualiam vaizde aptikto defekto įvertinimui (žr. 29 pav.). Gauti vaizdai išsaugomi tolimesniam nuotraukų palyginimui.



29 pav. Defektuoto paviršiaus vaizdo segmentacija, diskriminatoriaus apdorojimas ir gautų segmentų įvertinimo rekonstrukcija

Norint papildyti diskriminatoriaus modelį, reikalinga daugiau segmentų su defekto ir paviršiaus, kuris neturi defekto, pavyzdžiais. Tikslui pasiekti yra naudojamas generatoriaus modelis, kuris gali tuos segmentus sugeneruoti. Generatoriaus modelis veikia panašiu principu kaip ir diskriminatorius, tačiau jo tikslas yra priešingas. Generatorius turi generuoti kuo tikroviškesnius vaizdus, kuriuos diskriminatorius atpažintų kaip tikrus. Kadangi po apmokymo ir patikrinimo su realiais duomenimis yra žinomas diskriminatoriaus veikimas ir jo charakteristikos, galima kurti ir tobulinti generatoriaus modelį.

Generatoriui yra paduodamas atsitiktinis vektorius iš latentinės erdvės, kuris užteršiamas atsitiktiniu triukšmu. Generatorius apsirąšomas taip pat kaip ir diskriminatorius, tačiau jo modelis juda priešinga kryptimi. Iš vienmačio vektoriaus generatorius sukuria dviejų dimensijų paveikslą.

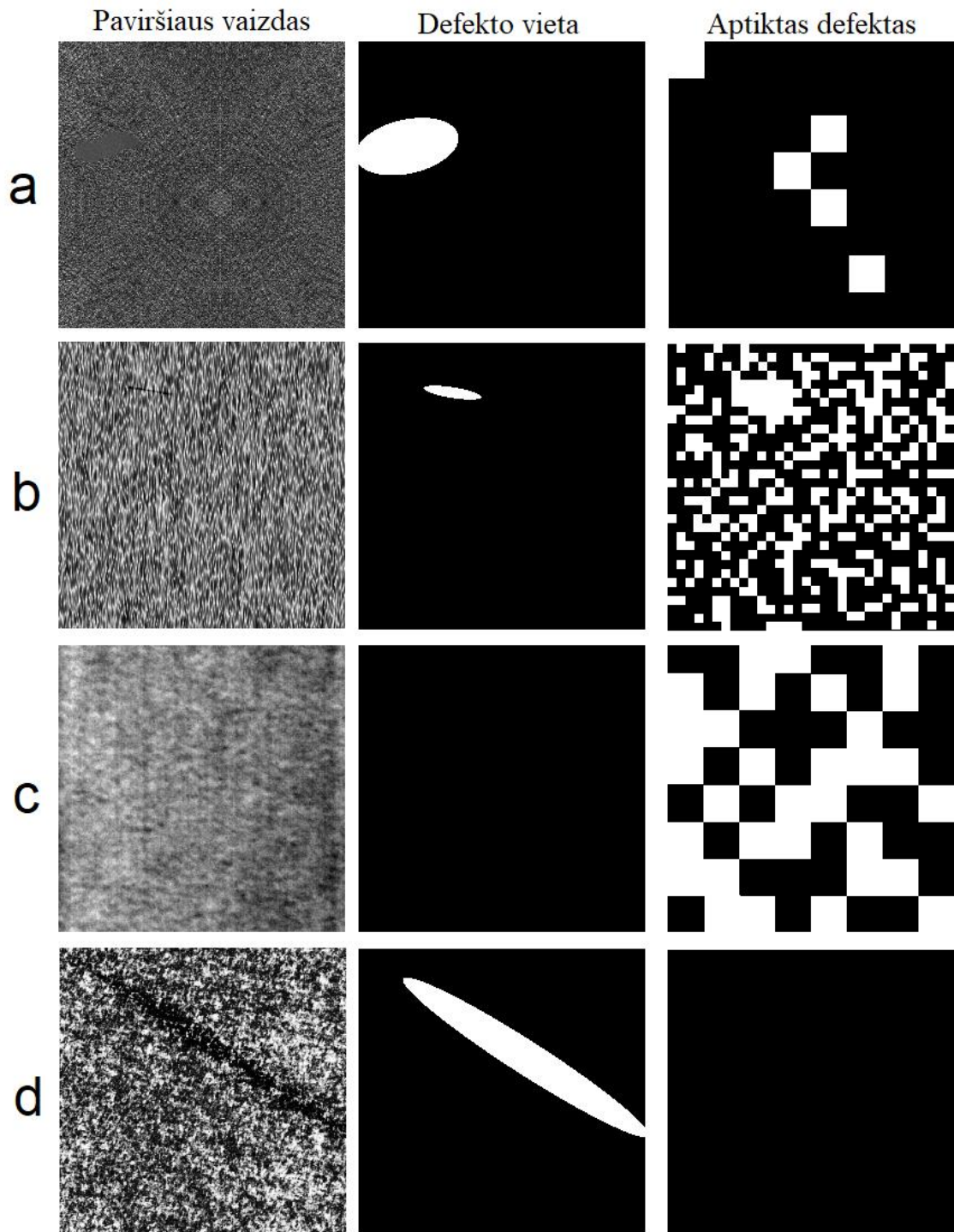
3. Rezultatai

Modeliuojant diskriminatoriaus modelį, buvo bandomos įvairios modelio konfigūracijos. Dalis išbandytų konfigūracijų pateikiama lentelėje žemiau. Pažymėtos tinkamiausios konfigūracijos, kurios buvo rastos bandymų metu.

5 lentelė. Dalis bandomų modelio konfigūracijų. Įtrauktos geriausias rezultatus parodžiusios konfigūracijos

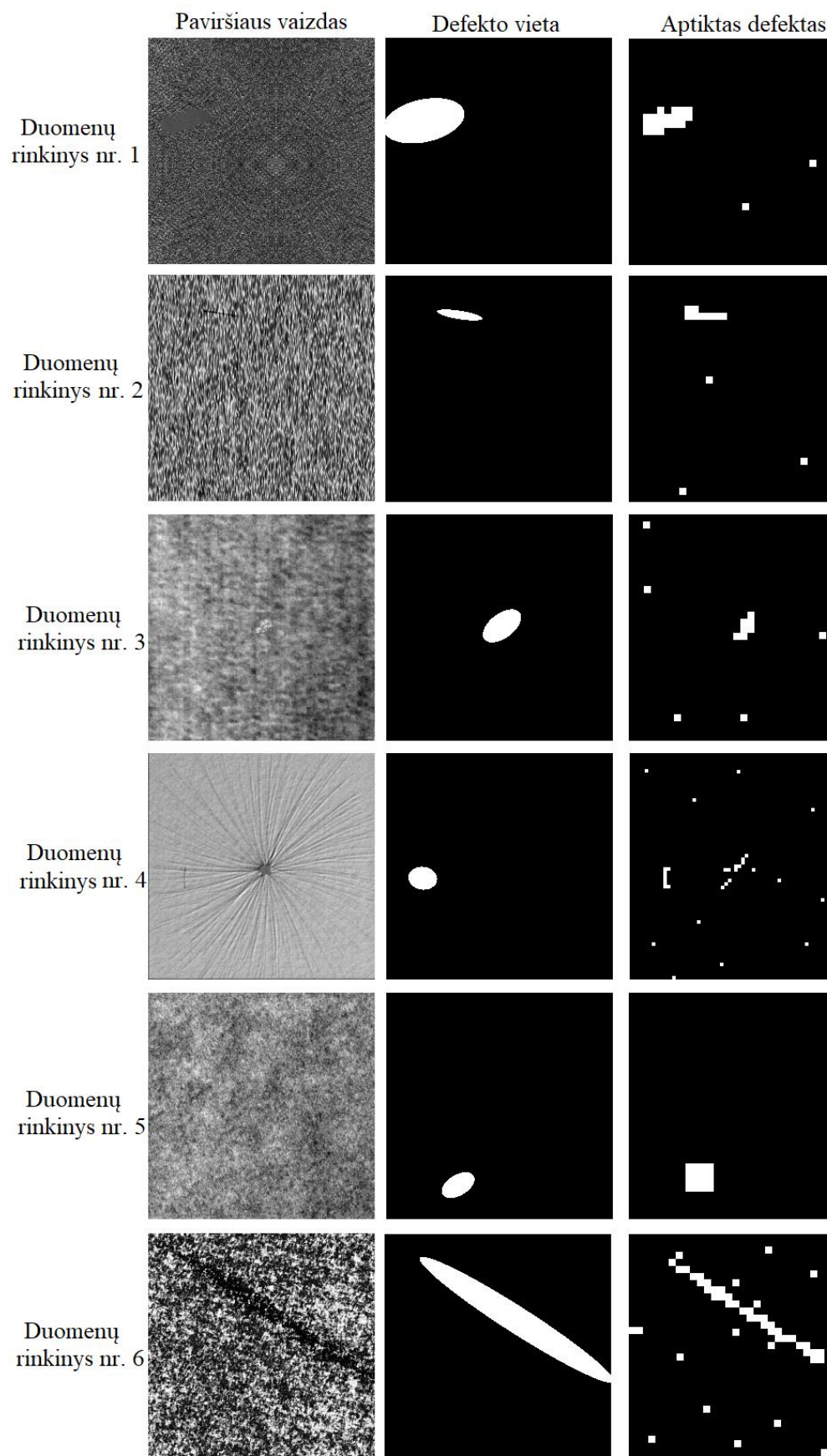
Eil. nr.	Konvoliucinio sluoksnio filtro dydis	Filtro branduolio dydis	Konv. sluoksnių skaičius	Tink. dydis	Normaliz. sluoksnių skaičius	Sutelkimo sluoksnių skaičius	Pilnai sujungtų sluoksnių skaičius	Modelis tinkamiausias duomenų rinkiniui
1	32x32	3x3	2	32x32	1	2	3	
2	32x32	3x3	2	16x16	2	3	4	
3	32x32	3x3	2	64x64	2	2	3	
4	32x32	3x3	2	8x8	1	4	4	
5	32x32	3x3	2	32x32	1	2	3	
6	32x32	3x3	2	16x16	2	3	2	
7	64x64	3x3	3	16x16	2	3	3	4
8	64x64	3x3	3	8x8	3	4	2	
9	64x64	3x3	3	16x16	2	2	4	
10	64x64	3x3	3	64x64	3	3	3	
11	64x64	3x3	3	64x64	2	2	2	
12	64x64	3x3	3	32x32	2	2	2	
13	64x64	3x3	3	16x16	1	3	4	
14	64x64	3x3	3	8x8	1	3	3	
15	64x64	3x3	3	64x64	2	2	2	5
16	64x64	3x3	3	32x32	2	2	2	1, 2, 3, 6
17	64x64	3x3	3	16x16	3	3	2	
18	64x64	3x3	3	8x8	3	4	2	
19	64x64	3x3	4	64x64	3	2	3	
20	64x64	3x3	4	32x32	3	3	4	
21	128x128	5x5	2	16x16	1	3	2	
22	128x128	5x5	2	8x8	1	4	3	
23	128x128	5x5	2	32x32	2	4	2	
24	128x128	5x5	2	32x32	2	2	3	
25	128x128	5x5	3	64x64	2	3	4	
26	128x128	5x5	3	32x32	2	2	2	
27	128x128	5x5	3	16x16	2	2	2	
28	128x128	5x5	4	8x8	3	5	3	
29	128x128	5x5	4	64x64	2	3	2	
30	128x128	5x5	4	32x32	2	3	3	
31	128x128	5x5	5	16x16	3	4	4	

Žemiau pateikiamos nepavykę diskriminatoriaus modelių išėjimo pavyzdžiai, kurie buvo gauti dėl netinkamos modelio konfigūracijos. 30 pav. a) pateikiamas pavyzdys, kai diskriminatoriaus modelis visiškai neaptiko defekto, ir klaidingai spėjo segmentus kaip turinčius defektą, nors toje vietoje defekto nebuvo. 30 pav. b) matyti, kad defektas buvo patiktas, tačiau tuo pačiu metu diskriminatoriaus modelis pateikia daugybę neteisingai aptiktų defekto segmentų. 30 pav. c) pateikiamas pavyzdys, kai paviršiaus vaizde defekto nebuvo, tačiau diskriminatoriaus modelis aptiko daugiau negu pusę defektuotų segmentų. 30 pav. d) pavyzdyje modelis visai neaptiko defekto, nors defektas užima santykinai didelį paviršiaus plotą paviršiaus vaizde, kuris turi defektą.



30 pav. Nepavykusių modelių išėjimo pavyzdžiai

31 pav. pateikiamas atrinktų diskriminatoriaus modelių išėjimo vaizdas. Modelių konfigūracijos buvo gautos bandymų metu.

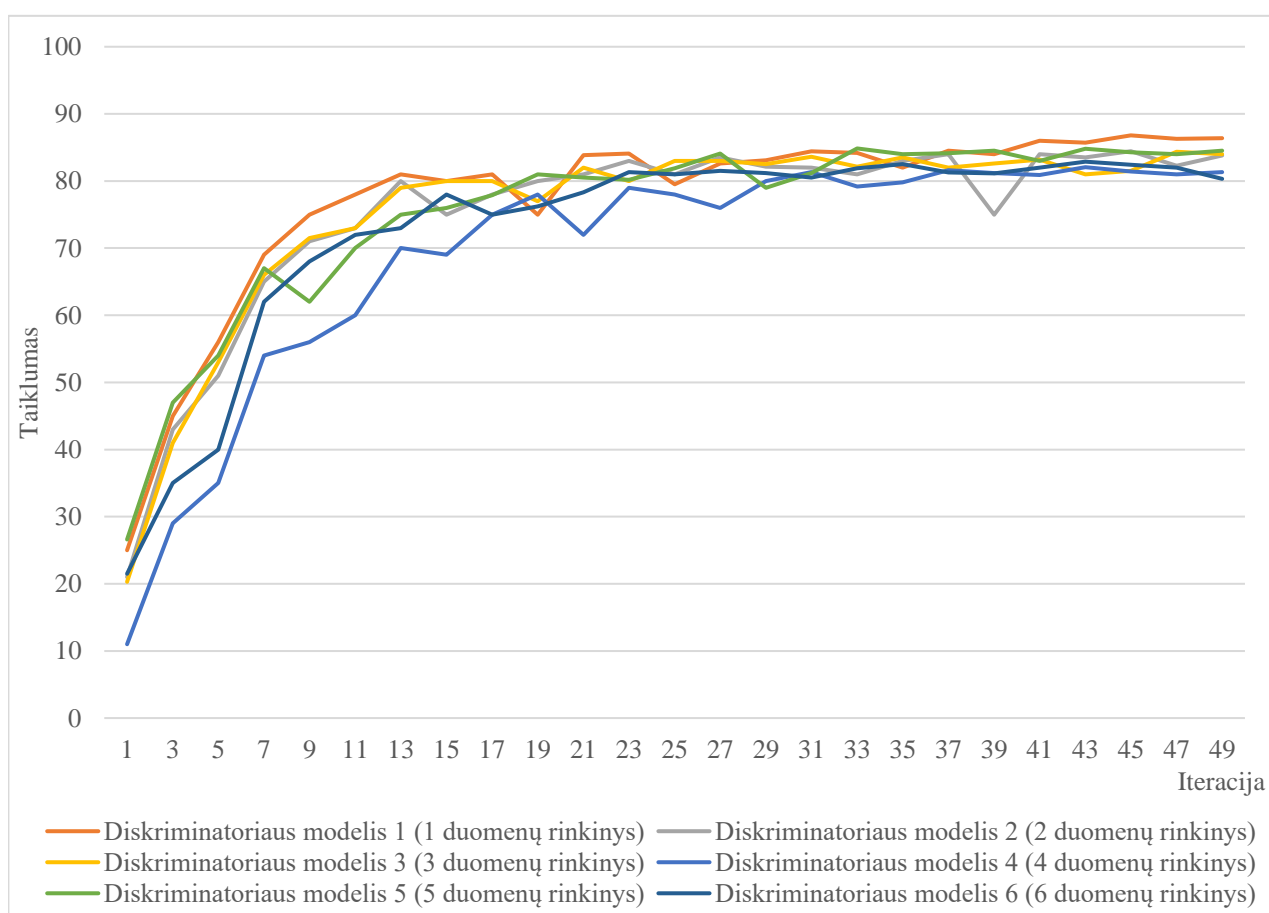


31 pav. Atrinktų diskriminatoriaus modelių išėjimai kiekvienam duomenų rinkiniui

6 lentelė. Bandymų metu nustatytos geriausios kombinacijos diskriminatoriaus modeliai kiekvienam duomenų rinkiniui

Diskriminatoriaus modelis	FN	FP	TN	TP	β	Prec	Acc	F1
1	0,7748	12,4248	85,2542	1,5462	0,6662	66,62%	86,80%	0,6662
2	0,4298	15,1256	83,4964	0,9482	0,6881	68,81%	84,44%	0,6881
3	0,4668	15,1690	82,6654	1,6988	0,7844	78,44%	84,36%	0,7844
4	0,4650	17,4737	81,4880	0,5733	0,5522	55,22%	82,06%	0,5522
5	1,1544	13,9982	82,4480	2,3994	0,6752	67,52%	84,85%	0,6752
6	1,3566	15,7681	79,0012	3,8741	0,7406	74,06%	82,88%	0,7406

Kaip matyti iš pateiktų modelių išėjimo rezultatų, geriausią taiklumą, 86,80%, turėjo diskriminatoriaus modelis, kuris apdorojo pirmo duomenų rinkinio duomenis. Šis diskriminatoriaus modelis taip pat buvo naudojamas ir 2, 3 bei 6 duomenų rinkiniui. Su šiuo diskriminatoriaus modeliu prasčiausiai buvo apdorojamas 6 duomenų rinkinys – 82,88%, o 2 ir 3 duomenų rinkiniai pasiekė atitinkamai 84,44 % ir 84,36 % taiklumą. Žemiau grafike pateikiamas tyrimo metu gautų diskriminatoriaus modelių mokymosi procesas.



32 pav. Tyrimo metu gautų diskriminatoriaus modelių mokymo procesas

Tyrimo metu buvo atliktos konfigūracijos diskriminatoriaus (D) modeliui, kurios pateiktos 7 lentelėje. Lentelėje surašytos galutinės diskriminatoriaus modelio konfigūracijos vertės.

7 lentelė. Tinklo modelis po modifikacijos

Modifikacija	1 D modelis	2 D modelis	3 D modelis	4 D modelis	5 D modelis	6 D modelis
Konvoliucinio sluoksnio filtro dydis	64x64	64x64	64x64	128x128	64x64	64x64
Filtro branduolio dydis	3x3	3x3	3x3	5x5	3x3	3x3
Konvoliucinių sluoksnių skaičius	3	4	3	4	4	4
Tinkelio dydis	32x32	32x32	32x32	16x16	32x32	32x32
Normalizavimo sluoksnių skaičius	3	3	3	3	3	3
Sutelkimo sluoksnių skaičius	3	2	3	3	2	3
Pilnai sujungtų sluoksnių skaičius	3	3	2	3	2	2
Minimali defekto užimamo ploto riba	85%	90%	85%	90%	85%	85%
Patikrai skirtas vaizdų skaičius	5%	5%	5%	5%	5%	5%

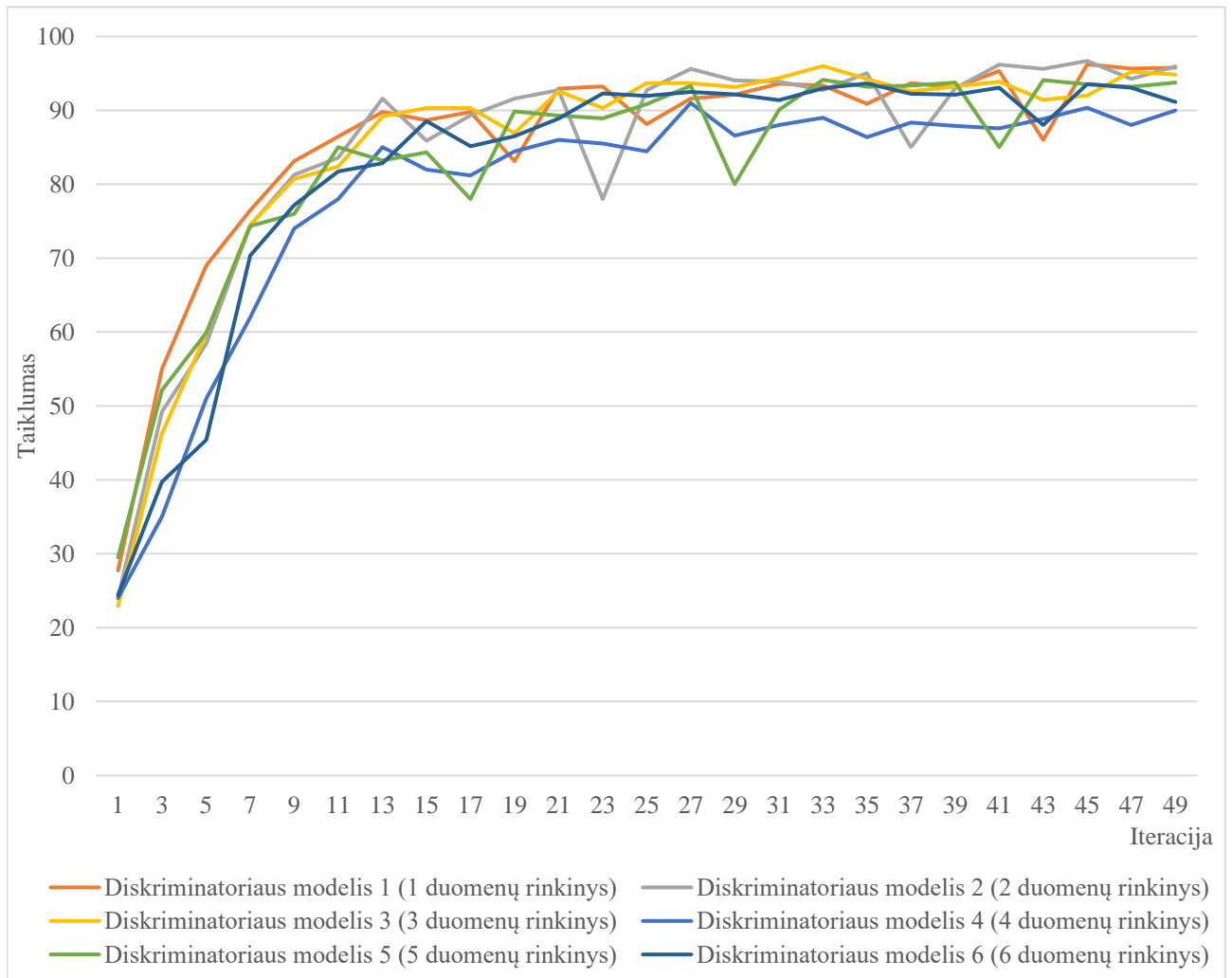
Kaip matyti iš pateiktų modelių išėjimo rezultatų, geriausią taiklumą, 98,96%, pavyko pasiekti su antru diskriminatoriaus modeliu, kuris apdorojo antro duomenų rinkinio duomenis (žr. 8 lentelę). Labai panašų taiklumą pavyko gauti ir su pirmuoju diskriminatoriaus modeliu. Prasčiausiai realizuotas 4 diskriminatoriaus modelis, kurio taiklumas siekė tik 90,31%. Kiekvieno modelio taiklumas nuo pirminės realizacijos pagerėjo atitinkamai: 10,86%, 14,51%, 12,89%, 8,25%, 10,94% ir 13,51%. Geriausią tikslumą, 96,78%, pavyko pasiekti su pirmu diskriminatoriaus modeliu, kuris apdorojo pirmo duomenų rinkinio duomenis (žr. 8 lentelę). Labai panašų tikslumą pavyko gauti ir su pirmuoju diskriminatoriaus modeliu. Prasčiausiai realizuotas 5 diskriminatoriaus modelis, kurio tikslumas siekė tik 84,42%. Kiekvieno modelio tikslumas nuo pirminės realizacijos pagerėjo atitinkamai: 30,16%, 25,54%, 9,34%, 30,94%, 19,91% ir 10,36%.

F_1 parametro reikšmė pateikta lentelėje, tačiau nevertinama, nes esant dideliame brokuotų ir gerų paviršių santykiui, šis parametras nenusako tikrojo diskriminatoriaus modelio veikimo rezultato.

8 lentelė. Po atliktų modifikacijų gautų diskriminatoriaus modelių rezultatai

Diskriminatoriaus modelis	FN	FP	TN	TP	β	Prec	Acc	F_1
1	0,0748	1,2624	96,4166	2,2462	0,9678	96,78%	98,66%	0,9678
2	0,0778	0,9656	97,6564	1,3002	0,9435	94,35%	98,96%	0,9435
3	0,2646	2,4804	95,3540	1,9010	0,8778	87,78%	97,26%	0,8778
4	0,1437	9,5454	89,4163	0,8946	0,8616	86,16%	90,31%	0,8616
5	0,4470	3,7608	92,6854	3,1068	0,8742	87,42%	95,79%	0,8742
6	0,8147	2,8028	91,9665	4,4160	0,8442	84,42%	96,38%	0,8442

Žemiau grafike pateikiamas tyrimo metu gautų modifikuotų diskriminatoriaus modelių mokymosi procesas.



33 pav. Tyrimo metu gautų diskriminatoriaus modelių mokymo procesas

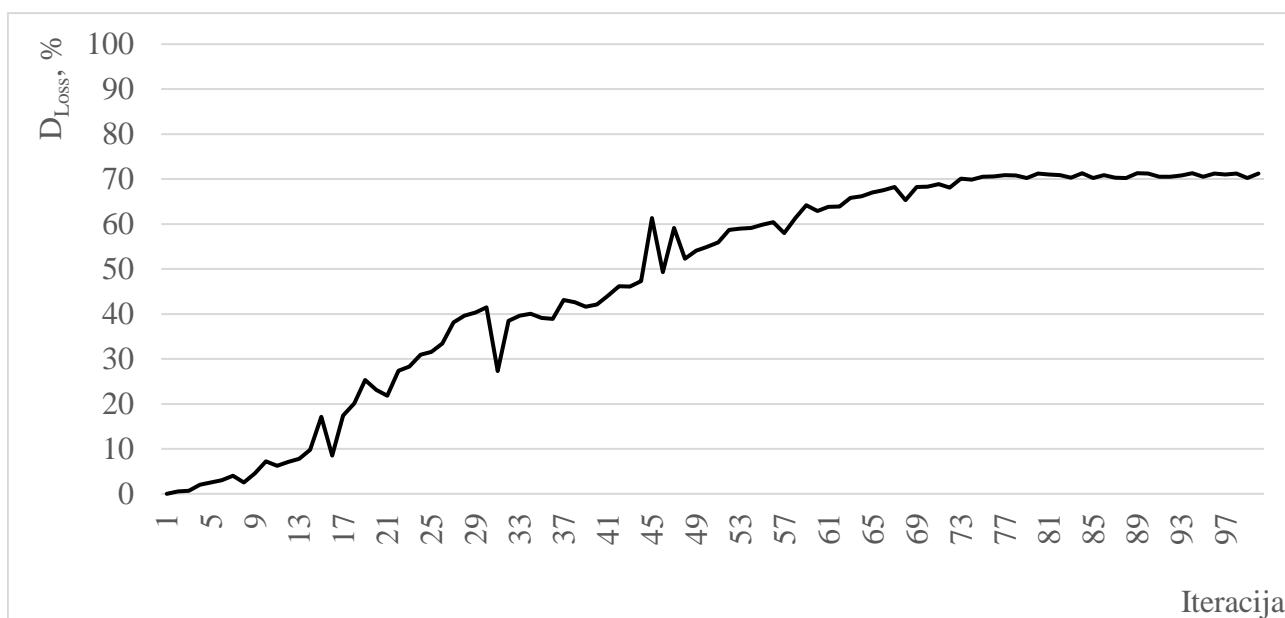
Turint žinomą diskriminatorių buvo tiriamas generatoriaus modelis. Modeliui realizuoti buvo naudojamos įvairios giliųjų neuroninių tinklų kombinacijos, kurių išėjimo vaizdas buvo perduodamas diskriminatoriui. Diskriminatorius įvertina sugeneruotą vaizdą jį įvardijant geras paviršius, ar turintis defektą (1 arba 0).

Generatorius buvo modeliuojamas tik antram duomenų rinkiniui, kadangi antro duomenų rinkinio diskriminatoriaus modelio gauti rezultatai buvo geriausi. Generatoriaus pirminis tikslas buvo generuoti kuo realesnius vaizdus, kurie neturėtų paviršiaus defekto. Geriausias generatoriaus modelis pateiktas lentelėje žemiau.

9 lentelė. Generatoriaus modelis skirtas antram duomenų rinkinio paviršiaus segmentams generuoti

Pilnai sujungtų sluoksnių skaičius	Konvoliucinio sluoksnio filtro dydis	Filtro branduolio dydis	Konvoliucinių sluoksnių skaičius	Normaliz. sluoksnių skaičius	Sutelkimo sluoksnių skaičius
3	32x32	3x3	4	3	2

Gauti sujungto generatoriaus ir diskriminatoriaus duomenys pateikiami žemiau. Generatoriaus modelis nusistovėjo ties 71,26 % riba. Tai yra, kai generatorius buvo apmokytas generuoti defekto neturinčius paviršiaus vaizdus, jo sugeneruotas nuotraukas diskriminatoriaus modelis 28,74 % nustatydavo kaip turinčias defektą. Dėl santykinai aukšto vaizdų skaičiaus, kurie yra atpažįstami neteisingai, generatoriaus modelis nebuvo naudojamas. Diskriminatoriaus modelis buvo apmokomas tik su vaizdais iš duomenų bazės.



34 pav. Generatoriaus modelio mokymo funkcija

Lyginant su kitų tyrėjų modeliu [20] (giliuoju regresiniu neuroniniu tinklu skirtu industrinių paviršiaus defekto aptikimui), kuris taip pat buvo taikomas DAGM2007 duomenų bazei analizuoti, pirmojo ir antrojo duomenų rinkinio diskriminatoriaus modeliai sugebėjo įgyti didesnę tikslumą, β ir F1 koeficientus (žr. 10 lentelę). Pirmojo ir antrojo duomenų rinkinio diskriminatoriaus modeliai atitinkamai apsiekė 96,78 % ir 94,35 % tikslumą, o kitų tyrėjų modelis pasiekė 89,1 % tikslumą (atitinkamai 7,68 % ir 5,25 % mažesnę tikslumą). Tačiau tyrėjų modelis pasiekė didesnę tikslumą lyginant su 3, 4, 5 ir 6 duomenų rinkinio diskriminatoriaus modeliu.

10 lentelė. Gautų diskriminatoriaus modelių parametrų palyginimas su kitų tyrėjų giliuoju regresiniu neuroniniu tinklu pagrįsto modeliu

Diskriminatoriaus modelis	β	Prec	F ₁
1	0,9678	96,78%	0,9678
2	0,9435	94,35%	0,9435
3	0,8778	87,78%	0,8778
4	0,8616	86,16%	0,8616
5	0,8742	87,42%	0,8742
6	0,8442	84,42%	0,8442
Giliojo regresinio neuroninio tinklo modelis	0,908	89,1%	0,897

Išvados

1. Apžvelgtos paviršiaus defekto nustatymo priemonės. Nustatyta, kad dabar dažniausiai pasirenkamos automatinės sistemos, kurios yra patikimesnės ir greitesnės;
2. Išanalizuotas ir realizuotas GAN tinklo diskriminatoriaus ir generatoriaus modelis. Diskriminatoriaus modelis realizuotas taikant įvairias konfigūracijas (keičiant konvoliucinio sluoksnio filtro dydį, filtro branduolio dydį, konvoliucinių sluoksnių skaičių, tinklelio dydį, normalizavimo sluoksnių skaičių, sutelkimo sluoksnių skaičių ir pilnai sujungtų sluoksnių skaičių);
3. Kiekvienam paviršiui (duomenų rinkiniui) nustatyta labiausiai tinkama diskriminatoriaus modelio konfigūracija. Po nustatymo, konfigūracija kiekvienam paviršiui buvo priderinama atskirai;
4. Po priderinimo individualiems paviršiams, buvo pasiektas 98,96 % taiklumas atpažįstant turintį defektą paviršių, naudojant antro paviršiaus duomenų rinkinį ir jam pritaikytą diskriminatoriaus modelį. Prasčiausi rezultatai pasiekti su ketvirto duomenų rinkinio diskriminatoriaus modeliu, kuris pasiekė 90,31 % taiklumą;
5. Palyginus gautus diskriminatoriaus modelius su kitų tyrėjų realizuotu gilioju regresiniu neuroniniu tinklo modeliu, pastebima, kad pirmas ir antras diskriminatoriaus modeliai pasiekia 7,68 % ir 5,25 % (atitinkamai didesnę 96,78 % ir 94,35 %) tikslumą nei tyrėjų modelis (89,1 %). Visi modeliai lyginami su rezultatais pasiektais juos taikant su DAGM2007 duomenų baze;

Literatūros sąrašas

1. https://www.researchgate.net/publication/324179731/figure/fig7/AS:668663598297088@1536433233187/Different-types-of-defects-in-cotton-fabric-from-a-to-f-the-defects-are-classified_W640.jpg
2. Goodfellow I. NIPS 2016 tutorial: Generative adversarial networks. arXiv preprint arXiv:1701.00160. 2016 Dec 3.
3. <https://dwfritz.com/wp-content/uploads/2019/06/DWF22.png>
4. Wang Z, She Q, Ward TE. Generative adversarial networks: A survey and taxonomy. arXiv preprint arXiv:1906.01529. 2019 Jun 4.
5. 3. Brownlee J. Generative Adversarial Networks with Python: Deep Learning Generative Models for Image Synthesis and Image Translation. Machine Learning Mastery; 2019 Jul 11.
6. Brownlee J. A gentle introduction to generative adversarial networks (GANs). 2019 June 17.
7. Murphy KP. Machine learning: a probabilistic perspective. MIT press; 2012 Sep 7.
8. Goodfellow I, Bengio Y, Courville A. Deep learning. MIT press; 2016 Nov 10.
9. Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434. 2015 Nov 19.
10. Li L, Vakanski A. Generative adversarial networks for generation and classification of physical rehabilitation movement episodes. International journal of machine learning and computing. 2018 Oct;8(5):428.
11. Li Q, Qu H, Liu Z, Zhou N, Sun W, Sigg S, Li J. AF-DCGAN: Amplitude Feature Deep Convolutional GAN for Fingerprint Construction in Indoor Localization Systems. IEEE Transactions on Emerging Topics in Computational Intelligence. 2019 Nov 5.
12. Fang W, Zhang F, Sheng VS, Ding Y. A method for improving CNN-based image recognition using DCGAN. CMC: Comput. Mater. Continua. 2018 Oct 1;57(1):167-78.
13. <https://hci.iwr.uni-heidelberg.de/content/weakly-supervised-learning-industrial-optical-inspection>
14. Yinka-Banjo C, Ugot OA. A review of generative adversarial networks and its application in cybersecurity. Artificial Intelligence Review. 2019 Jun 1:1-6.
15. Ruppin E, Yeshurun Y. An attractor neural network model of recall and recognition. In Advances in Neural Information Processing Systems 1991 (pp. 642-648).
16. Weimer D, Thamer H, Scholz-Reiter B. Learning defect classifiers for textured surfaces using neural networks and statistical feature representations. Procedia CIRP. 2013 Jan 1;7:347-52.
17. Mera C, Orozco-Alzate M, Branch J, Mery D. Automatic visual inspection: An approach with multi-instance learning. Computers in Industry. 2016 Dec 1;83:46-54.
18. Staar B, Lütjen M, Freitag M. Anomaly detection with convolutional neural networks for industrial surface inspection. Procedia CIRP. 2019 Jan 1;79(1):484-9.
19. Huang Y, Qiu C, Wang X, Wang S, Yuan K. A Compact Convolutional Neural Network for Surface Defect Inspection. Sensors. 2020 Jan;20(7):1974.
20. He Z, Liu Q. Deep Regression Neural Network for Industrial Surface Defect Detection. IEEE Access. 2020 Feb 19;8:35583-91.
21. Racki D, Tomaževic D, Skocaj D. Towards surface anomaly detection with deep learning.

22. Xiao Z, Leng Y, Geng L, Xi J. Defect detection and classification of galvanized stamping parts based on fully convolution neural network. In Ninth International Conference on Graphic and Image Processing (ICGIP 2017) 2018 Apr 10 (Vol. 10615, p. 106150K). International Society for Optics and Photonics.
23. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556. 2014 Sep 4.
24. Weimer D, Scholz-Reiter B, Shpitalni M. Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. CIRP Annals. 2016 Jan 1;65(1):417-20.
25. Markowska-Kaczmar U, Skiba M. Is Convolutional Network Competitive for Vision Method in the Furniture Dowel Quality Control?. In 2018 Innovations in Intelligent Systems and Applications (INISTA) 2018 Jul 3 (pp. 1-8). IEEE.
26. Zeiler MD, Fergus R. Visualizing and understanding convolutional networks. In European conference on computer vision 2014 Sep 6 (pp. 818-833). Springer, Cham.
27. Siebel NT, Sommer G. Learning defect classifiers for visual inspection images by neuro-evolution using weakly labelled training data. In 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence) 2008 Jun 1 (pp. 3925-3931). IEEE.
28. Siebel NT, Sommer G. Learning defect classifiers for visual inspection images by neuro-evolution using weakly labelled training data. In 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence) 2008 Jun 1 (pp. 3925-3931). IEEE.
29. Yu Z, Wu X, Gu X. Fully convolutional networks for surface defect inspection in industrial environment. In International Conference on Computer Vision Systems 2017 Jul 10 (pp. 417-426). Springer, Cham.