



Kauno technologijos universitetas
Matematikos ir gamtos mokslų fakultetas

**Šešėlinio muaro juostų su kintamu fono triukšmu
klasifikavimas taikant konvoliucinius neuroninius tinklus**

Baigiamasis magistro projektas

Andrius Ridikas
Projekto autorius

Doc. dr. Paulius Palevičius
Vadovas

Kaunas, 2020



Kauno technologijos universitetas
Matematikos ir gamtos mokslų fakultetas

**Šešėlinio muaro juostų su kintamu fono triukšmu
klasifikavimas taikant konvoliucinius neuroninius tinklus**

Baigiamasis magistro projektas
Taikomoji matematika (6211AX006)

Andrius Ridikas
Projekto autorius

Doc. dr. Paulius Palevičius
Vadovas

Doc. dr. Loreta Saunorienė
Recenzentė

Kaunas, 2020



Kauno technologijos universitetas
Matematikos ir gamtos mokslų fakultetas
Andrius Ridikas

**Šešėlinio muaro juostų su kintamu fono triukšmu
klasifikavimas taikant konvoliucinius neuroninius tinklus**

Akademinio sąžiningumo deklaracija

Patvirtinu, kad mano, Andriaus Ridiko, baigiamasis projektas tema „Šešėlinio muaro juostų su kintamu fono triukšmu klasifikavimas taikant konvoliucinius neuroninius tinklus“ 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)

Turinys

Įvadas	10
1. Literatūros apžvalga	11
1.1. Metrologija	11
1.2. Kontaktiniai metodai.....	12
1.3. Nekontaktiniai metodai	13
1.3.1. Muaro juostų metodai.....	15
1.3.2. Muaro metodų taikymai.....	17
1.3.3. Skaitmeninis vaizdo apdorojimas optinėje metrologijoje.....	17
1.4. Mašininis mokymasis.....	19
1.4.1. Mašininio mokymosi metodų klasifikacija.....	19
1.4.2. Mokymasis su mokytoju	21
1.4.3. Mašininio mokymo metodai vaizdų klasifikavimui.....	22
1.4.4. Dirbtiniai neuroniniai tinklai ir jų klasifikacija.....	23
1.5. Muaro metodų ir neuroninių tinklų pritaikymų apžvalga	24
2. Medžiagos ir tyrimų metodai.....	25
2.1. Šešėlinis muaras	25
2.2. Vaizdinių šešėlinio muaro duomenų generavimo strategija	26
2.3. Dirbtiniai neuroniniai tinklai	28
2.4. Aktyvacijos funkcijos	30
2.5. Tinklo parametrų inicijavimas.....	32
2.6. Klaidos skleidimo atgal algoritmas tinklo apmokymui	33
2.7. Mokymo optimizavimo algoritmai	34
2.8. Regularizavimas	36
2.9. Konvoliuciniai neuroniniai tinklai.....	36
2.10. Naudojamos konvoliucinių neuroninių tinklų architektūros.....	39
2.11. Programiniai įrankiai ir įranga.....	40
3. Tyrimų rezultatai ir jų aptarimas	41
3.1. Tyrimo duomenys.....	41
3.2. Konvoliucinio neuroninio tinklo realizacija.....	42
3.2.1. Optimizavimo algoritmai.....	45
3.2.2. Aktyvacijos funkcijos	46
3.2.3. Mokymosi greitis.....	46
3.2.4. Mokymo duomenų rinkinio dydis	47

3.2.5. Reguliarizavimas	48
3.2.6. Duomenų papildymas	50
3.2.7. Galutinė tinklo architektūra ir palyginimas	50
3.2.8. Tinklo jautrumas foninio vaizdo kitimui	52
Išvados.....	53
Literatūros sąrašas.....	54

Paveikslų sąrašas

1 pav. Optinės trimačių objektų metrologijos metodų klasifikacija [16].....	14
2 pav. Lygiagrečių juostų ir apskritimo gardelės interferencinės juostos	16
3 pav. Mašininio mokymo metodų klasifikacija	20
4 pav. Mokymo su mokytoju proceso schema	22
8 pav. Dirbtinių neuroninių tinklų architektūrų klasifikacija [51].....	24
10 pav. Šešėlinio muaro juostų sudarymo schema [55].....	25
11 pav. Paraboloido pavyzdys	27
12 pav. Paraboloido topografinio žemėlapiu pavyzdys	27
13 pav. Pradinės, deformuotos bei apjungtų gardelių pavyzdys	28
14 pav. Biologinio neurono supaprastinta schema	28
15 pav. Dirbtinio neurono schema.....	29
16 pav. Vienasluoksnių perceptrono schema	30
17 pav. Supaprastintas daugiasluoksnių tiesioginio sklaidimo neuroninis tinklas	30
18 pav. Sigmoidinė funkcija.....	31
19 pav. Hiperbolinio tangento funkcija.....	31
20 pav. ReLU funkcija.....	32
21 pav. Leaky ReLU funkcija	32
22 pav. Konvoliucijos operacijos supaprastinta schema [63]	37
23 pav. Surinkimo operacijos pavyzdys naudojant maksimalios reikšmės funkciją.....	38
24 pav. Konvoliucinio neuroninio tinklo architektūros pavyzdys [64].	39
25 pav. AlexNet tinklo architektūros schema.....	40
26 pav. Vaizdų generavimui naudojamos spausdintų elektroninių plokščių nuotraukos	41
27 pav. Pirmos (a), antros (b), trečios (c), ketvirtos (d) klasės vaizdų pavyzdžiai.....	42
28 pav. AlexNet modifikacijų su 64, 128 ir 256 neuronų pilno sujungimo sluoksniais validacijos tikslumas.....	43
29 pav. AlexNet modifikacijų su 64, 128, 256 neuronų pilno sujungimo sluoksniais validacijos klaidos funkcijos reikšmės	44
30 pav. AlexNet modifikacijų su 64, 128, 256 neuronų pilno jungimo sluoksniais mokymo klaidos funkcijos reikšmės	44
31 pav. Optimizavimo algoritmų palyginimo validacijos tikslumo reikšmės.....	45
32 pav. ReLU, leaky ReLU, PReLU funkcijų apmokomų tinklų validacijos tikslumo reikšmės	46
33 pav. Tinklų su skirtingais mokymo greičio parametrais validacijos tikslumo reikšmės.....	47
34 pav. Tinklų su skirtingais mokymo rinkinių dydžiais validacijos tikslumo reikšmės	48
35 pav. Tinklų su išmetimo metodais validacijos tikslumo reikšmės.....	49

36 pav. Tinklų su rinkinio normalizavimo metodu validacijos tikslumo reikšmės.....	49
37 pav. Tinklų, apmokyto su papildytais duomenų rinkiniais, validacijos tikslumo reikšmės.....	50
38 pav. AlexNet tinklo modifikacijos ir LeNet-5 tinklų rezultatų palyginimas.....	52

Lentelių sąrašas

1 lentelė. Optimizavimo algoritmų mokymų trukmė.....	45
2 lentelė. Naudojama AlexNet modifikacijos galutinė architektūra.....	51
3 lentelė. AlexNet tinklo modifikacijos rezultatų palyginimas su LeNet-5 architektūra	51

Ridikas Andrius. Šešėlinio muaro juostų su kintamu fonu triukšmu klasifikavimas taikant konvoliucinius neuroninius tinklus. Magistro baigiamasis projektas / vadovas doc. dr. Paulius Palevičius; Kauno technologijos universitetas, Matematikos ir gamtos mokslų fakultetas.

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

Reikšminiai žodžiai: šešėlinis muaras, konvoliuciniai neuroniniai tinklai.

Kaunas, 2020. 58 p.

Santrauka

Darbe nagrinėjami paviršių metrologijos metodai koncentruojantis į optinių metodų klasę. Apžvelgiamos šių metodų pritaikymų sritys tiriant paviršių deformacijas. Pasitelkiant optinius metodus sukurtiems vaizdams tirti, apžvelgiami mašininio mokymo metodai skirti vaizdų atpažinimui, klasifikavimui.

Metodologinėje dalyje detalizuojama šešėlinio muaro metodika paviršiaus topografijai tirti. Aprašoma strategija šešėlinio muaro interferencinių juostų vaizdų generavimui. Šių vaizdų klasifikavimo uždaviniui spręsti pasirenkama konvoliucinių neuroninių tinklų metodologija. Detalizuojami šių tinklų parametrai, algoritmai bei naudojamų AlexNet ir LeNet architektūrų pagrindai. Darbe tiriamų tinklų programinė realizacija atliekama Python programavimo kalba, naudojant PlaidML karkasą ir Keras biblioteką.

Tiriamajoje dalyje atliekamos pasirinktos konvoliucinio neuroninio tinklo architektūros modifikacijos, siekiant ją pritaikyti nagrinėjamam uždaviniui. Toliau seka optimalių tinklo parametrų nustatymo tyrimas. Galiausiai, atliekamas rezultatų palyginimas ir foninio vaizdo pasikeitimo įtakos tinklo rezultatams tyrimas.

Ridikas Andrius. Classification of shadow moiré interference fringes with variable background noise using convolutional neural networks. Master's Final Degree Project / supervisor Paulius Palevičius; Faculty of mathematics and natural sciences, Kaunas University of Technology.

Study field and area (study field group): Mathematics, Applied mathematics.

Keywords: shadow moiré, convolutional neural networks.

Kaunas, 2020. 58 pages.

Summary

This paper examines methods of surface metrology focusing on a class of optical methods. Application areas for these methods in the study of surface characteristics such as deformations are reviewed. Also, to study the images created by optical methods, machine learning methods for image recognition and classification are analyzed.

The methodological part of this paper details the shadow moiré methodology for surface topography research. A strategy for generating images of shadow moiré interference fringes is described. To process these images, methodology of convolutional neural networks is chosen to solve the problem of classification. The parameters, algorithms and basics of AlexNet and LeNet architectures used by these networks are detailed. The software implementation used in the work is performed in Python programming language, using PlaidML framework and Keras library.

In the exploratory part, the modification of the chosen convolutional neural network architecture is adapted to the task and the research of determining the optimal parameters is performed. Finally, a comparison and study of the effect of background image change on network results is performed.

Įvadas

Objektų paviršiaus charakteristikų tyrimų poreikis nuolatos auga tiek pramonės, tiek moksliniuose pritaikymuose. Tyrimai atliekami dirbant su naujomis medžiagomis, mikro elementais ar ypač sudėtinga elektronika, kurios tinkamam veikimui būtina užtikrinti aukštą kokybę. Šiam tikslui pasiekti dažnai žvelgiama į optinius metodus, kurie leistų įvertinti objektus nerizikuojant jų pažeisti. Vis dažniau pritaikomas ir tiriamas metodas yra šešėlinis muaras (angl. *shadow moiré*), kuris leidžia įvertinti įvairias objektų paviršių charakteristikas ir kurio realizacija reikalauja mažų kaštų [4]. Viena iš elektronikos sričių, kurioje aktualios tokių paviršiaus charakteristikų kaip deformacija tyrimas, yra spausdintų elektroninių plokščių gamyba. Jos yra sudarytos iš keleto sluoksnių bei gamybos procese yra veikiamos aukštos temperatūros. Tai nulemia įvairių paviršių defektų atsiradimą, kurie gali būti sunkiai pastebimi akimi, tačiau turi didelę įtaką patikimumui. Šiame darbe nagrinėjama būtent spausdintų elektronikos plokščių deformacijos problema.

Naudojant optinius metodus dažniausiai susiduriama su vaizdo apdorojimo, atpažinimo, klasifikavimo problemomis. Šių uždavinių sprendimui pritaikomi šiuo metu ypač populiarūs mašininio mokymo metodai (angl. *machine learning*) [39]. Vienas iš plačiausiai tiriamų ir taikomų su vaizdo apdorojimu susijusių metodų yra konvoliuciniai neuroniniai tinklai (angl. *convolutional neural networks*). Būtent šis metodas darbe tiriamas nagrinėjant optinių metodų sukurtų efektų vaizdus.

Šio darbo tikslas yra pasiūlyti metodiką šešėlinio muaro su kintamu fono vaizdu klasifikavimui bei ją iširti siekiant kuo didesnio efektyvumo, tikslumo ir laiko atžvilgiu. Šio tikslo įgyvenimui iškeliami uždaviniai:

1. Atlikti vaizdų klasifikavimo ir paviršių tyrimų optinių metodų analitinę apžvalgą.
2. Realizuoti šešėlinio muaro vaizdų generavimo metodą tyrimo duomenų imties generavimui.
3. Pasiūlyti bei realizuoti konvoliucinio neuroninio tinklo architektūrą sprendžiamam uždaviniui.
4. Atlikti pasirinkto tinklo parametrų ir metodų palyginamąją analizę.

1. Literatūros apžvalga

Darbe apžvelgiami optiniai interferometriniai bei mašininio mokymosi metodai. Plačiau nagrinėjant tokius interferometrinius muaro metodus, kaip šešėlinį muarą. Taip pat, apžvelgiami neuroninių tinklų bei jų apmokymo metodai. Apžvelgiamos šių metodų tyrimų ir taikymų sritys.

1.1. Metrologija

Metrologija yra „matavimo mokslas, apimantis tiek eksperimentinius, tiek teorinius matavimus esant bet kokiam neapibrėžtumo lygiui bet kurioje mokslo ir technologijos srityje, kaip apibrėžė tarptautinis svorio ir matų biuras (pranc. *Bureau International des Poids et Mesures*). Metrologiją galima suskirstyti į tris kategorijas:

- mokslinė metrologija;
- taikomoji metrologija;
- teisinė metrologija.

Teisinė metrologija yra eilės pabaigoje ir ji susijusi su nusistovėjusių matavimų, matavimo priemonių norminiais reikalavimais, siekiant apsaugoti vartotojus bei užtikrinti sąžiningą prekybą. Taikomoji metrologija, tai matavimo mokslas plėtojamas gamybos ir kitų procesų srityje, užtikrinant matavimo priemonių tinkamumą, jų kalibravimą ir kokybės kontrolę. Mokslinė metrologija yra visų kitų dalių pagrindas susijęs su naujų matavimo metodų kūrimu, matavimo standartų įgyvendinimu ir šių standartų perdavimu vartotojams [31].

Šiame darbe aktuali paviršiaus metrologija, kuri yra viena iš mokslinės metrologijos sričių. Jos svarba stipriai išaugo per paskutinius dešimtmečius. Dvidešimtame amžiuje paviršiaus charakteristikos pirmiausia buvo matuojamos metodu naudojančiu kontaktinės adatos sekimą per visą paviršių stebint vertikalų adatos judėjimą, kol ji vedama per paviršiaus nelygumus. Dažniausiai buvo matuojama tik viena linija arba paviršiaus profilis ir tai leido gauti pakankamai informacijos tam tikrų komponentų gamybai kontroliuoti, tačiau toks metodas buvo lėtas ir sudėtingas platesniam pritaikymui [34].

Dvidešimto amžiaus pradžioje ir viduryje buvo pradėti kurti optiniai instrumentai, skirti matuoti paviršių tekstūrą ir formą. Tarp jų buvo įprasti Michelson‘o ir Twyman-Green‘o interferometrai [65], Schmalz‘o šviesos pjaustymo mikroskopai, Linnik‘o mikrointerferometrai, Tolansk‘o kelių spindulių interferometrai, ir vienodos chromatinės eilės interferometrai. Šie interferometrai sugeneruodavo labai tikslus paviršiaus charakteristikų raštų vaizdus, kurie tiksliai nusakydavo paviršių viršūnes ir įdubimus su didele vertikalia raiška. Tačiau, tik sukūrus pakankamą skaičiuojamąją galią turinčius aparatus šis procesas galėjo būti automatizuotas ir naudingumu aplenkti mechaninius metodus [34]. Pirmieji tokie aparatai buvo fazių kitimo interferometrinis (angl. *phase shifting interferometry*)

mikroskopas devintojo dešimtmečio pradžioje, kuris visų pirma buvo naudingas matuojant lygius optinius paviršius [3]. Dešimtojo dešimtmečio pradžioje sekė koherencijos nuskaitymo interferometrinis (angl. *coherence scanning interferometry*) mikroskopas [19].

Bendrai, reikalavimai energijos, informacijos pernešimo ir saugojimo srityse vedė technologijų vystymą. Panašu, kad tai tęsis ir ateityje. Daugelis anksčiau nurodytų reikalavimų tiesiogiai priklauso nuo tam tikrų komponentų dydžio, jų mažumo. Mažėjant įvairiems komponentams buvo vis sudėtingiau įvertinti jų paviršių charakteristikas, todėl vis daugiau dėmesio buvo skiriama būtent paviršiaus metrologijos metodams vystyti [58].

Paviršiaus metrologijoje tiriami tokie paviršiaus požymiai kaip: nelygumai, struktūra, banguotumai ir kt. Šie tyrimai pritaikomi nuo didelių industrinių objektų paviršių tyrimo iki mikroelektromechaninių sistemų (MEMS) (angl. *microelectromechanical systems*) [9].

Pastebima, kad šiame skyriuje aptartus paviršiaus metrologijos metodus galima suskirstyti į dvi kategorijas: kontaktinius ir nekontaktinius [28]. Pirmuoju atveju norint iširti paviršiaus charakteristikas būtinas tiesioginis instrumentų kontaktas su paviršiumi. Tuo tarpu, antruoju atveju galima atlikti tą patį tyrimą paviršiaus nepalietus. Tolimesniuose skyriuose plačiau apžvelgsiu šias kategorijas ir jose naudojamus metodus [13].

1.2. Kontaktiniai metodai

Visi kontaktinių metodų principai yra pagrįsti mechaniniu kontaktu su tiriamu objektu. Šie metodai naudojami tirti objektų paviršių formas, įtampas, deformacijas ir poslinkius.

Paviršiaus formoms ir deformacijoms tirti dažniausiai naudojamas kontaktinis metodas yra paremtas matavimo adatos judėjimu per tiriamąjį paviršių, kuomet adatos vertikalus judėjimas yra paverčiamas į tam tikrą signalą. Tai yra vienas seniausių iki šiol tebenaudojamų metodų. Tačiau, šis metodas turi trūkumų. Matavimo adata gali pažeisti paviršių priklausomai nuo nuskaitymo paviršiaus kietumo palyginti su įprastine adatos kontakto su juo jėga ir adatos lietimosi taško dydžio [55]. Nors ir šio metodo pagrindas per laiką išliko panašus, tačiau, buvo sukurta ne viena metodo variacija padedanti tiksliau surinkti informacija apie paviršių ar kontroliuoti matavimo adatą. Trys svarbiausios šio metodo vystymo sritys gali būti laikomos šiomis: koordinačių matavimo aparatas (angl. *coordinate measuring machine*), apvalumo matavimo aparatas, sukurtas T. H. Talyrond'o ir paviršiaus šiurkštumo matavimo aparatai [3].

Ganėtinai dažnai kontaktiniai metodai taikomi ir objekto įtampai, krūviui matuoti. Šie metodai dažniausiai pritaikomi pakankamai atspariems objektams, kadangi atliekant įtampos matavimus jie yra veikiami kur kas didesnės jėgos nei paviršiaus charakteristikų matavimo atveju [15]. Vieni pagrindinių prietaisų, naudojamų pramonėje, yra tempimo jėgos matuokliai. Jie idealiai tinka matuoti kietų medžiagų paviršiaus įtampą. Du labiausiai paplitę tokio tipo metodai arba prietaisai yra

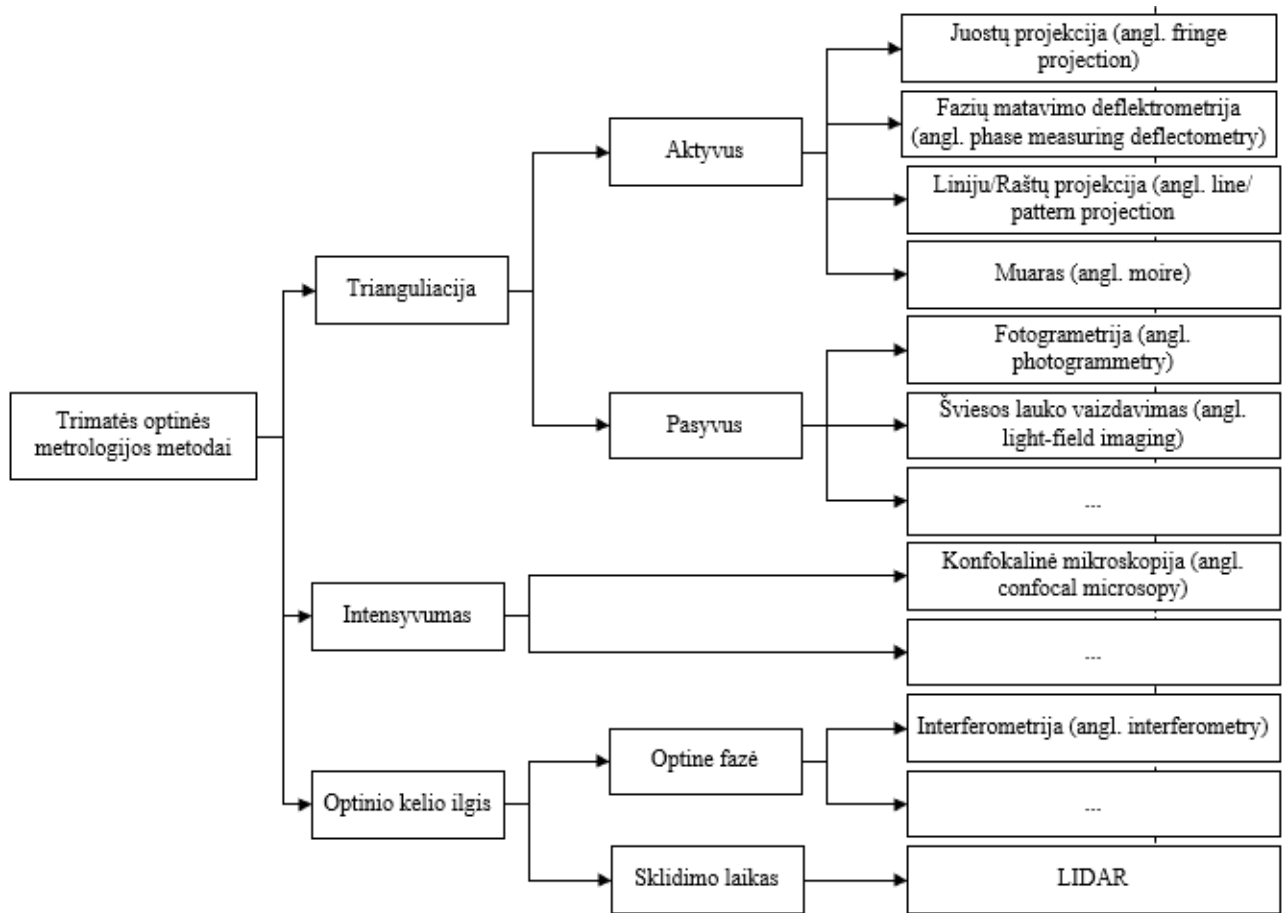
graviruotos folijos jutiklis (angl. *etched-foil sensor*) ir puslaidininkių matuoklis (angl. *semiconductor gage*). Šie prietaisai dažniausiai naudoja veikimo principą vadinama inžinerine įtampa (angl. *engineering strain*), kuri apibrėžiama objekto ilgio pasikeitimus nuo pradinio ilgio. Minėti jutikliai matuoja ne ilgio pasikeitimą, o varžos pokyčius medžiagoje [51].

Kita svarbi metodų kategorija paviršiaus ar medžiagos įtampos matavime yra ekstensometrai (angl. *extensometers*). Jie suteikia patogiausią būdą išmatuoti vidutinę įtampą objekte [51].

1.3. Nekontaktiniai metodai

Nekontaktinių metrologinių metodų pagrindu dažniausiai laikomi optiniai metodai. Literatūroje dažnai vadinama optine metrologija (angl. *optical metrology*). Žvelgiant iš istorinės perspektyvos, optinė metrologija prasidėjo tuo metu, kai pirmą kartą buvo suprasta, kad nepasiekiamo objekto plotį ir aukštį galima išmatuoti pasitelkus keletą matymo kampų. Šis beveik tiesinių šviesos ir šešėlio geometrinių savybių principas šiandien yra atvaizduojamas daugybėje instrumentų. Šiuolaikinės optinės metrologijos pradžia galima laikyti interferometrijos sukūrimu. Tai suteikė prieigą prie šviesos bangos ilgio kaip tikslaus žymeklio. Be ploto matavimų, interferometrija, taip pat, taikoma formos ir šiurkštumo matavimams atlikti [10].

Nepaisant to, kad didelė dalis nekontaktinių metodų priklauso optinei metrologijai, literatūroje nekontaktiniai metodai trimačių objektų metrologijoje skirstomi į optinius ir neoptinius metodus [28]. Tuo tarpu, optiniai metodai gali dar būti skirstomi į trianguliacijos, intensyvumo ir optinio kelio ilgio metodus, kurie pavaizduoti 1 pav. [48].



1 pav. Optinės trimačių objektų metrologijos metodų klasifikacija [48]

Tarp neoptinių metodų dažniausiai minimas yra rentgeno spindulių pritaikymas. Rentgeno spindulių difrakcija yra metodika galinti išmatuoti elastinę deformaciją visose medžiagos kristalinėse fazėse, kurią galima paversti įtempiumi naudojant tinkamas elastines konstantas. Naudojant laboratorines priemones tipinis rentgeno spindulių įsiskverbimo gylis yra mikronų lygyje, o gilesnį gylį galima įvertinti naudojant destruktivius sluoksnių pašalinimo metodus arba didesnės energijos rentgeno spindulius, ypač iš sinchrotrono šaltinių. Difrakcija taip pat suteikia papildomos informacijos apie kristalografinę tekstūrą ir plastinę deformaciją. Visgi, naudojantis šiuo metodu ir siekiant didelio tikslumo svarbu atsižvelgti į dažnai pasitaikančias paklaidas [51].

Apžvelgiant optinius metodus, dažnai sutinkamas skaitmeninės vaizdų koreliacijos metodas (DIC) (angl. *digital image correlation*). Jis taikomas tiek dvimačiams deformacijos, tiek trimačiams formos ir deformacijų mikro, makro objektų matavimams. Šis metodas pagrįstas vaizdo sekimu. Stebint pikselių blokus, sistema gali išmatuoti paviršiaus poslinkius ir sudaryti viso lauko dvimačius ar trimačius deformacijų vektorius, deformacijų žemėlapius. Norint, kad DIC veiktų efektyviai, pikselių blokai turi būti atsitiktiniai ir unikalūs, turintys įvairaus lygio kontrastą bei intensyvumą. Tam nereikia specialaus apšvietimo. Daugeliu atvejų natūralus konstrukcijos ar komponento paviršiaus vaizdas yra pakankamas, kad DIC galėtų veikti be jokio specialaus paviršiaus paruošimo [37].

Svarbu paminėti holografiją kaip vieną šiuo metu vis daugiau tyrinėjamą metodą optinėje metrologijoje. Holograma yra fizinis interferencinių raštų modelio vaizdas, gaunamas naudojant difrakciją atgaminti trimatį šviesos lauką ir išlaikantis gylį, paralaksą bei kitas originalaus lauko savybes. Tuo tarpu, holografija yra metodas išgauti šiuos vaizdus panaudojant šviesos lauką (kuris paprastai yra šviesos šaltinio skleidžiamos šviesos atsispindėjimo nuo objekto rezultatas), kuris turi būti registruojamas ir vėliau rekonstruojamas [53]. Šį metodą galima pritaikyti nuo tokių dydžių objektų kaip pastatai iki kelių mikronų dydžio objektų, tačiau augiausiai dėmesio šis metodas susilaukė būtent ypač mažų objektų poslinkių ir deformacijų matavimams [51].

Taip pat, dėl paprastumo pramonėje bei moksliniuose taikymuose paplitę yra muaro metodai. Šių metodų pagrindas – šviesai pralaidžios gardelės panaudojimas apšviečiant bei fiksuojant tam tikru kampu vaizdą bei stebint susidariančias interferencines šešėlių ir gardelės juostas. Muaro metodams pritaikyti naudojama palygintinai paprasta įranga ir išgaunamas tikslumas gali būti pritaikytas net mikro objektams tirti. Panaudojimai dažniausiai būna objektų paviršių įtampos, deformacijų, formos nusitaikymo srityse. Šis metodas toliau bus apžvelgtas plačiau.

1.3.1. Muaro juostų metodai

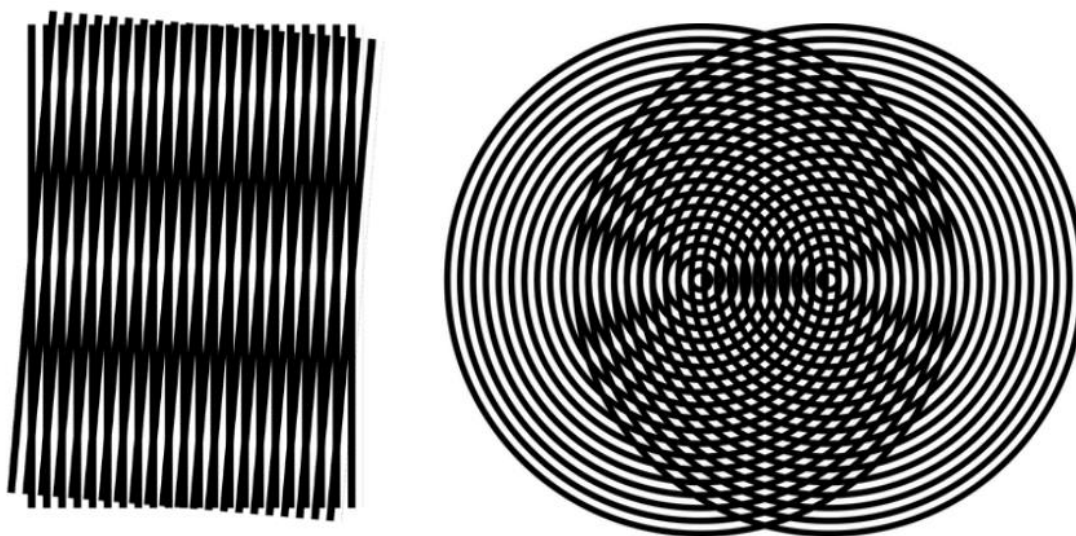
Terminas „muaras“ yra kilęs iš prancūzų kalbos, kur taip vadinama austinė šilko juostelė. Šis šilkas, dar vadinamas Muaro šilku, yra sudarytas iš dviejų sujungtų audinio sluoksnių [51]. Audiniui lankstantis sluoksniai persidengia vienas su kitu ir taip gaunamas interferencinių raštų vaizdas [6]. Natūralias muaro juostas kasdieniame gyvenime galima pamatyti žiūrint į judančią nailoninę užuolaidą arba žiūrint pro lygiagrečių laidų tinklelio tvorą, taip pat, uždėjus vieną ant kito persviečiamus panašaus rašto piešinius ar audinius. Kaip ir iš angoros vilnos audinių gauti raštai, geometriniai raštai (tokie kaip apskritimai, lygiagrečios ir spindulinės linijos) taip pat suformuoja interferencinės juostas, kai sudedami keli šių raštų sluoksniai.

Eksperimentinėje mechanikoje, muaro metodai yra apibrėžiamas kaip muaro raštų panaudojimas paviršiaus poslinkiui matuoti. Šio metodo vystymo pradžia galima laikyti J. Guild'o 1956 m. aprašytas metodikas [20]. Tačiau ilgą laiką šie metodai turėjo mažą praktinę vertę, kadangi šios metodikos be modernių prietaisų tikslumas buvo per mažas platesniam pritaikymui [56].

Šio metodo pagrindas gali būti apibūdinamas taip: paviršiaus deformacijos būseną charakterizuota modelio gardele. Ši gardele veikia kaip pamatinis paviršiaus pokyčių elementas iš pradinės ar atskaitos būsenos į galutinę ar deformuotą būseną. Šie pokyčiai apibūdina tiriamojo paviršiaus atitinkamas deformacijas. Norint nustatyti modelio gardelės geometrijos pokyčius, įtraukiama antroji juostų gardele arba pagrindinė gardele. Dviejų grotelių superpozicija sukuria interferencines juostas arba muaro raštus. Norint stebėti paviršiaus poslinkius, įtampą reikia žinoti santykį tarp gautų muaro raštų, stebėjimo sistemos ir paviršiaus, kuris yra tiriamas. Šis santykis priklauso nuo naudojamos

metodikos. Gardelė ir raštai gali būti graviruojami stebimame paviršiuje, tačiau dažniausiai naudojamas optinis metodas – projekcija paviršiuje. Tokiu atveju galima išgauti efektą, kai besikeičiant paviršiuje atitinkamai keičiasi ir projekcija [50].

2 pav. pavaizduotas geometrinio muaro efektas, kai sudėjus du linijų masyvus arba gardeles susidaro muaro interferencinės juostos.



2 pav. Lygiagrečių juostų ir apskritimo gardelės interferencinės juostos

Muaro metodai gali būti grupuojami į dvi pagrindines grupes:

- geometrinis muaras;
- muaro interferometrija.

Geometrinių muarą galima skirstyti į dar kelias kategorijas, kaip plokštuminis muaras (angl. *in-plane moiré*), kurio atveju stebimi poslinkiai dvimatėje plokštumoje arba erdvinis muaras (angl. *out-of-plane moiré*), kuriuo stebimi trimačiai poslinkiai, deformacijos, įtampas. Pastarosios kategorijos dažniausiai naudojamas šešėlinio muaro metodas [50].

Tiriant erdvinis objektus dažniausiai naudojami metodai yra:

- projekcinis muaras;
- šešėlinis muaras;
- atspindžio muaras.

Atspindžio muaras yra konstrukcinių komponentų nuolydžio ar sukimosi, arba pokyčių nuolydžio matavimo metodas. R. Ritter aprašė šio metodo panaudojimą, matuojant plokštumos lenkimo metu atsirandančias įtampas [44].

Projekcinio ir šešėlinio muaro taikymai dažniausiai susidaro iš deformacijų nustatymo, erdvinio kūno formos dvimačio žemėlapiu sudarymo. Šie metodai dažniausiai aptariami kartu dėl panašaus

teorinio pagrindo, tačiau pritaikant dažniausiai yra naudojamas būtent šešėlinis muaras dėl savo paprastumo. Taip pat, teoriškai naudojantis juo galima pasiekti geresnį tikslumą [42].

Muaro interferometrija yra labai didelio jautrumo optinis metodas, dažniausiai naudojamas matuoti poslinkius plokštumoje. Aukšto dažnio difrakcijos gardelės pakartojamos ant tiriamo paviršiaus ir interferuoja su gardele, sudaryta naudojant keletą kolimizuotų lazerių spindulių.

Pagrindinis skirtumas tarp geometrinio ir interferometrinio muaro yra gardelės dažnis. Interferometriniame muare, siekiant didelio tikslumo, arba atliekant bandymus su mažais objektais, reikia naudoti, lyginant su geometrinio muaru, didelio dažnumo gardele [4].

1.3.2. Muaro metodų taikymai

Pritaikant muaro interferencines juostas, dažniausiai galima išvystyti du pagrindinius tikslus:

- turimų raštų analizė, siekiant ištirti objekto įtempimus, postūmius ar deformacijas interferencinių juostų centruose;
- iš kelių turimų vaizdų sintezės proceso metu išgauti norimas interferencines juostas sudedant turimus vaizdus.

Gaunami ar turimi muaro raštai gali būti pritaikomi navigacijoje, interferometrijoje [35], topografijoje [59], steganografijoje [30], mikroskopijoje [49], klastotės prevencijoje, siekiant atpažinti, kuomet bandoma atpažinti kameros vaizdo klastojimą [2], kriptografijoje, norint paslėpti, užkoduoti informaciją vaizduose [40].

L. Konyang'as ir kiti ištyrė dėl tiesinio Fresnelio objektyvios bei LCD ekranų superpozicijos susidarancias interferencines juostas. Remiantis šio tyrimo rezultatais, galima efektyviai pašalinti muaro interferencines juostas iš įvairių ekranų, kuriuose naudojamos optinės plėvelės, turinčios kvaziperiodines struktūras [36].

P. G. Ifju'as ir B. Han'as, aprašė atvejus, kuriais naudojamos muaro medžiagų charakteristikos, mikromechanikos, mikroelektronikos prietaisų, perteklinio įtempio, lūžių mechanikos, kompozicinių medžiagų ir biomechanikos srityse [27]. Juose apžvelgiama ir pristatyta metodika, kuri leidžia suformuoti gardeles ir atvaizduoti jų interferenciją ant objektų, turinčių sudėtingas geometrines formas [21].

H. Bongtae pristatė metodiką, skirtą mikromechanikos tyrinėjimams, kuri leido padidinti muaro metodų tikslumą ir buvo daugelio interferometrinių muaro tyrimų pagrindas [22].

1.3.3. Skaitmeninis vaizdo apdorojimas optinėje metrologijoje

Nemažai tokių optinės metrologijos metodų kaip holografinė interferometrija, taškinė metrologija (angl. *speckle metrology*), juostų projekcija (angl. *fringe projection*) ir muaro, turi bendrą savybę – rezultate sudaryti juostų raštus. Šiuose intensyvumo svyravimuose tokie dominantys rodikliai kaip

koordinatės, poslinkiai, lūžio rodiklis ir kiti, yra užkoduojami juostų periodo skalėje. Taigi, viena iš pagrindinių užduočių, kurias reikia išspręsti apdorojant šiuos rezultatus galima apibrėžti kaip juostų modelio pavertimą ištisine fazių schema, atsižvelgiant į kvazi-sinusoidinį intensyvumo pasiskirstymo pobūdį [38].

Nuotraukos apdorojimo pradžioje galima taikyti vieną iš paprastų nuotraukos pradinio apdorojimo metodų, kurie leistų panaikinti nepageidautinus trikdžius (netiesiškumus, triukšmą, geometrinius iškraipymus ir kt.). Šie metodai gali būti suskaidyti į tris pagrindines kategorijas atsižvelgiant į pikselių kaimynystės dydį, kuris naudojamas apskaičiuojant jo naują savybę.

- Pikselių transformacija. Remiantis apibrėžta transformacijos taisykle, keičiamas esamo pikselio ryškumas, spalva arba padėtis išvestiniame vaizde: šešėliavimo pataisa, binarizacija, histogramos modifikacija ir tokios geometrinės transformacijos kaip pikselių koordinatinių transformacijos.
- Vietinės transformacijos. Šiuo atveju, nedidelė grupė pikselių įvesties vaizde naudojami sugeneruoti naują vieno iš grupės pikselių pilkumo reikšmę arba spalvą: tiesinės išlyginimo operacijos, tokios kaip vidurkinimas ir žemųjų dažnių filtravimas; netiesinės išlyginamosios operacijos, tokios kaip mediana ir kitos eiliškumo operacijos; briaunų aptikimą per tokius gradiento operatorius kaip Robertsono, Laplaso ir Sobelo.
- Globalios transformacijos. Globalus operatorius paveikia visą vaizdą. Priešingai nei vietinė transformacija, tokios globalios transformacijos kaip Furjė, kosinuso, Hilberto ir Hough'o transformacijos paverčia vaizdą į visiškai kitokią atvaizdavimo formą.

Pagrindinis tokio apdorojimo tikslas yra apdoroti duotą vaizdą taip, kad gautas vaizdas būtų tinkamesnis už originalų vaizdą konkrečiam taikymui. Taigi vaizdo tobulinimas paprastai yra orientuotas į problemą [51].

Norint atlikti kiekybinį vaizdo įvertinimą optinėje metrologijoje dažniausiai taikomos tokios metodikos kaip intensyvumo modelio konstravimas. Intensyvumo modelis, atsižvelgiant į pagrindinius vaizdo formavimo proceso parametrus, yra gana naudingas apdorojant ir analizuojant juostų raštų modelius. Be to, modelis apima įvairius trikdžius, tokius kaip foninė šviesa ir triukšmas, kurie daro įtaką rekonstruoto fazės pasiskirstymo tikslumui ir jo nustatymui taikytino metodo pasirinkimui. Kadangi atskirti įvairių rūšių triukšmo padarinius realiame vaizde yra gana neįmanoma, fizinis modelis, apibūdinantis vaizdo formavimo procesą su visais jo komponentais, suteikia galimybę apskaičiuoti dirbtinius vaizdus, kurie žingsnis po žingsnio apytiksliai atspindi sudėtingą realaus vaizdo struktūrą. Taikant šią dirbtinę aplinką, algoritmų ir vaizdų apdorojimo priemonių veikimas gali būti tiriamas labai efektyviai. Taip pat, patogesniais tampa klaidų aptikimas arba parametrų parinkimas [51].

Pramonėje dažniau sutinkamas kokybinis vaizdo vertinimas. Kokybinio vertinimo metodai buvo sėkmingai pritaikyti pramoniniu požiūriu aktualioms problemoms įvairiose srityse: plokščių ir elektroninių modulių kokybės kontrolėje, palydovinių degalų bakų ir slėginių indų patikrinimuose, padangų, stiklo ir anglies pluoštu armuotos medžiagos bandymai, automobilių variklių ir kėbulų patikra, pastatų ir meno dirbinių apžiūra, ir kt.. Šiuo metu kokybiniam vertinimui dažniausiai taikomas vienas iš dviejų metodų. Pirmasis taiko žinių surinkimu grįstas sistemas arba neuroninius tinklus, kad „išmoktų“ įvairių rūšių informaciją iš sumodeliuotų ar praktinių pavyzdžių. Antrasis metodas sujungia modeliais pagrįstus modeliavimo metodus ir praktinius matavimus. Tačiau daugeliu atvejų automatinis vertinimas neveikia tobulai, nes gedimus rodančių raštų atpažinimas ir klasifikavimas juostų modeliuose dažnai yra sudėtingesnis nei kiekybinė fazės rekonstrukcija, taikoma nustatant poslinkius ir matuojant formas [51].

1.4. Mašininis mokymasis

Mašininis mokymasis yra mokslo sritis, tirianti algoritmus, kurie tobulėja pasitelkdami patirtį. Mašininio mokymosi algoritmai sukuria matematinį modelį, pagrįstą imties duomenimis, vadinamais „mokymosi duomenimis“, kad būtų galima prognozuoti ar priimti sprendimus, jei jie nėra tiesiogiai užprogramuoti. Ši sritis taip pat artima statistikos, matematinės analizės ir optimizavimo, kurie duoda metodinį ir teorinį pagrindą mašininiam mokymuisi [7].

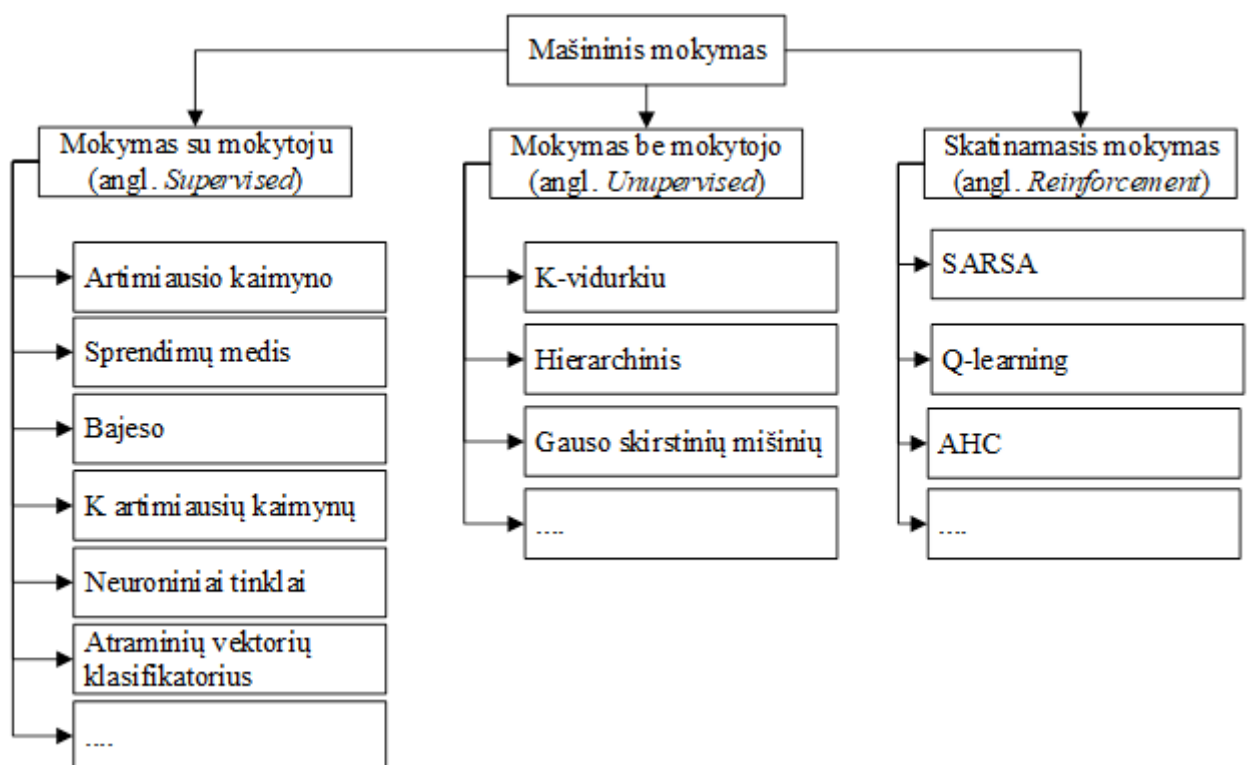
Mašininis mokymasis, nors ir tyrinėjimas jau ilgą laiką, tačiau dėl skaičiuojamųjų pajėgumų išaugimo moderniuose kompiuteriuose yra išpopuliarėjęs pastaraisiais dešimtmečiais. Vienas pirmųjų mašininio mokymo pritaikymų buvo atliktas A. Samuel'o 1952 m. Panaudodamas nesudėtingus algoritmus, jis sukūrė programą žaidžiančią šaškių žaidimą bei pirmą kartą pavartojo terminą „mašininis mokymas“ [47]. Vėliau F. Rosenblatt'as pasiūlė pirmąjį neuroninio tinklo mokymo algoritmą – perceptroną (angl. *perceptron*), kurio metu mokymas buvo vykdomas minimizuojant skirtumą tarp išėjimo vektoriaus ir norimo gauti vektoriaus [45][46].

1.4.1. Mašininio mokymosi metodų klasifikacija

Mašininio mokymosi algoritmai skirstomi į klases pagal veikimo principus, kontrolės aparatą. Dažnai išskiriamos trys pagrindinės kategorijos [39]:

- Mokymas su mokytoju (angl. *supervised*) – algoritmas, kurį galima dar įvardinti kaip „mokymasis iš patirties“. Šis algoritmas naudojamas, kai turimi sužymėti duomenys vadinami mokymo duomenimis. Modeliai mokomi atsižvelgiant į sužymėtus duomenis ir nauji duomenys vertinami pagal surinktą patirtį. Šis mokymo algoritmas dažniausiai naudojamas klasifikavimo arba regresijos uždaviniuose [11].

- Mokymasis be mokytojo (angl. *unsupervised*) – algoritmas, kuris apdoroja duomenis neturinčius sužymėjimo ir naudojamas klasterizavimo uždaviniams spręsti. Dėl šios priežasties galima atlikti tik tokius veiksmus kaip grupavimas arba taisyklių atradimas [11]. Pavyzdžiui, grupavimas yra naudingas segmentuojant pirkėjus pagal jų pirkimo įpročius, taisyklių atradimas gali padėti parodyti, kad pirkėjai, kurie perka vieną produktą, turi tendenciją pirkti ir kitą produktą.
- Skatinamasis mokymas (angl. *reinforcement*) – tai metodas, kuris nuolatos atnaujinama arba keičia elgsenos strategiją priklausomai nuo pateiktų žinių apie aplinką. Galima sakyti, kad mokymosi procesas šiam metodui yra nuolatinis ir turimas modelis yra nuolatos atnaujinamas pagal naujus duomenis [5].



3 pav. Mašininio mokymo metodų klasifikacija

Be šių pagrindinių kategorijų, galima išskirti dar keletą algoritmų, kurie pritaikomi rečiau:

- Dalinai prižiūrimas mokymas (angl. *semi-supervised*) – algoritmas, kuris mokosi turint dalį sužymėtų ir dalį nesužymėtų duomenų. Tokiu atveju yra sujungiami pažymėti ir nepažymėti duomenys atitinkamai funkcijai arba klasifikatoriui išgauti [11].
- Daugelio užduočių (angl. *multitask learning*) – algoritmas, kurio pagrindinis tikslas - padėti kitiems besimokantiems pasiekti geresnių rezultatų. Kai tam tikroje užduotyje pritaikomi daugelio užduočių algoritmai, jie įsimena, kaip problema buvo išspręsta arba kaip buvo pasiektas tam tikras rezultatas. Tada algoritmas šią patirtį pritaiko panašiai problemai ar užduočiai spręsti. Ši vieno algoritmo pagalba kitam gali būti vadinama induktyviu žinių

perdavimo mechanizmu. Jei besimokantieji dalijasi savo žiniomis, jie gali mokytis vienu metu arba kartu, tai turėtų įtakos greitesniam apsimokymui [8].

- Signalo keitimo (angl. *transduction*) – algoritmas panašus į mokymą su mokytoju, tačiau nebūtinai sukonstruoja funkciją. Vietoje to, bando prognozuoti naujus rezultatus pagal mokymo duomenis, mokymo rezultatus ir naujus įvesties duomenis [62].

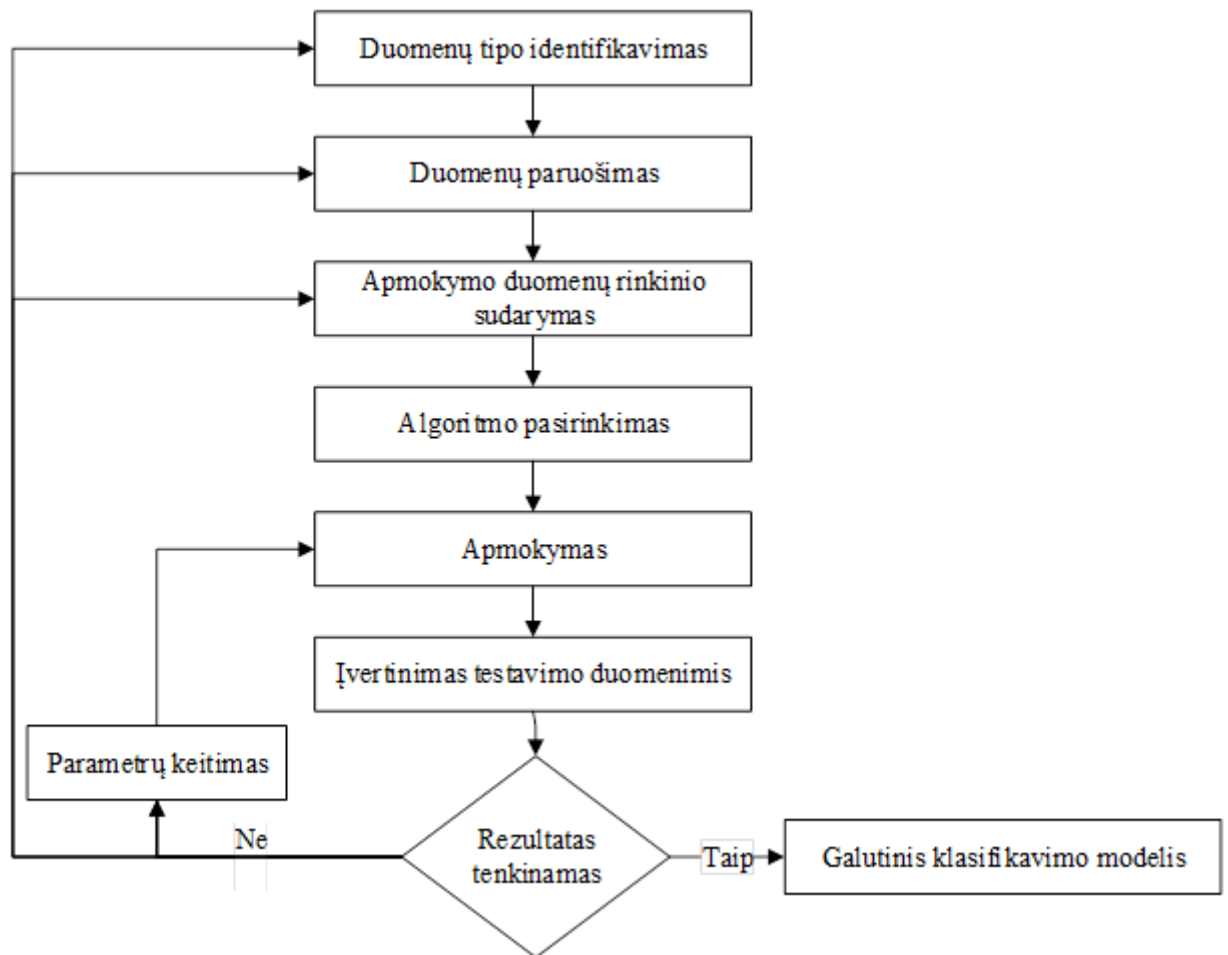
Taip pat yra daug smulkių kategorijų, kurios taikomos tik konkrečiais atvejais ir panaudojimas nėra toks platus kaip aptartų kategorijų. Toliau bus plačiau apžvelgiamas mokymo su mokytoju algoritmas, dėl aktualumo sprendžiant turimą uždavinį.

1.4.2. Mokymasis su mokytoju

Įprastame mašininio mokymo modelyje mokymo procesas dažniausiai yra padalintas į dvi dalis: mokymą ir testavimą. Mokymo procese imami stebėjimai kaip įvestis ir naudojant mokymo algoritmą sudaromas modelis [64]. Gauti modeliai dažniausiai taikomi klasifikavimo uždaviniams spręsti.

Mokymas su mokytoju yra metodika dažniausiai naudojama mokant neuroninius tinklus ir sprendimų medžius. Abu šie metodai yra stipriai priklausomi nuo pradinio informacijos suklasifikavimo arba sužymėjimo. Neuroninių tinklų atveju, pradinė klasifikacija naudojama nustatyti tinklo paklaidai ir ją vėliau minimizuoti. Tuo tarpu, sprendimo medžio atveju pradinė klasifikacija leidžia nuspręsti kurie atributai suteikia daugiausiai informacijos ir gali būti naudojami naujos informacijos klasifikavime [62].

Mokymą su mokytoju galima išskirstyti etapais, kaip pavaizduota 4 pav. Pirmasis etapas yra duomenų surinkimas ir svarbios informacijos arba laukų išrinkimas. Toliau seka duomenų paruošimo ir apdorojimo etapas, kuomet yra panaikinami stipriai išsiskiriantys ar neteisingi duomenys. Apmokymo duomenų rinkinio sudarymo metu atskiriama dalis duomenų modelio apmokymui ir dalis testavimui. Nuo šio žingsnio gali priklausyti ar neįvyks persimokymas. Pasirinkus algoritmą, bei apmokius modelį atliekamas įvertinimas naudojant testavimo duomenis. Galiausiai jei rezultatai patenkinami, tuomet gaunamas modelis arba klasifikatorius, priešingu atveju grįžtama redaguoti algoritmo parametrų arba į pirmus žingsnius.



4 pav. Mokymo su mokytoju proceso schema

1.4.3. Mašininio mokymo metodai vaizdų klasifikavimui

Atraminų vektorių klasifikatorius (AVK) (angl. *support vector machines*) yra populiarus metodas klasifikuojant vaizdus. Pastaraisiais metais atsirado nauja metodo variacija pagrįsta statistiniu mokymusi, kuris naudoja struktūrinės rizikos minimizavimo teoriją ir yra labai apibendrintas [26]. AVK demonstruoja gerus rezultatus, turi gerą teorinį pagrindimą bei buvo taikomi daugelyje sričių. Iš pradžių šis metodas buvo naudojamas binarinės klasifikacijos problemai spręsti. Atsižvelgiant į duomenų rinkinį, kuriame vienas įrašas priklauso vienai iš dviejų klasių, AVK bando rasti hiperplokštumą, kad atskirtų pavyzdžius iš dviejų skirtingų kategorijų ir padidintų atstumą tarp dviejų skirtingų klasių [57].

K artimiausių kaimynų (angl. *k-nearest neighbours*) yra paprastas, tačiau gerai žinomas klasifikavimo metodas, kurio klasifikavimui naudojama informacija iš k artimiausių duomenų taškų. Šis algoritmas naudoja tam tikrus atstumo matavimus, kad palygintų atstumą ar panašumą tarp dviejų duomenų taškų, pvz., Euklido atstumą, Manhatto atstumą ir Kanberos atstumą. Klasifikuojamo duomenų taško žymėjimai turėtų atitikti daugumos aplink šį duomenų tašką esančius žymėjimus. K artimiausių kaimynų metodas buvo pritaikytas daugelyje sričių, tarp kurių yra didieji duomenys [12], biologija [52].

Sprendimų medis yra metodas naudojantis sprendimų medį kaip prognozavimo modelį, kuris sujungia pastabas apie objektą išreikštą šakomis. Išvados apie elemento siektina vertę vaizduojama lapais. Tai yra vienas iš prognozavimo modeliavimo metodų naudojamų statistikoje, mašiniame mokymesi. Medžio modeliai, kuriuose priklausomas kintamasis gali įgauti baigtinę reikšmių aibę, yra vadinami klasifikuojančiais medžiais. Tai palygintinai paprastas metodas, kuris dažnai lyginamas su anksčiau minėtais metodais [41].

Dirbtiniai neuroniniai tinklai šiuo metu yra viena labiausiai tyrinėjamų sričių vaizdo klasifikavimo problemoje. Viena iš šio metodo modifikacijų yra konvoliuciniai neuroniniai tinklai, kuri atkartodama akies regėjimo principus leidžia pasiekti labai gerų rezultatų. Šie metodai bus apžvelgiami toliau.

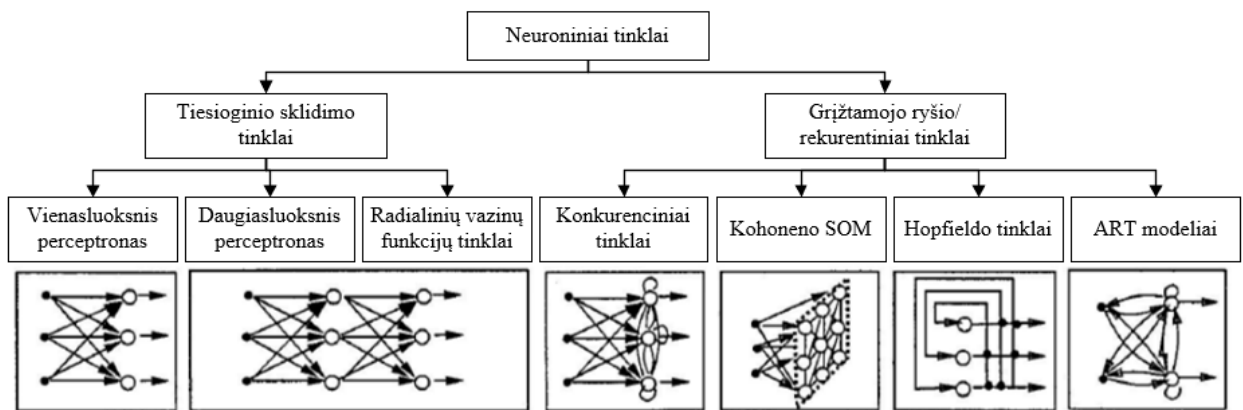
1.4.4. Dirbtiniai neuroniniai tinklai ir jų klasifikacija

Neuroniniai tinklai (NN) (angl. *neural networks*) arba dirbtiniai neuroniniai tinklai (DNT) (angl. *artificial neural networks*) – tai informacijos apdorojimo struktūros sukurtos imituojant gyvųjų organizmų smegenyse vykstančius biologinius informacijos apdorojimo procesus. NN sudaromi iš keleto tarpusavyje sujungtų paprastų elementų, vadinamų neuronais. Šie elementai apjungti vieni su kitais tam tikro stiprumo jungtimis yra apytikris biologinių neuronų tinklo modelis. Neuroniniai tinklai pradėti nagrinėti XX a. šeštajame dešimtmetyje, tačiau iki devintojo dešimtmečio vidurio jie nebuvo plačiai taikomi ir naudojami. Tik sukūrus greitus ir galingus modelių apmokymo mechanizmus bei atsiradus pakankamo skaičiavimo pajėgumo kompiuterius NN galėjo spręsti realius uždavinius [1].

NN dažniausiai skirstomi į dvi pagrindines kategorijas:

- tiesioginio sklidimo tinklai (angl. *feedforward*);
- grįžtamojo ryšio tinklai (angl. *feedbackward*).

Tiesioginio sklidimo tinkluose signalas juda tik viena kryptimi – į priekį. Tokiems tinklams priskiriami vienasluoksniai ir daugiasluoksniai tinklai. Tuo tarpu, grįžtamojo ryšio tinkluose signalo sklidimas galimas į visas puses. Platesnis galimas dirbtinių neuroninių tinklų klasifikavimas 8 pav.



8 pav. Dirbtinių neuroninių tinklų architektūrų klasifikacija [54]

1.5. Muaro metodų ir neuroninių tinklų pritaikymų apžvalga

Vieni pirmųjų šešėlinio muaro ir neuroninių tinklų metodų pritaikymai buvo atlikti medicinos srityje. H. Kim'as, S. Ishikawa, M. Khalid'as ir kt. panaudojo šešėlinio muaro metodą atliekant pacientų nugaros srities nuotraukas. Šiose nuotraukose nugaros sritis buvo atskiriama į dvi dalis pagal centrinę liniją. Kiekvienoje srityje tiriant binarizuotų pikselių dažnį buvo išskiriami centroidai bei atstumai tarp jų. Gauti parametrai panaudoti klasifikuojant nuotraukas kuriose yra fiksuojamos stuburo patologijos ir kuriose patologijos nėra [32].

Taip pat, ši sritis buvo sėkmingai pritaikyta industrijoje, kurioje atsižvelgiant į objektų formą ir deformacijas jie buvo klasifikuojami pagal dydžius arba nustatinėjami defektai. Vienas iš pavyzdžių yra M. M. Ratman'o atliktas darbas kuriant automatinę kiaušinių klasifikavimo procedūrą. Jos metu buvo atliekamos kiaušinių nuotraukos naudojant šešėlinio muaro metodiką ir gautuose vaizduose išskiriami parametrai (atstumai nuo centro) ant pasirinktų interferencinių juostų. Pasitelkiant neuroninius tinklus buvo atlikta šių kiaušinių dydžių klasifikacija [43].

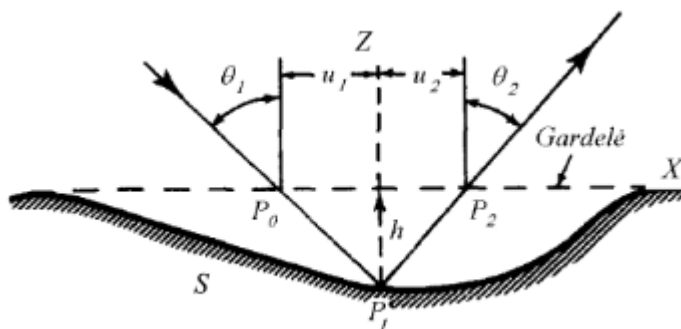
W. H. Woo, M. M. Ratnam'as, ir K. S. Yen'as 2015 metais pirmą kartą aprašė metodiką, kurioje pasinaudojant neuroniniu tinklu iš nuotraukos išskiriamos muaro interferencinės juostos centras. Šie metodai leidžia atsisakyti arba dalinai atsisakyti ekspertinio muaro interferencinių juostų sužymėjimo bei pagreitinti nuotraukų apdorojimo procesą [60].

Vienas naujų darbų apjungiantis neuroninius tinklus ir muaro juostas yra E. Abraham'o darbas nuotraukų klastočių atpažinime. Fotografuojant ekraną dažnai persidengia ekraną ir fotoaparato sensorių sudarantys tinkleliai ir sukuriamos aukšto dažno interferencinės – muaro juostos. Šiam efektui atpažinti panaudoti neuroniniai tinklai [3].

2. Medžiagos ir tyrimų metodai

2.1. Šešėlinis muaras

Muaro juostos, kaip aptarta 1.2 skyriuje, susidaro, kai dvi gardelės (dvi tiesių lygiagrečių brūkšnių sistemos) uždedamos viena ant kitos tam tikru kampu. Jeigu gardelės juostos yra pakankamai plonos, o gardelės žingsnis pakankamai mažas, tai atskiros juostos yra nematomos, o išryškėja tik kombinacinės arba muaro juostos. Žingsnis tarp šių juostų priklauso nuo gardelių susikirtimo kampo ir įprastai būna didesnis negu naudojamų gardelių juostų žingsnis. Šis metodas puikiai tinka matuoti įvairiems poslinkiams ir deformacijoms, tačiau, siekiant išmatuoti objektų formą ir deformacijas statmenai jų paviršiui naudojama muarų juostų metodo modifikacija. Nuo klasikinio metodo ši modifikacija skiriasi tuo, kad muaro juostos gaunamos ne panaudojant dvi gardelės ir stebint jų superpoziciją, bet projektuojant šešėlinį vienos gardelės vaizdą ant paviršiaus kampu θ_1 ir jį stebint per tą pačią gardelę kitu kampu θ_2 , kaip pavaizduota 10 pav.. Šis metodas vadinamas šešėliniu muaru [18].



10 pav. Šešėlinio muaro juostų sudarymo schema [18]

Matematiškai tai galima aprašyti taip, kad taškas P_0 , esantis ant gardelės, yra projektuojamas į tiriamojo objekto paviršiaus tašką P_1 . Stebint iš θ_2 , taškas P_1 susiprojektuoja į tašką P_2 esantį ant gardelės plokštumos. Tai yra lygu gardelės poslinkiui jo šešėlio atžvilgiu dydžiu, kurį pažymėsime u

$$u = u_1 + u_2 = h(x, y)(\operatorname{tg}\theta_1 + \operatorname{tg}\theta_2),$$

čia h žymimas atstumas nuo gardelės plokštumos iki paviršiaus taško P_1 . Daugumą gardelių sudaro nepermatomos, tamsios ir permatomos, skaidrios linijos. Jų laidumo arba atspindžio koeficientas keičiasi pagal stačiakampės bangos funkciją. Tačiau Furjė analizė leidžia manyti, kad matematiškai bet kokia gardelė yra sudaryta iš tam tikro skaičiaus sinusinių gardelių, kurios gali būti aprašomos taip:

$$t_1(x) = a + \operatorname{acos}\left(\frac{2\pi}{p}x + \psi(x)\right),$$

čia a – laidžio amplitudė; p – žingnis; ψ – moduluojančioji funkcija. Jeigu gardelė griežtai sinusinė, tai moduluojančioji funkcija lygi nuliui:

$$t_2(x) = a + a \cos\left(\frac{2\pi}{p}x\right),$$

čia manoma, kad abiejų gardelių linijos lygiagrečios y ašiai. Tačiau rezultatas nesikeis, jeigu linijos bus pakreiptos kampu y ašiai. Abiem gardelėms persidengiant, suminis laidis keičiasi kaip abiejų gardelių laidžių sandauga:

$$t = t_1 t_2 = a^2 \left\{ 1 + \cos\frac{2\pi}{p}x + \cos 2\pi \left[\frac{x}{p} + \psi(x) \right] + \frac{1}{2} \cos 2\pi \left[\frac{2x}{p} + \psi(x) \right] + \frac{1}{2} \cos 2\pi \psi(x) \right\},$$

čia pirmieji trys nariai atvaizduoja pradines gardeles, ketvirtasis išreiškia dvigubo dažnio pirmąją gardelę, o penktasis atspindi tik moduluojančiąją funkciją.

Jeigu abi gardelės paėliui eksponuojamos ant to paties paviršiaus, tai jų laidžiai susideda:

$$t = t_1 + t_2 = 2a \left\{ 1 + \cos \pi \psi(x) \cos 2\pi \left[\frac{x}{p} + \frac{1}{2} \psi(x) \right] \right\}.$$

Iš šios išraiškos matoma, kad ji atitinka pirmąją amplitudę moduluotą gardelę. Galima pastebėti, kad šviesos juostos bus ten, kur $\psi(x) = n$, kai $n = \dots, -2, -1, 0, 1, 2, \dots$, o tamsios juostos bus ten, kur $\psi(x) = n + \frac{1}{2}$.

Šešėlinio muaro atveju, poslinkis u apibrėžia moduliavimo funkciją:

$$\psi(x) = \frac{u}{p} = \frac{h(x, y)}{p} (tg\theta_1 + tg\theta_2).$$

Šviesos juostos bus ten, kur $\psi(x) = n$, kai $n = 0, 1, 2, \dots$, arba ten, kur

$$h(x, y) = \frac{np}{tg\theta_1 + tg\theta_2},$$

o tamsios juostos, kur

$$h(x, y) = \frac{\left(n + \frac{1}{2}\right)p}{tg\theta_1 + tg\theta_2}.$$

Taigi, gaunamas objekto vaizdas, kuris yra padengtas topografinėmis linijomis. Kaip pavyzdį galima pateikti H. Kim'o, S. Ishikawa, M. Khalid'o ir kitų šešėlinio muaro pritaikymo pavyzdį medicinoje, kuriuo buvo sukurta metodika, įvertinti stuburo pakitimus, stebint asmens nugaros asimetriškumą [32].

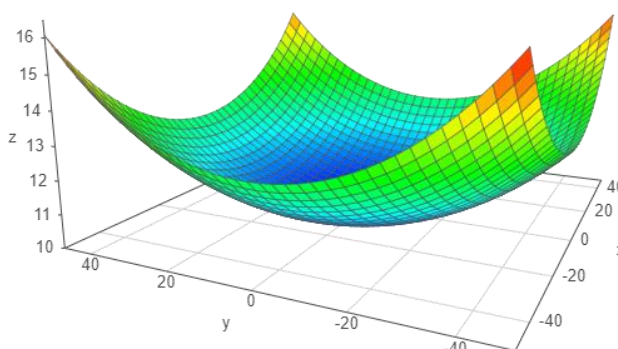
2.2. Vaizdinių šešėlinio muaro duomenų generavimo strategija

Šiame darbe klasifikavimui bus naudojami programiškai sugeneruoti vaizdai, todėl svarbu apibrėžti jų generavimo strategiją. Šios strategijos pagrindas bus sugeneruoti šešėlinio muaro juostas

pagal tam tikrą paviršiaus funkciją bei ją atvaizduoti ant pasirinkto fono, taip imituojant foninio paviršiaus deformaciją.

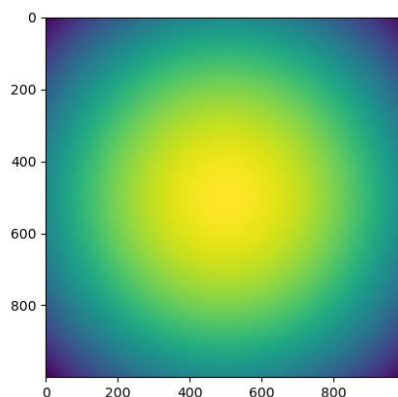
Pirmasis žingsnis yra pasirinkti imituojamo paviršiaus funkciją, kurios parametrus atsitiktinai keičiant bus išgaunami skirtingi paviršiai arba deformacijos. Šiai užduočiai naudosime elipsinį paraboloidą (11 pav.):

$$z(x, y) = \alpha e^{-\beta \left(\left(\frac{x-u}{p} \right)^2 + \left(\frac{y-v}{q} \right)^2 \right)}.$$



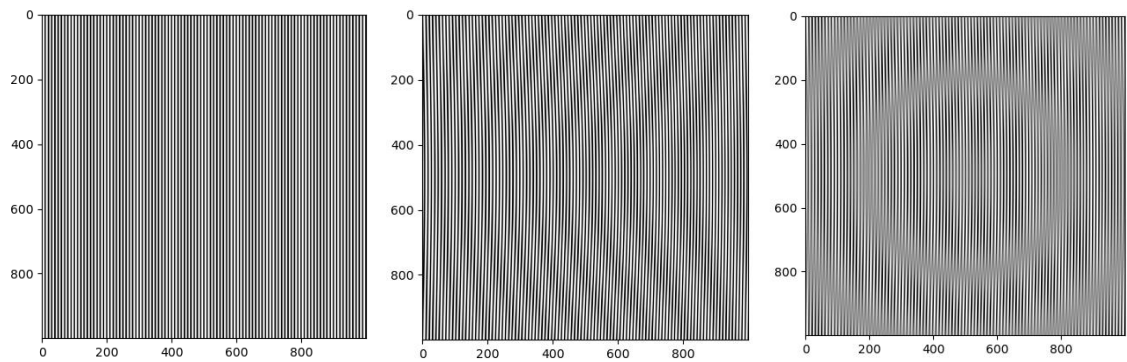
11 pav. Paraboloido pavyzdys

Kitas žingsnis yra sugeneruoti paviršiaus šešėlinio muaro juostas. Tai atliekama sugeneruojant pradinę gardelę, jos kopiją deformuojant pagal pasirinktą paviršių bei šias gardeles atvaizduojant kartu. Pirmiausia, generuojama skaičių aibė $G_x = \{x_1, x_2, \dots, x_n\}$, kuri atitiks pradinės gardelės X ašį. Toliau pasirenkame sinusoidę $F(x)$, kuri panaudojama iš G_x narių sudaryti gardelę. Turint pradinę gardelę, sugeneruojamas paviršiaus topografinis žemėlapis $Z(X, Y)$, pavaizduotas 12 pav.



12 pav. Paraboloido topografinio žemėlapis pavyzdys

Šis žemėlapis apdorojamas funkcija $\Delta Z = (1 - Z) \tan \lambda$, kur λ yra šviesos šaltinio kampas. Tuomet, gaunama matrica $D_z = G_x + \Delta Z$, kuri apdorojama $F(x)$ bei gaunama deformuota gardelė.

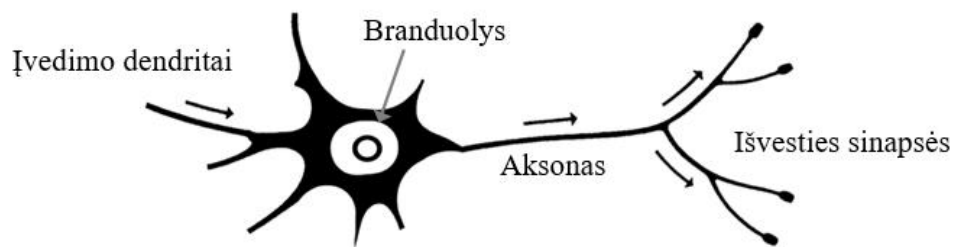


13 pav. Pradinės, deformuotos bei apjungtų gardelių pavyzdys

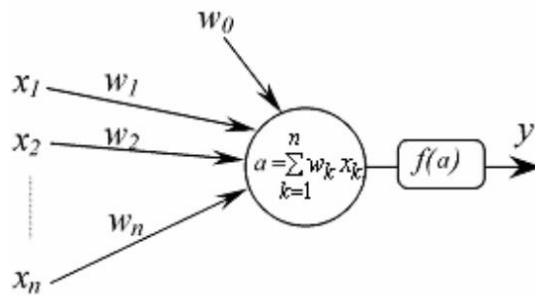
Galiausiai, turint interferencines šešėlinio muaro juostas, jas galima panaudoti padengiant foninį vaizdą. Tam pasirenkamos elektroninių plokščių nuotraukos ir galutinis vaizdas gaunamas naudojant tokius nuotraukų apdorojimo įrankius kaip kontrasto padidinimas, spalvų paletės konvertavimas į juodą ir baltą spalvas arba iš trijų kanalų į vieno kanalo spalvas.

2.3. Dirbtiniai neuroniniai tinklai

Šiame skyriuje aptariama NN matematinė interpretacija. Pirma, svarbu apibrėžti dirbtinį neuroną, kuris yra neuroninio tinklo pagrindas. 14 ir 15 pav. atitinkamai atvaizduotos biologinio ir dirbtinio neuronų schemas. Biologinis neuronas susideda iš dendritų, kurie priima signalus, branduolio, kuris tą informaciją apdoroja, ir aksono, per kurį perduodamas signalas kitiems neuronams. Neuronas aktyvuojamas kuomet priimant signalus iš aplinkinių neuronų gaunamas signalo lygis viršija tam tikrą reikšmę, kuri vadinamas sužadavimo slenksčiu. Dirbtiniu neuronų bandoma atkartoti šią struktūrą bei veikimą [29].



14 pav. Biologinio neurono supaprastinta schema



15 pav. Dirbtinio neurono schema

Pavaizduotoje schemoje neuroną sudaro 5 pagrindiniai komponentai.

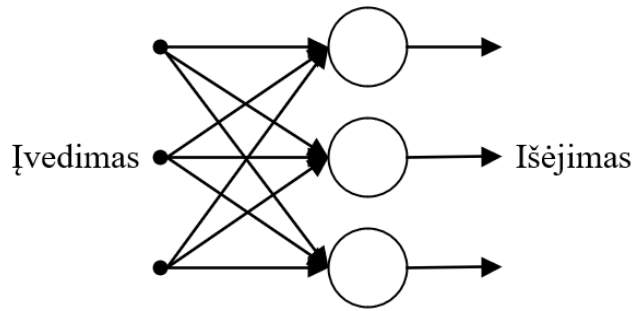
1. Įėjimų (angl. *input*) aibė per kurią neuronui perduodamos reikšmės x_1, x_2, \dots, x_n .
2. Sviurių (angl. *weight*) aibė. Kiekvienas įeities taškas turi savo perdavimo koeficientą arba svorį w_1, w_2, \dots, w_n . Tai dažniausiai realūs skaičiai, kurie nurodo kiekvieno įėjimo svarbą kitų įėjimų atžvilgiu. Neuronu branduolyje skaičiuojama įėjimo reikšmių x ir sviurių w sandaugų suma:

$$a = w_1 x_1 + w_2 x_2 + \dots + w_n x_n = \sum_{k=1}^n w_k x_k.$$

3. Poslinkis (angl. *bias*) arba slenkstis (angl. *threshold*). Šiuo atveju pažymėsime b . Ši reikšmė nustato ribą įėjimo reikšmei nuo kurios yra aktyvuojamas neuronas pridėdant arba atimant iš įėjimo reikšmių ir sviurių sandaugų sumos.
4. Aktyvacijos funkcija. Pagal sužadavimo signalą a perdavimo funkcija $f(a)$ apskaičiuojama neurono išėjimo reikšmė y . Dirbtinio neurono modelyje gali būti naudojamos įvairios aktyvacijos funkcijos. Priklausomai nuo neuroninio tinklo, gali būti taikomos įvairios aktyvacijos funkcijos.
5. Neuronu išėjimo (angl. *output*) arba rezultato reikšmės. Apskaičiuojamos pagal išraišką:

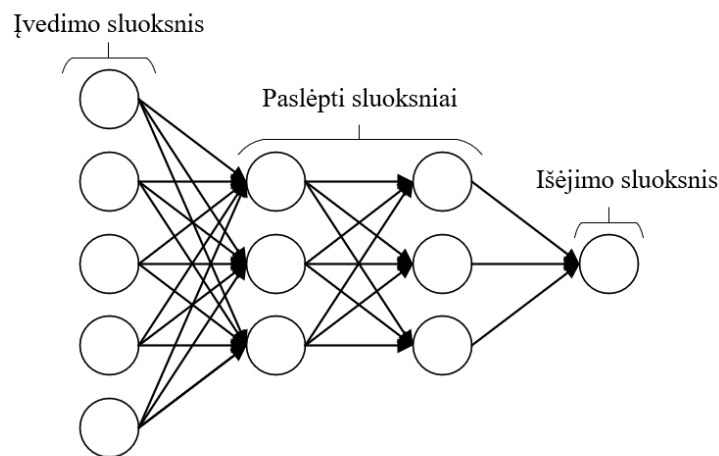
$$y = f(a + b).$$

Neuroninį tinklą sudaro daugelis apjungtų neuronų. Tuo tarpu, paprasčiausias neuroninis tinklas yra vienasluoksnis perceptronas. Jis sudaromas iš vieno neuronų sluoksnio, kuris atlieką informacijos į tinklą įvedimo ir išvedimo funkciją. Toks tinklas pavaizduotas 16 pav. ir dažniausiai yra skirtas spręsti mažo sudėtingumo klasifikavimo uždavinius [29].



16 pav. Vienasluoksnio perceptrono schema

Sudėtingesniems uždaviniais spręsti naudojama daugiasluoksnis tinklas. Dažniausia daugiasluoksnio tinklo struktūra – vienas po kito einantys neuronų sluoksniai, kurių kiekvienas neuronas sujungtas su visais kito sluoksnio neuronais ir neturintis jokių kitų jungčių. Tiesioginio sklaidimo neuroninis tinklas turintis daugiau nei vieną neuronų sluoksnį vadinamas daugiasluoksniais tiesioginio sklaidimo neuroniniais tinklais (angl. *multilayer feedforward neural networks*). Jo supaprastintas modelis atvaizduotas 17 pav. Toks tinklas turi vieną arba daugiau paslėptų sluoksnių.



17 pav. Supaprastintas daugiasluoksnis tiesioginio sklaidimo neuroninis tinklas

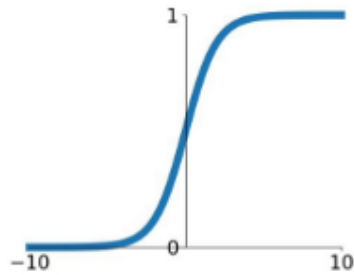
2.4. Aktyvacijos funkcijos

Svarbu plačiau apžvelgti aktyvacijos funkciją, kadangi nuo jos pasirinkimo priklauso modelio apsimokymo kokybė. Aktyvacijos funkcija yra matematinė operacija apdorojanti įvesties signalą taip, kad išvesties signalas patektų į tam tikrą intervalą. Dažniausiai šie intervalai būna $(0,1)$ ir $(-1,1)$. Toliau apžvelgsiu dažniausiai pritaikomas aktyvacijos funkcijas.

Sigmoidinė arba logistinė funkcija apibrėžiama taip:

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

Ji pavaizduota 18 pav.

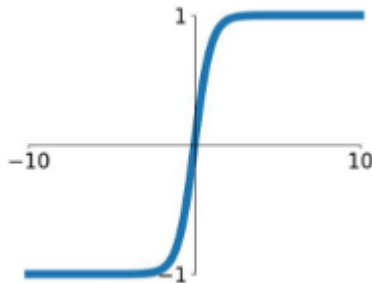


18 pav. Sigmoidinė funkcija

Sigmoidinė funkcija įgauna reikšmes intervale $(0, 1)$. Tiriant dažnai reikalingos ir neigiamos reikšmės. Tokiam atvejui dažnai taikoma hiperbolinio tangento aktyvacijos funkcija, kuri išreiškiama

$$f(x) = \frac{2}{1 + e^{-2x}} - 1.$$

Ji įgauna reikšmes intervale $(-1; 1)$ ir yra pavaizduota 19 pav.

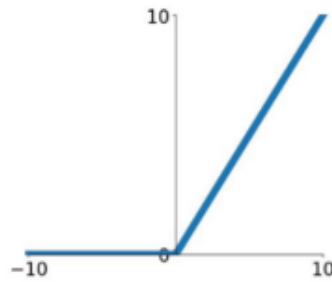


19 pav. Hiperbolinio tangento funkcija

Šiuo metu tarp populiariausių aktyvacijų funkcijų yra lyginanti (ReLU) (angl. *rectifier*) funkcija, kuri išreiškiama taip:

$$f(x) = \max(0, x).$$

ReLU funkcija pavaizduota 20 pav. Naudojant šią aktyvacijos funkciją neuroninio tinklo mokymo paklaida konverguoja daug greičiau nei naudojant sigmoidinę ar hiperbolinio tangento funkcijas. Taip pat, kadangi šioje funkcijoje naudojamos labai paprastos operacijos padidinamas neuroninio tinklo apmokymo greitis bei išnaudojama mažiau resursų.

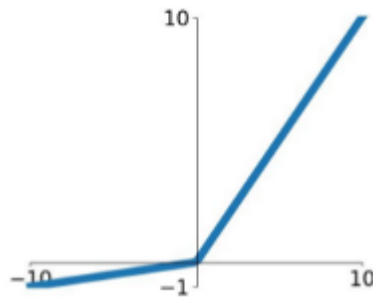


20 pav. ReLU funkcija

Visgi, ReLU tam tikrai atvejais gali sukurti situaciją, kurioje neuronas yra nebeapmokomas. Jei neurono išvestis yra neigiama su visomis įvestimis ReLU reikšmė tampa lygi 0 ir pagal atgalinės sklaidos algoritmo svorių atnaujinimą į neuroną įeinantys svoriai nebebus apmokomi. Šiai problemai spręsti buvo sukurta ReLU modifikacija leaky ReLU. Šioje funkcijoje įvedamas papildomas parametras α . Šis parametras dažnai pasirenkamas tarp 0,1 ir 0,01. Ši funkcija išreiškiama

$$f(x) = \max(\alpha x, x).$$

Leaky ReLU pavaizduota 21 pav.



21 pav. Leaky ReLU funkcija

Taip pat, sukurta parametrinė ReLU (PReLU) modifikacija (angl. *parametric ReLU*), kuri apibrėžiama kaip ir leaky ReLU, tačiau šiuo atveju parametras α yra mokomas tinklo parametras. Tiek leaky ReLU, tiek PReLU tam tikrais atvejais turėtų leisti pasiekti geresnius apmokymo rezultatus nei ReLU [61].

2.5. Tinklo parametų inicijavimas

Neuroninio tinklo pradinių svorių inicijavimui dažnai naudojamas normalusis ar tolygusis skirstinys. Siekiant geresnių rezultatų buvo pasiūlytos šių metodų modifikacijos. Viena iš jų yra Glorot'o normalusis ir Glorot'o tolygusis skirstinys. Pradiniai svoriai naudojant Glorot'o normalųjį skirstinį inicijuojami naudojant išraišką

$$W \sim N(0, \sigma), \text{ kur } \sigma = \sqrt{\frac{2}{n_j + n_{j+1}}},$$

kur n_j ir n_{j+1} yra neuronų skaičius dabartiniame ir sekančiame neuroniniame sluoksnyje. Tuo tarpu, svoriai pagal Glorot'o tolygų skirstinį inicijuojami pagal išraišką

$$W \sim U \left[-\frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}, \frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}} \right].$$

Taip pat, dažnai ReLU tipo aktyvacijos funkcijoms yra naudojamas He inicijavimo metodas [23]. Naudojantis šiuo metodu kiekvienam tinklo sluoksniui pradinės reikšmės parenkamos pagal tai kiek jis turi įvesčių. Pagal He normalųjį skirstinį parametrai inicijuojami naudojant išraišką

$$W \sim N(0, \sigma), \text{ kur } \sigma = \sqrt{\frac{2}{n}}.$$

Tyrimė bus naudojamas būtent He skirtinio parametų inicijavimas.

2.6. Klaidos skleidimo atgal algoritmas tinklo apmokymui

Šis algoritmas vadinamas klaidos skleidimo atgal algoritmu, kadangi jis paskirsto klaidą tarp neuroninio tinklo elementų pradėdant nuo viršaus ir tolyn į apačią. Šis algoritmas susideda iš dviejų operacijų: įėjimo reikšmių skleidimas į priekį tarp įėjimų ir išėjimų sluoksnių, ir paklaidos skleidimas atgal tarp išėjimų ir įėjimų sluoksnių.

Šiame algoritme panaudojama mokymo su mokytoju strategija. Pradiniai duomenys yra mokymo reikšmių vektorius ir norimų gauti reikšmių vektorius. Tiesioginio skleidimo metu kiekvienas elementas skaičiuoja svorinę įėjimo elementų sumą, kuri žymima

$$a_j = \sum_i w_{ji} z_i,$$

čia a_j yra svorinė j -tojo neurono išėjimų suma, z_i - įėjimo į neuroną reikšmė arba žemesnio sluoksnio išėjimo reikšmė, w_{ji} - svoriai tarp nagrinėjamo sluoksnio ir žemesnio sluoksnio neuronų elementų.

Svorinė įėjimo elementų suma transformuojama perdavimo funkcija

$$z_j = f(a_j).$$

Neuroninio tinklo mokymo metu stengiamasi minimizuoti tam tikra klaidos funkciją apibrėžtą klaidą. Šiuo atveju klaida apskaičiuojama pagal suminę klaidą ir ji bus žymima E_s , kur s yra duomenų vektoriaus indeksas. Klaida E_s šiuo atveju priklauso nuo w_{ji} per svorinę įėjimų į j -ąjį neuroną sumą žymima a_j . Tuomet, galima išreikšti dalines klaidos išvestines pagal w_{ji} :

$$\frac{\partial E_s}{\partial w_{ji}} = \frac{\partial E_s}{\partial a_j} \cdot \frac{\partial a_j}{\partial w_{ji}}.$$

Tuomet, pažymėkime neurono išėjimo klaidą

$$\delta_j = \frac{\partial E_s}{\partial a_j}.$$

Išreiškiame tinklo jėjimą

$$\frac{\partial a_j}{\partial w_{ji}} = \frac{\partial (\sum_i w_{ji} z_i)}{\partial w_{ji}} = z_i.$$

Gaunama dalinės klaidos išraiška:

$$\frac{\partial E_s}{\partial w_{ji}} = \delta_j z_i.$$

Toliau, išėjimo sluoksnių klaidą žymime δ_k . Žinodami, kad $z_k = f(a_k)$ gauname klaidą išėjimo sluoksniams:

$$\delta_k = \frac{\partial E_s}{\partial a_k} = \frac{\partial E_s}{\partial z_k} \cdot \frac{\partial z_k}{\partial a_k} = f'(a_k) \frac{\partial E_s}{\partial z_k}.$$

Paslėptiems sluoksniams:

$$\delta_i = \frac{\partial E_s}{\partial a_j} = \sum_k \frac{\partial E_s}{\partial a_k} \cdot \frac{\partial a_k}{\partial a_j}.$$

Kai visi svoriai būna pakeisti, tuomet procesas kartojamas su nauju mokymo vektoriumi. Algoritmas sustabdomas arba pagal iš anksto nustatytą paklaidos funkcijos slenksčio reikšmę, arba pagal nurodytą iteracijų skaičių [17].

2.7. Mokymo optimizavimo algoritmai

Gradientinis nusileidimas yra klasikinis mokymo optimizavimo algoritmas, plačiai apžvelgtas literatūroje bei naudojamas kaip daugelio kitų metodų pagrindas. Šio metodo taikymo metu tinklo išmokstamų parametrų reikšmių atnaujinimas vyksta po kiekvienos epochos apskaičiavus klaidos funkcijos reikšmę kiekvienam mokymosi rinkinio elementui. Šio algoritmo reikšmių atnaujinimas gali būti apibrėžtas k-toje iteracijoje taip:

$$x_{k+1} = x_k - t_k \Delta f(x_k)^{(1:n)},$$

čia Δf - klaidos funkcijos gradientas, t_k - mokymosi greitis iteracijoje.

Įprasto gradientinio nusileidimo trūkumas yra tai, jog jis dažnai užstringa lokaliame optimumo taške ir negali iš jo ištrūkti. Šiai problemai spręsti buvo sukurta šio metodo modifikacija – gradientinis nusileidimas su pagreičiu (Momentum) (angl. *gradient decent with momentum*). Šiame metode įvedamas parametras m , kuris nurodo kiek iteracijų anksčiau apskaičiuoti gradientai bus yra įtraukti į dabartinį atnaujinimą. Šio algoritmo atnaujinimas apibrėžiamas taip:

$$v_k = m v_{k-1} + t_k \Delta f(x_k),$$

$$x_{k+1} = x_k - v_k.$$

Dažniausiai m parametro reikšmė laikoma 0,5 kol pradinis apsimokymas stabilizuojasi ir toliau padidinama iki 0,9 arba daugiau, kad ieškoti išėjimų iš lokalaus optimumo taško [63].

Tiek gradientinį nusileidimą, tiek gradientinį nusileidimą su pagreičiu galima laikyti baziniais algoritmais kituose algoritmuose. Toliau, ypač populiarius ir gerus rezultatus pasiekiantys yra adaptyvaus mokymosi greičio metodai. Šių metodų veikimas pagrįstas tuo, jog mokymo greitis apskaičiuojamas atskirai kiekvienam parametrai ir naudojama informacija apie prieš tai naudotus gradientus parametru atnaujinimui. Pavyzdžiui, parametrams su dideliais susietais gradientais mokymosi greitis sumažinamas, o parametrams su mažais susietais gradientais greitis padidinamas.

Nesterov'o pagreitinto gradiento (NAG) (angl. *Nesterov accelerated gradient*) algoritmas yra gradientinio nusileidimo su pagreičiu modifikacija, kurio esmė yra aproksimuoti būsimas parametru reikšmes prieš skaičiuojant gradientą. Tuomet gradientas apskaičiuojamas ne pagal dabartines parametru reikšmes, o pagal tai kur jos bus kitame žingsnyje. Šio metodo žingsnis apibūrinamas lygtimi:

$$v_k = mv_{k-1} + t_k \Delta f(x_k - mv_{k-1}),$$

$$x_{k+1} = x_k - v_k.$$

Buvo atlikti tyrimai tarp 7 populiariausių optimizavimo algoritmų, naudojant 3 skirtingus vaizdų rinkinius. Rezultatuose galima pastebėti, kad visais atvejais geriausius rezultatus pasiekdavo adaptyvaus pagreičio įvertinimo (Adam) (angl. *adaptive moment estimation*) ir Nesterov'o pagreitintas Adam (Nadam) (angl. *Nesterov-accelerated adaptive moment estimation*) [16].

Adam algoritme yra keturi reguliuojami parametrai: β_1 , β_2 - eksponentinio slopinimo koeficientai, t – žingsnio dydis ir ε – labai mažas skaičius dalybos iš nulio išvengimui. Dažniausiai, rekomenduojama naudoti standartines šių parametru reikšmes: $\beta_1 = 0,9$, $\beta_2 = 0,999$, $t = 0,001$, $\varepsilon = 10^{-8}$. Šiuo atveju, kiekvienas apmokomas parametras yra siejamas su parametrai m ir v , kurie yra pirmo ir antro momentų įverčiai. Algoritmo atnaujinimas apibūrinamas taip:

$$m_k = \beta_1 m_{k-1} + (1 - \beta_1) \Delta f(x_k),$$

$$v_k = \beta_2 v_{k-1} + (1 - \beta_2) \Delta f(x_k)^2,$$

$$\hat{m}_k = \frac{m_k}{1 - \beta_1^k},$$

$$\hat{v}_k = \frac{v_k}{1 - \beta_2^k},$$

$$x_{k+1} = x_k - \frac{t}{\sqrt{\hat{v}_k} + \varepsilon} \hat{m}_k.$$

Nadam apjungia Adam ir NAG algoritmus. Jis apibūrinamas papildant Adam algoritmo apibūrinimą būsimų parametru aproksimavimu pagal NAG:

$$m_k = \beta_1 m_{k-1} + (1 - \beta_1) \Delta f(x_k),$$

$$v_k = \beta_2 v_{k-1} + (1 - \beta_2) \Delta f(x_k)^2,$$

$$\hat{m}_k = \frac{m_k}{1 - \beta_1^k},$$

$$\hat{v}_k = \frac{v_k}{1 - \beta_2^k},$$

$$x_{k+1} = x_k - \frac{t}{\sqrt{\hat{v}_k} + \varepsilon} \left(\beta_1 \hat{m}_k + \frac{(1 - \beta_1) \Delta f(x_k)}{1 - \beta_1^k} \right).$$

2.8. Reguliarizavimas

Reguliarizavimas yra metodas, kuris atlieka mažas modifikacijas apsimokymo algoritme, kad modelis nepersimokytų ir neapsimokytų per mažai. Dažniausiai naudojami yra L2, L1 arba išmetimo (angl. *dropout*) metodai.

L2 ir L1 metodai veikia įvesdami regularizavimo reikšmę, kuri yra pridedama prie klaidos funkcijos. L2 atveju ši reikšmė yra apibrėžiama $\lambda \sum w^2$, čia λ - regularizavimo parametras, kuris gali būti optimizuojamas geresniems rezultatams pasiekti, $\sum w$ - tinklo svorių suma. L1 atveju regularizavimo reikšmė yra $\lambda \sum |w|$. Abu metodai atlieką panašią funkciją – mažina svorių reikšmes. L2 atveju svoriai negali pasiekti nulinės reikšmės, o L1 atveju, kadangi atimama skaičiuojama absoliutinių svorių suma, kai kurie svoriai gali būti sumažinti iki nulinės reikšmės. Tokiu būdu siekiama, kad tinklo rezultatai nebūtų nulemti tik keleto svorių su didelėmis reikšmėmis.

Išmetimo metodo veikimas pagrįstas tinklo struktūros modifikavimu kiekvienai mokymo iteracijai. Kiekvienos iteracijos metu pasirinktuose tinklo sluoksniuose su nurodyta tikimybe atsitiktinai parenkami neuronai ir jų visi svoriai, kurie dabartinėje iteracijoje nedalyvaus. Taip į mokymo procesą įvedamas atsitiktinumas, sumažinantis priklausomybę tarp neuronų. Išmetimo metodo tikimybė yra tinklo parametras, kurį reikia optimizuoti norint pasiekti optimalių rezultatų.

Vienas iš naujesnių metodų naudojamų konvoliuciniuose neuroniniuose tinkluose yra rinkinio normalizavimas (angl. *batch normalization*). Šis metodas veikia normalizuojant ankstesnio neurono aktyvacijos funkcijos išvesties reikšmę, atimant rinkinio vidurkį ir padalijant iš rinkinio standartinio nuokrypio. Teoriškai šis metodas turėtų įvesti regularizavimo į tinklą bei pagreitinti jo apsimokymą.

2.9. Konvoliuciniai neuroniniai tinklai

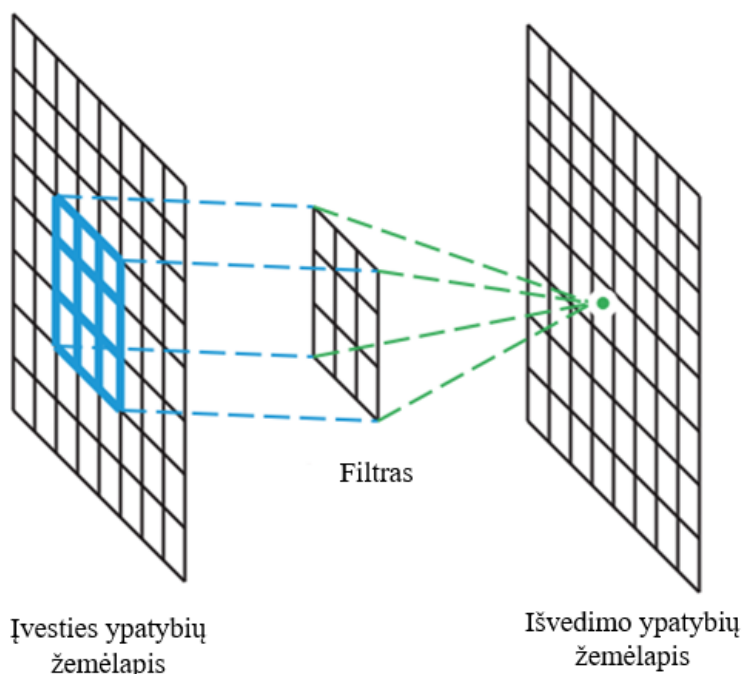
Vaizdų apdorojimui iki šiol aptarti neuroniniai tinklai (dažniausiai naudojami daugiasluoksniai tiesioginio sklaidimo tinklai) nėra efektyvus, kadangi augant duomenų apimčiai, apdorojamų vaizdų detalumui ir dydžiui, stipriai išauga tinklo jungčių skaičius. Dėl to išauga ir tinklo apmokymo laikas. Taip pat, dažnai vaizduose galime įžvelgti gretimų pikselių priklausomybę nuo vienas kito, bet iki šiol aptartų neuroninių tinklų architektūrose tolimi pikseliai turi tokia pat įtaką kaip ir artimi. Šiems neuroninių tinklų trūkumams išspręsti buvo sukurti konvoliuciniai neuroniniai tinklai [33].

Konvoliuciniai neuroniniai tinklai, kaip ir įprasti neuroniniai tinklai sudaryti iš įvedimo, išvedimo ir kelių paslėptų sluoksnių. Paslėpti sluoksniai sudaromi iš konvoliucinio sluoksnio, surinkimo sluoksnio (angl. *pooling*) ir pilnai sujungtų sluoksnių. Atsižvelgiant į tai, jog tinklo įvestis yra vaizdas, neuronai šiuose sluoksniuose išdėstomi pagal tris dimensijas – plotį, aukštį, gylį. Plotis ir aukštis apibrėžiamas pagal nuotraukos dydį, o gylis atitinka spalvų kanalų skaičių. Pavyzdžiui, nagrinėjant nuotrauką, kurios rezoliucija yra 100×100 pikselių RGB spalvų erdvėje ir kurioje yra trys spalvų kanalai: raudona, žalia ir mėlyna, sudarant įvesties sluoksnį, jos dimensijos būtų $100 \times 100 \times 3$.

Konvoliucinio sluoksnio pagrindinis tikslas yra išgauti vaizdo ypatybes arba šablonus. Tai atliekama apmokomais filtrais. Tuo tarpu, šių filtrų skaičius nurodo sluoksnio gylį. Svarbus filtrų parametras yra jų dydis, kuris, kaip ir sluoksnis, aprašomas trimis dimensijomis. Įprastai šių filtrų dydis pasirenkamas kur kas mažesnis nei tiriamo vaizdo dydis. Pavyzdžiui, galime pažvelgti į AlexNet architektūros konvoliucinį tinklą, kuris 2012 metais laimėjo ILSVRC-2012 konkursą skirtą vaizdų klasifikavimui ir stipriai aplenkė klasikinius klasifikavimo metodus. Šiuo atveju $224 \times 224 \times 3$ dydžio vaizdui buvo pritaikomas $11 \times 11 \times 3$ filtras [33]. Konvoliucijos operaciją dvimačiam vaizdui galima apibrėžti taip:

$$S(p, q) = (I * K)(p, q) = \sum_{i=1}^N \sum_{j=1}^M I(i, j)K(p - i, q - j),$$

čia I – dvimatis vaizdas, o K – dvimatis branduolys arba filtras.



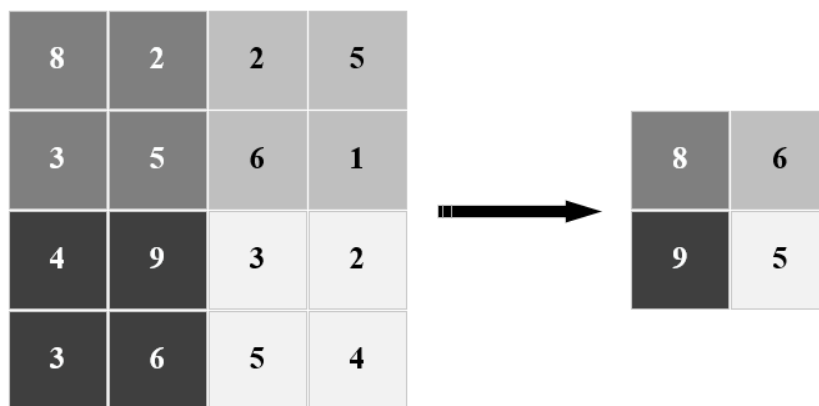
22 pav. Konvoliucijos operacijos supaprastinta schema [25]

Konvoliucijos operacija atliekama slenkant filtro dydžio langą per įvesties vaizdą tarp lango ir filtro reikšmių. Supaprastinta schema pateikta 22 pav. Gaunamas rezultatas vadinamas ypatybių

žemėlapiu (angl. *feature map*). Svarbi konvoliucinio tinklo savybė yra poslinkio invariantiškumas (angl. *shift invariant*), kuri reiškia jog tinklas tam tikrą savybę, pavyzdžiui tam tikrą briauną, gali surasti nepriklausomai nuo jos vietos vaizde. Konvoliucinį sluoksnį galime apibrėžti šiais hiperparametrais: filtro dydis ($F \times F$), filtrų skaičius (K), filtro slinkimo žingsnio dydis (S) (angl. *stride*). Kai kuriais atvejais įvedamas krašto padengimas nuliais (angl. *zero-padding*), kuris leidžia filtrui panaudoti nulines reikšmes esančias už vaizdo ribų, šiuo atveju žymimas (N). Konvoliucinio sluoksnio dimensija, esant įvesties dimensijai $W \times H \times D$, gali būti apibrėžiama taip:

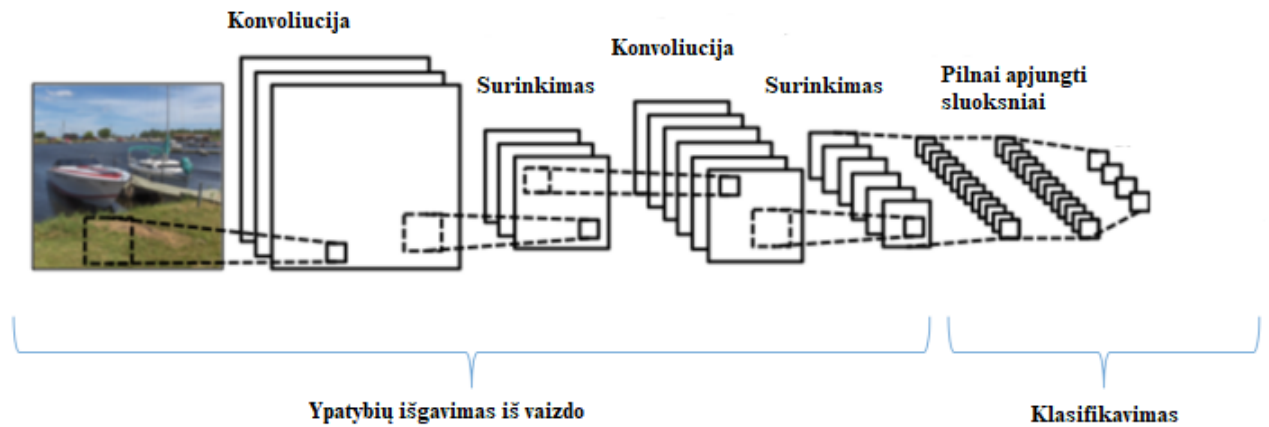
$$\left(\frac{W - F}{S} + 1\right) \times \left(\frac{H - F}{S} + 1\right) \times D.$$

Po konvoliucijos operacijos dažnai seka surinkimo arba telkimo operacija, kurios pagrindinis tikslas yra sumažinti įvedimo dydį sumažinant tinklo parametrų skaičių, pagreitinant apsimokymą bei kontroliuojant persimokymą. Surinkimo operacijos metu yra parenkamas lango dydis ir jis slenkamas per įvesties matricą pritaikant tam tikrą funkciją į langą patekusioms reikšmėms, kurios rezultatas dažniausiai būna viena reikšmė. Dažniausiai naudojama funkcija yra maksimalios reikšmės, kuri gražina didžiausia tarp lange esančių reikšmių. Priešingai nei konvoliucijos operacijos vykdyme, surinkime langai nepersidengia.



23 pav. Surinkimo operacijos pavyzdys naudojant maksimalios reikšmės funkciją

Konvoliucinių neuroninių tinklų architektūra dažniausiai susideda iš tam tikro skaičiaus konvoliucinių sluoksnių besikeičiančių su surinkimo sluoksniais bei galiausiai pilnai sujungtais sluoksniais bei išvedimo sluoksniu.



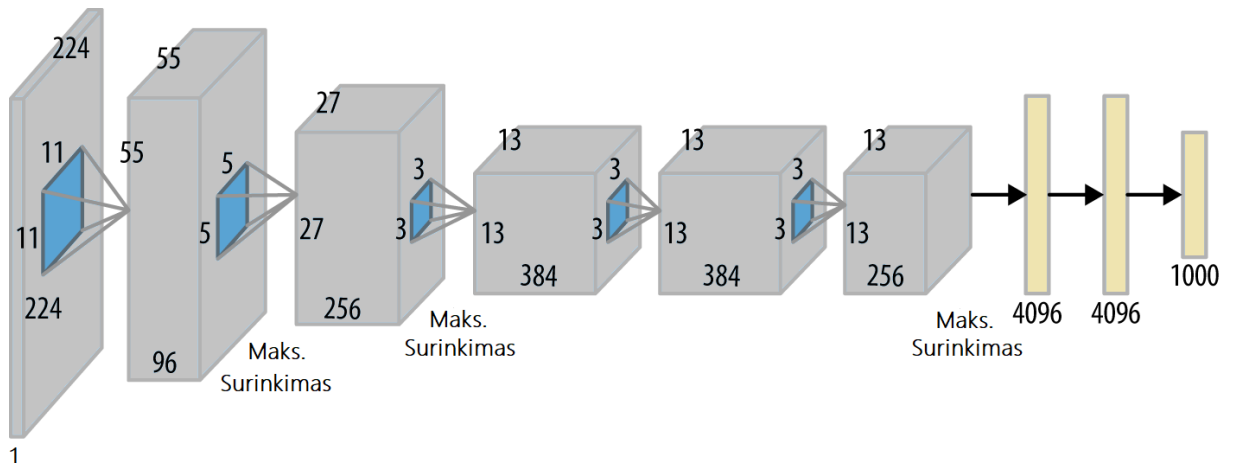
24 pav. Konvoliucinio neuroninio tinklo architektūros pavyzdys [24].

Šio tinklo apmokymui dažniausiai naudojamas klaidos skleidimo atgal principas.

2.10. Naudojamos konvoliucinių neuroninių tinklų architektūros

LeNet-5 yra viena paprasčiausių konvoliucinių neuroninių tinklų architektūra, kuri buvo pristatyta dar 1998 metais. Ji sukurta $32 \times 32 \times 3$ dydžio vaizdams ir bendrai yra sudaryta iš 7 sluoksnių. Šioje architektūroje naudojami 2 konvoliuciniai sluoksniai, kuriuose naudojama atitinkamai 6 ir 16 filtrų, kurių dydžiai yra 5×5 ir žingsnis lygus 1. Po kiekvieno konvoliucinio sluoksnio seka telkimo sluoksniai, kurių filtro dydis yra 3×3 su žingsnio dydžiu lygiu 1. Po šių sluoksnių seka 3 pilno jungimo sluoksniai, kurių neuronų skaičius yra atitinkamai 120, 84 ir 10. Ji panaudota skaitinių vaizdų klasifikavimui [14].

Taip pat, bus pritaikoma jau minėta AlexNet architektūra. Ji sukurta aukštesnės raiškos vaizdų klasifikavimui, todėl įvesties sluoksnio dimensijos yra $224 \times 224 \times 3$. Šioje architektūroje yra 5 konvoliuciniai sluoksniai, kuriuose filtrų skaičius atitinkamai yra 96, 258, 384, 384 ir 258. Filtrų dydis yra atitinkamai 11×11 , 5×5 ir paskutinių trijų 3×3 . Pirmasis sluoksnis turi žingsnio dydį lygų 4, tuo tarpu, likusieji lygų 1. Šis tinklas išsiskyrė tuo, kad telkimo sluoksniai nebuvo pritaikyti po kiekvieno konvoliucinio sluoksnio. Telkimo sluoksniai buvo taikomi po pirmųjų dviejų ir paskutinio konvoliucinio sluoksnio su 3×3 dydžio filtru ir žingsnio dydžiu lygiu 2. Po šių sluoksnių sekė 3 pilno jungimo sluoksniai, kurie atitinkamai turėjo 4096, 4096 ir 1000 neuronų. Šios architektūros schema pavaizduota 25 pav.



25 pav. AlexNet tinklo architektūros schema

2.11. Programiniai įrankiai ir įranga

Šiuo metu, viena populiariausių bibliotekų skirtų mašininio mokymosi algoritmų realizavimui yra Tensorflow, sukurta Google kompanijos 2015 metais. Šios bibliotekos populiarumą nulėmė išsami dokumentacija, laisvai prieinamas programinis kodas, naudojimo paprastumas ir stiprus palaikymas iš kūrėjų. Tensorflow palaiko daugelį standartinių mašininio mokymo metodų: įvairios architektūros neuroninius tinklus, atraminių vektorių mašinas, atsitiktinius miškus ir kitus. Šios bibliotekos trūkumas yra toks, jog skaičiavimai, kurie pagreintintų tinklo apsimokymą, naudojant vaizdo plokštes, yra apribotos tik vieno gamintojo – NVIDIA, naudojant CUDA skaičiavimų aplinką. Norint panaudoti kito gamintojo įrangą, galima pasitelkti vieną populiariausių alternatyvų – PlaidML biblioteka. Tai yra atvirojo kodo biblioteka veikianti panašiu principu, kaip ir Tensorflow. Nors PlaidML nėra taip gerai palaikoma, dokumentuota bei dažnai reikalaujanti daugiau įdirbio norint ją pasinaudoti, tačiau naudojanti atvirojo kodo OpenCL karkasą (alternatyva CUDA), kuris yra plačiai palaikomas įvairios programinės įrangos ir suteikia galimybę panaudoti įvairių gamintojų techninę įrangą. Būtent ši biblioteka bus naudojama tyrimo skaičiavimuose naudojant Python programavimo kalbos sąsają.

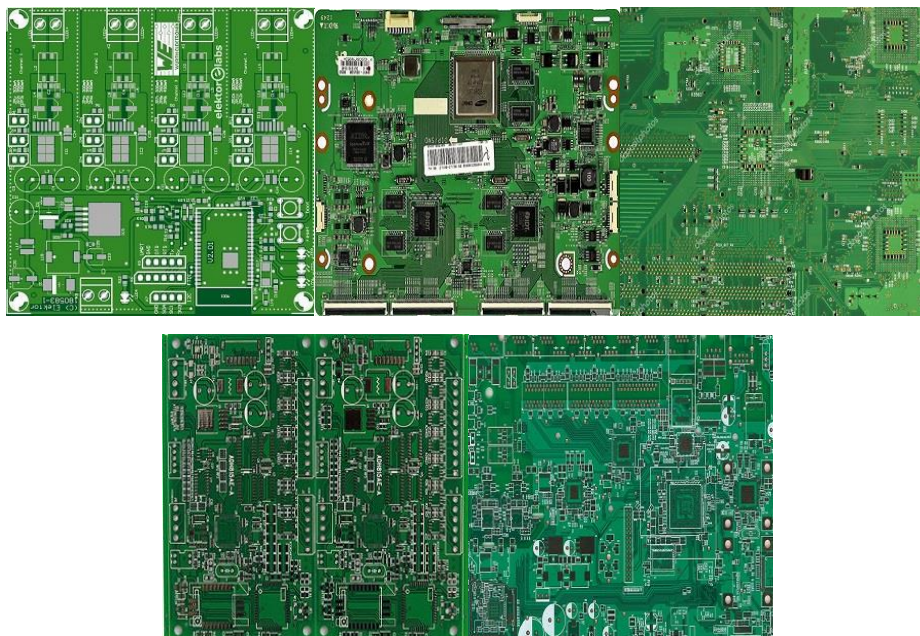
Keras yra dar viena populiari biblioteka, palaikoma tiek Tensorflow, tiek PlaidML. Ši biblioteka supaprastina neuroninių tinklų kūrimą pasinaudojant kitų bibliotekų pagrindu. Ši biblioteka leidžia paprastai kurti tinklus, prototipus bei juos testuoti. Taip pat, ji palaiko įvairias sujungimo schemas – daugybinius įvedimo ir išvedimo modelius, sluoksnių dalinimąsi ir kt.

Tyrimo įgyvendinimui naudojama Python programavimo kalba (3.7 versija). Darbui su neuroniniais tinklais naudojamos PlaidML (0.7.0 versija) ir Keras (2.2.4 versija) bibliotekos. Bandymai atliekami kompiuteryje su Windows 10 64 bitų operacinė sistema. Jo specifikacija: AMD Ryzen 5 2600 3.4 GHz 6 branduolių procesorius, AMD Radeon RX 580 plokštė su 8GB GDDR5 darbinės atminties, 8GB DDR4 darbinė atmintis, SSD kietasis diskas.

3. Tyrimų rezultatai ir jų aptarimas

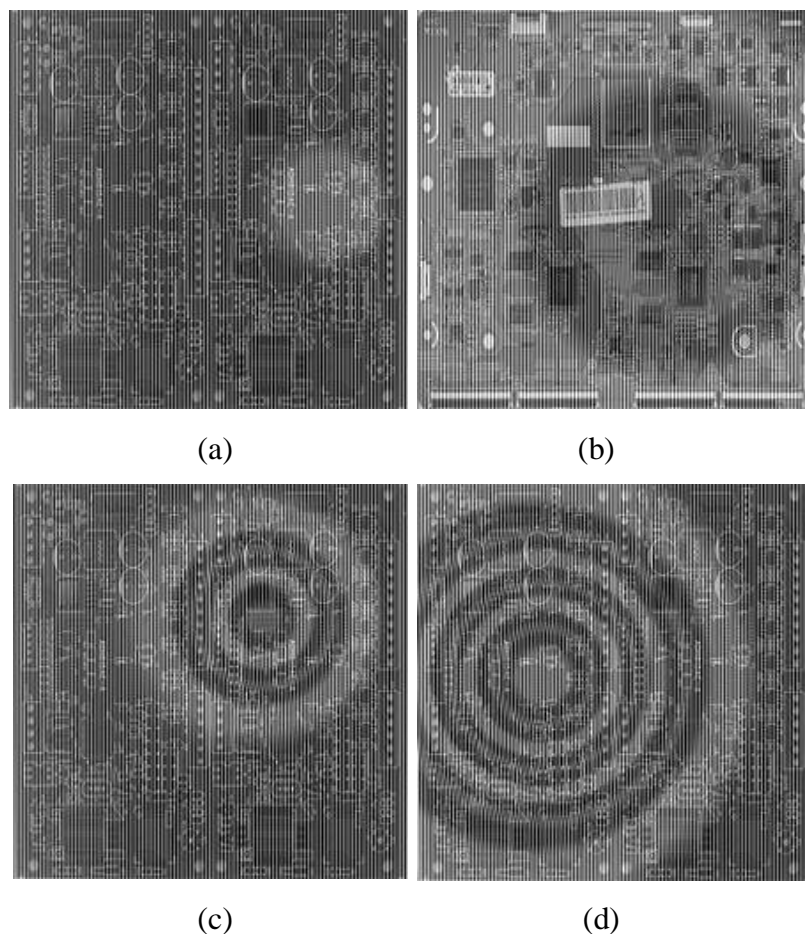
3.1. Tyrimo duomenys

Tyrimo duomenys bus generuojami pagal 2.2 skyrelyje aprašytą metodiką. Interferencinėmis juostomis bus padengiami penki foniniai spausdintų elektronikos plokščių vaizdai, pavaizduoti 26 pav.



26 pav. Vaizdų generavimui naudojamos spausdintų elektroninių plokščių nuotraukos

Vaizdai bus skirstomi į 4 klases: nedeformuota, silpnai deformuota, vidutiniškai deformuota ir stipriai deformuota detalė. Vaizdai suskirstomi į klases pagal aukščiausią arba žemiausią deformacijos tašką. Taip pat, vaizdai apdorojami konvertuojant juos į vieną spalvinį kanalą - juodai baltus vaizdus. Tai atliekama, kadangi šiuo atveju tik foniniuose vaizduose yra spalvinė informacija, kuri klasifikavimui nėra aktuali. Taip sumažinama reikiama tinklo apimtis ir bei panaikinama nenaudinga informacija. 27 paveiksle pavaizduoti kiekvienos klasės galutinių vaizdų pavyzdžiai.



27 pav. Pirmos (a), antros (b), trečios (c), ketvirtos (d) klasės vaizdų pavyzdžiai

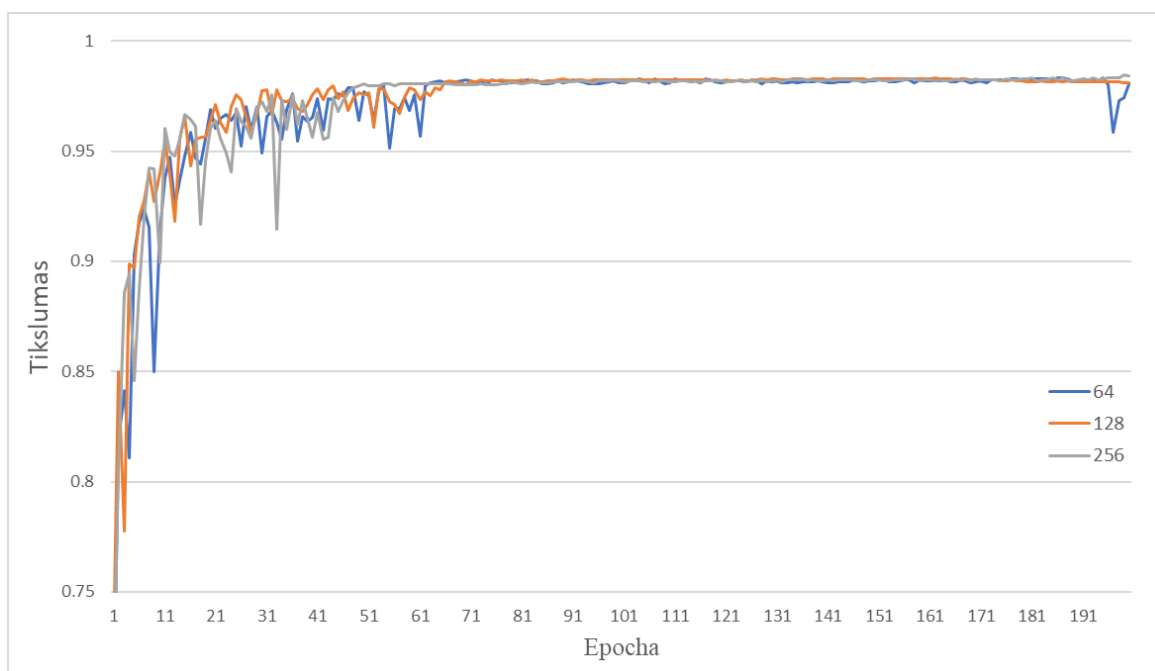
Modelio mokymo ir testavimo duomenims, panaudojant 4 foninius elektronikos plokščių vaizdus, bus sugeneruota 10 tūkstančių vaizdų. Iš jų kiekvienai klasei priklausys po 2500 vaizdus. Šis duomenų rinkinys dalinamas į mokymo ir testavimo santykiu 80/20. Klasėse esančių vaizdų kiekis tolygiai pasiskirstęs abiejose imtyse. Panaudojant likusį foninį vaizdą sugeneruojami papildomi 2000 vaizdų su tolygiai pasiskirsčiusiu klasėse esančių vaizdų skaičiumi. Šis duomenų rinkinys bus naudojamas tinklo jautrumui foninio vaizdo pokyčiams įvertinti.

3.2. Konvoliucinio neuroninio tinklo realizacija

Atsižvelgiant į tai, jog vaizdų rinkinys yra pakankamai mažas, jį sudaro tik 4 klasės ir ribotas kiekis ypatybių, kurios gali būti išskiriamos vaizduose, realizacijai buvo pasirinkta AlexNet architektūra, kuri yra viena paprasčiausių iš modernių architektūrų. Šį tinklą sudaro 5 konvoliuciniai sluoksniai, kuriuose filtrų skaičius atitinkamai yra 96, 258, 384, 384 ir 258. Pirmojo filtro žingsnio dydis yra 4, o likusiųjų 1. Filtrų dimensijos atitinkamai yra 11×11 , 5×5 ir likusiųjų 3×3 . Taip pat, tarp konvoliucinių sluoksnių įterpiami 3 apjungimo sluoksniai, kurie yra įterpiami atitinkamai po pirmojo, antrojo ir penktojo konvoliucinių sluoksnių. Apjungimo langai yra 3×3 dydžio. Trys paskutiniai sluoksniai yra pilno jungimo, kurių atitinkamai neuronų skaičius yra 4096, 4096 ir 1000, kur paskutinis sluoksnis yra išvesties.

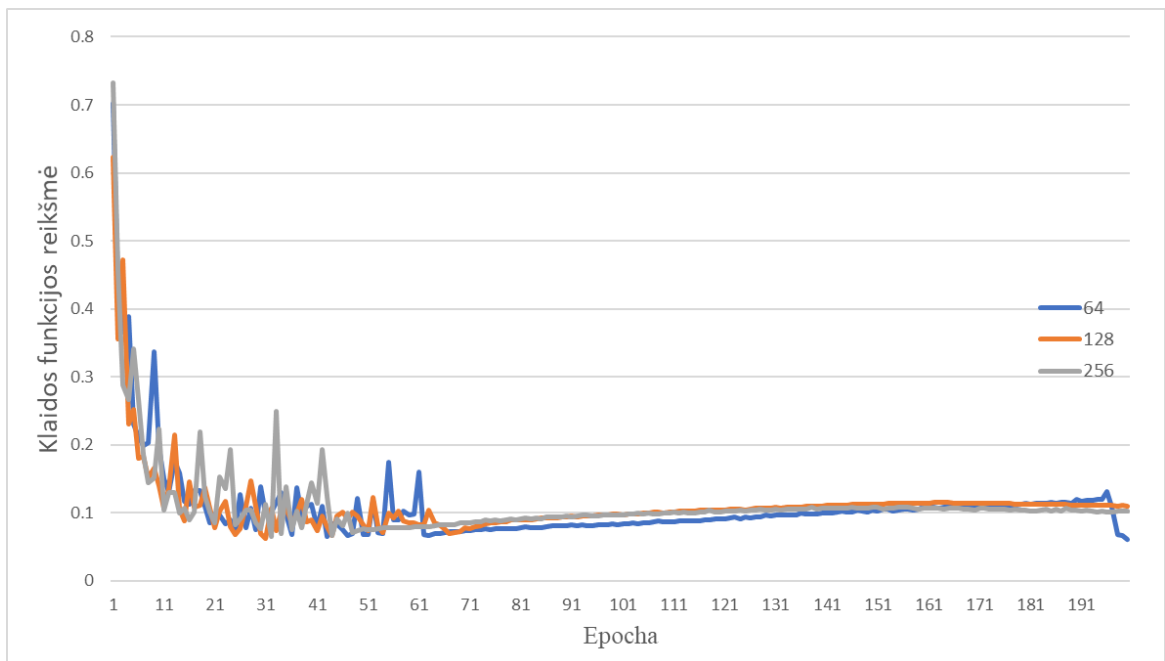
Visgi, norint apmokyti šios architektūros tinklą turimais duomenimis, jis turi būti modifikuojamas. Standartiškai šios architektūros įvesties dimensijos yra $224 \times 224 \times 3$, o naudojamų duomenų - $224 \times 224 \times 1$, todėl įvesties sluoksnis buvo pritaikytas vieno spalvinio kanalo duomenims. Ši architektūra buvo kuriama ir apmokoma ImageNet duomenų rinkiniu, kurį sudaro duomenys suskirstyti į 1000 klasių, todėl tiek išvesties dimensija, tiek pilnai sujungti sluoksniai yra per dideli tiriamam uždaviniui. Išvesties sluoksnio neuronų skaičius keičiamas į 4. Likusiems dviem pilnai sujungtiems sluoksniams buvo atliekami bandymai atitinkamai su 512, 256, 128 ir 64 neuronais. 512 neuronų pločio tinklui trūko informacijos ir jis praktiškai nesimokė, todėl detaliau buvo žvelgiama į 256, 128 ir 64 neuronų pločio sluoksnius.

Tinklas buvo mokomas 200 epochų. Po kiekvienos epochos vykdomas atsitiktinis įvedamų vaizdų sumaišymas. Konvoliuciniuose ir pilnai sujungtuose sluoksniuose naudojama ReLU aktyvacijos funkcija. Klaida apskaičiuojama pagal kategorinės kryžminės entropijos funkciją ir naudojamas Adam optimizavimo algoritmas.



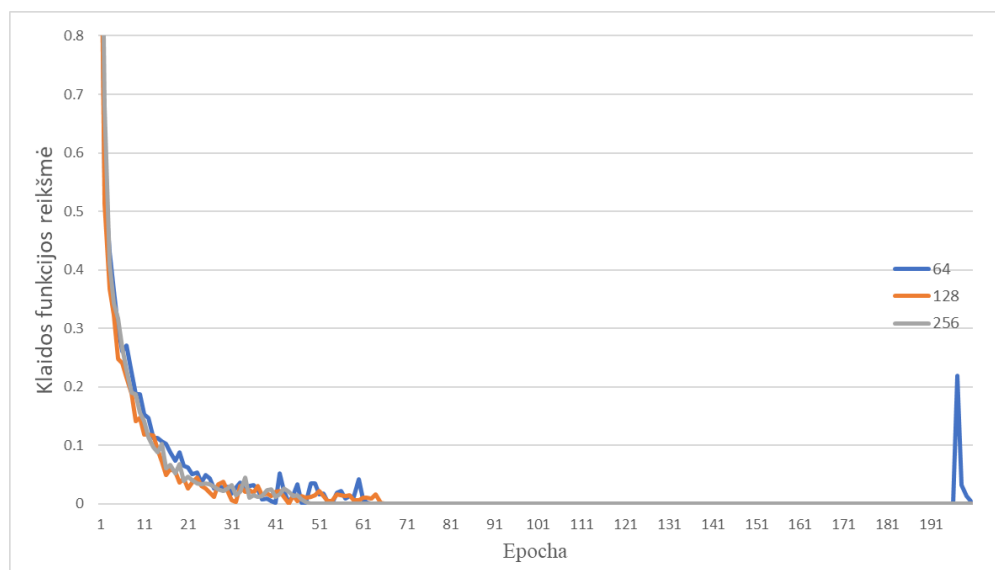
28 pav. AlexNet modifikacijų su 64, 128 ir 256 neuronų pilno sujungimo sluoksniais validacijos tikslumas

Tinklų apmokymas truko nuo 2h 29 min. mažiausios apimties tinklui ir iki 2 val. 51 min didžiausiam. 28 pav. matoma, kad visi tinklai, 64, 128, 256 neuronų, po 200 epochų pasiekė labai panašų tikslumą, atitinkamai 98,35%, 98,35% ir 98,4%. Mokymo procese visi trys tinklai elgėsi panašiai ir iki vienodo lygio apsimokyti užteko iki 70 epochų.



29 pav. AlexNet modifikacijų su 64, 128, 256 neuronų pilno sujungimo sluoksniais validacijos klaidos funkcijos reikšmės

Pagal klaidos funkcijos reikšmės galima numanyti, kad mažesnės apimties tinklas persimokė. Tai galima išvelgti ir palyginus modelio validacijos ir apmokymo klaidos funkcijos reikšmes mokymo metu. Po 53 epochos apmokymo klaida priartėjo prie 0.



30 pav. AlexNet modifikacijų su 64, 128, 256 neuronų pilno jungimo sluoksniais mokymo klaidos funkcijos reikšmės

Pagal gautus rezultatus galime teigti, jog visi trys tinklai persimokė apmokymo duomenimis. Šiuo atveju, didesnės apimties tinklas persimokė greičiau, o 128 neuronų tinklas mokėsi stabiliausiai. Kadangi visi trys tinklai veikė labai panašiai, tolimesniems bandymams pasirenkamas 128, sumažinant apmokymo laiką lyginant su 256 neuronų tinklu ir išlaikant galimybę geriau išnaudoti konvoliucinių sluoksnių informaciją lyginant su 64 neuronų tinklu.

3.2.1. Optimizavimo algoritmai

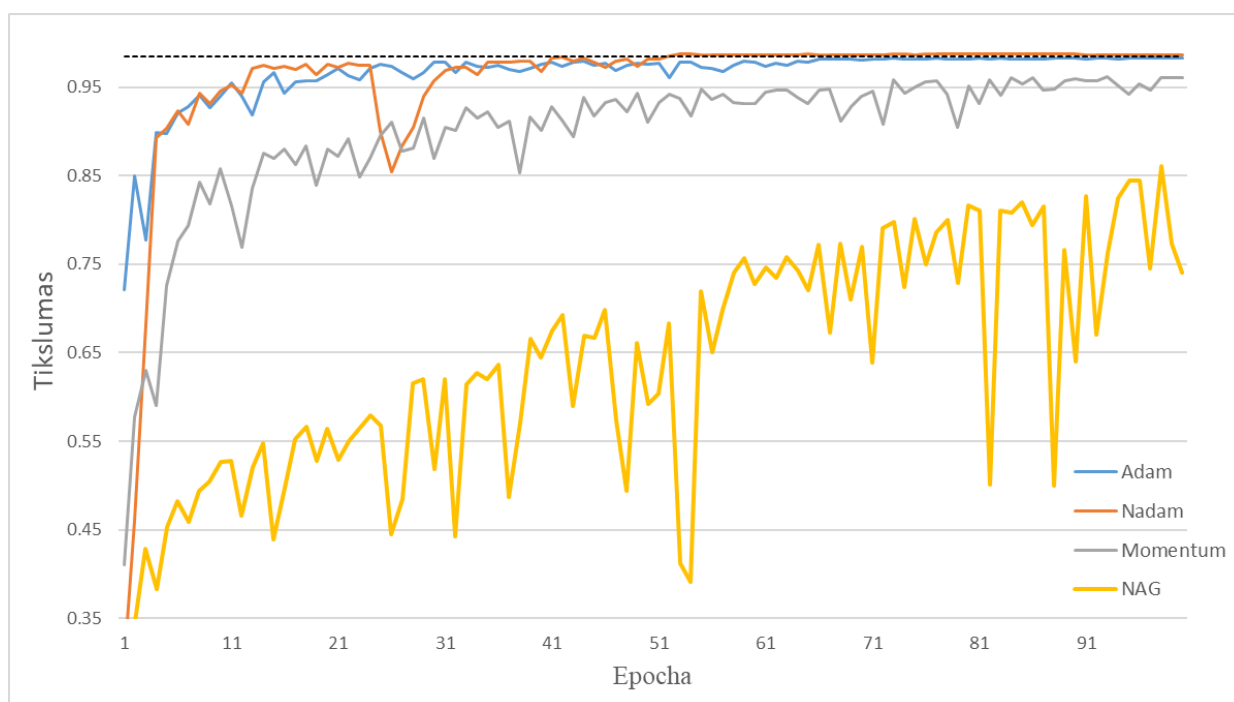
Tinklo apmokymo greičiui ir rezultatams svarbus tinkamos klaidos funkcijos optimizavimo algoritmo parinkimas. Bandymai buvo atliekami tarp 2.6 skyriuje aprašytų Adam, Nadam, NAG, Momentum algoritmų. Adam algoritmo rezultatai buvo panaudoti iš anksčiau atliktų tyrimų. Adam ir Nadam naudotos parametrų reikšmės - $\beta_1 = 0,9$, $\beta_2 = 0,999$, $\varepsilon = 10^{-8}$. Momentum parametras $m = 0,9$. Mokymosi greitis visiems algoritmams buvo taikomas $t = 0,0001$.

Tinklai su skirtingais optimizavimo algoritmais buvo apmokomi 100 epochų. Apmokymo greičiai pateikti 1 lentelėje.

1 lentelė. Optimizavimo algoritmų mokymų trukmė

Algoritmas	Trukmė (min.)
Adam	59,1
Nadam	58,5
NAG	51,15
Momentum	50,10

Bandymų metu gautos validacijos tikslumo reikšmės pateiktos 31 pav.



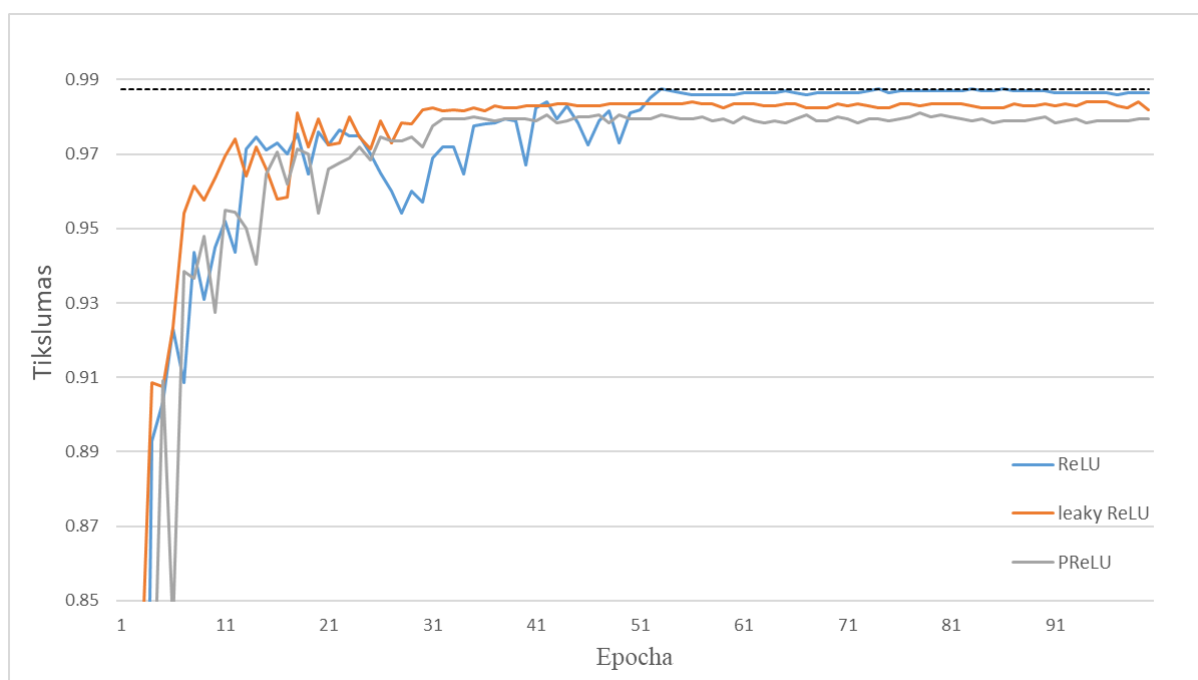
31 pav. Optimizavimo algoritmų palyginimo validacijos tikslumo reikšmės

31 pav. pavaizduotame grafike galime įžvelgti, kad NAG ir Momentum algoritmai apsimoko žymiai lėčiau ir per 100 epochų nepriartėjo prie Adam ir Nadam. Adam ir Nadam algoritmai apsimokymų greičiu buvo panašūs. Nepaisant tikslumo kritimo po maždaug 20 epochų, Nadam

algoritmą naudojantis tinklas pagerino iki šiol geriausią validacijos tikslumo reikšmę (98,75%). Tolimesniuose bandymuose bus naudojamas Nadam klaidos funkcijos optimizavimo algoritmas.

3.2.2. Aktyvacijos funkcijos

Šiuo metu ReLU aktyvacijos funkcija yra viena plačiausiai naudojamų bei pasiekianti gerus rezultatus, todėl bandymams buvo pasirinktos šios funkcijos modifikacijos leaky ReLU ir PReLU. Leaky ReLU aktyvacijos funkcijai buvo pasirinkta dažniausiai naudojama $\alpha = 0,1$ reikšmė, tuo tarpu, PReLU atveju ši reikšmė apmokoma atskirai kaip sluoksnio parametras. Bandymai buvo vykdomi tinklą mokant 100 epochų.



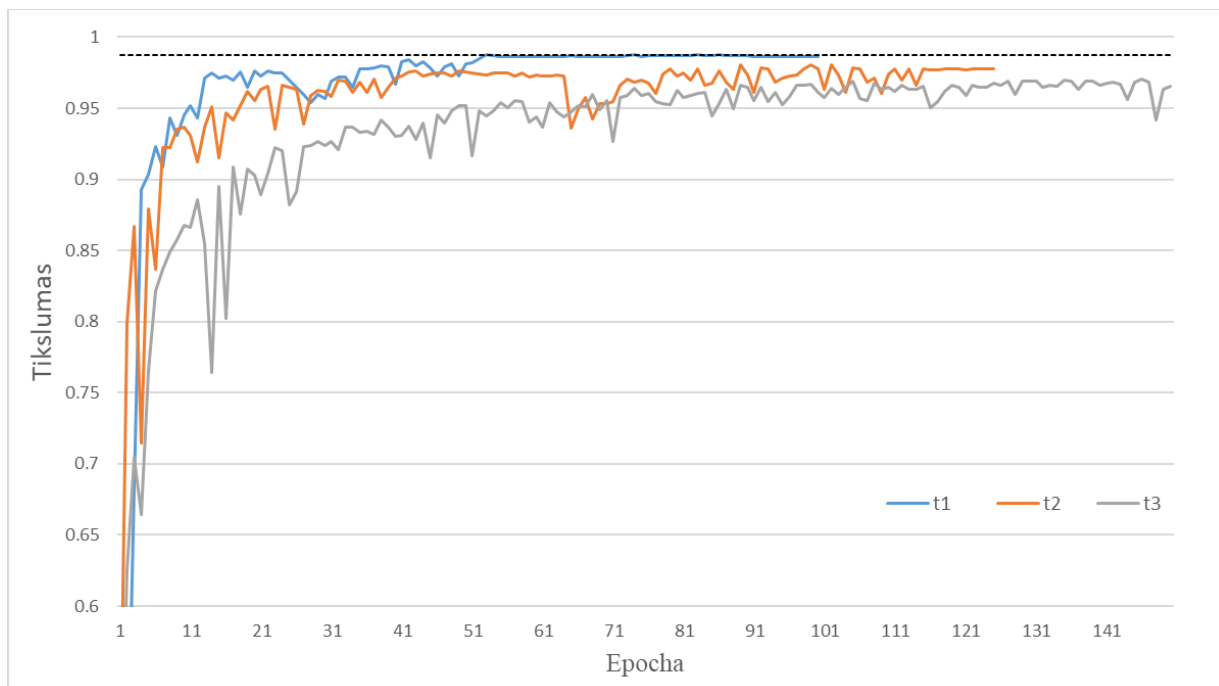
32 pav. ReLU, leaky ReLU, PReLU funkcijų apmokomų tinklų validacijos tikslumo reikšmės

32 pav. galime išvelgti, kad leaky ReLU funkcija apsimokė nežymiai greičiau nei likusios aktyvacijos funkcijos, tačiau tai nepadėjo pasiekti geresnių rezultatų. PReLU ir leaky ReLU pasiektas tikslumas buvo atitinkamai 98,4% ir 98,1%. PReLU atveju dėl papildomų parametų apmokymo prailgėjo mokymo laikas – 100 epochų apmokyti užtruko 1 val. 11 min. Tolimesniuose bandymuose ir toliau bus naudoja ReLU aktyvacijos funkcija.

3.2.3. Mokymosi greitis

3.2.2 skyrelyje buvo apžvelgti keli optimizavimo algoritmai ir pasirinktas Nadam tolimesniems tyrimams. Šiame skyriuje apžvelgiami bandymai su šio algoritmo mokymosi greičio parametru t , siekiant nustatyti kokią įtaką jis turi tinklo mokymo rezultatams. Iki šiol bandymai buvo atliekami su $t_1 = 0,0001$ parametro reikšme. Anksčiau gautus rezultatus palyginsime su $t_2 = 0,00005$ ir $t_3 =$

0,00001. Didesnių reikšmių nei t_1 detaliau nebus bandomos, kadangi su jomis tinklas nesimoko dėl per greitai keičiamų svorių reikšmių. Bandymai buvo atlikti tinklą mokant iki 150 epochų.



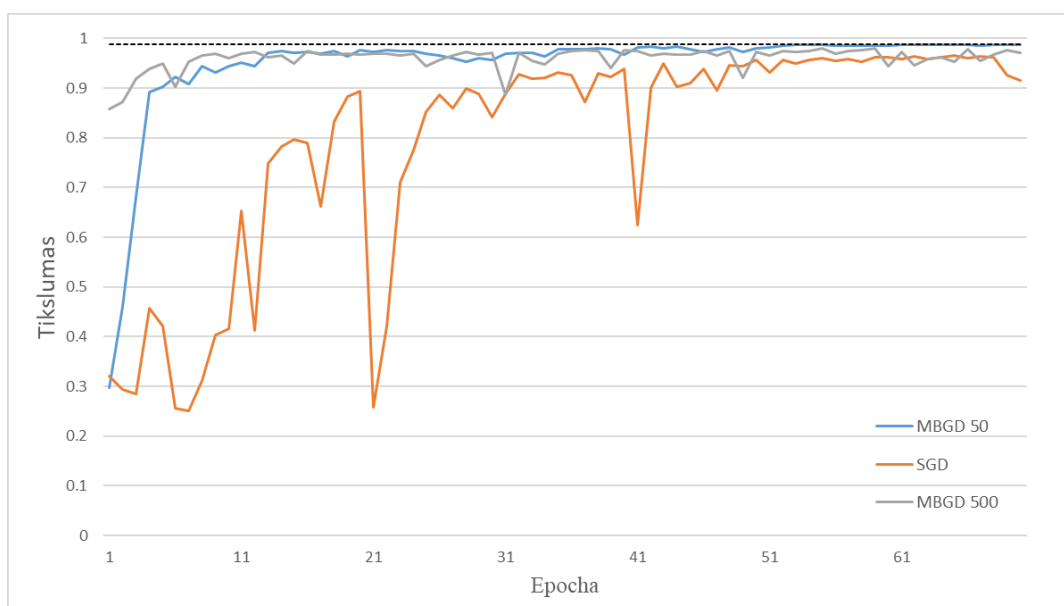
33 pav. Tinklų su skirtingais mokymo greičio parametrais validacijos tikslumo reikšmės

Žvelgiant į 33 pav., akivaizdu, kad mažinant tinklo mokymo greičio parametą tinklo persimokymas nusikeldavo vėliau. Naudojant t_2 parametą buvo stebimas ankstyvas persimokymas, po kurio sekdamo staigus tikslumo kritimas bei mokymasis iki kito persimokymo. Ši elgsena išliko ir atliekant pakartotinius bandymus. Nepaisant lėtesnio apsimokymo, gauti rezultatai neprilygo naudojant t_1 mokymo greitį.

3.2.4. Mokymo duomenų rinkinio dydis

Dažnai nagrinėjama tema mokant tinklą yra mokymo duomenų imties padalinimas į mokymo rinkinius (angl. *batch*). Jis nusako kiek duomenų iš mokymo imties bus panaudota vienam tinklo svorių atnaujinimui. Dažniausiai šių dydžių pasirinkimas skirstomas į kategorijas: rinkinio gradientinis nusileidimas (BGD) (angl. *batch gradient descent*), stochastinis gradientinis nusileidimas (SGD) (angl. *stochastic gradient descent*) ir minimalaus rinkinio gradientinis nusileidimas (MBGD) (angl. *mini-batch gradient descent*). Rinkinio gradientiniame nusileidime vaizdų skaičius po kurio atnaujinami svoriai yra viso mokymo rinkinio dydis, stochastinio – 1, o minimalaus rinkinio tarp viso mokymo rinkinio ir 1. Viso mokymo imties panaudojimas vienam svorių atnaujinimui yra pakankamai retas, kadangi tai reikalauja daug operatyvinės atminties. Šio parametro įtakai mokymo procesui patikrinti, buvo atlikti bandymai naudojant SGD bei MBGD, kurio atitinkamai pasirinkta 50 ir 500 vaizdų vienam svorių atnaujinimui. 50 vaizdų duomenys

paimami iš prieš tai atliktų bandymų. Apmokoma 70 epochų. SGD tinklui apmokyti prireikė 1 val. 53 min., MBGD su 50 vaizdų rinkiniu 55 min., MBGD su 500 vaizdų rinkiniu 43 min.

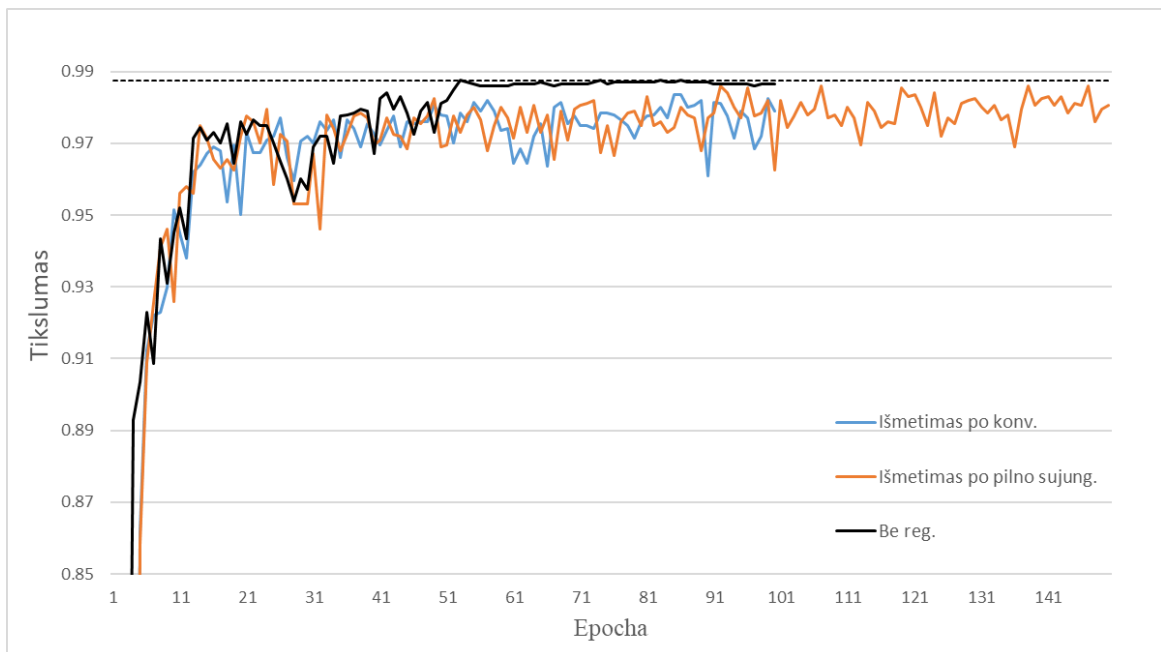


34 pav. Tinklų su skirtingais mokymo rinkinių dydžiais validacijos tikslumo reikšmės

34 pav. galima pastebėti, kad atnaujinant tinklo svorius su kiekvienu vaizdu mokymasis vyksta netolygiai, dažnai įvyksta tinklo kokybės kritimai. Taip įvyksta dėl netiksliai apskaičiuojamo gradiento. Didėjant duomenų rinkiniui, mokymo procesas tampa vis stabilesnis, tačiau prieš tai pasiekti rezultatai nebuvo pagerinti (SGD – 96,5% , MBGD su 500 vaizdų - 97,8%), todėl tolimesniuose bandymuose bus naudojama 50 vaizdų rinkinys.

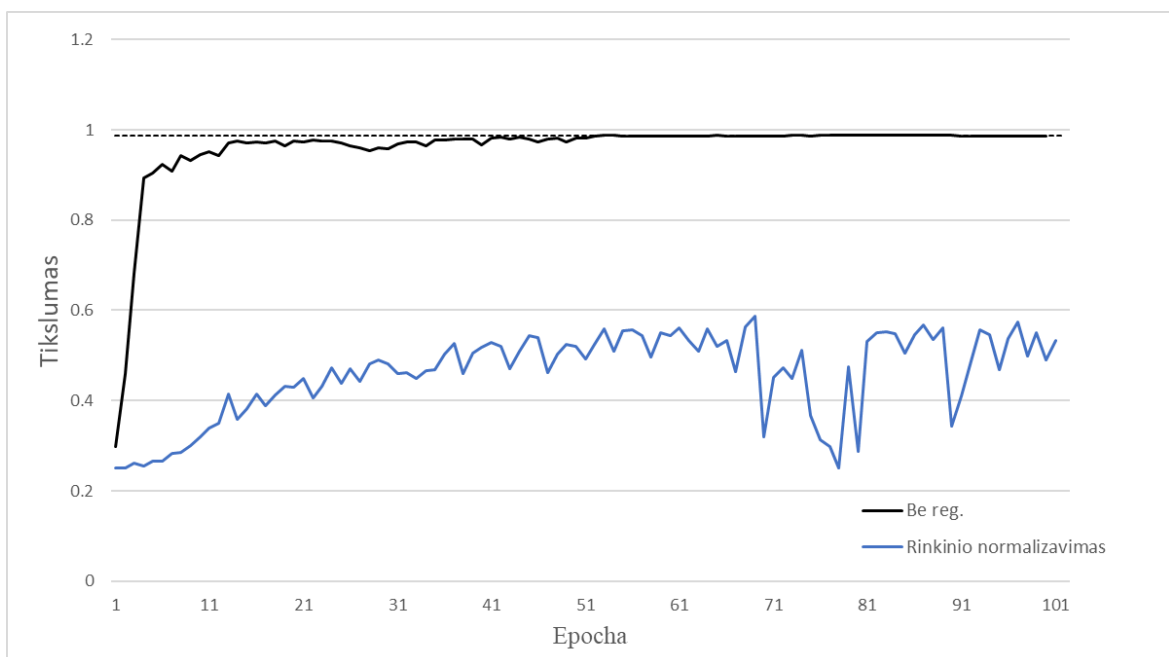
3.2.5. Reguliarizavimas

Maža pradinė mokymosi duomenų aibė padidina tinklo persimokymo galimybę. Šiai galimybei sumažinti panaudojamos reguliarizavimo technikos. Bandymuose buvo išbandomi du metodai: išmetimo ir rinkinio normalizavimo. Išmetimo metodas buvo išbandytas dviem būdais: pritaikant tik po konvoliucinių sluoksnių ir tik po pilnai sujungtų sluoksnių. Naudota neurono išmetimo iš mokymo iteracijos tikimybė buvo 0,2. Rinkinio normalizavimo metodas buvo taikomas po kiekvieno konvoliucinio sluoksnio prieš pritaikant aktyvacijos funkciją. Buvo atlikta atitinkamai nuo 100 iki 150 epochų. Išmetimo metodo rezultatai pateikti 35 pav.



35 pav. Tinklų su išmetimo metodais validacijos tikslumo reikšmės

Lyginant tinklų su išmetimo metodu mokymosi eigą su tinkle be šio metodo, akivaizdu, kad tinklai naudojantys išmetimo metodiką buvo mažiau linkę persimokyti. Visgi, buvo pastebimas nusistovėjimas tam tikrame tikslumo diapazone. Tinklas su pritaikymu po pilno sujungimo sluoksnio savo geriausią tikslumą 98,25% pasiekė per 55 epochas ir buvo sustabdytas ties 100 epochų, taip ir nepagerinus šio rezultato. Pritaikius metodą po konvoliucinių sluoksnių geriausias rezultatas buvo pasiektas 92 epochoje 98,6% ir mokymas sustabdytas ties 150 epochų.



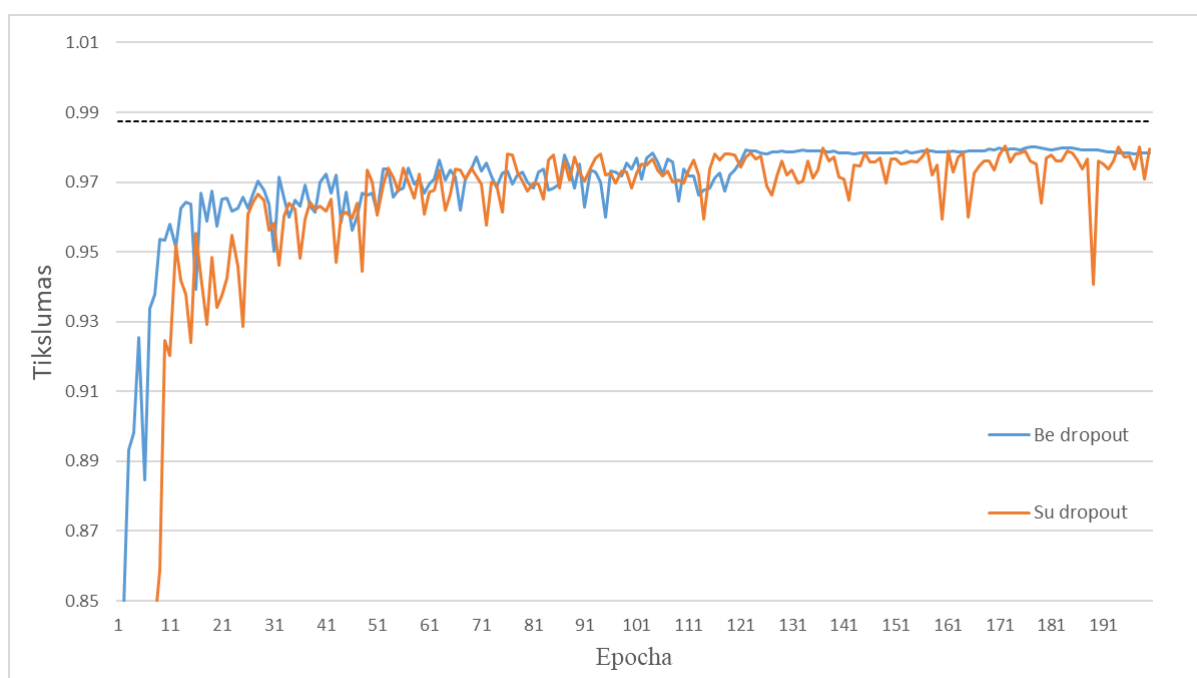
36 pav. Tinklų su rinkinio normalizavimo metodu validacijos tikslumo reikšmės

Rinkinio normalizavimo atveju tinklo apmokymas vyko labai lėtai ir nestabiliai. Buvo atlikti keli bandymai didinant mokymosi greitį, tačiau buvo gaunami panašūs rezultatai. Taip pat, naudojant

rinkinio normalizavimą išaugo mokymo laikas. Įvykdyti 100 epochų užtruko 1 val. 58 min. Pagal gautus rezultatus toliau bandymuose bus naudojamas tinklas be reguliarizavimo metodų.

3.2.6. Duomenų papildymas

Duomenų papildymas vyko naudojant du vaizdo apdorojimo metodus: vaizdo pasukimą ir veidrodinį vaizdo atspindį. Vaizdo pasukimas buvo atliekamas naudojant atsitiktinai parenkamą 90 arba 270 laipsnių pasukimo kampą. Atspindys buvo atliekamas atsitiktinai pasirinkus vertikalią arba horizontalią ašį. Duomenų rinkinyje vaizdų kiekis buvo padidintas 3 kartus iki 30000 vaizdų, atsižvelgiant į tai, jog naudojamos rezoliucijos vaizdų kiekis yra ribojamas išnaudojamos darbinės atminties. Tinklas buvo apmokomas 200 epochų. Pastebėjus, kad tinklas persimoko, buvo išbandytas tinklas su pridėtais išmetimo sluoksniais po konvoliucinių operacijų. Apmokymas atitinkamai užtruko 5 val. 34 min. tinklui be išmetimo operacijų ir 5 val. 41 min. tinklui su šia operacija.



37 pav. Tinklų, apmokyto su papildytais duomenų rinkiniais, validacijos tikslumo reikšmės

Tinklo be išmetimo operacijos aukščiausias pasiektas tikslumas 98% ir mokymasis vyko tik iki 120 epochos, po kurios tinklas buvo persimokęs mokymo duomenimis. Išmetimo operacija padėjo tinklui nepersimokėti, tačiau tai neleido pasiekti geresnių rezultatų. Šio tinklo pasiektas tikslumas – 98,03%. Taip pat, pasiekus šį rezultatą tikslumo ir klaidos parametruose atsirado staigūs kokybės kritimai.

3.2.7. Galutinė tinklo architektūra ir palyginimas

Architektūra, kurią naudojant buvo pasiekti geriausi rezultatai, pateikta 2 lentelėje. Tinklo optimizavimui naudojamas Nadam algoritmas su mokymo greičio parametru $t = 0,0001$. Sviurių

inicijavimui naudojamas He skirstinys. Tinklas apmokytas pradiniais 10000 vaizdų, svorius atnaujinant kas 50 vaizdų. Naudojant šią architektūrą buvo pasiektas 98,75% testavimo tikslumas.

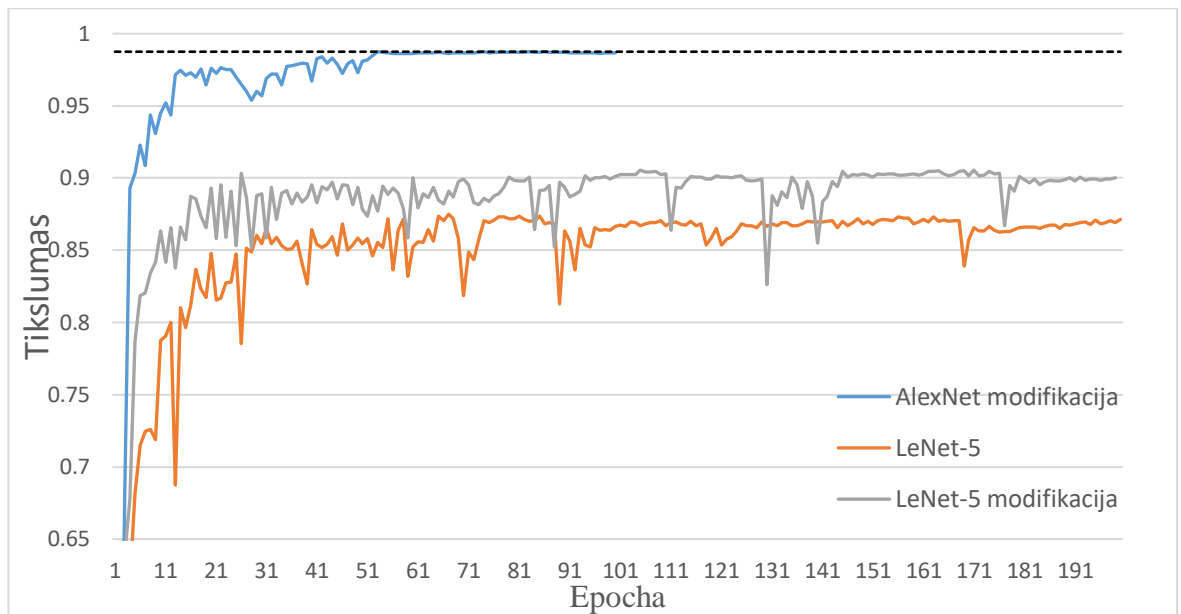
2 lentelė. Naudojama AlexNet modifikacijos galutinė architektūra

	Sluoksnis	Dydis	Gylis	Filtrai	Nulinis užpildas	Žingsnis	Aktyvacija
1	Įvedimas	224x224x1	1	-	-	-	-
2	Konvoliucijos	54x54x96	96	11x11	-	4	ReLU
3	Maks. telkimo	26x26x96	-	3x3	-	2	ReLU
4	Konvoliucijos	26x26x256	256	5x5	2	1	ReLU
5	Maks. telkimo	12x12x256	-	3x3	-	2	ReLU
6	2 × Konvoliucijos	12x12x384	384	3x3	1	1	ReLU
8	Konvoliucijos	12x12x256	256	3x3	1	1	ReLU
9	Maks. telkimo	5x5x256	-	3x3	-	2	ReLU
10	Pilno jungimo	128	-	-	-	-	ReLU
11	Pilno jungimo	128	-	-	-	-	ReLU
12	Pilno jungimo	4	-	-	-	-	SoftMax

Įvertinti tinklo veikimui ir uždavinio sudėtingumui buvo pasirinkta jį palyginti su sąlyginai paprastu konvoliuciniu neuroniniu tinklu, atitinkančiu LeNet-5 architektūrą. Ši tinklą sudaro 2 konvoliuciniai sluoksniai, kurių gylis yra atitinkamai 6 ir 16, naudojami filtrai 5×5 . Po konvoliucinių sluoksnių seka maksimalaus telkimo sluoksniai, kurių filtrų dydis yra 2×2 . Galiausiai, seka 3 pilno jungimo sluoksniai su atitinkamai 120, 84 ir 4 neuronais. Taip pat, LeNet-5 pritaikymuose dažnai padidinamas filtrų skaičius konvoliuciniuose sluoksniuose iki atitinkamai 32 ir 64 filtrų, todėl bus išbandoma ir ši modifikacija. Naudojamos ReLU aktyvacijos funkcijos, He skirstinio svorių inicijavimas ir Adam optimizavimo algoritmas, su mokymosi greičiu $t = 0,001$. Mokymo ir testavimo vaizdai buvo konvertuojami, kad atitiktų įvedimo dimensijas, kurios yra $32 \times 32 \times 1$.

3 lentelė. AlexNet tinklo modifikacijos rezultatų palyginimas su LeNet-5 architektūra

	AlexNet pritaikymas	LeNet-5	Praplėstas LeNet-5
Epochų skaičius	100	200	300
Mokymo laikas	59 min.	21 min.	34 min.
Mokymo tikslumas	98,75%	87,5%	90,3%



38 pav. AlexNet tinklo modifikacijos ir LeNet-5 tinklų rezultatų palyginimas

LeNet-5 architektūros tinklų mokymai buvo stabdomi kai jie persimokydavo ir bent 50 epochų nebe pagerindavo rezultatų. Akivaizdu, kad daugiau parametrų turinti LeNet-5 modifikacija su gilesniais konvoliuciniais sluoksniais iki persimokymo užtruko daugiau laiko. Nepaisant to, AlexNet architektūra pagrįstas tinklas pasiekė kur kas geresnį tikslumą.

3.2.8. Tinklo jautrumas foninio vaizdo kitimui

Iki šiol tinklas buvo apmokomas naudojant vaizdų rinkinius sugeneruotus pagal 4 foninius vaizdus. Norint patikrinti ar apmokytas tinklas yra jautrus foninio vaizdo kitimui buvo panaudotas vaizdų rinkinys sugeneruotas naudojant foninį vaizdą kuriuo tinklas nebuvo mokomas. Šį rinkinį sudaro 2000 vaizdų tolygiai pasiskirsčiusių tarp 4 klasių. Bandymai atlikti įvertinant pasiekiamą tinklo tikslumą. Vaizdų rinkinys išbandytas tarp tinklo, su kuriuo mokymo metu buvo pasiektas 98,75% tikslumas ir to paties tinklo apmokyto papildytais duomenimis, kuris pasiekė 98% tikslumą.

Įvertinant aukščiausią tikslumą pasiekusį tinklą duomenimis sugeneruotais panaudojant naują foną gautas 97,90% tikslumas. Tuo tarpu, naudojant tinklą, kurio tikslumas buvo žemesnis ir kuris buvo apmokomas su papildytais duomenimis, pasiektas 98,05% tikslumas. Akivaizdu, kad tiksliausiai pradiniais duomenimis apmokytas tinklas dalinai išmoko ir foninio vaizdo ypatybes. Tuo tarpu, ilgiau ir su didesnę duomenų imtimi apmokytas tinklas buvo nejautrus foninio vaizdo pasikeitimui ir net pagerino mokymo metu pasiektą validacijos tikslumą.

Išvados

1. Atlikta vaizdų klasifikavimo ir optinių paviršių tyrimo metodų apžvalga. Vaizdo klasifikavimo uždaviniui spręsti pasirinkti konvoliuciniai neuroniniai tinklai.
2. Realizuotas šešėlinio muaro vaizdų generavimo metodas. Naudojant pasirinktą elipsinio paraboloido paviršiaus funkciją sugeneruojamos šešėlinio muaro interferencinės juostos, kuriomis padengiamas foninis spausdintų elektroninių plokščių vaizdas. Sugeneruoti vaizdai skirstomi į 4 klases, atsižvelgiant į paviršiaus funkcijos aukščiausius ir žemiausius taškus. Taip sugeneruota 10000 vaizdų imtis tinklo mokymui ir testavimui.
3. Konvoliucinio neuroninio tinklo realizacijai pasirinktas AlexNet architektūros pagrindas. Ši architektūra pritaikyta sprendžiamam uždaviniui modifikuojant įvedimo ir išvedimo sluoksnius, pritaikant tinklo pilno jungimo sluoksnių gylį. Ištyrus skirtingo gylio sluoksnius pasirinkta naudoti 128 neuronų pilno jungimo sluoksnius.
4. Atliktas realizuoto tinklo optimizavimo algoritmų, aktyvacijų funkcijų, mokymosi greičio, duomenų rinkinio dydžio, reguliarizavimo metodų, duomenų papildymo tyrimas. Nustatyta, kad geriausi rezultatai buvo pasiekti naudojant Nadam optimizavimo algoritmą su mokymo greičio parametru 0,0001 bei ReLU aktyvacijos funkciją. Pastebėta, kad Adam ir Nadam optimizavimo algoritmai daro didelę teigiamą įtaką tinklo mokymosi greičiui. Reguliarizavimo ir duomenų papildymo metodai padėjo išvengti tinklo persimokymo, tačiau nebuvo pagerinti tikslumo rodikliai. Vaizdų klasifikavimo geriausias pasiektas tikslumas – 98,75%. Norint pasiekti didesnę tikslumą būtų tikslinga ryškiau apibrėžti klasių ribas, kadangi tiriamu atveju labai mažas pokytis deformacijos aukštyje galėjo nulemti priskiriamą klasę. Taip pat, pastebėta, kad tinklas apmokytas papildytais duomenimis buvo mažiau jautrus foninio vaizdo pasikeitimui ir pasiekė 98,05%, nors su testavimo imtimi tikslumas metu siekė 98%. Tuo tarpu mokymo imtimi mokyto tinklo tikslumas įvedus naują foninį vaizdą buvo 97,90%.

Literatūros sąrašas

1. ABIODUN, O.I. ir kt. State-of-the-art in artificial neural network applications: A survey. Iš *Heliyon* . 2018. t. 4, nr. 11, pp. 938. ISBN 2405-8440.
2. ABRAHAM, E. Moiré Pattern Detection using Wavelet Decomposition and Convolutional Neural Network. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)* . 2018. p. 1275–1279. Prieiga per: doi:10.1109/SSCI.2018.8628746.
3. ALI, S.H.R. Advanced Nanomeasuring Techniques for Surface Characterization. In *ISRN Optics* [interaktyvus]. 2012. [žiūrėta 2020-04-30]. Prieiga per: <https://www.hindawi.com/journals/isrn/2012/859353>.
4. ANASTASI, R.F. An Introduction to Moire Methods with Applications in Composite Materials. In [interaktyvus]. 1992. [žiūrėta 2020-04-15]. Prieiga per: <https://www.semanticscholar.org/paper/An-Introduction-to-Moire-Methods-with-Applications-Anastasi/f520371ddba40313a6ed7edeacf67e8970ac4c8c>.
5. ANTUNES, M. *Knowledge extraction from semi-structured sources*. 2018. Prieiga per: doi: 10.13140/RG.2.2.25430.70726.
6. BAUBLIENĖ, R. *Chaotinė vizualinė kriptografija* [interaktyvus]. Kauno technologijos universitetas, 2015. [žiūrėta 2020-04-08]. Prieiga per: <https://epubl.ktu.edu/object/elaba:8694185/index.html>.
7. BISHOP, C. *Pattern Recognition and Machine Learning* [interaktyvus]. New York: Springer-Verlag, 2006. ISBN 978-0-387-31073-2.
8. CARUANA, R. Multitask Learning. In *Machine Learning* . 1997. t. 28, nr. 1, pp. 41–75. Prieiga per: doi:10.1023/A:1007379606734 .
9. CONROY, M. - ARMSTRONG, J. A comparison of surface metrology techniques. Iš *Journal of Physics: Conference Series* . 2005. t. 13, pp. 458–465. Prieiga per: doi: 10.1088/1742-6596/13/1/106.
10. DEGROOT, P. Optical Metrology. In *digital Encyclopedia of Applied Physics* [interaktyvus]. American Cancer Society, 2004. [žiūrėta 2020-05-02]. ISBN 978-3-527-60043-4 Prieiga per: <https://onlinelibrary.wiley.com/doi/abs/10.1002/3527600434.eap638>.
11. DEY, A. Machine Learning Algorithms: A Review. Iš: *International Journal of Computer Science and Information Technologies*. 2016. t. 7, nr. 3 pp. 1174-1179. ISSN: 0975-9646.
12. DENG, Z. ir kt. Efficient kNN classification algorithm for big data. Iš *Neurocomputing* . 2016. Vol. 195, p. 143–148. Prieiga per: doi:10.1016/j.neucom.2015.08.112.
13. DEROSE, J. ir kt. Brief Introduction to Surface Metrology. [interaktyvus]. 2019. [žiūrėta 2020-04-28]. Prieiga per: <https://www.leica-microsystems.com/science-lab/brief-introduction-to-surface-metrology>.
14. DHILLON, A. - VERMA, G.K. Convolutional neural network: a review of models, methodologies and applications to object detection. In *Progress in Artificial Intelligence* [interaktyvus]. 2019. [žiūrėta 2020-05-03]. Prieiga per: <http://link.springer.com/10.1007/s13748-019-00203-0>.

15. DYSON, B.F. ir kt. *Materials Metrology and Standards for Structural Performance*. Springer Science & Business Media, 2012. 341 pp. ISBN 978-94-011-1264-2.
16. DOGO, E. ir kt. A Comparative Analysis of Gradient Descent-Based Optimization Algorithms on Convolutional Neural Networks. 2018. Prieiga per: doi:10.1109/CTEMS.2018.8769211.
17. DZEMYDA, G. ir kt. *Daugiamųjų duomenų vizualizavimo metodai: vadovėlis informatikos krypties doktorantams ir magistrantams* [interaktyvus]. Vilnius: Mokslo aidai [Matematikos ir informatikos institutas], 2008. ISBN 978-9986-680-42-0.
18. EITUTIS, A. *Matavimai biomechanikoje* [interaktyvus]. KTU Technologija, 2011. ISBN 978-609-02-0212-8.
19. GROOT, P. DE Coherence Scanning Interferometry. Iš LEACH, R.Sud. *Optical Measurement of Surface Topography* [interaktyvus]. Berlin, Heidelberg: Springer, 2011. pp. 187–208. [žiūrėta 2020-04-30]. ISBN 978-3-642-12012-1 Prieiga per: doi:10.1007/978-3-642-12012-1_9.
20. GUILD, J. The interference systems of crossed diffraction gratings: theory of moiré fringes. Iš Oxford at the Clarendon Press. 1956.
21. GUO, Y. ir kt. Solder Ball Connect (SBC) assemblies under thermal loading: I. Deformation measurement via moiré interferometry, and its interpretation. Iš *IBM Journal of Research and Development*. 1993. t. 37, nr. 5, pp. 635–648. Prieiga per: doi:10.1147/rd.375.0635.
22. HAN, B. Higher sensitivity moiré interferometry for micromechanics studies. Iš *Optical Engineering*. 1992. t. 31, nr. 7, pp. 1517–1526. Prieiga per: doi:10.1117/12.57681.
23. HE, K. ir kt. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. Iš *arXiv:1502.01852* [interaktyvus]. 2015. [žiūrėta 2020-05-23]. Prieiga per: <http://arxiv.org/abs/1502.01852>.
24. HUYNH, L.N. ir kt. D-Pruner: Filter-Based Pruning Method for Deep Convolutional Neural Network. Iš *Proceedings of the 2nd International Workshop on Embedded and Mobile Deep Learning - EMDL'18* [interaktyvus]. Munich, Germany: ACM Press, 2018. pp. 7–12. [žiūrėta 2020-05-06]. Prieiga per: <http://dl.acm.org/citation.cfm?doid=3212725.3212730>.
25. YAKURA, H. ir kt. Malware Analysis of Imaged Binary Samples by Convolutional Neural Network with Attention Mechanism. 2018. pp. 127–134. Prieiga per: doi:10.1145/3176258.3176335.
26. YANG, F. ir kt. Dynamic texture recognition by aggregating spatial and temporal features via ensemble SVMs. Iš *Neurocomputing*. 2016. t. 173, pp. 1310–1321. Prieiga per: doi:10.1016/j.neucom.2015.09.004.
27. IFJU, P.G. - HAN, B. Recent Applications of Moiré Interferometry. Iš *Experimental Mechanics*. 2010. t. 50, nr. 8, pp. 1129–1147. Prieiga per: doi:10.1007/s11340-010-9404-9.
28. YOSHIKAWA, T. *Handbook of Optical Metrology: Principles and Applications, Second Edition*. CRC Press, 2017. 744 pp. ISBN 978-1-351-83184-0.
29. JIANG, Y. ir kt. A Brief Review of Neural Networks Based Learning and Control and Their Applications for Robots. Iš *Complexity* [interaktyvus]. 2017. [žiūrėta 2020-05-03]. Prieiga per: <https://www.hindawi.com/journals/complexity/2017/1895897>.

30. JIANG, N. - WANG, L. A Novel Strategy for Quantum Image Steganography Based on Moire Pattern. Iš *International Journal of Theoretical Physics* . 2014. t. 54. Prieiga per: doi:10.1007/s10773-014-2294-3.
31. JORIO, A. - DRESSELHAUS, M.S. Nanostructured Materials: Metrology. In *Reference Module in Materials Science and Materials Engineering* [interaktyvus]. Elsevier, 2016. [žiūrėta 2020-04-28]. ISBN 978-0-12-803581-8 Prieiga per: <http://www.sciencedirect.com/science/article/pii/B9780128035818025911>.
32. KIM, H. ir kt. AUTOMATIC JUDGMENT OF SPINAL DEFORMITY BASED ON BACK PROPAGATION ON NEURAL NETWORK. Iš *International Journal of Innovative Computing, Information and Control*. 2006. 2(6). pp. 1271-1279.
33. KRIZHEVSKY, A. ir kt. ImageNet Classification with Deep Convolutional Neural Networks. Iš PEREIRA, F. ir kt. Sud. *Advances in Neural Information Processing Systems 25* [interaktyvus]. Curran Associates, Inc., 2012. pp. 1097–1105. [žiūrėta 2020-05-04]. Prieiga per: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>.
34. LEACH, R. Introduction to Surface Texture Measurement. Iš LEACH, R. Sud. *Optical Measurement of Surface Topography* [interaktyvus]. Berlin, Heidelberg: Springer, 2011. pp. 1–14. [žiūrėta 2020-04-30]. ISBN 978-3-642-12012-1 Prieiga per internetą: https://doi.org/10.1007/978-3-642-12012-1_1.
35. LI, F.C. - KISHEN, A. Deciphering dentin tissue biomechanics using digital moiré interferometry: A narrative review. Iš *Optics and Lasers in Engineering* . 2018. t. 107, pp. 273–280. Prieiga per: doi:10.1016/j.optlaseng.2018.03.032.
36. LI, K. ir kt. Global control of colored moiré pattern in layered optical structures. In *Optics Communications*. 2018. t. 414, pp. 154–159. Prieiga per: doi:10.1016/j.optcom.2018.01.013.
37. MCCORMICK, N. - LORD, J. Digital Image Correlation. Iš *Materials Today* . 2010. t. 13, nr. 12, pp. 52–54. Prieiga per: doi:10.1016/S1369-7021(10)70235-2.
38. MORA-GONZÁLEZ, M. ir kt. Image Processing for Optical Metrology. 2011. ISBN 978-953-307-907-3.
39. MUHAMEDYEV, R. Machine learning methods: An overview. Iš *CMNT* . 2015. t. 19, nr. 6 pp. 14–29.
40. MUÑOZ-RODRÍGUEZ, J.A. - RODRÍGUEZ-VERA, R. Image encryption based on moiré pattern performed by computational algorithms. Iš *Optics Communications*. 2004. t. 236, nr. 4, pp. 295–301. Prieiga per: doi:10.1016/j.optcom.2004.03.089.
41. NURWAUZIYAH, I. ir kt. Satellite Image Classification using Decision Tree, SVM and k-Nearest Neighbor. 2018. Prieiga per: https://www.researchgate.net/publication/326316293_Satellite_Image_Classification_using_Decision_Tree_SVM_and_k-Nearest_Neighbor.
42. PIRODDA, L. Shadow And Projection Moire Techniques For Absolute Or Relative Mapping Of Surface Shapes. Iš *Optical Engineering*. 1982. t. 21, nr. 4, pp. 214640. Prieiga per: doi:10.1117/12.7972959.

43. RATNAM, M.M. ir kt. Three-dimensional object classification using shadow moire and neural network. Iš *Optical Engineering*. 2001. t. 40, nr. 9, pp. 2036–2040. Prieiga per: doi:10.1117/1.1391260.
44. RITTER, R. Reflection Moire Methods For Plate Bending Studies. Iš *Optical Engineering*. 1982. t. 21, nr. 4, pp. 214663. Prieiga per: doi:10.1117/12.7972962.
45. ROSENBLATT, F. Perceptron Simulation Experiments. Iš *Proceedings of the IRE*. 1960. t. 48, nr. 3, pp. 301–309. Prieiga per: doi:10.1109/JRPROC.1960.287598.
46. ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. Iš *Psychological Review*. 1958. t. 65, nr. 6, pp. 386–408. Prieiga per: doi:10.1037/h0042519.
47. SAMUEL, A.L. Some Studies in Machine Learning Using the Game of Checkers. Iš *IBM Journal of Research and Development*. 1959. t. 3, nr. 3, pp. 210–229. Prieiga per: doi:10.1147/rd.33.0210.
48. SANDER, M. Optical measurement of partially specular surfaces by combining pattern projection and deflectometry techniques. Iš *ResearchGate* [interaktyvus]. 2015. [žiūrėta 2020-05-02]. Prieiga per: https://www.researchgate.net/publication/303444277_Optical_measurement_of_partially_specular_surfaces_by_combining_pattern_projection_and_deflectometry_techniques.
49. SCHMOLZE, D. ir kt. Advances in Microscopy Techniques. Iš *Archives of pathology & laboratory medicine*. 2011. t. 135, pp. 255–63. Prieiga per: doi:10.1043/1543-2165-135.2.255.
50. SCIAMMARELLA, C.A. The Moire Method--A Review. Iš *Experimental Mechanics*. 1982. t. 22, pp. 418-433. Prieiga per: doi:10.1007/BF02326823.
51. SHARPE, W.N. - JRSud. *Springer Handbook of Experimental Solid Mechanics* [interaktyvus]. Springer US, 2008. ISBN 978-0-387-26883-5.
52. SHEN, L. ir kt. A novel local manifold-ranking based K-NN for modeling the regression between bioactivity and molecular descriptors. Iš *Chemometrics and Intelligent Laboratory Systems*. 2016. t. 151, pp. 71–77. Prieiga per: doi:10.1016/j.chemolab.2015.12.005.
53. TIZIANI, H.J. Holographic Interferometry And Speckle Metrology: A Review Of The Present State. In *Industrial Applications of Laser Technology* [interaktyvus]. International Society for Optics and Photonics, 1983. pp. 2–10. [žiūrėta 2020-05-03]. Prieiga per: <https://www.spiedigitallibrary.org/conference-proceedings-of-spie/0398/0000/Holographic-Interferometry-And-Speckle-Metrology--A-Review-Of-The/10.1117/12.935348.short>.
54. VERMA, K. - SINGH, P. An Insight to Soft Computing based Defect Prediction Techniques in Software. Iš *International Journal of Modern Education and Computer Science*. 2015. t. 7, pp. 52–58. Prieiga per: doi:10.5815/ijmeccs.2015.09.07.
55. VORBURGER, T.V. ir kt. Comparison of optical and stylus methods for measurement of surface texture. Iš *The International Journal of Advanced Manufacturing Technology*. 2007. t. 33, nr. 1, pp. 110–118. Prieiga per: doi:10.1007/s00170-007-0953-8.
56. WALKER, C.A. A historical review of moiré interferometry. Iš *Experimental Mechanics*. 1994. t. 34, nr. 4, pp. 281–299. Prieiga per: doi:10.1007/BF02325143.

57. WANG, L. ir kt. Comparative analysis of image classification methods for automatic diagnosis of ophthalmic images. Iš *Scientific Reports*. 2017. t. 7, nr. 1, pp. 1–11. Prieiga per: doi:10.1109/CTEMS.2018.8769211.
58. WHITEHOUSE, D.J. Surface metrology. Iš *Measurement Science and Technology*. 1997. t. 8, nr. 9, pp. 955–972. Prieiga per: doi:10.1088/0957-0233/8/9/002.
59. WILLNER, S. Moiré topography - a method for school screening of scoliosis. Iš *Archives of Orthopaedic and Traumatic Surgery. Archiv Fur Orthopadische Und Unfall-Chirurgie*. 1979. t. 95, nr. 3, pp. 181–185. Prieiga per: doi:10.1007/bf00460117.
60. WOO, W.H. ir kt. Artificial neural network approach for moiré fringe center determination. Iš *Journal of Electronic Imaging*. 2015. t. 24, nr. 6, pp. 063021. Prieiga per: doi:10.1117/1.JEI.24.6.063021.
61. XU, B. ir kt. Empirical Evaluation of Rectified Activations in Convolutional Network. Iš *arXiv:1505.00853* [interaktyvus]. 2015. [žiūrėta 2020-05-15]. Prieiga per: <http://arxiv.org/abs/1505.00853>.
62. ZHANG, Y. *New Advances in Machine Learning*. BoD – Books on Demand, 2010. 375 pp. ISBN 978-953-307-034-6.
63. ZHANG, J. Gradient Descent based Optimization Algorithms for Deep Learning Models Training. Iš *arXiv:1903.03614* [interaktyvus]. 2019. [žiūrėta 2020-05-06]. Prieiga per: <http://arxiv.org/abs/1903.03614>.
64. A review on Machine Learning Techniques | Machine Learning | Statistical Classification. Iš *Scribd* [interaktyvus]. [žiūrėta 2020-04-17]. Prieiga per: <https://www.scribd.com/document/336172863/A-review-on-Machine-Learning-Techniques>.
65. Michelson - an overview | ScienceDirect Topics. [interaktyvus]. [žiūrėta 2020-04-30]. Prieiga per: <https://www.sciencedirect.com/topics/engineering/michelson>.