



Kauno technologijos universitetas

Elektros ir elektronikos fakultetas

**Neprižiūrimo dirbtinio intelekto autoenkoderinių modelių
taikymo objektų vizualinių defektų aptikimui tyrimas**

Baigiamasis magistro projektas

Vytautas Stakeliūnas

Projekto autorius

Doc. dr. Arūnas Lipnickas

Vadovas

Kaunas, 2022



Kauno technologijos universitetas

Elektros ir elektronikos fakultetas

Neprižiūrimo dirbtinio intelekto autoenkoderinių modelių taikymo objektų vizualinių defektų aptikimui tyrimas

Baigiamasis magistro projektas

Valdymo technologijos (6211EX014)

Vytautas Stakeliūnas

Projekto autorius

Doc. dr. Arūnas Lipnickas

Vadovas

Prof. habil. dr. Rimvydas Simutis

Recenzentas

Kaunas, 2022



Kauno technologijos universitetas

Elektros ir elektronikos fakultetas

Vytautas Stakeliūnas

Neprižiūrimo dirbtinio intelekto autoenkoderinių modelių taikymo objektų vizualinių defektų aptikimui tyrimas

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;
2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąrašė;
3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs (-usi);
4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Vytautas Stakeliūnas

Patvirtinta elektroniniu būdu

Stakeliūnas Vytautas. Neprižiūrimo dirbtinio intelekto autoenkoderinių modelių taikymo objektų vizualinių defektų aptikimui tyrimas. Magistro baigiamasis projektas. Vadovas doc. dr. Arūnas Lipnickas; Kauno technologijos universitetas, elektros ir elektronikos fakultetas.

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

Reikšminiai žodžiai: dirbtinis intelektas, neprižiūrimas mokymasis, autoenkoderiai, defektų aptikimas.

Kaunas, 2022. 50 p.

Santrauka

Šiuo metu inžinerijoje ir gamyboje kokybės įvertinimui, ieškant defektų, yra taikomi prižiūrimo mokymosi neuroninių tinklų algoritmai, kurie reikalauja ilgų darbo valandų atidžiai žymint sukauptus duomenis. Per pastaruosius keletą metų eksperimentinėje srityje anomalijoms objektuose aptikti vis daugiau dėmesio sulaukia neprižiūrimo mokymosi dirbtinio intelekto algoritmai, kurie aptinka defektus apsimokę be mokytojo. Vieni populiariausių tokių algoritmų - autoenkoderiai (AE) ir buvo analizuojami šiame darbe. Tyrimo tikslas - atrasti efektyviausią defektų objektuose tikrinimo AE modifikaciją, bei iširti skirtingų AE modifikacijų poveikumą rezultatų tikslumui. Pirmajame šio darbo skyriuje supažindinama su dažniausiai produktų kokybės inspekcijoje naudojamais gilaus neprižiūrimo mokymosi autoenkoderiniais modeliais ir analizuojamos šiuolaikinės defektams aptikti taikomos praktikos. Antrajame darbo skyriuje yra aprašoma pasirinkta duomenų bazė MVTEC AD, skirta būtent neprižiūrimo mokymosi algoritmams, randantiems broką objektuose tirti. Trečiajame skyriuje ieškant defektų medstraigčių, lazdyno riešuto ir stiklo butelio objektuose yra treniruojami ir ištestuojami skirtingų latentinės erdvės dydžių, architektūrų, nuostolių funkcijos autoenkoderiniai modeliai. Šių neuroninių tinklų anomalijų aptikimo rezultatai palyginami tiriamajame darbe naudojamais, nuo slenksčio nepriklausomais, kokybės įverčiais (AUROC, AUPRC ir AUIoU). Iš visų darbe tiriamų modelių aukščiausius AUPRC rezultatus segmentuojant broką medsriegyje parodė AE su didžiausiu kiekiu filtrų – 29,9%, segmentuojant broką lazdyno riešute ir stiklo butelyje parodė kontekstinis AE atitinkamai pasiekęs – 51,9% ir 41,4%. Taip pat pastebėta, jog skirtingiems objektams ir skirtingo tipo defektams aptikti neprižiūrimo mokymosi autoenkoderiniai algoritmai rekonstravo vaizdus nevienodai tiksliai. AE modeliai su papildomai reguliarizuojamomis klaidos funkcijomis bei kontekstinis AE efektyviau rasko anomalijas esant ryškaus kontrasto defektams ir aptakiuose objektuose, kai tuo tarpu $L2$ -tipo autoenkoderiai geriau aptikdavo anomalijas nesimetriško vaizdo nuotraukose, segmentuodami medstraigčių defektus.

Stakeliūnas Vytautas. Investigation of the unsupervised artificial intelligence autoencoder models application for the object's visual defects detection. Master's Final Degree Project. Supervisor Doc. Dr. Arūnas Lipnickas; Faculty of Electrical and Electronics Engineering, Kaunas University of Technology.

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

Keywords: artificial intelligence, unsupervised learning, autoencoders, anomaly detection.

Kaunas, 2022. 50.

Summary

Currently, artificial neural network algorithms based on supervised learning are used in the field of engineering and manufacturing. They demand long and attentive labor for labeling data. During the last few years, in the experimental field of anomaly detection, more and more attention is given for unsupervised learning algorithms. Of which one the most popular – autoencoder (AE) was analyzed in this work. The aim of this study is to find the most effective AE for defect inspection and to determine how different modifications affect detections accuracy. In the first part of this work a short review is given about the most often employed deep learning autoencoders for the purpose of quality inspection. The second part describes selected MVTec AD database used in the training of unsupervised learning AE networks for object defect segmentation. Finally, after training various autoencoders with latent space, architectural and loss function differences on screw, hazelnut and bottle data, a study on accuracy was conducted with threshold independent characteristics (AUROC, AUPRC and AUIoU). The highest AUPRC score for detecting anomalies in screws was shown by AE with extended filter count – 29,9%, in hazelnut and in bottle was shown by context AE accordingly reaching - 51,9% and 41,4%. It was also observed that for different objects and different anomaly types, autoencoder performances were not persistent. AE models with additional regularization in loss function as well as context AE segmented high-contrast defects and reconstructed curved-edge objects more accurately while $L2$ loss function models performed better on asymmetrical image scenes segmenting screw defects more accurately.

Turinys

Lentelių sąrašas	8
Paveikslų sąrašas	9
Santrumpų ir terminų sąrašas	10
Įvadas.....	11
1. Metodų apžvalga.....	14
1.1. Vaizdų analizės sistemos	14
1.2. Vaizdų analizės algoritmai defektams aptikti.....	14
1.3. Dirbtiniai neuroniniai tinklai ir gilusis mokymasis	15
1.3.1. Atgalinė sklaida	17
1.3.2. Aktyvacijos funkcijos	17
1.3.3. Konvoliuciniai neuroniniai tinklai.....	18
1.3.4. Paruoštų modelių naudojimas.....	20
1.4. Prižiūravimo ir neprižiūravimo mokymosi neuroninių tinklų taikymo ypatumai	21
1.5. Pagrindiniai AE neprižiūravimo mokymosi algoritmų tipai	22
1.5.1. Standartinis AE algoritmas.....	22
1.5.2. Variacinis AE algoritmas	23
1.5.3. Kontekstinis AE algoritmas.....	24
1.5.4. Suvaržytas AE algoritmas	25
1.5.5. GAN algoritmas.....	25
1.6. Segmentavimo technikos neprižiūravimo mokymosi algoritmuose	26
1.7. Kokybės įverčiai.....	27
1.8. Apibendrinimas	28
2. MVTec AD Duomenų bazė	29
2.1. Apie MVTec AD	29
2.2. Duomenų paruošimas	30
3. Nagrinėjamų neprižiūravimo mokymosi algoritmų metodinė ir eksperimentinė dalis.....	32
3.1. Tyrime naudota įranga.....	32
3.2. Autoenkoderinių modelių sukūrimas ir apmokymas.....	32
3.2.1. Bazinis AE.....	33
3.2.2. AE su modifikuota latentine erdve	35
3.2.3. Architektūriškai modifikuoti AE	35
3.2.4. AE su modifikuotomis klaidos funkcijomis	36
3.2.5. AE su duomenų pakeitimais	37
3.3. Eksperimentai	37
3.3.1. AE modifikacijų greitaveika.....	38
3.3.2. AE latentinės erdvės	39
3.3.3. AE architektūros	40
3.3.4. AE klaidos funkcijos	41
3.3.5. AE su duomenų pakeitimu	42
3.3.6. Autoenkoderiai neprižiūravimo mokymosi kontekste	42
3.3.7. Kiekybės įverčių palyginimas	44
3.3.8. AE veiksmingumas su įvairiomis defektų kategorijomis.....	45
Išvados ir rezultatai	48
Literatūros sąrašas	49

Priedai.....	51
1 priedas. Autoenkoderio su 5×5 filtru architektūra.....	51
2 priedas. Autoenkoderio su padidintu filtru skaičiumi architektūra	51
3 priedas. AE Denner'o pasiūlyta architektūra.....	52
4 priedas. AE modifikacijos, su padidintu filtru skaičiumi, rezultatai	52

Lentelių sąrašas

1 lentelė. Nebaigtinis sąrašas klasikinių metodų tekstūriniais ir spalviniams defektams aptikti [2]	14
2 lentelė. Treniravimui ir validavimui skirtų nuotraukų skaičius prieš ir po augmentavimo.	31
3 lentelė. Palyginamas šiame darbe sukurto ir analogiško [14] straipsnyje naudoto AE.	33
4 lentelė. Bazinė AE architektūra. Čia c - tai spalvos kanalų skaičius. Stiklo butelio ir medsaigčių klasės nuotraukos naudotos nespalvotos, dėl to $c=1$, lazdyno riešuto – spalvotos, tai - $c=3$. Sluoksniai 1-10 sudaro enkoderį, o $10^{\prime}-1^{\prime}$ sudaro dekoderį.....	34
6 lentelė. Modelių greitaveika. Čia greitaveika apskaičiuota kaip vidutinė vertė iš tūkstančio modelio rekonstruotų vaizdų, neuroniniam tinklui paduodant po vieną nuotrauką.	38
7 lentelė. AE latentinės erdvės modifikacijų rezultatai. Geriausiai pasirodęs modelis paryškintas.	39
8 lentelė. AE architektūros modifikacijų rezultatai. Geriausiai pasirodęs modelis paryškintas.	40
9 lentelė. AE modifikacijų su skirtingomis klaidos funkcijomis rezultatai. Geriausiai pasirodęs modelis yra paryškintas.	42
10 lentelė. AE su duomenų pakeitimu rezultatai. Geriausias rezultatas paryškintas.	42
11 lentelė. AUPRC rezultatų palyginimas (%)	43
12 lentelė. Trijų skirtingų autoenkoderių defektų aptikimo rezultatai kiekvienai atskirai anomalijų kategorijai. Geriausiai aptinkamos defektų kategorijos paryškintos žalia spalva, blogiausiai – raudona spalva.....	46

Paveikslų sąrašas

1 pav. Neprižiūrimo mokymosi algoritmo taikymas anomalijoms rasti formacijos pramonėje [11]	12
2 pav. Dirbtinio neurono struktūros atvaizdavimas	16
3 pav. Atgalinės sklaidos veikimo schema [9].....	17
4 pav. Tipinio konvoliucinio neuroninio tinklo struktūra	19
5 pav. maksimalaus sutelkimo operacija.....	20
6 pav. Tipinė AE struktūra	22
7 pav. Variacinio AE struktūra.....	23
8 pav. Kontekstinio AE struktūra.....	24
9 pav. Darbe naudojamų ir iš MVTEC AD duomenų bazės parsisiųstų, trijų skirtingų objektų nuotraukos	30
10 pav. Vaizdų pavyzdžiai naudojami kontekstinio AE apmokymui	37
11 pav. Butelio su užteršimu anomalijų žemėlapis su skirtingų latentinių erdvių AE modeliais	40
12 pav. Medsriegio galvutės deformacijos anomalijų žemėlapis su skirtingų architektūrų AE modeliais.....	41
13 pav. Lazdyno riešuto su skylė anomalijų žemėlapis taikant skirtingų klaidos funkcijų AE modelius	41
14 pav. Trijų, ROC, PER ir IoU, įverčių grafikai parodantys skirtingų modelių efektyvumą aptikti defektus su atskirų objektų nuotraukomis	45
15 pav. Trijų skirtingų AE modelių defektų atpažinimo rezultatai su pavyzdžiu iš palankiausiai aptinkamų defektų kategorijų	47

Santrumpų ir terminų sąrašas

Santrumpos:

DL – gilus mokymasis (angl. *deep learning*).

ANN – dirbtinis neuroninis tinklas (angl. *artificial neural network*).

CNN – konvoliucinis neuroninis tinklas (angl. *convolutional neural network*).

AE – autoenkoderinis tinklas (angl. *autoencoder*).

GAN – generacinis priešiškas tinklas (angl. *generative adversarial network*).

VAE – variacinis autoenkoderinis tinklas (angl. *variational autoencoder*).

AUROC – TPR ir FPR priklausomybės funkcijos integralas (angl. *area under receiver operating characteristic*).

AUPRC – PRC ir TPR priklausomybės funkcijos integralas (angl. *area under precision recall curve*).

HSV – tai spalvinė koduotė, nusakanti: spalvinį toną, sodrumą ir ryškumą (angl. *hue saturation value*).

RGB – tai spalvinė koduotė, nusakanti: raudonos, žalios ir mėlynos spalvos vertes (angl. *red green blue*).

Terminai:

Duomenų augmentacija (angl. *data augmentation*) – tai procesas, leidžiantis ženkliai padidinti duomenų įvairovę, naudojamą neuroninio tinklo mokymuisi, neįvedant naujos informacijos.

Persimokymas (angl. *over-fitting*) – tai neuroninių tinklų mokymosi klaida, kai sistema remiasi per daug glaudžiai esama duomenų baze, todėl negeba suformuoti teisingų generalizacijos savybių.

Kernelis, šablonas, filtras (angl. *kernel* arba *filter*) – skaliarinės formos svorių matrica, naudojama konvoliucinių neuroninių tinklų ypatybėms išgauti.

Konvoliucija, sąsuka (angl. *convolution*) – matematinė operacija - įvesties ir filtro matricų sandaugos integralas.

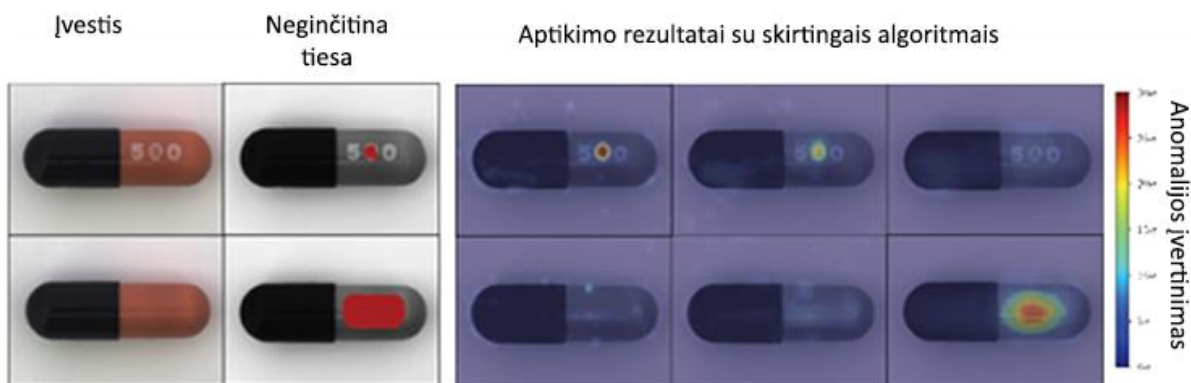
Įvadas

Vaizdo analizės sistema – tai tam tikra technologinė struktūra su informacijos apdorojimo metodais, leidžianti kompiuteriams ne tik manipuluoti vaizdais, kaip aibe duomenų, bet ir žmogui priimtinu būdu juos interpretuoti. Kompiuterinė rega plačiai taikoma gamyboje, nes tai padeda automatizuoti ir akivaizdžiai pagerinti procesą tuose sektoriuose, kuriuose reikia vizualiai atlikti įvairius išmatavimus, kokybės nustatymus ar kiekio įvardijimus. Vis dažniau vaizdų analizės sistemose yra panaudojami gilaus mokymosi (DL) algoritmai ir pagal gaunamus rezultatus tai yra vienas iš perspektyvesnių metodų. DL algoritmų potencialas yra ypač matomas produktų vaizdo kokybės įvertinimo sistemose. Tokios sistemos geba paimti neapdorotą informaciją iš kameros skaliarinėje matricos formoje ir atlikusios didžiulę seriją paralelių ir nuoseklių skaičiavimų nuspręsti, kokia yra objekto kokybė. Anksčiau tokie algoritmai buvo neįgyvendinami dėl milžiniško kiekio skaičiavimų, reikalingų sprendžiant funkcijas tarp dirbtinių neuronų įėjimų ir išėjimų kiekviename neuroninio tinklo sluoksnyje. Tačiau pastaraisiais dešimtmečiais sparčiai tobulėjant vaizdo plokščių pajėgumams bei dėl galimybės atlikti skaičiavimus šiuose procesoriuose paraleliai, gilaus mokymosi algoritmai tapo iš tiesų populiarūs ir susilaukia vis daugiau dėmesio.

Paradoksalu, jog tokia paprastame uždavinyje, kaip produkto kokybės įvertinimas, pamačius dar nesutiktą ar unikalų defektą, žmogus gana sėkmingai ir per trumpą laiką geba intuityviai nuspręsti, ar tai yra brokas, tačiau mašinoms ir kompiuteriams tai yra neįkandama ir slegianti užduotis. Iki šiol šiam uždaviniui atlikti kompiuterinėse regos sistemose buvo naudojami prižiūravimo gilaus mokymosi algoritmai. Nors jų pasiekiami rezultatai yra aukšti, tačiau šie algoritmai turi ir nemažai trūkumų. Prižiūravimo mokymosi dirbtiniams neuroniniams tinklams apmokyti reikia labai didelio kiekio sužymėtų duomenų, kas yra daroma rankiniu būdu - kruopščiai suskirstant nuotraukos regionus į atitinkamas klases. Kita problema atsiranda tuomet, kai turimoje net ir didelėje duomenų bazėje nepakanka tam tikro tipo defektų pavyzdžių imties ar iš viso jų nėra. Dėl šių priežasčių - bandant sukurti naujas ir modifikuoti esamas neprižiūravimo mokymosi struktūras ir taip pagerinti aukščiausią esamą rezultatą – sparčiai auga atliekamų tyrimų ir eksperimentų skaičius.

Neprižiūravimo mokymosi algoritmai, aptinkant prastos produkcijos gaminius ir atliekant jų klasifikaciją, remiasi dar nematytų duomenų, ženkliai besiskiriančių nuo apmokymo metu naudojamų, priskyrimo defektams principu. Vieni populiariausių neprižiūravimo mokymosi algoritmų modelių yra GAN – generuojantys-besivaržantys tinklai, AE – autoenkoderiniai tinklai, AnoGAN – autoenkoderinių ir generuojančių-besivaržančių tinklų hibridai. Visi jie naudoja pikselio tikslumo rekonstruotą nuotrauką ir įvairiomis matematinėmis lygtimis lygina ją su originalia fotografija, taip sprenddami, ar kiekvienas pikselis arba pikselių regionas priklauso anomalijai ar ne.

Šiame darbe yra taikomi autoenkoderiniai modeliai, nes būtent jie per paskutinius keletą metų nustatant gaminių anomalijas rodė aukščiausius rezultatus. Modernios vaizdų atpažinimo sistemos, kuri naudoja neprižiūravimo mokymąsi, sukūrimas yra aktualus tiems uždaviniams, kai neįmanoma skirti daug laiko žymėjimui ar tiesiog neužtenka resursų sukaupti išsamiai, turinčiai platų spektrą produkto nuotraukų su defektais, duomenų bazei. Efektyvus neprižiūravimo mokymosi algoritmas panaikintų žmogiškųjų išteklių, kai reikia žymėti duomenis, poreikį bei išspręstų kokybės inspekcijos problemą, kai pavyzdžių su anomalijomis trūksta.



1 pav. Neprižiūrimo mokymosi algoritmo taikymas anomalijoms rasti formacijos pramonėje [11]

Įprastai neprižiūrimo mokymosi dirbtiniai neuroniniai tinklai gali susidurti su aptikimo ir segmentavimo problemomis kai susidaro viena iš dviejų aplinkybių. Pirmoji aplinkybė lemia tai, jog tinklas būna per daug jautrus ir bet koks natūralus nuokrypis nuo apmokytų duomenų priverčia modelį manyti, jog tai yra anomalija. Antroji aplinkybė sąlygoja, kad neuroninis tinklas būna per mažai jautrus, todėl esant net ir dideliame nukrypimui nuo produkto standarto modelis mano, jog tai yra kokybiškas gaminys.

Šiuo metu vaizdo analizės algoritmų kūrėjai susiduria su iššūkiu – išmokyti dirbtinius neuroninius tinklus net ir susidūrus su ribiniais atvejais gebėti tiksliai įvertinti produkto kokybę. Nors šioje srityje per pastaruosius kelerius metus yra įvykę nemažai technologinių proveržių, tačiau mokslininkai bei tyrėjai ir toliau ieško optimalių sprendimų, kurie galėtų tapti galimybėmis, orientuotomis ne tik į dabarties perspektyvas bei inovacijų taikymą gamybos sektoriuje, bet ir į novatoriškas ateities technologijas, pažangius pokyčius inžinerijos praktiniame lauke.

Šio darbo aktualumas reiškiasi autoenkoderinių algoritmų defektoskopijos procese palyginimu, kas suteikia daugiau informacijos apie galimus neprižiūrimo mokymosi vaizdo analizės sprendimus ateities uždaviniams spręsti.

Tyrimo objektas: autoenkoderinių algoritmų, taikomų produktų kokybės užtikrinime, efektyvumas.

Darbo tikslas: ištirti neprižiūrimo mokymosi giliųjų dirbtinių neuroninių tinklų algoritmų, padedančių aptikti anomalijas produktų fotografijose, veiksmingumą.

Darbo uždaviniai:

- Išanalizuoti veikimo principus ir išskirti aukščiausius rezultatus rodančius neprižiūrimo mokymosi kompiuterinių regos sistemų algoritmus;
- Eksperimentiškai išnagrinėti autoenkoderio latentinės erdvės dydžio įtaką defektų segmentavimo tikslumui nuotraukoje;
- Palyginti anomalijų aptikimą tarp skirtingų architektūrų autoenkoderių;
- Nustatyti neprižiūrimo mokymosi autoenkoderinių modelių veiksmingumą naudojant $L1$, $L2$ bei kitas, labiau reguliuojamas, klaidos funkcijas;

- Įvertinti ar duomenų užtušavimo technika turi teigiamą įtaką autoenkoderio efektyvumui segmentuojant anomalijas;
- Išsiaiškinti AUPRC, AUIoU ir AUROC kokybės įverčių naudojimo tinkamumą neprižiūrimo mokymosi defektų radimo algoritmų tikslumui matuoti.

Gaminių nuotraukos paimtos iš *MVTec AD* - specialiai tokio tipo algoritmų testavimui sukurtos duomenų bazės.

Dirbtinių neuroninių tinklų apmokymui buvo naudojamos *Tesla K80*, *Tesla P100-PCIE* ir *Tesla T4* vaizdo plokštės ir virš 25 GB operatyvios atminties.

Gilaus mokymosi architektūrų sudarymui, treniravimui ir testavimui buvo naudota *Python* programavimo kalba, kuri yra ypatingai patogi savo skaitomumu ir palaiko vieną populiariausių mašininio mokymo bibliotekų *Tensorflow*. Programa buvo parašyta *Visual Studio 2022* integruota kūrimo aplinka ir *Google Colab* sąsiuviniu naršyklėje.

1. Metodų apžvalga

1.1. Vaizdų analizės sistemos

Vaizdas kompiuteriniams įrenginiams yra labai svarbi priemonė, padedanti gauti informaciją apie išorinį pasaulį. Yra nesuskaičiuojamas kiekis objektų atpažinimo, klasifikavimo, lokalizavimo ir įvertinimo darbų, kuriuos anksčiau atlikdavo operatorius – specialistas, nustatantis defektus gaminiuose, bet sparčiai vystantis technologijoms, atrankai reikalingi sprendimai buvo automatizuojami pasitelkiant kompiuterinės regos sistemas. Žmogus negali dirbti ilgų darbo valandų, yra veikiamas nuovargio, blaškomas išorinių sąlygų, priklausomas nuo turimų kompetencijų, kas veikia jo darbo kokybę bei rezultatus. Keičiant žmogaus atliekamą darbą į kompiuterizuotą seriją veiksmų yra siekiama sutaupyti laiko ir pinigų kaštų, minimalizuojant paklaidos procentą padaryti procesą dar efektyvesnį. Tokios sistemos principas yra paprastas – gauti duomenis iš vienos ar daugiau kamerų ir atliekant įvairias matematinės funkcijas, juos interpretuoti taip, kad būtų išgaunama prasminga užsibrėžtos paskirties informacija.

Skaitmeninė rega taikoma įvairiose srityse. Medicinoje vaizdo analizės sistemos yra naudojamos ligoms diagnozuoti, gamyklose tokios sistemos užtikrina produktų kokybę, duomenų analitikoje jos rūšiuoja informaciją į skirtingas grupes, miesto kelių infrastruktūroje stebi eismą bei pėsčiuosius ir garantuoja saugumą.

1.2. Vaizdų analizės algoritmai defektams aptikti

Viena dažniausių kompiuterinės regos užduočių yra defektų, turimoje produktų nuotraukoje, aptikimas. Tai gali būti defekto radimas tekstilės, medienos plokščių ar kitos struktūros paviršiuje, taip pat produkto anomalijų ar kokybės neatitikimo identifikavimas gamybos linijoje.

1 lentelė. Nebaigtinis sąrašas klasikinių metodų tekstūriniams ir spalviniams defektams aptikti [2]

Metodo būdas	Metodas
Statistinis	Histogramos savybių
	Ryšių matricos
	Regiono binarinės struktūros
	Pilkumo lygio statistikos
	Autokoreliacijos
	Vaizdų registravimo
Struktūrinis	Primityvaus matavimo
	Kraštų savybių
	Kraštų skeleto
	Morfologinių operacijų
Filtrais grindžiamas	Erdvės filtravimo
	Dažnio filtravimo
	Erdvės ir dažnio filtravimų
Modeliu grindžiamas	Fraktalų matematinio modelio
	Atsitiktinės erdvės modelio

Kompiuterinės sistemos vizualiai gamybos inspekcijai atlikti buvo pradėtos naudoti jau XX amžiaus pabaigoje. R. W. Connors [1] 1983 metų pranešime rašo apie paprastų vaizdo apdorojimo metodų, kaip RGB ir HSV spalvinės koduotės histogramų apskaičiavimo, ryšių matricių ar tiesiog ribinių pikselių intensyvumo verčių nustatymo, panaudojimą medienos paviršiaus defektams klasifikuoti.

Paprastos klasikinės - statistinės, morfologinės ir filtruojančios - funkcijos gali būti tinkamas sprendimas anomalijoms rasti, kai aplinkos sąlygos yra gerai kontroliuojamos, kai yra aiškiai žinomas galimų defektų spektras ir gaminio forma bei spalva smarkiai nevarijuoja. Tačiau tokias idealias sąlygas realiose pramoninėse sistemose sukurti yra labai sunku. Konvejeriuose dažnai pasitaiko įbrėžimų, dulkių ar tiesiog svetimkūnių, produktų defektai nenuspėjamai varijuoja, o ir objekto pozicija nuotraukoje yra ne visada kontroliuojama. Klasikiniai metodai (žr. 1 lentelę) dažniausiai išsprendžia klasifikavimo uždavinius, tačiau segmentavimo užduotys lieka neišspręstos. Kad būtų galima identifikuoti ir segmentuoti įvairias anomalijas kokybės įvertinimo uždaviniuose prieš daugiau nei dešimtmetį buvo pradėti taikyti dirbtiniai neuroniniai tinklai, netrukus atsirado ir gilieji dirbtiniai neuroniniai tinklai, kurie savo struktūra leido išspręsti net ir komplikuotas realių aplinkybių sukurtas situacijas.

1.3. Dirbtiniai neuroniniai tinklai ir gilusis mokymasis

Dirbtiniai neuroniniai tinklai (angl. *artificial neural network* - ANN), nors ir buvo pradėti tyrinėti XX amžiaus antroje pusėje, ilgą laiką nesusilaukė pakankamai dėmesio. Tai lėmė prastas kompiuterių skaičiavimų pajėgumas, didelių duomenų bazių stygius bei žinių apie efektyvias dirbtines neuronines jungtis stoka, dėl ko buvo susiduriama su problema, kai tikslo funkcija įstrigdavo prastame lokaliame minimume ar buvo susiduriama su nykstančio gradiento problema. Dėl šių priežasčių 10-ajame dešimtmetyje bei XXI amžiaus pradžioje ANN algoritmai buvo ignoruojami, nes vaizdo ir garso atpažinimo srityse dirbantys mokslininkai vengė juos taikyti.

Susidomėjimas buvo atgaivintas 2006 metais, kai mokslininkas G. Hinton su komanda pristatė algoritmą apie dirbtinio neuroninio tinklo modelio panaudojimą, kaip pradinį variantą gilesniam modeliui mokytis taikant DBN (angl. *deep belief networks*) [5]. Šis metodas davė pradžią gilesnių ANN modelių kūrimui vis pridėdant po vieną paslėptąjį sluoksnį ir apmokant jį pagal *Suvaržytos Bolcmano Mašinos* principus (angl. *restricted boltzmann machine* - RBM), ir vėliau atliekant tikslesnį derinimą atgalinės sklaidos metodu. Tokiu būdu, taikant DBN ir kuriant vis gilesnius ANN modelius, gilieji neuroniniai tinklai (angl. *deep neural networks*) buvo išpopuliarinti.

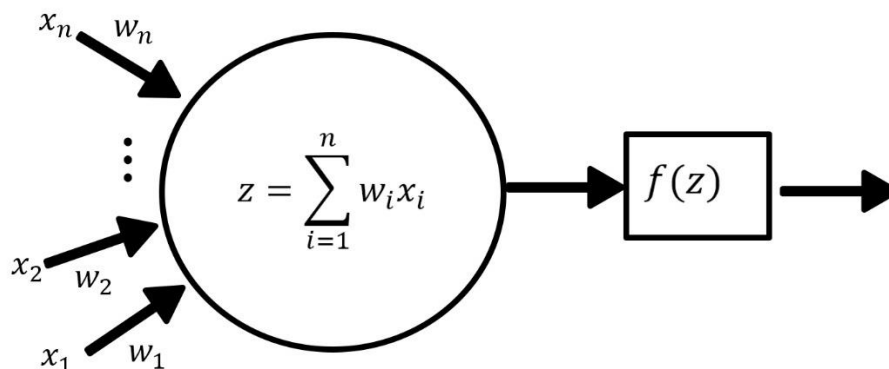
Per paskutinį dešimtmetį vaizdo plokščių pajėgumai toliau augo, vis labiau buvo skaitmenizuojama informacija, atsirado *didieji duomenys* (angl. *big data*), visa tai lėmė didelį atradimų pagreitį ANN algoritmų srityje.

Matematiškai analizuojant ANN - tai tam tikra tarpusavyje surištų skaičiavimų seka, gebanti iš įėjimų matricos gražinti išėjimą skaliarine forma, kuris atitinkamai pagal užduotį reprezentuotų reikšmingą informaciją. Tokie neuroniniai tinklai laikomi dirbtiniu intelektu dėl jų gebėjimo mokytis, tai yra treniravimosi metu keisti vidinius parametrus be žmogaus pagalbos.

ANN savo struktūra ir funkcionalumu primena biologinius neuronus. Kaip ir nerviniai raizginiai smegenyse, dirbtiniai neuronai geba sudaryti skirtingo stiprumo ryšius tarpusavyje. Dendritą galima tapatinti su įvestimi, pačią nervinę ląstelę - su aktyvacijos funkcija, aksoną - išvesčiai, o sinapses - jungtims tarp vieno neurono išėjimo ir kito įėjimo. Smegenyse neuronų siunčiamo signalo stiprumas

sinapsėmis varijuoja, o toks skirtumas dirbtinių neuronų tinkle pasimato per svorių parametrus. Tam kad ryšiai įgautų netiesiškumo savybę, įvesties ir svorių sandauga yra perleidžiama per *sigmoid*, *tanh*, *relu* ar kitokią aktyvacijos funkciją [6].

ANN struktūrų modifikacijų yra daugybė. Viena paprasčiausių ir labiausiai paplitusių yra tiesioginio sklaidimo tinklas (angl. *feed-forward*) arba kitaip - informaciją pirmyn skleidžiantis tinklas. Vienas iš pagrindinių jo principų yra formuoti neuronus taip, kad tinkle nesusidarytų ciklai, tai yra sluoksnio neuronai yra sujungiami su prieš tai buvusio sluoksnio neuronais, su tolimesnio sluoksnio neuronais, bet nėra sujungiami tame pačiame sluoksnyje.



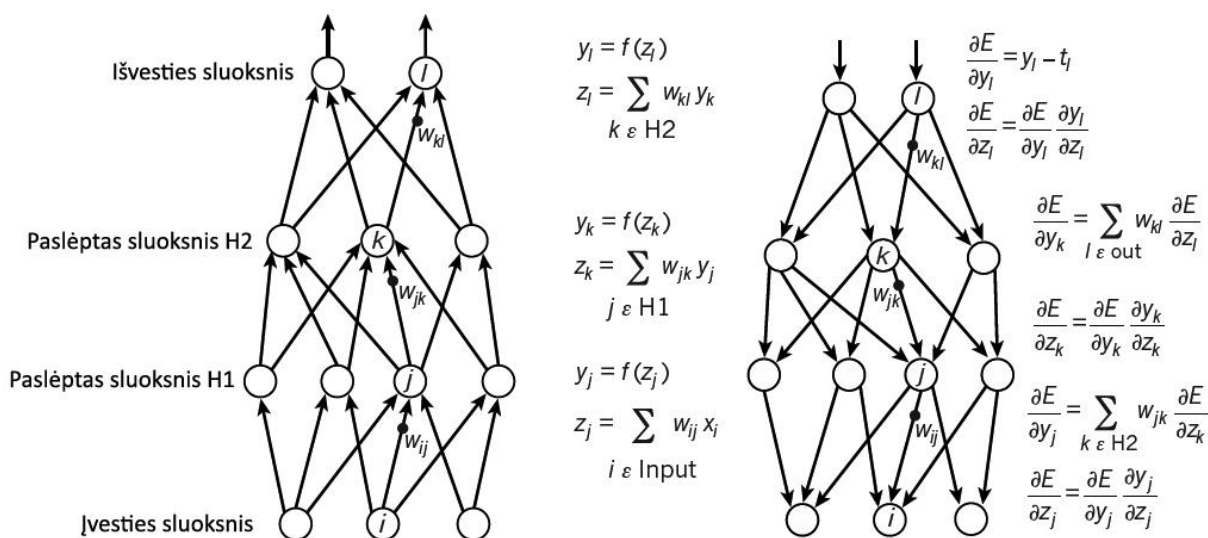
2 pav. Dirbtinio neurono struktūros atvaizdavimas

Gilias mokymosi (DL) modeliai yra poaibė ANN tinklų – tai tokie dirbtiniai neuroniniai tinklai, kurie turi daugiau nei kelis paslėptus sluoksnius, tai yra eksponentiškai didesnę kiekį nuoseklių apdorojimo skaičiavimų. Gilieji neuroninių tinklų algoritmai drastiškai pakeitė tradicines metodikas kalbos atpažinimo, vaizdo analizės, objektų aptikimo, medicinos diagnostikos ir kitose srityse bei tapo inovacijų dėmesio centru. DL modelis susideda iš daugybės sluoksnių, apdorojančių informaciją, kuomet kiekvienas jų geba išgauti prasmingas skirtingo abstrakcijos lygio savybes. Vaizdų analizės sistemose nuotraukos savybių ištraukimas po kiekvieno neuronų sluoksnio galimai atrodo taip: aptinkami kraštai ir staigūs spalviniai slenksčiai, vėliau formuojasi vaizdiniai motyvai, dar gilesniuose sluoksniuose šios savybės susijungia į dalis ir objektas yra nustatomas. DL algoritmai įdiegiami įvairiose technologinėse sistemose, nes yra efektyvūs ir adaptyvūs.

Pagrindinis veiksmingumo faktorius, lemiantis giliųjų neuroninių tinklų aukštus rezultatus, yra subalansuotos architektūros parinkimas ir plačios duomenų bazės sukaupimas. Kiekvienam uždaviniui spręsti yra parenkama unikali modelio struktūra. Atliekant struktūros parinkimo veiksmus yra daromas kompromisas tarp tikslesnių rezultatų, esant didesniai paslėptųjų sluoksnių skaičiui, bet lėtesnio veikimo, ir atvirkščiai – ne tokių tikslių rezultatų, esant mažesniai paslėptųjų sluoksnių skaičiui, tačiau didelės greitaveikos. Galima išskirti tai, kad kuo labiau vizualiai komplikotas tiriamasis objektas, tuo daugiau sluoksnių modelis reikalauja. Dažnai neuronų kiekis šiuose paslėptuose sluoksniuose parenkamas įvesties ir išvesties neuronų skaičiaus diapazone. Paskutiniame etape, parinkus atitinkamus parametrus, eksperimentavimo fazės metu jie yra tikslinami ir modifikuojami.

1.3.1. Atgalinė sklaida

Visas DL algoritmo intelektas ateina iš galimybės savarankiškai, be žmogaus pagalvos, išmokti kokios įvesties duomenų savybės yra svarbios tam tikrai užduočiai (iš)spręsti. Kad tinklas gebėtų ištraukti prasmingas savybes (angl. *feature extraction*) yra ieškoma tikslo funkcijos paklaidos minimumo taikant tam tikrą gradientinio nusileidimo metodo modifikaciją. Tačiau šis optimizavimo metodas nekeičia tinklo neuroninių jungčių stiprumo - tik nurodo ar mokomasi teisinga linkme ar ne. Tam, kad ANN gebėtų keisti svorius (vidinius neuronų jungčių stiprumo koeficientus) 8-ajame dešimtmetyje kelios skirtingos ir nepriklausomos individualių mokslininkų komandos šiai užduočiai atlikti pasiūlė *atgalinės sklaidos* algoritimą (angl. *backpropagation*) ir jau 1990 metais šią techniką efektyviai panaudojo Y. Le Cun su komanda ranka rašytų skaitmenų atpažinimui [3].



3 pav. Atgalinės sklaidos veikimo schema [9]

Atgalinės sklaidos algoritmas grandininės reakcijos seka nuo išvesties iki įvesties pasluoksniui skaičiuoja lokalius gradientus, kai nustatoma paklaidos (žymima E) dalinė išvestinė išėjimo (žymima y) ir svorių (žymima w) pokyčių atžvilgiais. Operacija nuosekliai kartojama su sekančio sluoksnio išėjimais, iki kol pasiekiamas įvesties sluoksnis. Tokiu būdu yra randamas kiekvieno sluoksnio išėjimų (y) ir jungčių koeficientų (w) gradientinis nusileidimas.

1.3.2. Aktyvacijos funkcijos

Tam, kad neuroninis tinklas išmoktų kompleksinių ir sudėtingų savybių, lemiančių vienokį ar kitokį užduoties sprendimą yra reikalinga, kad modelis būtų netiesinis. Būtent dėl to dauguma ANN sluoksnių - ypač konvoliuciniai ir informaciją pirmyn skleidžiantys sluoksniai - turi aktyvacijos funkcijas. Šios funkcijos aktyvuoja jungčių svorių ir praeito sluoksnio neuronų sandaugą (1), suteikiant netiesiškumo savybių.

$$z = \sum_i^n w_i x_i, \quad (1)$$

Kur z žymi jungčių svorių ir praeito sluoksnio neurono sandaugų sumą.

Neuroninių tinklų vystymosi pradžioje populiarios aktyvacijos funkcijos buvo *sigmoid* (2) ir *tanh* (3), tačiau laikui bėgant pastebėta, jog gilaus mokymosi algoritmuose naudojant paprastą *ReLU* (4) (angl. *rectified linear function*) aktyvaciją, klaidos funkcijos minimumas yra pasiekiamas dar greičiau. *ReLU* yra aprašoma kaip pusiau nuožulni funkcija.

$$f(z) = \frac{1}{1 + e^{-z}}, \quad (2)$$

$$f(z) = \tanh(z), \quad (3)$$

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

Pastaraisiais metais taip pat populiarėja *leakyReLU* (5) aktyvacijos funkcija, tai yra *ReLU* modifikacija, kai neigiama dalis yra lygi ne 0, bet mažo koeficiento α tiesinei funkcijai.

$$f(z) = \begin{cases} z, & \text{kai } z > 0, \\ \alpha z, & \text{kai } z \leq 0 \end{cases} \quad (5)$$

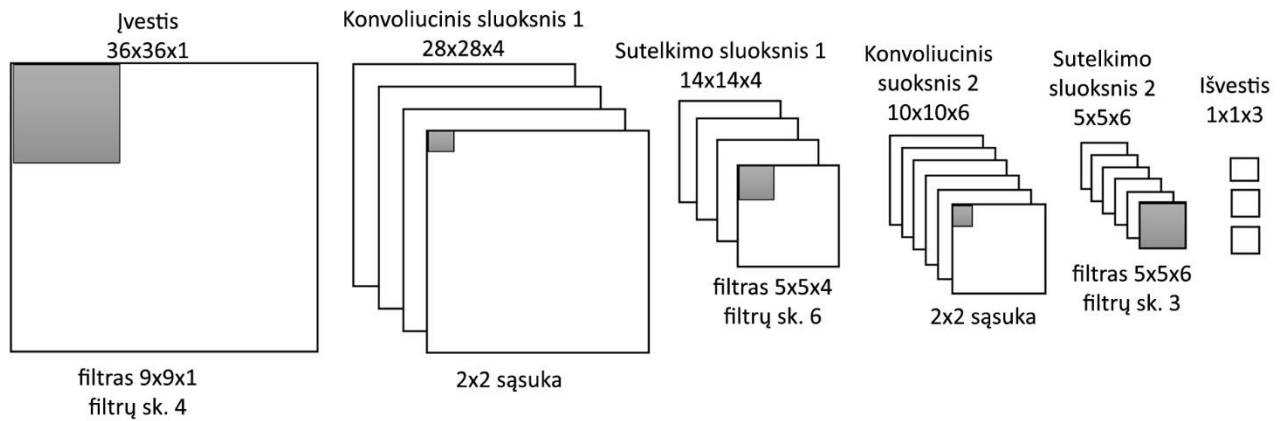
1.3.3. Konvoliuciniai neuroniniai tinklai

Vaizdo analizėje standartiniai ANN tipo tinklai su pilnai sujungtais sluoksniais dažniausiai nėra taikomi dėl prastos greitaveikos. Kad sumažinti parametrų skaičių ir būtų galima identifikuoti objekto savybes nepaisant pasisukimo kampo, apšvietimo variacijos, pozicijos kadre ir priartinimo, buvo išrasti *konvoliuciniai neuroniniai tinklai* (angl. *convolutional neural network* - CNN). Tai yra specifinė ANN atšaka, kuri taip pat yra sudaryta iš neuronų bei sluoksnių su aktyvacijos funkcijomis, tačiau šie neuronai nėra pilnai sujungti tarpusavyje.

Nors CNN pradėti naudoti jau 10-ajame dešimtmetyje [2], praktiškai nebuvo išpopuliarėję iki šio amžiaus pirmojo dešimtmečio. XXI amžiuje sparčiai besivystantys grafiniai procesoriai ir vis didėjanti kompiuterinė greitaveika sukėlė konvoliucinio neuroninio tinklo modelio renesansą. Pastaraisiais metais jis sulaukia vis daugiau dėmesio. Šis konvoliucinis ANN modelis yra plačiai naudojamas kompiuterių regoje - vaizdų apdorojimui ir objektų identifikavimui. Konvoliuciniai neuroniniai tinklai vis dažniau taikomi uždavinių sprendiniuose, kuriuose reikia prognozuoti, spėti, identifikuoti ar nustatyti objektą vaizde.

CNN privalumas pasireiškia gebėjime naudojant nedidelius kernelius, dar kitaip vadinamus šablonais arba filtrais, sėkmingai užfiksuoti globalias ir lokalias savybes. CNN gali grupėje pikselių bei painios fotografijos visumoje atrasti esminius užduočiai išspręsti reikalingus motyvus. Lyginant su pilnai sujungtu neuroniniu tinklu, CNN geba pasiekti panašius rezultatus su daug kartų mažesniu kiekiu svorių parametrų, taip minimizuojant apdorojimo laiką, kas leidžia šį algoritmą plačiai taikyti kompiuterinės regos uždaviniuose.

CNN architektūra yra sudaryta iš serijos blokų (žr. 4 paveikslą). Pirmųjų blokų pagrindą dažniausiai sudaro trys esminiai sluoksnių tipai: konvoliucinis, sutelkimo ir reguliarizavimo sluoksniai.



4 pav. Tipinio konvoliucinio neuroninio tinklo struktūra

Konvoliucinį sluoksnį sudaro kernelių rinkinys. Filtrai dažniausiai yra 3×3 matricos dydžio (nors gali būti ir 1×1 , 5×5 ar kt.) svorių masyvas, o jų skaičius nusako sluoksnio gylį. Per nuotrauką šis kernelio langas yra slenkamas nuosekliai - išilgai kiekvienu stulpeliu ir eile - atliekant matematinę konvoliucijos, dar vadinamą sąsukos, operaciją, t.y. atliekant kernelio ir pradinio vaizdo sandaugą, taip gaunant naujų transformuotų pikselių rinkinį. Pastarasis dar kitaip yra vadinamas ypatybių žemėlapiu (angl. *feature map*) (6). Priešingai nei įprastame pilnai sujungtų neuronų sluoksnyje, konvoliuciniame sluoksnyje atliekant sąsukos operacijas su atitinkama nuotraukos dalimi yra išgaunama prasminga vaizdo turinio informacija. Šių gautų ypatybių skaičius priklauso nuo sluoksnyje esančių filtrų skaičiaus.

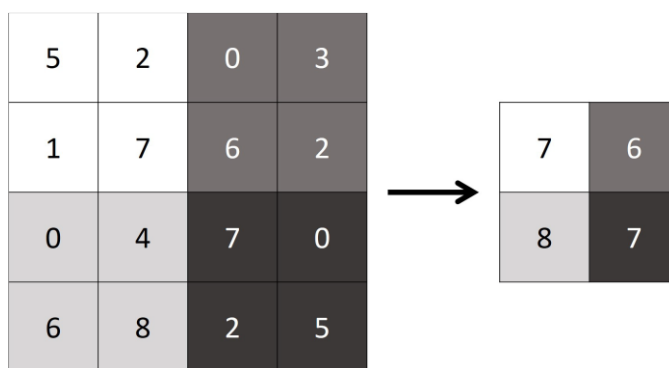
$$g(x, y) = \omega * f(x, y) = \sum_{dx=-a}^a \sum_{dy=-b}^b \omega(dx, dy) f(x + dx, y + dy) \quad (6)$$

$g(x, y)$ - gautas transformuotas vaizdas; ω - šablonas; $f(x, y)$ - pradinė nuotrauka.

Dažnai kompiuterinės regos ir vaizdų apdorojimo srityje filtrai naudojami vaizdui išblukinti, paaštrinti, objektų kraštams rasti ir t.t. Tačiau konvoliucinis neuroninis tinklas, priklausomai nuo šablono narių svorių, mėgina išgauti specifines fotografijos ypatybes. CNN modelis mokosi ieškoti tam tikrų reikšmingų savybių nuotraukoje ir pagal rezultatus sprendžia apie fotografijoje esantį vaizdą. Kad CNN veiktų efektyviai, dažniausiai šablonų kiekis sluoksnyje varijuoja tarp dešimčių ir tūkstančių, o tokių konvoliucinių sluoksnių kiekis būna didesnis nei du. CNN architektūra veikia hierarchiniu principu. Pirmieji konvoliuciniai sluoksniai aptinka tokias smulkias ir abstrakčias vaizdo savybes kaip linija, kraštas, spalvinis tonas, šviesumo gradientas, tačiau vėlesni sluoksniai CNN architektūroje rišdami turinio motyvus į objekto detales jau pradeda užfiksuoti aukšto sudėtingumo savybes, pavyzdžiui tokias, kaip žmogaus akis, šuns uodega ar kelio ženklas. Populiariausi CNN modeliai klasifikacijos uždaviniams atlikti yra *LeNet*, *AlexNet*, *VGGNet*, *ResNet* ir *ZFNet*, segmentavimui - *UNet*, o lokalizavimo uždaviniams - *YoloV4*.

Konvoliuciniai sluoksniai aptinka abstrakčius vaizdo bruožus, tačiau vien to neužtenka efektyviam neuroniniam tinklui sudaryti. Labai svarbus yra sutelkimo pagal maksimumą (angl. *max-pooling*) sluoksnio vaidmuo. Šio tipo sluoksnio tikslas - sutraukti esminę savybių informaciją į mažesnės dimensijos savybių žemėlapius, taip sumažinant parametrų skaičių bei prisirišimą prie pozicijos nuotraukoje. Maksimalaus sutelkimo metu savybių bruožai šiek tiek persislenka, atitinkamai pagal

tai kur buvo didžiausios vertės neuronas. Operacija vykdoma analogiškai kaip ir konvoliuciniame sluoksnyje, t.y. slenkant per nuotrauką nuosekliai - išilgai kiekvienu stulpeliu ir eile – tam tikrų matmenų plote (dažniausiai tai būna 2×2 dydis) yra randamas ir paliekamas didžiausias narys, o visi kiti nariai atmetami (žr. 5 paveikslą). Nors toks veikimo principas praktikoje taikomas dažniausiai, tačiau galima susidurti ir su metodu, kai vietoje sutelkimo sluoksnio yra naudojamas konvoliucinis su 2×2 filtro dydžiu ir žingsniu lygiu dviem [12].



5 pav. maksimalaus sutelkimo operacija

Vykstant mokymui dažna tendencija yra modelio prisitaikymas prie tam tikrų neuroninių kelių, kai kelios jungtys įgauna dideles vertes, o likę svoriai beveik nedaro įtakos. Kad išvengti tokio disbalanso ir pagerinti tinklo generalizavimo savybes, yra taikoma svorių reguliarizacija. Pastaraisiais metais yra išpopuliarėję rinkinio normalizacijos (angl. *batch normalization*) [10] bei atsitiktinio išmetimo (angl. *Dropout*) metodai. Rinkinio normalizacijos metodas pakoreguoja svorių vertes taip, kad jų matematinė viltis būtų 0, o vidutinis standartinis nuokrypis - 1. O atsitiktinio išmetimo metodas - pagal Gauso skirstinio tikimybę - priskiria jungčiai svorį lygų nuliui.

Konvoliucinio, sutelkimo ir normalizacijos sluoksnių perdengimo struktūra yra kartojama dažniausiai iki tol, kol susikuria bent 3 tokie blokai. Toks ANN architektūros tipas defektų aptikimo uždaviniuose taikomas dažniausiai.

1.3.4. Paruoštų modelių naudojimas

Sudėtingiems nekontroliuojamų aplinkybių uždaviniams spręsti dirbtinis neuroninis tinklas privalo turėti puikias generalizavimo savybes ir mokėti išskirti prasmingus skirtingo abstrakcijos lygio motyvus. Šį modelį reikia treniruoti su labai plačia duomenų baze ir architektūrą parinkti iš daugiau nei poros blokų - dažniausiai jų reikia bent dešimties. Dėl šios priežasties sukurti efektyvų gilus mokymosi modelį užtrunka labai daug laiko. O ir apmokius modelį, procesą tenka kartoti kiekvieną kartą, kai norima pridėti papildomą klasę segmentavimui. Šios problemos gilus mokymosi architektūras, sudarytas iš ne mažiau kaip 10 blokų, padaro nepraktiškas naudojimui situacijose, kai laikas ir darbo kaštai yra vieni svarbiausių faktorių.

Ilgo apmokymo problema yra sprendžiama sujungiant jau apmokytą neuroninį tinklą (angl. *pretraining*) su vienu ar dviem naujai pridėtais blokais ir tęsiant mokymą su nauja, dažniausiai nedidele, bet uždaviniui specifine, duomenų baze. Toks procesas dar vadinamas perdavimo mokymusi (angl. *transfer learning*). Jo metu yra paimamas jau apmokytas labai gilus neuroninis tinklas, kaip *VGG-16* [8], turintis virš 138 milijonų parametru, skirtas klasifikavimo užduočiai spręsti. Šis modelis, ištreniruotas naudojant nepaprastai didelę duomenų bazę, kaip *ImageNet* [7], tampa naujojo neuroninio tinklo pradinė būseną. Iš gilus neuroninio tinklo *VGG-16* ar kito analogiškai

gilus ANN modelio yra pašalinami keli paskutiniai – dažniausiai pilnai sujungti – neuroniniai sluoksniai, o likę yra “užšaldomi“. Prie šios apmokyto tinklo architektūros yra prijungiami nauji sluoksniai su šviežiai inicializuotais svoriais ir toks tinkas jau mokomas specifinei ir sudėtingai užduočiai spręsti.

Perdavimo mokymosi būdą defektų segmentavimui medienos paviršiaus nuotraukose efektyviai pritaikė tyrėjas Ting He su komanda [13] ir įvairių produktų gaminių anomalijoms aptikti - mokslininkas P. Bergmann su bendraautorais [11], taip pagerindami moderniausių ir pažangiausių to meto algoritmų rezultatus.

1.4. Prižiūravimo ir neprižiūravimo mokymosi neuroninių tinklų taikymo ypatumai

Visus dirbtinius neuroninius tinklus, pagal mokymosi tipą, galima skirti į tris grupes – prižiūravimo, neprižiūravimo ir hibridinio mokymosi.

Algoritmai, kurie naudoja žymėjimus objektui klasifikuoti, segmentuoti ar lokalizuoti, vadinami prižiūravimo mokymosi (angl. *supervised learning*) arba mokymosi su mokytoju algoritmais. Tokie modeliai būdavo treniruojami taikant duomenų bazes, kurias sudarydavo vaizdai su atitinkamu žymėjimu. Mokymosi metu modelis pasikoreguoja pagal tai ar vaizde objektas nustatomas teisingai, ar klaidingai. Prižiūravimo mokymosi gilieji CNN tinklai jau šio amžiaus pirmame dešimtmetyje suteikė naujas galimybes objektų segmentavimo, lokalizavimo ir klasifikavimo uždavinių sprendimams.

Lyginant su klasikiniiais vaizdo apdorojimo metodais prižiūravimo mokymosi neuroninis tinklas su plačia duomenų baze rodo reliatyviai aukštus rezultatus. Vaizdo analizėje objekto aptikimo ir segmentavimo uždaviniuose mokymasis su mokytoju yra populiariausias sprendimo būdas nes tokio tipo modeliai yra patikimi ir rezultatai tikslūs. Nors prižiūravimo mokymosi ANN algoritmai yra veiksmingi, tačiau turi vieną didelį trūkumą. Tokiam mokymosi algoritmo modeliui reikalinga sužymėta duomenų bazė – jis turi žinoti, kokie objektai ir kur nuotraukoje yra randami.

Daugumą realiai sprendžiamų uždavinių komplikuoja tai, kad susidaro atitinkamos sąlygos, kurios neleidžia surinkti pakankamo kiekio duomenų su ieškamu objektu vaizde arba duomenų bazė yra labai didelė, o ją žymėti rankiniu būdu kainuotų daug laiko ir pinigų. Būtent dėl šios priežasties yra vis labiau taikomi ir tobulinami neprižiūravimo mokymosi algoritmai. Mokslininkas Y. LeCun [9] teigia, kad ateityje gilieji neprižiūravimo mokymosi algoritmai susilauks vis daugiau dėmesio. Tyrėjas aiškina, kad norint pasiekti žmogaus suvokimo galimybes ANN modeliai turėtų išmokti analizuoti informaciją stebėdami pasaulį, o ne remiantis duotomis žymėmis. Būtent neprižiūravimo mokymosi algoritmai šią sąlygą – nors ir netobulai - bet išpildo.

Neprižiūravimo mokymosi (angl. *unsupervised learning*) algoritmai – tai tokie algoritmai kurie, priešingai nei prižiūravimo mokymosi algoritmai, nenaudoja papildomos informacijos apie turinį, ar tai būtų nuotrauka, ar garso takelis, bet geba analizuoti duomenis savarankiškai. Pastaraisiais metais ženkliai išaugo tyrimų, kurie bando efektyviai identifikuoti anomalijas vien tik iš kokybiškų produktų vaizdų, skaičius. Kokybiško gaminio duomenų bazę suformuoti užtrunka daug trumpiau ir paprasčiau nei produkto su defektais. Dėl to mėginama taikyti neprižiūravimo mokymo algoritmus.

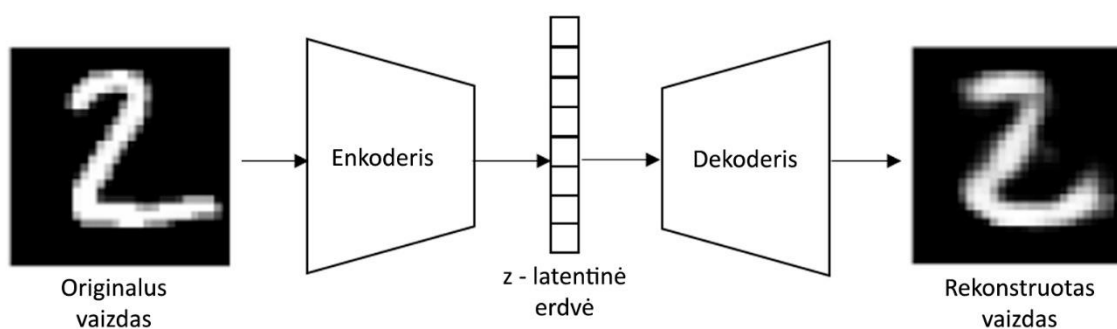
Neprižiūravimo mokymosi modelių tipų yra daug ir įvairių. Populiariausi jų remiasi autoenkoderių (angl. *autoencoder* - AE) ir generacinių priešiškujų tinklų (angl. *generative adversarial network* -

GAN) veikimo principais. Ir AE, ir GAN modeliai dažnai naudojami defektų identifikavimo uždaviniams spręsti, tačiau priešingai nei prižiūravimo mokymosi neuroniniai tinklai, kurie ieško jau išmokyto defektų, šie anomalijos sprendinius pateikia aptikdami vaizde prieš tai dar nesutiktą objektą, kas ir atitinka neprižiūravimo mokymosi veikimo mechanizmą.

AE ir GAN tipo neuroniniai tinklai bando transformuoti gautą vaizdą į kokybiško gaminio vaizdą. Tuomet pakeisti įvesties rezultatai yra palyginami su originaliu atvaizdu ir pagal pikselių verčių skirtumą yra daroma išvada, ar nuotraukoje yra anomalija. Neprižiūravimo mokymosi algoritmai, identifikuojantys defektus, ir šiomis dienomis susiduria su aukšto klaidingai teigiamo santykio (FPR) problema, kai kokybiškas objektas nustatomas kaip defektas. Kaip sumažinti FPR ir pagerinti identifikavimą yra esminis, reikalaujantis sprendimo, iššūkis anomalijų atpažinimo srityje.

1.5. Pagrindiniai AE neprižiūravimo mokymosi algoritmų tipai

Labiausiai paplitęs neprižiūravimo mokymosi algoritmų tipas anomalijoms rasti yra autoenkoderiai. Jie užkoduoja nuotraukos informaciją į mažų dimensijų latentinę erdvę, ši suspausta informacija vėliau yra atkoduojama ir naudojant pasirinktą metodiką yra lyginama įvestis su išvestimi taip nustatant defekto būvimo tikimybę [28]. AE algoritmo veikimo hipotezė yra tokia, jog modelis, apmokytas tik užkoduoti ir dekoduoti kokybiško gaminio duomenis, negalės rekonstruoti nuotraukos, jeigu ši turės defektų. Kuriant autoenkoderį ir projektuojant latentinės erdvės dydį visada yra daromas kompromisas tarp svarbiausių savybių išgryninimo mažinant z matricą ir efektyvaus atkūrimo tikslumo didinant z matricą.



6 pav. Tipinė AE struktūra

1.5.1. Standartinis AE algoritmas

Jie suspaudžia nuotraukos informaciją į vektorinę latentinės erdvės informaciją, kuri po to yra dekoduojama. Veikimo principas yra sutraukti didžiulį kiekį vaizdinės informacijos $x \in R^{H \times W \times D}$ į siaurą matricą $z \in R^K$, kuri reprezentuoja esminius ir prasmingiausius nuotraukos turinio bruožus. Latentinės erdvės masyvas dažniausiai būna vektorius tipo, tačiau galima ir trimatė matrica $z \in R^{m \times n \times K}$. Pastarasis atvejis dažniausiai naudojamas didesnės rezoliucijos vaizdams, kad enkoderis išsaugotų daugiau kontekstinės informacijos. Iš šios esminės informacijos yra stengiamasi atgaminti originalų vaizdą $\hat{x} \in R^{H \times W \times D}$. Algoritmas susideda iš dviejų konvoliucinių neuroninių tinklų: enkoderio, turinčio parametrus ϕ , ir dekoderio, turinčio parametrus θ . Modelis yra treniruojamas taip,

kad sutraukiant ir išskleidžiant sveikų objektų vaizdinius duomenis, būtų minimizuojami rekonstravimo nuostoliai \mathcal{L} (7).

$$\operatorname{argmin}_{\varphi, \theta} \mathcal{L}_{AE \text{ tinklas}}^{\varphi, \theta}(x, \hat{x}) = l_1(x, \hat{x}) = \|x - \hat{x}\| \quad (7)$$

, kur l_1 reiškia vidutinį nuokrypį. Be (7) klaidos funkcijos dar yra taikoma ir vidutinio kvadratinio nuokrypio (8) ar į ją panaši kita klaidos funkcija.

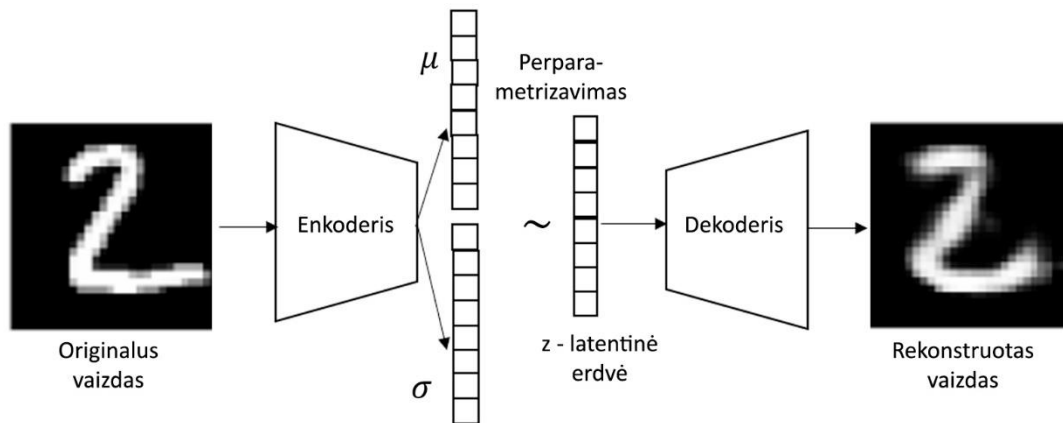
$$\operatorname{argmin}_{\varphi, \theta} \mathcal{L}_{AE \text{ tinklas}}^{\varphi, \theta}(x, \hat{x}) = l_2(x, \hat{x}) = \|x - \hat{x}\|^2 \quad (8)$$

, kur l_2 reiškia vidutinį kvadratinį nuokrypį.

Standartinio tipo AE modelius ligos diagnozavimui iš magnetinio rezonanso smegenų nuotraukų sėkmingai pritaikė tyrėjai C. Baur ir B. Wiestler [25].

1.5.2. Variacinis AE algoritmas

Ženklus pokytis autoenkoderių taikymo vaizdo analizės tyrimuose buvo variacinė autoenkoderių (VAE) modifikacija. Šis pakeitimas pagerino modelio reprezentacijos gebėjimus, priverčiant tinklą suspausti latentinę erdvę pagal Bajeso analizę į tikimybinį pasiskirstymą - dažniausiai Gauso skirstinį. Kaip ir pats pavadinimas duoda nuorodą, jog modelis nebėra pilnai deterministinis, bet tampa variacinis – įgauna stochastinių bruožų. Vietoj kietai apmokytos matricos z autoenkoderių tinkle, kuri atitinka tam tikrą įvestį pagal x , VAE modelis erdvę z atvaizduoja pagal užduotą atsitiktinio pasiskirstymo dėsnį, kai duodama įvestis x . Tai yra daroma tam, kad latentinė erdvė kistų nuosekliai, pereinų tolygiai, kai įvestis keičiasi po truputį.



7 pav. Variacinio AE struktūra

Veikimas pagrįstas principu, jog kiekvieną galimą įvestį x atitinka matematinės vilties ir dispersijos vektoriai, kurie, pasinaudojant konstanta, yra perparametrizuojami į z vektorių [29]. Norimas latentinės z erdvės skirstinys yra nustatomas iš anksto, tada modeliui reikia rasti tokius enkoderio ir dekoderio parametrus, su kuriais latentinė erdvė pasiskirstytų pagal užduotą dėsnį. Skaičiuojant klaidos funkciją (8) yra įvedama regularizacijos dedamoji, kuri ir priverčia modelį užkoduoti įvestį į tolygią ir prasmingą informaciją. Ši dedamoji tai Kulbako ir Leiblerio (angl. *Kullback and Leibler*) funkcija skirstinių nuokrypiui rasti.

Pilna VAE modelio nuostolių funkcija:

$$\operatorname{argmin}_{\varphi, \theta} \mathcal{L}_{VAE}^{\varphi, \theta}(\mathbf{x}, \hat{\mathbf{x}}) = l_2(\mathbf{x}, \hat{\mathbf{x}}) + \lambda_{KL} D_{KL}(q(z) || p(z)) \quad (9)$$

$p(z)$ – skirstinys nusakantis užsibrėžtą tikimybę;

$q(z)$ – gautas skirstinys užkodavus įvestį x ;

D_{KL} – Kulbako ir Leblerio funkcija;

λ_{KL} - Lagranžo daugiklis, kuris suteikia svarbos, kiek smarkiai norime, kad latentinė erdvė atitiktų užsibrėžtą skirstinį.

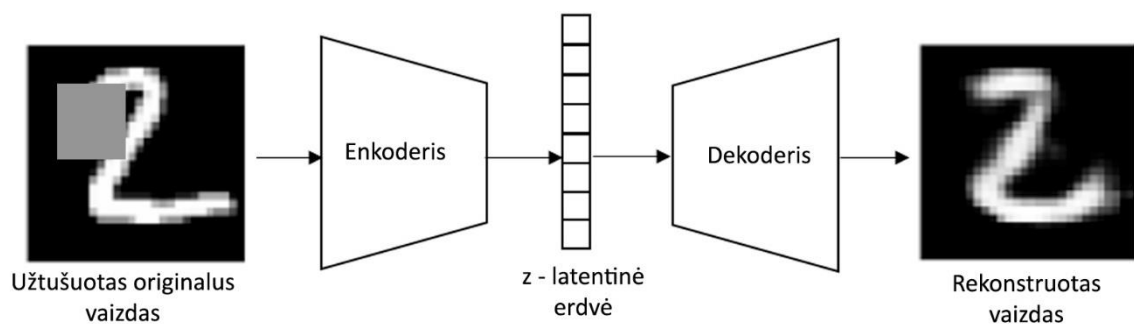
Dažniausiai λ_{KL} narys yra prilyginamas 0.5, užduotas skirstinys - gausinis, o parametrai $\mu = 0$ ir $\sigma = 1$. Tada klaidos funkcijos dedamoji latentinei erdvei reguliuoti yra gaunama tokia:

$$D_{KL} = \mu^2 + \sigma^2 - \log(\sigma) - 1 \quad (10)$$

Didinat Lagranžo daugiklį, z erdvė tampa vis tolydesnė ir prasmingesnė, bet rekonstruotas vaizdas abstraktesnis.

1.5.3. Kontekstinis AE algoritmas

Dar vienas dažnai taikomas būdas efektyviai išmokyti autoenkoderį užkoduoti ir dekoduoti informaciją yra duomenų, kuriais jis yra apmokomas, iškarpymas darant prielaidą, jog tai padarys modelį atsparų triukšmui. Atsitiktinių vaizdo regionų paslėpimas mokymosi metu yra puiki metodika norint, kad AE modelis mokėtų atkurti nuotrauką esant triukšmui. Pirminiai, nuotraukų duomenų apkarpymas autoenkoderiniams tinklams apmokyti buvo naudojamas uždažymo užduočiai atlikti [17] - sugeneruoti tas vaizdinio dalis, kurios dėl vienos ar kitos priežasties buvo prarastos, tačiau pastaraisiais metais radosi ir daugiau panaudojimo sričių. Defektų aptikimo uždaviniuose kontekstinio AE modelio atsparumas triukšmui pasireiškė kaip dar didesnis gebėjimas atkurti kokybišką objektą iš galimai defektą turinčio objekto nuotraukos, taip aiškiai išskiriant anomaliją originaliame vaizde. Mokslininkas D. Zimmer ir kt. [18] tyrė kontekstinį autoenkoderį, kuris aptiko pažeidimus iš magnetinio rezonanso smegenų nuotraukų, ir iš gautų eksperimentų rezultatų padarė išvadą, jog metodas yra daug žadantis ir turi potencialo paspartinti medicininių ligų diagnostiką ateityje.



8 pav. Kontekstinio AE struktūra

1.5.4. Suvaržytas AE algoritmas

Defektų aptikimui, naudojant neprižiūrimą mokymąsi, tyrėjų D. Pathak su komanda [19] pasiūlytas autoenkoderis su taip vadinama *suvaržyta* (angl. *constrained*) klaidos funkcija yra taikomas rečiau. Algoritmo modifikacijos veikimas grįstas modelio, papildant klaidos funkciją dar viena algebrine dedamąja, suvaržymu.

Nuostolių funkcijoje yra naudojama standartinė l_2 rekonstrukcijos dedamoji (8), prie jos yra pridamas vidutinis kvadratinis skirtumas tarp latentinių erdvių, kai iš pirminės užkoduotos įvesties yra atimama užkoduota, rekonstruota ir vėl užkoduota įvestis (11).

$$\operatorname{argmin}_{\varphi, \theta} \mathcal{L}_{VAE}^{\varphi, \theta} \text{ tinklas}(x, \hat{x}) = l_2(x, \hat{x}) + l_2(z, \hat{z}) = \|x - \hat{x}\|^2 + \|z - \hat{z}\|^2 \quad (11)$$

1.5.5. GAN algoritmas

Generaciniai besivaržantys tinklai (GAN) anomalijų aptikime buvo pradėti taikyti dar prieš AE modelius. GAN tinklas yra sudarytas iš dviejų posistemių, generatoriaus (G) ir diskriminatoriaus (D). GAN paskirtis yra išmokyti duomenų pasiskirstymą, tendencijas ir bruožus taip, kad būtų galima generuoti naujus duomenis bei, keičiant įvesties reikšmes, kontroliuoti jų turinį. Šio neprižiūrimo mokymosi modelio, kuris savyje turi du skirtingus modelius, genialumas yra galimybėje treniruotis pačiam. Kaip teigia ir pats pavadinimas, mokymasis vyksta varžymosi principu. Generatoriaus posistemė mokosi generuoti kuo realistiškesnius duomenis taip, kad diskriminatoriaus posistemė neatskirtų, ar šie duomenys yra tikri, ar ne, o tuo tarpu diskriminatoriaus posistemė treniruojasi atskirti tikrus duomenis nuo sukurtų taip, kad generatoriaus posistemės duomenys būtų identifikuojami kaip netikri [30].

GAN neuroninis tinklas turi du išskirtinius privalumus: gebėjimą mokytis be mokytojo ir tolyginio pasiskirstymo duomenų sugeneravimą, kai įvestis kinta tolygiai, sugeneruojami duomenys irgi keičiasi tolygiai. Lyginant su autoenkoderiniais tinklais pagrindinis GAN trūkumas yra nestabilus mokymosi procesas. Dėl architektūrinių savybių yra nemaža tikimybė, kad mokymosi metu modelis išsibalansuos ir viena iš posistemių pasieks be galo dideles nuostolių funkcijos reikšmes, o algoritmas negebės generuoti tikroviškų duomenų.

GAN klaidos funkcija yra išvesta iš binarinės kryžminės entropijos funkcijos (12).

$$\mathcal{L}_{BCE} = y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}) \quad (12)$$

kur y – tikra reikšmė, o \hat{y} – algoritmo gauta reikšmė. Ši formulė GAN modeliui apmokyti užsirašo kaip:

$$\mathcal{L}_{GAN} = \min_G \max_D \left[\log(D(x)) + \log(1 - D(G(z))) \right] \quad (13)$$

GAN tinklai yra dažnai naudojami kūrybinėje industrijoje, kur norima sukurti naujų vaizdo ar garso medžiagų, pavyzdžiui, turint vieną žmogaus veido nuotrauką yra sugeneruojama visa aibė to veido išraiškų. GAN tinklai yra eksploatuojami ne tik kūrybiniams tikslams, juos taip pat bandoma taikyti ir anomalijų aptikimo srityje. Defektų aptikimui GAN yra apmokomas tiksliai su anomalijų neturinčiais objektais. Testavimo metu generatoriaus modelis sugeneruoja kuo panašesnę vaizdą į analizuojamą nuotrauką, ir, apskaičiavus skirtumą tarp sukurto bei testuojamo vaizdo, yra gaunamas anomalijų rezultatų žemėlapis.

Mokslininkų komanda, vadovaujama C. Baur [25], bei tyrėjas T. Schlegl su bendraautoriais [16], analizavo įvairių GAN tipo modelių taikymą anomalijų aptikimui iš medicininių - smegenų magnetinio rezonanso ir tinklainės optinės koherentinės tomografijos - nuotraukų. Abi tyrėjų komandos padarė išvadą, jog gilieji GAN tinklai gali būti naudojami anomalijoms rasti anatomijos vaizdiniuose. Kitoje srityje tyrimus vykdęs P. Bergmann su komanda [15] priėjo priešingų pastarosioms išvadų. Mokslininkai, ieškodami defektų pramoninės kilmės gaminiuose, gavo ženkliai žemesnius kiekybinius įverčius.

Kadangi šiame darbe neprižiūrimo mokymosi autoenkoderiniai neuroniniai tinklai yra tiriami su ta pačia duomenų baze, kurią naudojo ir pastaroji mokslininkų komanda, gavusi prastus rezultatus su GAN algoritmu, dėl to generaciniai besivaržantys tinklai nebuvo įtraukti į tolimesnį praktinį tyrimą.

1.6. Segmentavimo technikos neprižiūrimo mokymosi algoritmuose

Neprižiūrimo gilaus mokymosi dirbtiniai neuroniniai tinklai defektams segmentuoti naudoja įvairias technikas. Pati paprasčiausia ir populiariausia yra rekonstrukcijos technika:

$$r = |x - \hat{x}|, \quad (14)$$

kur x – įvestis;

\hat{x} – užkoduota ir dekoduoja įvestis;

r - liekamoji vertė pikselių atžvilgiu.

Taip pat segmentavimui galima naudoti ir *Monte Karlo* metodą, kai rezultatas gaunamas rekonstrukcijos principu, bet ne iš vieno vaizdo, o iš aibės mėginių. Šie mėginiai - tai skirtingų latentinių erdvių matricos, kai algoritmas yra VAE bei GAN tipo, arba tos pačios latentinės erdvės, bet su atsitiktinai išmestais neuronais. Su kiekvienu dekoduojamu latentinės erdvės pavyzdžiu yra randama rekonstrukcija ir gauta suma padalinama iš imties skaičiaus:

$$r = \frac{1}{N} \sum_{n=1}^N |x - \hat{x}| \quad (15)$$

Mažiau populiari, tačiau verta dėmesio, technika yra gradientinės paieškos metodas. Tai technika, kombinuojanti rekonstrukcijos metodo rezultatą su atgalinės sklaidos išvestinės rezultatais (8). Šį algoritmą patologijoms iš medicininių nuotraukų segmentuoti pasiūlė ir ištyrė mokslininkas D. Zimmer su komanda [18], nustatę, jog variaciniai autoenkoderiai, kurie naudojo abi – rekonstrukcijos ir gradientinės paieškos – technikas veikė efektyviausiai, nepaisant su kuria duomenų baze buvo atliekami testai.

$$h \left(|x - \hat{x}|, \left| \frac{\partial (L_{KL} + L_{VAE}(x, z))}{\partial x} \right| \right), \quad (16)$$

Kur h – gradientinės paieškos funkcija;

L_{KL} - Kulbako ir Leiblerio funkcija skirstinių klaidai rasti;

L_{VAE} – variacinio AE klaidos funkcija.

Kadangi atliekamame tyrime yra nagrinėjami daugiausia deterministinio pobūdžio autoenkoderiai, dėl to defektų segmentavimui buvo pasirinkta naudoti rekonstrukcijos technika.

1.7. Kokybės įverčiai

Efektyviam defektų aptikimo algoritmui sukurti nepakanka turėti duomenų bazę ir sudarytą bei apmokytą modelį - dar reikia nuspręsti, kokią metriką naudoti veiksmingumui išmatuoti. Pagrindinis kuriamos sistemos kokybės kriterijus yra atpažinimo tikslumas. Defektų segmentavimo tikslumo įvertinimui nėra apibrėžtos vienos, viską nusakančios, metrikos. Vaizdo sistemų analitikai ir mokslininkai šio tikslumo nustatymui naudoja skirtingus įverčius, kuriuos galima priskirti dviem grupėm – priklausantys nuo slenkstinės vertės įverčiai ir nuo jos nepriklausantys.

Slenkstis – tai reikšmė, kuri binarizuoja analoginį (dažniausiai tarp 0 ir 1 pasiskirsčiusį) vaizdo segmentavimo algoritmo išėjimą. Dirbtinio intelekto modelių išvestis testavime yra lyginama su neginčytinos tiesos (angl. *ground truth*) vaizdais, kurie turi tik dvi reikšmes. Vienetas reiškia, jog defektas yra, nulis - kad jo nėra. Ankstesniame skyriuje (1.6) analizuotos technikos – rekonstrukcijos, Monte Karlo ir gradientinės paieškos - suformuoja anomalijų žemėlapi, kur kiekvienas analizuojamo vaizdo pikselis yra išreiškiamas slenkančio kablelio verte, parodančia defekto buvimo tikėtinumą. Būtent dėl šios priežasties algoritmų išėjimus reikia binarizuoti naudojant slenkstinę vertę. Tai padarius ir palyginus spėjimo rezultatus su teisingomis reikšmėmis, yra gaunama sumaišymo matrica (angl. *confusion matrix*). Pastarąją sudaro keturios klasės:

- TP – testo rezultatas, kuris teisingai nustato sąlygos buvimą (angl. *true positive*);
- TN – testo rezultatas, kuris teisingai nustato sąlygos nebuvimą (angl. *true negative*);
- FP – testo rezultatas, kuris klaidingai nustato sąlygos buvimą (angl. *false positive*);
- FN – testo rezultatas, kuris klaidingai nustato sąlygos nebuvimą (angl. *false negative*).

Visos keturios klasės yra naudojamos kaip pamatinės vertės prasmingiems tikslumo įverčiams apskaičiuoti.

Defektus aptinkančių algoritmų tikslumui nusakyti yra naudojami du pagrindiniai nuo slenkščio priklausomi įverčiai – $F1$ (17), žinomas kaip *Dice* indeksas, ir IoU (18), dar sutinkamas kaip *Jaccard* indeksas.

$$F1 = \frac{2TP}{2TP + FP + FN} \quad (17)$$

$$IoU = \frac{TP}{TP + FP + FN} \quad (18)$$

IoU ir $F1$ metrikos gali būti interpretuojamos kaip panašumo matavimo priemonė tarp baigtinio skaičiaus aibių. Pavyzdžiui, dviem A ir B aibėm *Jaccard* indeksas gali būti apibrėžiamas kaip:

$$IoU = J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (19)$$

Kuo panašesnės aibės, tai yra kuo kompiuterinės regos algoritmo surastas defektų plotas artimesnis neginčytinai tiesai, tuo didesnis IoU ir $F1$ įvertis. Kadangi šios funkcijos turi beveik vienodą charakteristiką ir rezultatus atspindi santykinai taip pat, dėl to tyrimuose naudojamas tik vienas tų įverčių.

Jaacard ir *Dice* slenksčio įverčiai puikiai nusako algoritmo, klasifikuojančio ir segmentuojančio defektus, efektyvumą, tačiau turi vieną pagrindinį trūkumą – yra priklausomi nuo to, kaip sėkmingai bus parinkta slenkstinė vertė. Tam, kad išvengti binarizavimo ribos parinkimo žingsnio, naudojamos kitos trys efektyvios metrikos – *AUROC*, *AUPRC* ir *AUIoU*. Visi trys įverčiai skaičiuoja kreivės integralą (*AU* – angl. *area under*).

AUROC brėžia atpažinimo (*TPR*) priklausomybę nuo FP santykio (*FPR*), *AUPRC* - preciziškumo (*PRC*) nuo atpažinimo (*TPR*) priklausomybę, tuo tarpu *AUIoU* – panašumą (*IoU*) nuo FP santykio (*FPR*). Pati charakteristika sudaroma iš rezultatų, apskaičiuotų prie daug skirtingų slenksčio verčių. Verta pabrėžti, jog objektų defektai dažniausiai užima tik kelis procentus viso vaizdo ploto, dėl to *FPR* reikšmės, nepaisant slenkstinės vertės, beveik visada būna aukštos, ir *AUROC* įvertis tampa mažiau reprezentatyvus algoritmo efektyvumui, kai tuo tarpu *AUPRC* įvertis išlieka informatyvus [20].

$$TPR = \frac{TP}{TP + FN} \quad (20)$$

Atpažinimas, arba *TPR* (angl. *true positive rate*) – santykis tarp TP ir sąlygą tenkinančių testų sumos.

$$FPR = \frac{FP}{FP + TN} \quad (21)$$

Klaidingas atpažinimo santykis, arba *FPR* (angl. *false positive rate*) – santykis tarp FP ir sąlygą netenkinančių testų sumos.

$$PRC = \frac{TP}{TP + FP} \quad (22)$$

Preciziškumas, arba *PRC* (angl. *precision*) – santykis tarp TP ir TP bei FP sumos.

Šios nepriklausomos nuo slenksčio vertinimo metrikos, *AUROC*, *AUPRC* ir *AUIoU*, tyrimuose veikia dažniausiai ir yra naudojamos kaip pagrindinis kriterijus vertinant neprižiūravimo mokymosi algoritmų, aptinkančių defektus, tikslumą.

1.8. Apibendrinimas

Neprižiūravimo mokymosi algoritmų panaudojimo defektams aptikti srityje aiškaus efektyviausio metodo nėra - vieni modeliai labiau pritaikyti atpažinti anomalijas tekstūrose, antri - specifiniuose objektuose, tretį - tam tikro tipo, kaip MRI, nuotraukose. Dauguma dirbtinio intelekto sprendimų veikia tiksliai aiškiai kontroliuojamose sąlygose ir nėra universalūs. Dirbtinių neuroninių tinklų, treniruojamų be mokytojo, tyrimų sritis yra eksperimentinėje stadijoje, todėl žinios kaip juos kurti ir taikyti dar tik formuojamos.

Buvo prieita išvada, jog mokymosi metu defektų aptikime naudojant tik nuotraukas be anomalijų, autoenkoderiniai modeliai pasiekia ypač gerus rezultatus ir yra dažnai pateikiami kaip tinkamiausias sprendimo būdas. Tačiau nėra pilnai aišku, kurios modifikacijos yra pranašesnės, o kurios - mažiau veikmingos, todėl šio tiriamojo darbo eksperimentinėje dalyje yra nagrinėjama įvairių AE tinklų skirtumų įtaka gebėjimui tiksliai segmentuoti anomalijas.

2. MVTec AD Duomenų bazė

Tam kad neprižiūravimo mokymosi algoritmų efektyvumą būtų galima lyginti vieną su kitu, reikia, kad tyrimų metu sąlygos išliktų panašios. Lyginant skirtingus prižiūravimo mokymosi modelius vaizdo klasifikavimo uždaviniams spręsti dažnai yra pasirenkamos naudoti *MNIST* ar *ImageNet* duomenų bazės, segmentavimo uždaviniams - *Coco* duomenų bazė, tuo tarpu lyginant neprižiūravimo mokymosi modelius defektų aptikimui vis dažniau naudojama *MVTec AD* duomenų bazė [20]. Tiriant anomalijų aptikimą paviršiuose su neprižiūravimo mokymosi algoritmais galima naudoti *NanoTWICE* nano-pluoštų fotografijų duomenų bazę [26] arba Y. Huang ir jo komandos pateiktą plytelių nuotraukų duomenų bazę [27], tačiau tiriant anomalijų aptikimą daiktuose su algoritmais, naudojančiais neuroninius tinklus be mokytojo, *MVTec AD* yra vienintelė tam tikslui pritaikyta duomenų bazė.

2.1. Apie MVTec AD

Ši buvo sukurta ir pavišinta 2019 metais. Duomenų bazę sudarė *MVTec* kompanija, siekdama, jog pastaroji pamėgdžiotų vaizdus, kurie yra užfiksuojami realiomis pramoninėmis sąlygomis.

MVTec AD duomenų bazė yra sukurta specialiai neprižiūravimo mokymosi algoritmams testuoti, dėl to yra pripažįstama tarp dirbtinio intelekto srities mokslininkų, tiriančių anomalijų aptikimą. Duomenų bazę sudaro apie 5 tūkstančiai nuotraukų, didžioji jų dalis yra kokybiškų gaminių ir tekstūrų be žymių, likusi dalis - sužymėtos nuotraukos su defektą turinčiais objektais ir paviršiais. Vaizdai be žymių yra naudojami modeliams apmokyti, o vaizdai su žymėmis - modeliams įvertinti.

Visą duomenų bazę sudaro penki skirtingi paviršiai (kilimo, tinklelio, odos, plytelių ir medienos dangos) bei dešimt objektų (butelis, kabelis, kapsulė, lazdyno riešutas, metalinė veržlė, tabletė, medsraigtis, dantų šepetėlis, tranzistorius ir užtrauktukas). Dauguma nuotraukų yra 1024×1024 pikselių dydžio, kai kurių kategorijų vaizdai - žemesnės rezoliucijos. Šiame darbe buvo pasirinkta tirti autoenkoderiniai giliųjų neuroninių tinklų modeliai defektams rasti naudojant tris skirtingus objektus: medsraigtis, stiklinis butelis ir lazdyno riešutas.

Defektai *MVTec AD* objektuose pasireiškia įvairiai, pavyzdžiui, varžtai gali būti su įlenkimu, nutrupėjusiu sriegiu ar nuskilusiu kotu, lazdyno riešutai - turėti kevalo įtrūkimų, įpjovimų, skylių bei uždažymų, stiklo butelis - būti įskilęs arba apsiteršęs. Vienos anomalijos nustatomos iš spalvos, kitos - iš formos, o defekto plotas varijuoja nuo smulkiausio, kaip sriegio nulūžimas, iki aprėpiančio visą objektą, kaip stiklo suskeldėjimas.



9 pav. Darbe naudojamų ir iš MVTec AD duomenų bazės parsisiųstų, trijų skirtingų objektų nuotraukos

2.2. Duomenų paruošimas

Prieš sukuriant ir apmokant neprižiūrimo mokymosi neuroninius tinklus, buvo parsisiųsta MVTec AD duomenų bazė ir paruošta treniravimuisi bei testavimui.

Daugumos naudojamos duomenų bazės originalus vaizdų dydis - 1024×1024 pikselių. Kad apmokomi modeliai turėtų mažiau vidinių sluoksnių ir taip atitinkamai mažiau parametrų bei, kad apmokymo procesas vyktų sparčiau, nuotraukos buvo sumažinamos iki 256×256 pikselių rezoliucijos. Šis nuotraukos dydis pasirinktas todėl, kad ir toliau mažėjant vaizdai, būtų prarandama vertinga informacija apie objektą. Taip pat fotografijos buvo normalizuojamos verčių mastelį nustatant tarp 0 ir 1.

Visų kategorijų objektų nuotraukų kiekis yra ribotas. Tam, kad treniravimosi metu modeliai per daug neprisitaikytų prie siauros aibės įvesčių, ir, kad būtų išvystytos aukšto lygio generalizavimo savybės, vaizdai yra augmentuojami, taip gausiai padidinant duomenų imtį.

Duomenų augmentavimo procesas buvo atliktas pagal [15] tyrime aprašytas gaires. Visi pasirinkti objektai buvo apdorojami trimis technikomis:

- 1) vaizdą vartant vertikaliai ir horizontaliai;
- 2) sukant paveikslą aplink centrą įvairiu kampu;
- 3) perstumiant vaizdą atsitiktine, tarp 0 ir 5 pikselių, verte ir naują jo plotą užpildant kraštinių pikselių reikšmėmis.

Vaizdas buvo augmentuojamas iki sukaupiant 20 tūkstančių kiekvieno objekto pavyzdžių, iš kurių ketvirtadalis buvo skirtos validavimui. Atitinkamai objektų vaizdų skaičių limitavo pasirinktos įrangos skiriamos operatyviosios RAM atminties ribojimai.

2 lentelė. Treniravimui ir validavimui skirtų nuotraukų skaičius prieš ir po augmentavimo.

	Pradinis nuotraukų sk.	Augmentuotų nuotraukų sk.	Padidinta kartų
Medsraigtis	320	20 000	62.5
Stiklinis butelis	209	20 000	95.69
Lazdyno riešutas	391	20 000	51.15

3. Nagrinėjamų neprižiūravimo mokymosi algoritmų metodinė ir eksperimentinė dalis

Šiame darbe gaminių kokybės nustatymui yra tiriamas autoenkoderinių neprižiūravimo mokymosi neuroninių tinklų efektyvumas. Eksperimentinės dalies procesas dalinamas į keturis etapus:

- įrangos pasirinkimas;
- autoenkoderinių modelių sukūrimas ir apmokymas;
- algoritmų rezultatų, identifikuojančių anomalijas, palyginimas;
- tyrimo apibendrinimas.

3.1. Tyrime naudota įranga

Pastaraisiais metais išaugus debesies sprendimų paklausai nemažai IT kompanijų pradėjo teikti paslaugas, padedančias per internetą vartotojui naudotis nuotoliniu serveriu. Vienas tokių produktų yra *Google Colaboratory* arba *Colab* – tai *Jupyter* tipo sąsiuvinis, įgalinantis vartotoją interaktyviai rašyti programas Python kalba ir tai daryti iš bet kurio kompiuterio. Kadangi mašininio mokymosi algoritmų treniravimasis reikalauja daug resursų, o *Colab* suteikia prieigą prie galingų vaizdo procesorių, būtent dėl to tyrimui ir buvo pasirinkta ši kūrimo aplinka. Eksperimente tiriami dirbtiniai neuroniniai tinklai buvo apmokomi naudojant atitinkamai vieną iš trijų vaizdo plokščių: *Tesla K80*, *Tesla P100-PCIE* ir *Tesla T4*, turinčių 25.46 GB prieinamą operatyviają atmintį.

Duomenų manipuliacijai ir apdorojimui naudota *Numpy* biblioteka, modelių kūrimui ir mokymuisi – *Tensorflow* biblioteka, o rezultatams atvaizduoti – *Matplotlib*, treniravimo proceso atvaizdavimui ir įraymui – *Tensorboard*. Visos jos yra populiaros atviro naudojimosi bibliotekos, suteikiančias apstų ir įvairiapusį spektrą funkcijų.

3.2. Autoenkoderinių modelių sukūrimas ir apmokymas

Šiame tiriamajame darbe buvo taikomi skirtingi autoenkoderiniai gilieji neprižiūravimo mokymosi neuroniniai tinklai. Kad tiriant skirtingas metodikas būtų galima lyginti vienus modelius su kitais, visos algoritmų struktūros buvo išvedamos iš bazinio autoenkoderio.

Šiame darbe išvestinių algoritmų modifikacijos yra skirstytos į keturias kategorijas:

- algoritmų su latentinės erdvės pokyčiais;
- algoritmų su architektūriniais pokyčiais;
- algoritmų su duomenų pokyčiais;
- algoritmų su nuostolių funkcijos pokyčiais;

Kad mokymosi metu būtų pasiektas maksimalus rezultatas, svarbu nesustabdyti mokymosi per anksti, kai dar nepasiektas gradientinis minimumas, ir ne per vėlai, kai modelis tampa persimokęs (angl. *overfitting*). Tam tikslui šiame darbe buvo naudojamos dvi priemonės: ankstyvas stabdymas ir geriausiai pasirodžiusio modelio išsaugojimas.

Pirmoji priemonė užtikrina tai, kad modelis nesustos per anksti, dar nebaigęs mokytis – tai yra, užduodamas *kantrybės* (angl. *patience*) koeficientas (P), kuris nurodo, kiek nesėkmingų iteracijų reikia mokymosi stabdymui. Kuo didesnis koeficientas, tuo ilgiau modelis laukia kol įvyks patobulėjimas, tai yra treniravimo ar validavimo nuostolių funkcijos vertės sumažėjimo. Tiriamajame darbe P koeficientas

pasirinktas dešimčiai tūkstančių iteracijų, tai yra tokį skaičių iteracijų algoritmo mokymasis tęsiasi, laukiant ar atsirado naujas lokalus minimumas.

Antroji priemonė – geriausias rezultatą rodančio modelio išsaugojimas - apsaugo nuo persimokymo. Tam tikru metu visi ANN tinklai pradeda prisitaikyti prie treniravimosi duomenų ir taip pradeda nykti generalizavimo savybės. Kad šito išvengti, autoenkoderiniai modeliai buvo saugomi ne po kiekvienos epochos, bet tik parodžius aukštesnius validavimo rezultatus nei prieš tai buvusio išsaugoto geriausio modelio.

Prieš atliekant modelių treniravimus buvo pasirinkti mokymuisi reikalingi parametrai. Nuspręsta, jog visų AE modelių apmokymas daromas su po 20 pavyzdžių turinčiais rinkiniais (angl. *batchsize*). Žinoma, didesnis jų skaičius reikštų greitesnę neuroninio tinklo konvergavimą į minimumą, tačiau tai reikalautų galingesnės vaizdo plokštės.

Buvo pasirinktas 2×10^{-4} mokymosi greičio koeficientas (angl. *learning rate*), nes eksperimentiškai pastebėta, jog VAE ir keli kiti modeliai, dvigubai didinant ar mažinant šį koeficientą, įstringa prastuose lokaliuose minimumuose ar netgi tampa nestabilūs, tai yra klaidos funkcijos vertės pasiekia begalybę.

Mokymosi optimizavimui naudojamas Adam algoritmas, nes būtent jis buvo sėkmingai panaudotas moksliniuose neprižiūrimo mokymosi defektų aptikimo tyrimuose ([4], [14]).

3.2.1. Bazinis AE

Tyrime naudojamas pagrindinis modelis buvo pasirinktas atsižvelgiant į mokslininko P. Bergmann ir bendraautorių pranešme [14], apie SSIM AE algoritmą, pasiūlytą *L2* autoenkoderio architektūrą. Kaip rašo autoriai, būtent tokia struktūra – su tokiais *hiperparametrais* (angl. *hyperparameters*) - yra optimali. Verta paminėti, jog sudarinėjant bazinį modelį, nors ir architektūra bei mokymosi metodika buvo ruošiami pagal anksčiau minimame straipsnyje pateiktą informaciją, rezultatas skiriasi nuo P. Bergmann ir bendraautorių gauto rezultato (žr. 3 lentelę).

3 lentelė. Palyginamas šiame darbe sukurtas ir analogiško [14] straipsnyje naudoto AE.

	Parametrų sk., mln.	Medsraigtis	Lazdyno riešutas	Stiklo butelis
		AUPRC, %	AUPRC, %	AUPRC, %
AE <i>L2</i> (tiriamajame darbe sukurtas bazinis modelis)	1,08	25,9	50,6	31,7
AE <i>L2</i> [P. Bergmann ir kt. 14]	1,20	14,7	59,0	30,8

Bazinį autoenkoderį su *L2* klaidos funkcija sudaro vien konvoliucinio tipo sluoksniai - atsisakyta pilnai sujungtų *Dense* tipo ir sutelkimo pagal maksimumą sluoksnių. Filtrų skaičius, slenkant enkoderio sluoksniais gilyn, didėja, o dekoderiu – mažėja. Išsami šio neuroninio tinklo architektūra yra aprašyta 4 lentelėje.

Iš viso bazinį autoenkoderio modelį sudaro 1,08 mln. parametrų, dešimt užkoduojančių ir dešimt dekoduojančių sluoksnių.

4 lentelė. Bazinė AE architektūra. Čia c - tai spalvos kanalų skaičius. Stiklo butelio ir medsraigčių klasės nuotraukos naudotos nespalvotos, dėl to $c=1$, lazdyno riešuto – spalvotos, tai - $c=3$. Sluoksniai 1-10 sudaro enkoderį, o 10'-1' sudaro dekoderį.

Sluoksnis						Sluoksnio išvesties dydis
x - įvestis						(256, 256, c)
Sluoksnio Nr.	Sluoksnio pavadinimas	Filtrų sk.	Filtro dydis	Žingsnis	Aktyvacijos funkcija	
1	Konvoliucinis	32	(4x4)	2	LeakyReLU	(128, 128, 32)
2	Konvoliucinis	32	(4x4)	2	LeakyReLU	(64, 64, 32)
3	Konvoliucinis	32	(4x4)	2	LeakyReLU	(32, 32, 32)
4	Konvoliucinis	32	(3x3)	1	LeakyReLU	(32,32, 32)
5	Konvoliucinis	64	(4x4)	2	LeakyReLU	(16, 16, 64)
6	Konvoliucinis	64	(3x3)	1	LeakyReLU	(16, 16, 64)
7	Konvoliucinis	128	(4x4)	2	LeakyReLU	(8, 8, 128)
8	Konvoliucinis	64	(3x3)	1	LeakyReLU	(8, 8, 64)
9	Konvoliucinis	32	(3x3)	1	LeakyReLU	(8, 8, 32)
10	z - Konvoliucinis	100	(8x8)	1	-	(1, 1, 100)
10'	Dekonvoliucinis	32	(8x8)	1	LeakyReLU	(8, 8, 32)
9'	Konvoliucinis	64	(3x3)	1	LeakyReLU	(8, 8, 64)
8'	Konvoliucinis	128	(3x3)	1	LeakyReLU	(8, 8, 128)
7'	Dekonvoliucinis	64	(4x4)	2	LeakyReLU	(16, 16, 64)
6'	Konvoliucinis	64	(3x3)	1	LeakyReLU	(16, 16, 64)
5'	Dekonvoliucinis	32	(4x4)	2	LeakyReLU	(32, 32, 32)
4'	Konvoliucinis	32	(3x3)	1	LeakyReLU	(32, 32, 32)
3'	Dekonvoliucinis	32	(4x4)	2	LeakyReLU	(64, 64, 32)
2'	Dekonvoliucinis	32	(4x4)	2	LeakyReLU	(128, 128, 32)
1'	\hat{x} - Dekonvoliucinis	c	(1x1)	2	sigmoid	(256, 256, c)

Visi sluoksniai turi pritaikytą *padding* techniką, neleidžiančią sluoksnio išvesčiai sumažėti daugiau nei nurodo žingsnis.

$$w_{out} = \frac{(w_{in} - f_w + 2p)}{s} + 1, \quad (23)$$

w_{out} – sluoksnio išėjimo kraštinės dydis;

w_{in} – sluoksnio įėjimo kraštinės dydis;

f_w – filtro dydis;

s – žingsnis;

$2p$ – du papildomi *padding* neuronai.

Kai *padding* technika nėra pritaikoma, $p=0$ ir sluoksnio išėjimo kraštinės dydžio formulė tampa:

$$w_{out} = \frac{(w_{in} - f_w)}{s} + 1 \quad (24)$$

Visų *LeakyReLU* aktyvacijos funkcijų neigiamos ašies koeficientas yra 0.2:

$$f(x) = \begin{cases} x, & \text{kai } x > 0, \\ 0.2x, & \text{kai } x \leq 0 \end{cases} \quad (25)$$

3.2.2. AE su modifikuota latentine erdve

Galima teigti, jog vienas svarbiausių AE veikimo *hiperparametru* - latentinės erdvės z dydis. Tyrime buvo analizuojamos keturių skirtingų dydžių latentinės erdvės. Be bazinio AE su $z \in R^{1 \times 1 \times 100}$ taip pat buvo tiriami atitinkamų dydžių latentinės erdvės autoenkoderiai:

- 1) $z \in R^{1 \times 1 \times 50}$
- 2) $z \in R^{1 \times 1 \times 300}$
- 3) $z \in R^{2 \times 2 \times 8}$

Dažnai (pavyzdžiui, [21] ir [14]) darbuose yra naudojami 100 neuronų dydžio *butelio kaklelio* (angl. *bottleneck*) autoenkoderiai, tačiau jie gali varijuoti – pasirenkami tiek 128-uonių ([4]), tiek ir 512-os ([24]) neuronų dydžiai. Moksliniame darbe [25] rašoma apie erdvinės z formos autoenkoderį ir gautus aukštesnius rezultatus nei su tipiniu z vektoriumi diagnozuojant patologijas iš MRI nuotraukų. Tiriamajame darbe, norint išanalizuoti skirtingo dydžio butelio kaklelio neuronų įtaką tikslumui, šis parametras buvo atitinkamai manipuluojamas (žr. 4 lentelės 10-as sluoksni), o rezultatai lyginami tarpusavyje. Tai yra, sukurtas autoenkoderis su 1) dvigubai trumpesniu z vektoriumi; 2) tris kartus didesniu z vektoriumi; 3) erdvine z matrica.

Teoriškai labai siauras butelio kaklelis (1 atvejis) turėtų priversti modelį išskirti tiktai pačią svarbiausią informaciją apie vaizdą, o išvestis – negebėti atkurti smulkių detalių, kai tuo tarpu platus butelio kaklelis (2 atvejis) turėtų leisti autoenkoderiui užkoduoti daugiau informacijos apie įvestį, taip atkuriant detalesnį vaizdą, bet tikėtina kartu atkuriant ir pačius defektus, kas yra nepageidautina. Analizuojant erdvinį butelio kaklelio autoenkoderį (3 atvejis) tikimasi, jog neuroninis tinklas gebės įsisavinti daugiau globalių savybių apie vaizdą ir taip tiksliau atkurti objektą be anomalijų.

3.2.3. Architektūriškai modifikuoti AE

1) AE su *Dense*-tipo sluoksniais

Moksliniame darbe „Neprižiūrimo mokymosi autoenkoderiai anomalijų segmentavimui iš MR nuotraukų: lyginamasis tyrimas“ (angl. „*Autoencoders for unsupervised anomaly segmentation in brain MR images: A comparative study*“) [4] aprašomi autoenkoderiniai modeliai bei daugelis populiariausių prižiūrimo mokymosi konvoliucinių neuroninių tinklų, kurie dažniausiai gilesniuose sluoksnuose, sumažėjus filtrų dimensijoms, pereina iš konvoliucinių į *Dense*-tipo pilnai sujungtus sluoksnius. Tyrime [23] mokslininkai panaudojo vien tik *Dense* tipo sluoksnių AE modelį ir taikant *MNIST* bei *MNIST-fashion* duomenų bazes parodė itin aukštus rezultatus. Šiame darbe buvo išbandyta modifikacija, kai 10-as ir 10⁶-as sluoksniai (žr. 4 lentelę) buvo pakeisti į pilnai sujungtus, informaciją pirmyn skleidžiančius neuroninius sluoksnius, kuomet neuronų kiekis – 1024, bei

pridėtas dar vienas, vienuoliktas, *Dense* tipo sluoksnis su 100 neuronų, taip išgaunant panašų z erdvės vektorių kaip ir baziniame AE.

2) AE su 5x5 filtrais

Bazinis AE su 3x3 dydžio kerneliu konvoliuciniame sluoksnyje yra populiarus pasirinkimas modeliuojant autoenkoderį. Tačiau tai nėra būtinybė. Jau minėtame tyrime apie AE panaudojimą ligoms diagnozuoti iš MRI nuotraukų [4] yra analizuojami neuroniniai tinklai, naudojantys 5x5 dydžio filtras. Didesnių filtrų privalumas pasireiškia gebėjime ištraukti globalią vaizdo informaciją, kai tuo tarpu mažesni filurai reprezentuoja lokalius vaizdo motyvus. Siekiant išsiaiškinti didesnių filtrų poveikį anomalijų identifikavimui buvo tiriamas autoenkoderis, turintis 5x5 dydžio konvoliucinių sluoksnių filtras (žr. architektūrą priede nr. 1).

3) AE su padidintu filtrų skaičiumi

Dar viena tiriama autoenkoderių modifikacija - filtrų skaičiaus padidinimas. Bazinio modelio (žr. 4 lentelę) enkoderio ir dekoderio filtrų skaičius buvo atitinkamai dvigubai padidintas filtras (žr. architektūrą priede nr. 2). Esant papildomam kiekiui kernelių, neuroninis tinklas įsisavina daugiau nuotraukos savybių, o vienintelis šios modifikacijos trūkumas – suprastėjusi greitimeika ir lėtesnis mokymasis.

4) AE Denner'o architektūra

Bazinis Bergmann'o ir jo komandos [14] pasiūlytas AE šiame darbe taip pat buvo lyginamas su Denner'o ir bendraautorių [4] rekomenduojamu architektūros autoenkoderiu. Pagrindiniai skirtumai yra tame, jog pastarajame AE modelyje visuose konvoliuciniuose sluoksniuose filtrų poslinkio žingsnis - 2, filtrų dydis - 5x5, kiekvienas toks sluoksnis yra reguliuojamas rinkinio normalizavimo technika, enkoderio pabaigoje ir dekoderio pradžioje naudojami *Dense*-tipo sluoksniai ir latentinės erdvės vektorius dydis yra 128 neuronai filtras (žr. architektūrą priede nr. 3).. Rinkinio normalizacijos sluoksnis yra gan dažnas pasirinkimas neuroninio tinklo reguliarizavimui, jį taip pat aptinkant objektų defektus naudojo ir tyrėjas H. Chung su bendraautoriais parodę aukštus neprižiūrimo mokymosi algoritmų rezultatus [22].

3.2.4. AE su modifikuotomis klaidos funkcijomis

Be architektūrinių pokyčių, taip pat buvo modifikuojama klaidos funkcija.

1) *L1* AE

L1 nuostolių funkcija:

$$\operatorname{argmin}_{\varphi, \theta} \mathcal{L}_{AE \text{ tinklas}}^{\varphi, \theta}(x, \hat{x}) = l_1(x, \hat{x}) \quad (26)$$

2) Variacinis AE

Nuostolių funkcija su latentinės erdvės skirstinio reguliarizavimu:

$$\operatorname{argmin}_{\varphi, \theta} \mathcal{L}_{VAE \text{ tinklas}}^{\varphi, \theta}(x, \hat{x}) = l_2(x, \hat{x}) + \lambda_{KL} D_{KL}(q(z)||p(z)) \quad (27)$$

Kur $\lambda_{KL} = 0,5$.

3) AE suvaržytas

Nuostolių funkcija su latentinės erdvės reguliarizavimu:

$$\operatorname{argmin}_{\varphi, \theta} \mathcal{L}_{AE \text{ tinklas}}^{\varphi, \theta}(x, \hat{x}, z, \hat{z}) = l_2(x, \hat{x}) + \lambda_{\text{suvaržymo}} l_2(z, \hat{z}) \quad (28)$$

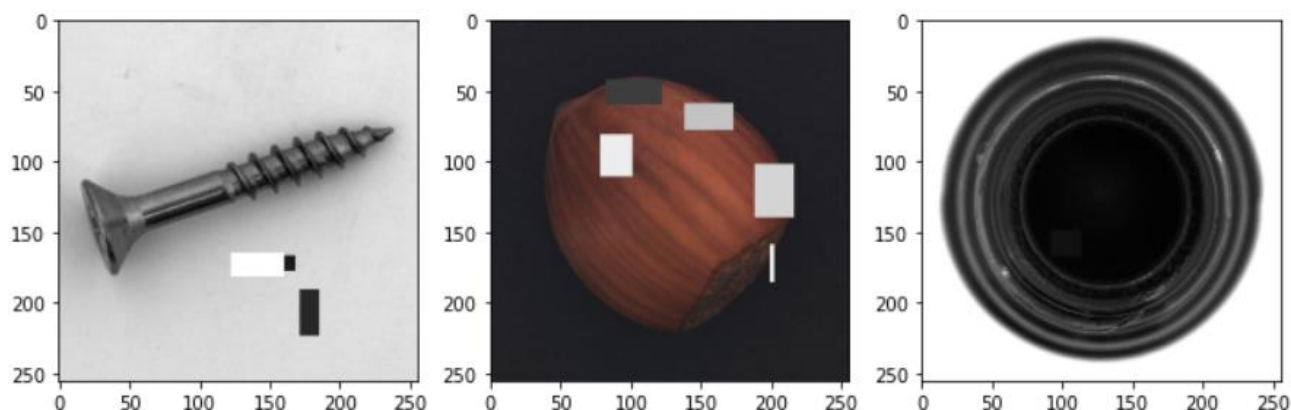
Kur $\lambda_{\text{suvaržymo}} = 1$.

Naudojant LI daroma prielaida, jog ši nuostolių funkcija mažiau koreguos vidinius svorius pagal rezultatus ekstremumuose, nes atgalinės sklaidos technika keis parametrus proporcingai klaidai tarp x ir \hat{x} , taip pagerinant autoenkoderinio modelio rekonstrukcijos gebėjimus. Tuo tarpu su variacinio AE ir suvaržyto AE tinklo klaidos funkcijomis daroma prielaida, kad tai padės latentinę erdvę padaryti tolydesnę ir taip skatins modelį įsisavinanti tik esmines objekto savybes.

3.2.5. AE su duomenų pakeitimais

Autoenkoderio modifikacijos architektūra su duomenų pakeitimais yra identiška bazinio AE modelio architektūrai, tik modelis treniruojamas su duomenimis, kurie yra atsitiktinai užtušuojami.

Maskavimo procesas vyksta pakeičiant įvesties nuotraukos atsitiktinių pikselių regionus į vientisos spalvos pilko atspalvio dėmes. Kiekvienai treniravime naudojamai fotografijai yra atsitiktinai užduodama: regionų skaičius nuo 0 iki 5; regiono dydis nuo 3×3 iki 40×40 ; pozicija vaizde; regiono spalva – baltos, pilkos ir juodos atspalviai.



10 pav. Vaizdų pavyzdžiai naudojami kontekstinio AE apmokymui

3.3. Eksperimentai

Bandymai buvo atliekami su medsraigčių, lazdyno riešutų ir stiklo butelių fotografijomis, iš kurių didžioji dalis turėjo vieną ar daugiau anomalijų, priklausančių skirtingoms defektų kategorijoms (žr. 5 lentelėje). Analizuojamiems autoenkoderio modeliams rekonstravus vaizdą buvo skaičiuojami trys įverčiai AUROC, AUPRC ir AUIoU. Kadangi keičiant binarizavimo slenkstį FPR santykis kinta eksponentiškai, dėl to slenkstinė reikšmė buvo didinama taip pat eksponentiškai pagal formulę:

$$t = 5 \times 10^{-5} \times i^2 \quad (29)$$

Kur t – slenkstinė vertė;

i – iteracija, kintanti nuo 0 iki 140.

Tokiu būdu slenkstinė reikšmė vis didėjančiu žingsniu buvo keičiama nuo 0 iki 0.98 vertės ir iš gautų binarinių verčių skaičiuojamos TPR, FPR, PR ir IoU reikšmės. Pagal šiuos rezultatus sudarytos priklausomybių charakteristikos ir apskaičiuotos AUROC, AUPRC ir AUIoU vertės.

Kadangi IoU įvertis prie aukštesnių FPR reikšmių yra labai mažas ir gaunamas AUIoU rezultatas tampa žemos vertės bei tarp skirtingų algoritmų gaunamos mažai varijuojančios AUIoU vertės, dėl to skaičiuojamas IoU integralas buvo apribojimas iki 30% FPR, o gautas ruožas normalizuojamas.

5 lentelė. Testavimui skirti duomenys

Objektas	Nuotraukų be defektų sk.	Nuotraukų su defektais sk.	Defektų kategorijų sk.
Stiklo butelis	20	63	3
Lazdyno riešutas	40	70	4
Medsraigtis	41	119	5

3.3.1. AE modifikacijų greitaveika

Rekonstruojant tūkstantį nuotraukų iš visų trijų objektų kategorijų ir matuojant vidutinį aptikimo laiką su kiekviena AE modifikacija buvo atliktas ir greitaveikos tyrimas. Nors vaizdus įvedant į modelį grupėmis, tarkim po 32, GPU procesorius greičiau atliktų rekonstrukciją nei įvedant po vieną, tačiau šiame eksperimente buvo iširta greitaveika nuotraukas rekonstruojant atskirai, taip atkartojant panaudojimą realiose sąlygose. Rezultatai pateikiami 6 lentelėje.

6 lentelė. Modelių greitaveika. Čia greitaveika apskaičiuota kaip vidutinė vertė iš tūkstančio modelio rekonstruotų vaizdų, neuroniniam tinklui paduodant po vieną nuotrauką.

Modifikacija	Parametrų sk., mln.	Vidutinis rekonstrukcijos laikas, ms
Bazinis AE	1,08	40
AE $z \in R^{1 \times 1 \times 50}$	0,88	39
AE $z \in R^{1 \times 1 \times 300}$	1,90	40
AE $z \in R^{8 \times 8 \times 2}$	0,73	39
AE $f=5 \times 5$	1,17	39
AE su padidintu filtrų sk.	3,73	40
AE su Dense sluoksniais	0,90	40
AE Denner'o architektūra	2,47	52
LI AE	1,08	40
Suvaržytas AE	1,08	39
Variacinis AE	1,29	51
Kontekstinis AE	1,08	39

Iš gautų greitaveikos trukmių matyti, jog autoenkoderinių algoritmų vyksmo sparta nepriklauso nuo parametrų skaičiaus. Visi modeliai, išskyrus AE Denner'o ir variacinį AE, parodė panašų rekonstrukcijos laiką – apie 39-40 milisekundžių. Kontekstinis, suvaržytas, LI ir bazinis autoenkoderiai turi identišką architektūrą ir turi po lygiai parametrų dėl to greitaveika buvo vienoda. Tuo tarpu skirtingų architektūrų ir skirtingo dydžio AE modeliai su parametrų skaičiumi,

varijuojančiu nuo 0,73 (erdvinės z matricos autoenkoderis) iki 3,73 (AE su padidintu skaičiumi filtrų) milijonų, parodė neįtikėtinai panašų rekonstravimo laiką. Tai galima paaiškinti tuo, jog GPU procesoriai naudoja paralelinius skaičiavimus ir dėl to matricų daugyba su skirtingais filtrais bei pavienėmis jungtimis, nepaisant jų kiekio, yra atliekama vienu metu ir dėl to greitaveika išlieka nepakitusi, nepaisant skirtingo bendro modelio parametrų kiekio.

Ilgiau užtruko variacinis autoenkoderis - vidutiniškai 51 milisekundę. Variacinis AE modelis turi atlikti perparametrizavimo žingsnį, verčiant μ ir σ tikimybę nusakančius vektorius į konvencinę latentinę erdvę, kas greičiausiai ir lėtina šios struktūros algoritmą. Taip pat neženkliai ilgiau nei dauguma algoritmų rekonstrukciją atliko Denner'o architektūros autoenkoderis - vidutiniškai užtrukdamas 52 milisekundes. Nėra tiksliai aišku kas sukėlė lėtesnę greitaveiką, tačiau galima daryti prielaidą, jog tam įtakos turėjo reguliarizavimui skirti rinkinio normalizavimo sluoksniai.

Apibendrinant, atlikus eksperimentą, galima teigti, jog visos autoenkoderio modifikacijos rekonstrukciją atliko gan sparčiai - trukdamos neilgiau nei vieną dešimtąją sekundės.

3.3.2. AE latentinės erdvės

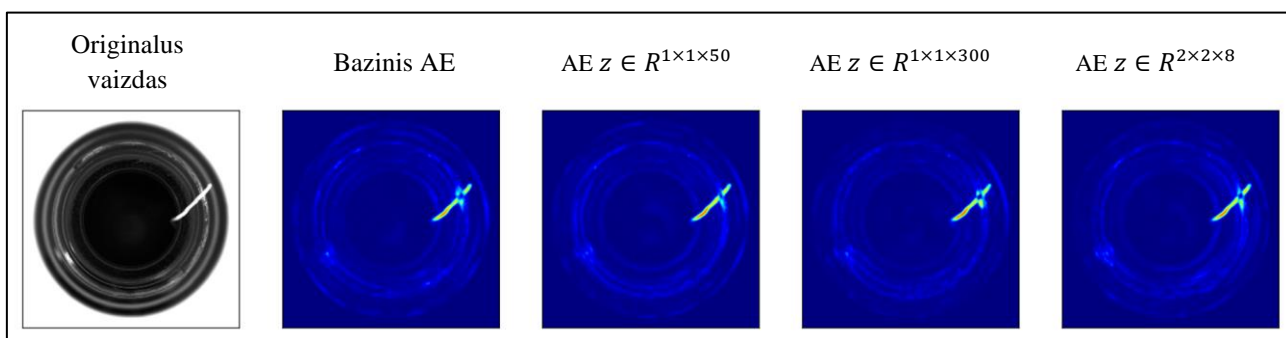
Ištirus skirtingų dimensijų latentinės erdvės AE modelius, pastebėta, kad vieno tiksliausio algoritmo nėra (žr. 7 lentelę ir 11 pav.). Nors tikėtasi, jog su mažesnių dimensijų z matrica autoenkoderis rekonstruos vaizdą prasčiau, tačiau visi rezultatai rodo, jog tikslumas išlieka aukštas. Taip pat matyti, jog hipotezė, kad erdvinis butelio kaklelis (AE su $z \in R^{8 \times 8 \times 2}$) leis modeliui įsisavinti daugiau vaizdo savybių ir taip tiksliau atkurti objektą be defektų, nesutapo su rezultatais - ir AUROC, ir AUPRC, ir AUIoU charakteristikos rodė panašias vertes. Tarp bazinio AE ir jo trijų modifikacijų tikslumo reikšmės mažiausiai tarpusavyje skyrėsi AUROC ir AUIoU, o didžiausias skirtumas iš trijų įverčių buvo pastebėtas tarp AUPRC reikšmių.

7 lentelė. AE latentinės erdvės modifikacijų rezultatai. Geriausiai pasirodęs modelis paryškintas.

Modifikacija	Objektai								
	Medsriegis			Lazdyno riešutas			Stiklo butelis		
	Tikslumo įverčiai, %								
	AUROC	AUPRC	AUIoU	AUROC	AUPRC	AUIoU	AUROC	AUPRC	AUIoU
Bazinis AE	94,10	25,91	2,64	92,92	50,61	14,43	84,81	31,69	18,37
AE $z \in R^{1 \times 1 \times 50}$	94,44	26,89	2,55	92,80	51,79	14,12	84,67	31,01	18,28
AE $z \in R^{1 \times 1 \times 300}$	93,73	25,45	2,60	92,25	45,92	14,16	84,99	32,95	18,37
AE $z \in R^{8 \times 8 \times 2}$	93,74	25,45	2,50	92,50	49,04	14,50	85,69	34,18	19,29

Taip pat iš gautų tikslumo įverčių galima įžvelgti ir priklausomai nuo objekto skirtingus modelių gebėjimus aptikti defektus. AE su $z \in R^{1 \times 1 \times 50}$ latentine erdve labiausiai varijavo aptinkant anomalijas tarp skirtingų objektų - gauta aukščiausia AUPRC 26,89% reikšmė, aptinkant defektus medsriegyje, ir žemiausia - AUPRC 31,01% reikšmė, aptinkant defektus stiklo butelyje. Ne mažiau varijuojantys rezultatai gauti ir su AE $z \in R^{8 \times 8 \times 2}$ latentinės erdvės modeliu, kuris parodė beveik prasčiausius -

25,45% AUPRC rezultatus, ieškant medsriegių defektų, ir geriausius - 34,18% AUPRC rezultatus, randant stiklo butelio anomalijas.



11 pav. Butelio su užteršimu anomalijų žemėlapis su skirtingų latentinių erdvių AE modeliais

3.3.3. AE architektūros

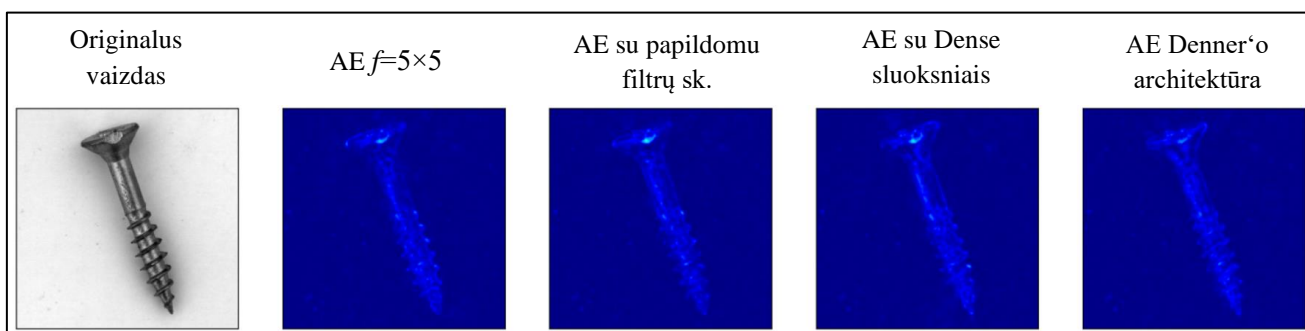
Vertinant architektūrinių pokyčių įtaką tikslumui, galima pastebėti, jog beveik prie visų trijų įverčių su visomis trimis objektų kategorijomis iš architektūriškai modifikuotų autoenkoderių Denner-tipo modifikacija parodė žemiausius rezultatus (žr. 8 lentelę). Ši architektūra nuo bazinio modelio skiriasi filtrų didumu (5×5), filtrų kiekiu, pilno sujungimo ir reguliarizacijos sluoksnių panaudojimu. Visi šie skirtumai, išskyrus rinkinio normalizacijos sluoksnį, buvo izoliuoti ir ištirti kaip atskiros modifikacijos, kurios parodė panašesnius į bazinio autoenkoderio rezultatus, dėl to labiausiai tikėtina, jog AE Denner'o neprižiūrimo mokymosi modelis veikė prasčiau nei kiti būtent dėl eksploatuojamo rinkinio normalizacijos defektams aptikti sluoksnio.

8 lentelė. AE architektūros modifikacijų rezultatai. Geriausiai pasirodęs modelis paryškintas.

Modifikacija	Objektai								
	Medsriegis			Lazdyno riešutas			Stiklo butelis		
	Tikslumo įverčiai, %								
	AUROC	AUPRC	AUIoU	AUROC	AUPRC	AUIoU	AUROC	AUPRC	AUIoU
Bazinis AE	94,10	25,91	2,64	92,92	50,61	14,43	84,81	31,69	18,37
AE $f=5 \times 5$	94,13	26,40	2,64	92,87	50,51	14,46	83,03	27,58	16,18
AE su papildomu filtrų sk.	94,50	29,91	2,73	93,02	51,35	14,56	86,33	36,16	19,96
AE su Dense sluoksniais	94,00	25,21	2,62	92,97	49,68	14,23	84,28	29,86	17,70
AE Denner'o architektūra	93,10	18,82	2,48	92,38	47,72	13,84	83,39	30,60	17,46

Taip pat galima išskirti AE modifikaciją su didesniu kiekiu filtrų. Ji AUROC, AUPRC ir AUIoU įverčiais parodė didžiausią tikslumą, ieškant defektų visose trijose objektų kategorijose. Hipotezė,

jog AE su platesne filtrų apimtimi kiekviename sluoksnyje geba nuodugniau įsisavinti objektų vaizdines savybes, taip tiksliau rekonstruojant gaminį be anomalijų, pasitvirtino.

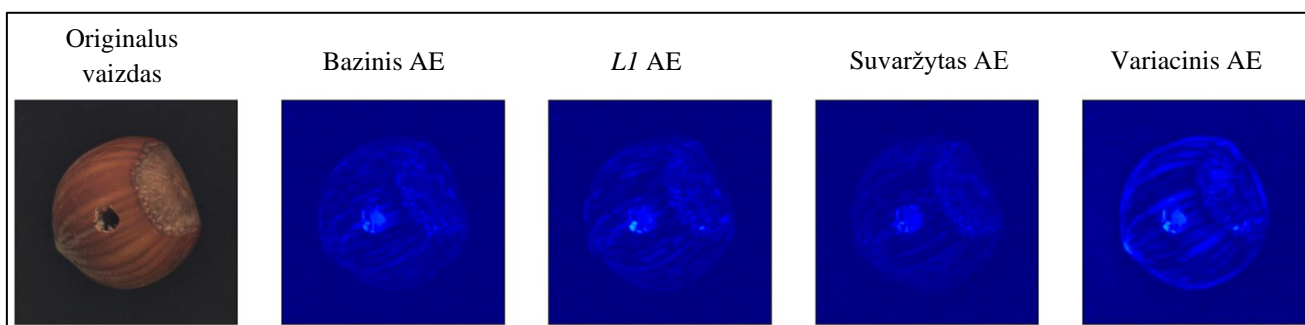


12 pav. Medsriegio galvutės deformacijos anomalijų žemėlapis su skirtingų architektūrų AE modeliais

AE modifikacijos su pilnai sujungtais sluoksniais butelio kaklelyje rezultatai bei AE modifikacijos su pakeistu filtrų dydžiu į 5×5 rezultatai yra gana panašūs į bazinio autoenkoderio. Iš tyrimo rezultatų galima daryt prielaidą, jog konvoliucinių sluoksnių pakeitimas į *Dense*-tipo sluoksnius enkoderio pabaigoje ir dekoderio pradžioje, kaip ir didesnės dimensijos kernelių naudojimas, didelės įtakos rezultatams neturėjo. Tik ieškant anomalijų stiklo butelyje matomas nežymus skirtumas tarp bazinio ir 5×5 filtrų turinčio modelio, kur AE neuroninis tinklas su 5×5 filtrais mažiau tiksliai identifikavo defektus.

3.3.4. AE klaidos funkcijos

Analizuojant skirtingų klaidos funkcijų autoenkoderius nustatyta, jog bazinio modelio *L2* nuostolių funkcija yra efektyviausia, rodanti aukštus AUROC, AUPRC ir AUIoU rezultatus medsriegio, lazdyno riešuto ir stiklo butelio anomalijų aptikimuose (žr. 9 lentelę). Jei atsižvelgsime tik į lazdyno riešuto ir stiklo butelio defektų radimo rezultatus, tai galima matyti, jog ir *L1* AE, ir suvaržytas AE, ir variacinis AE tikslumu statistiškai baziniam autoenkoderiui nenusileidžia ar net lenkia patį. Tačiau reikšmingas skirtumas stebimas medsriegių nuotraukų kategorijoje. Čia, pagal visas tris kokybes metrikas, tiriami autoenkoderiniai modeliai su modifikuota klaidos funkcija parodė prastesnio tikslumo rezultatus. Pagrindinis pastebimas vaizduose skirtumas tarp medsriegio ir lazdyno riešuto bei stiklo butelio yra vaizdo simetriškumas. Sukant butelį ir riešutą fotografijoje objektas vis vien užpildo panašią erdvę ir veidrodiniu principu lankstant vaizdą yra daugiau ar mažiau gaunamas simetriškumas, kai tuo tarpu medsriegio pozicija vis keičiasi ir stebimas simetrijos nebuvimas.



13 pav. Lazdyno riešuto su skyle anomalijų žemėlapis taikant skirtingų klaidos funkcijų AE modelius

9 lentelė. AE modifikacijų su skirtingomis klaidos funkcijomis rezultatai. Geriausiai pasirodęs modelis yra paryškintas.

Modifikacija	Objektai								
	Medsriegis			Lazdyno riešutas			Stiklo butelis		
	Tikslumo įverčiai, %								
	AUROC	AUPRC	AUIoU	AUROC	AUPRC	AUIoU	AUROC	AUPRC	AUIoU
Bazinis AE	94,10	25,91	2,64	92,92	50,61	14,43	84,81	31,69	18,37
<i>LI</i> AE	93,80	19,25	2,48	92,63	50,78	14,53	85,56	32,31	18,59
Suvaržytas AE	92,15	17,39	2,39	90,76	48,06	13,74	85,94	32,62	19,56
Variacinis AE	92,15	17,34	2,31	90,51	50,71	13,80	86,18	35,66	18,85

Galima prielaidyti, kodėl suvaržytas ir variacinis autoenkoderiai geba tiksliau aptikti simetriškus ir kiek aptakesnius objektus, yra tai, kad abu šie modeliai savo klaidų funkcijomis bando sugretinti latentinės erdvės reikšmes taip, kad keičiant tolygiai įvestyje esantį objektą, z vertės irgi kistų tolygiai vienodai.

3.3.5. AE su duomenų pakeitimu

Paskutinis tiriamasis autoenkoderis - kontekstinis AE. Kaip ir buvo numatyta, atsitiktinių regionų duomenyse užtušavimas privertė modelį ieškoti globalesnių savybių ir efektyviau nuspėti, kaip turi būti užkoduojama ir dekoduojama informacija, kad rekonstrukcija būtų kuo artimesnė objekto vaizdui be anomalijų. Iš rezultatų, pateiktų 10 lentelėje, matyti, jog pagal AUPRC įvertį kontekstinis AE tiksliau aptiko defektus už bazinį autoenkoderį. Pastarasis tik neženkliu AUIoU įverčio skirtumu buvo pranašesnis, aptinkant anomalijas medsriegiuose, ir nereikšmingu AUROC kokybės mato skirtumu pranašesnis, identifikuojant anomalijas lazdyno riešute.

10 lentelė. AE su duomenų pakeitimu rezultatai. Geriausias rezultatas paryškintas.

Modifikacija	Objektai								
	Medsriegis			Lazdyno riešutas			Stiklo butelis		
	Tikslumo įverčiai, %								
	AUROC	AUPRC	AUIoU	AUROC	AUPRC	AUIoU	AUROC	AUPRC	AUIoU
Bazinis AE	94,10	25,91	2,64	92,92	50,61	14,43	84,81	31,69	18,37
Kontekstinis AE	94,25	28,48	2,58	92,88	51,92	14,66	87,42	41,41	21,10

3.3.6. Autoenkoderiai neprižiūravimo mokymosi kontekste

Defektams aptikti atlikto autoenkoderinių modelių tyrimo rezultatai pagal AUPRC kokybės matą buvo palyginti tarpusavyje ir taip pat su moderniaisiais ir visuotinai pripažintais pažangiaisiais

algoritmais pristatytais „MVTEC anomalijų aptikimo duomenų bazė: išsami realaus pasaulio duombazė neprižiūravimo mokymosi algoritams“ (angl. „The MVTEC Anomaly Detection Dataset: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection“) [20] pranešime (žr. 11 lentelę).

11 lentelė. AUPRC rezultatų palyginimas (%)

Modifikacijos tipas	Modelis	Medsriegis	Lazdyno riešutas	Stiklo butelis	Vidurkis
-	Bazinis AE	25,9	50,6	31,7	36,1
Latentinės erdvės pakeitimai	AE $z \in R^{1 \times 1 \times 50}$	26,9	51,8	31,0	36,8
	AE $z \in R^{1 \times 1 \times 300}$	25,5	45,9	33,0	34,8
	AE $z \in R^{8 \times 8 \times 2}$	25,5	49,0	34,2	36,2
Architektūriniai pakeitimai	AE $f=5 \times 5$	26,4	50,5	27,6	34,8
	AE su padidintu filtrų skaičiumi	29,9 (2)	51,4	36,2	39,2
	AE su Dense sluoksniais	25,2	49,7	29,9	34,9
	AE Denner'o architektūra	18,8	47,7	30,6	32,4
Klaidos funkcijos pakeitimai	AE $L1$ nuostolių funkcija	19,3	50,8	32,3	34,1
	Suvaržytas AE	17,4	48,1	32,6	32,7
	Variacinis AE (šiam darbe tiriamas)	17,3	50,7	35,7	34,6
Duomenų pakeitimai	Kontekstinis AE	28,5 (3)	51,9	41,4	40,6 (3)
MVTEC AD pranešime [20] pasiūlyti algoritmai	$L2$ AE	14,7	59 (2)	30,8	34,8
	f-AnoGAN	6,2	52,6	16	24,9
	Savybių žodyno algoritmas	1,7	40,4	81,4 (1)	41,2 (2)
	Studento-mokytojo algoritmas	35,8 (1)	58,5 (3)	77,5 (2)	57,3 (1)
	$SSIM$ AE	5	31,2	30,9	22,7
	Tekstūrinės inspekcijos algoritmas	5,2	68,9 (1)	28,5	33,3
	Variacinis AE	13,8	48,5	53,6 (3)	38,6

Iš šiamo darbe nagrinėjamų autoenkoderinių modelių galima išskirti dvi geriausias rezultatus pateikusias AE modifikacijas - AE su daugiau filtrų, AUPRC vidurkis yra 39,2%, ir kontekstinis AE, AUPRC vidurkis 41,2%. Abu modeliai, ieškant anomalijos medsriegių ir stiklo butelių vaizdiniuose, reikšmingu tikslumo įverčio skirtumu pirmuoja prieš likusius, nagrinėjamus autenkoderinius neuroninius tinklus.

Vertinant tyrimo rezultatus išryškėjo dar vienas aspektas – visi analizuojami autoenkoderiai lazdyno riešutą rekonstravo panašiu tikslumu - gautos AUPRC reikšmės svyravo ties 50% verte ir turėjo palyginti mažą vidutinį kvadratinį nuokrypį.

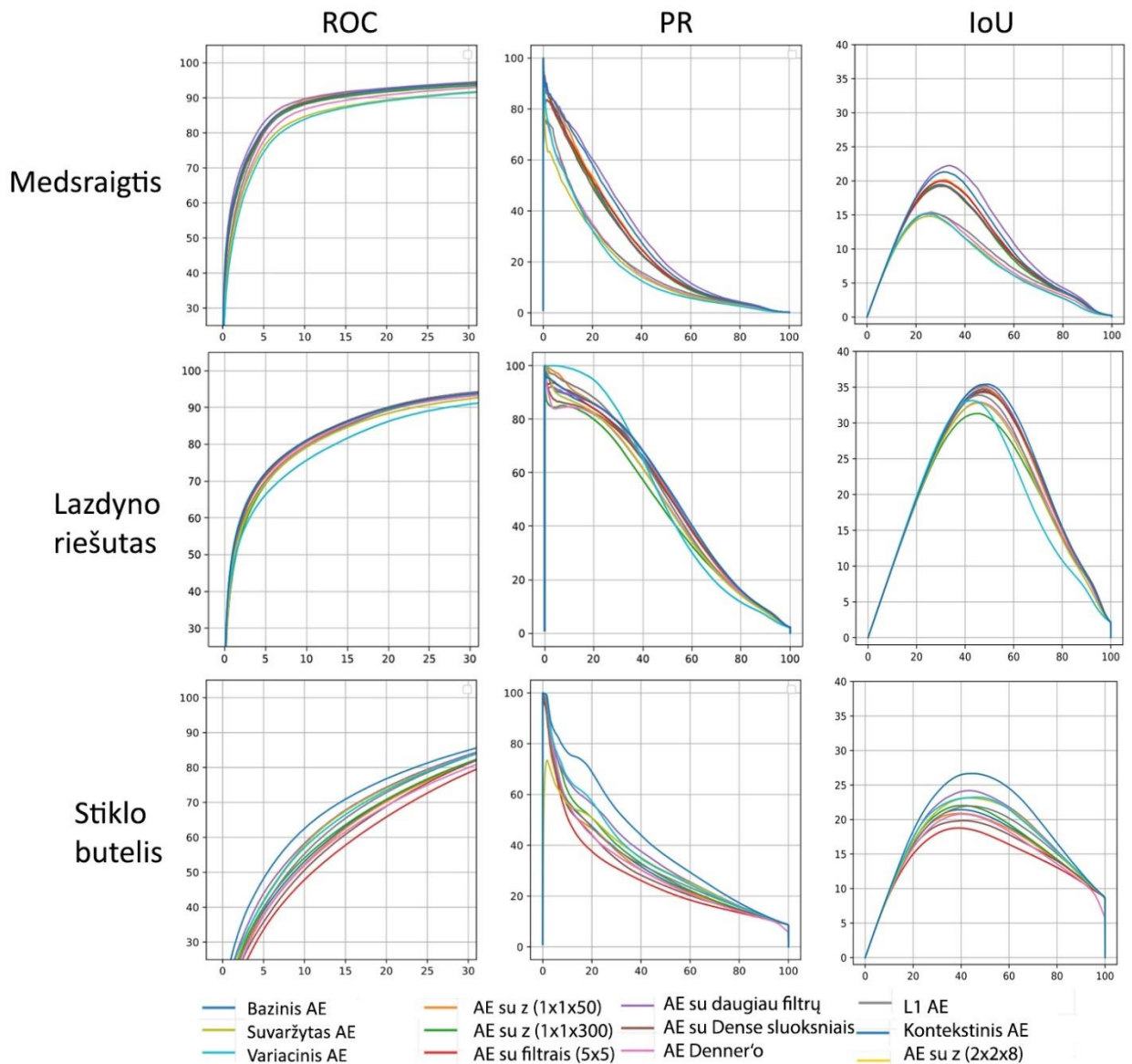
Lyginant su moderniausias neprižiūravimo mokymosi algoritmais defektams aptikti, nei viena autoenkoderio modifikacija nepriėjo 2020 metais pasiūlytam Studento-Mokytojo algoritmui [11], kuris rodo stabiliai aukštus rezultatus su visais trimis vaizde esančiais objektais. Iš 3.3.6 lentelėje pateiktų moderniausių algoritmų (pilka spalva) verta išskirti $L2$ AE ir variacinį AE, nes abu jie turi atitinkamus, šiame darbe nagrinėjamus, modelius, ir nepaisant atitinkamai panašių architektūrų bei gan vienodai parinktų *hiperparametru* AUPRC tikslumo įverčiai rezultate skiriasi. Tai yra šiame tyrime nagrinėjamas bazinis AE parodė ženkliai aukštesnius - 25,9% tikslumo rezultatus, aptinkant defektus medsriegyje, ženkliai žemesnius - 50,6% rezultatus, aptinkant defektus lazdyno riešute, ir panašius - 31,7% rezultatus, aptinkant defektus stiklo butelyje, lyginant su atitinkamomis 14,7%, 59% ir 30,8% $L2$ AE modelio reikšmėmis. Neatitinkamas matomas ir tarp struktūriškai panašių - šiame darbe analizuojamų variacinio AE bei literatūroje taikyto ([20]) variacinio AE - modelių AUPRC tikslumo rezultatų.

Iš 11-oje lentelėje pateiktų algoritmų pavyzdžių galima išvelgti pasikartojančią tendenciją, jog autoenkoderio principu veikiančios neprižiūravimo mokymosi algoritmai gan gerai segmentuoja anomalijas medsriegiuose ir prasčiau lazdyno riešutuose bei stiklo buteliuose, lyginant su algoritmais, naudojančiais regioninius aprašus (angl. *regional descriptors*) bei iš anksto paruoštus modelius, kaip studento-mokytojo ir savybių žodyno algoritmai, kurie santykinai prastai identifikuoja defektus medsriegiuose ir santykinai tiksliai – lazdyno riešutuose bei stiklo buteliuose.

3.3.7. Kiekybės įverčių palyginimas

Darbe tiriamų autoenkoderinių modelių rezultatai vertinami trimis įverčiais - AUROC, AUPRC ir AUIoU (žr. 7-10 lenteles). Mažiausia kokybinių matų variacija buvo gauta su AUROC tikslumo charakteristika. Visų modifikacijų autoenkoderiai parodė panašias ROC kreivės reikšmes dėl to jų atvaizdavimas 11-ame paveiksle buvo susiaurintas iki 30-100% TPR ir 0-30% FPR regiono. Taip pat tarp skirtingų modelių mažai varijavo ir IoU kreivės. Daugmaž su visais AE nagrinėjamais neuroniniais tinklais gauta AUIoU defektų aptikimo reikšmė buvo panaši į tarpusavyje lyginamų AE modelių reikšmes - medsriegių nuotraukose - 2,6%, lazdyno riešuto nuotraukose - 14% ir stiklo butelio nuotraukose - 38%. Kaip ir ROC kreivės, kad geriau išryškėtų skirtumai tarp modelių, IoU kreivės atvaizdavimas buvo taip pat susiaurintas (žr. 14 paveikslą). Skirtingai nei AUROC ar AUIoU, AUPRC rezultatai tarp skirtingų modifikacijų kontrastavo stipriau. Greičiausiai tai lėmė, jog preciziškumo (PRC) matas, kuris eina AUPRC charakteristikos y ašimi, nusako tikslumą su FP vietoj FN įverčiu lygtyje, o kadangi defektai užima palyginti nedidelį viso vaizdo plotą ir yra esminis anomalijų identifikavimo uždavinio tikslas, dėl to modelio gebėjimas sumažinti FP kiekį tampa ypač svarbus.

Taip pat reiktų paminėti, jog nors visų trijų tikslumo įverčių skaitinės reikšmės nesiekė 95%, vis dėlto taikyti nuo slenkščio nepriklausomus kokybės matus, kurie rodo ne tokias dideles skaliarines vertes, yra tikslingiau nei skaičiuojant įverčius prie konkrečios slenkstinės vertės, nes realiomis sąlygomis pramonėje sutinkamuose uždaviniuose ne visada yra galimybė atlikti defektų analizę ir parinkti geriausią vertę algoritmo rekonstrukcijos binarizavimui.



14 pav. Trijų, ROC, PER ir IoU, įverčių grafikai parodantys skirtingų modelių efektyvumą aptikti defektus su atskirų objektų nuotraukomis

3.3.8. AE veiksmingumas su įvairiomis defektų kategorijomis

Galiausiai buvo pasirinkti trys labiausiai savo veikimu tarpusavyje besiskiriantys autoenkoderių algoritmai ir analizuojami defektų aptikimų rezultatai su kiekviena izoliuota objekto anomalijų kategorija. Iš gautų kiekybės įverčių, kurie pateikti 12-oje lentelėje, galima pastebėti daugybę prasmingų tendencijų.

Visų pirma, aiškiai išvelgiama, jog visų trijų modelių geriausių ir prasčiausių AUPRC bei AUIoU tikslumo įverčių aptikimų rezultatai sutapo tarp skirtingų kategorijų anomalijų. Tai yra aukščiausios PRC ir IoU kreivių integralų reikšmės buvo gautos aptinkant kevalo skilimą lazdyno riešute (išimtis suvaržyto autoenkoderio AUPRC vertė, kuri pagal tikslumą buvo antra), o prasčiausios PRC ir IoU kreivių integralo reikšmės apskaičiuotos aptinkant medsriegio srieginės dalies matomos šonu deformaciją. Su pastarosiomis reikšmėmis AUROC įvertis nesutapo. Geriausius rezultatus atitinkamoje defektų kategorijoje - aptinkant medsriegio srieginės dalies deformaciją iš viršaus - parodė visi trys autoenkoderiai: kontekstinis, suvaržytas ir didesnę filtrų skaičių turintis

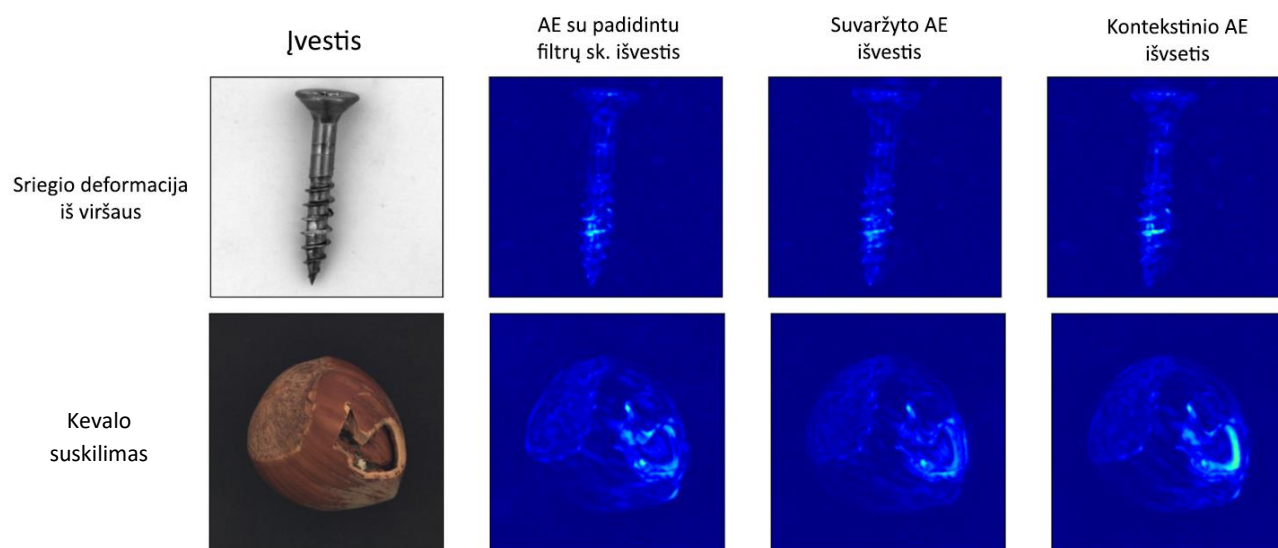
autoenkoderis. Pastarųjų prasčiausias rezultatas atspindėjo aptinkant didžiulius stiklo butelio suskilimus. Pastaroji defektų kategorija, lyginant su kitomis, labiausiai varijavo tarp tikslumo įverčių. Čia, skirtingai nei AUROC, kur gauti žemiausi rezultatai, AUPRC ir AUIoU vertės buvo gautos gana aukštos.

12 lentelė. Trijų skirtingų autoenkoderių defektų aptikimo rezultatai kiekvienai atskirai anomalijų kategorijai. Geriausiai aptinkamos defektų kategorijos paryškintos žalia spalva, prasčiausiai – raudona spalva.

Objektai	Defekto tipas	AE daugiau filtrų			Svaržytas AE			Kontekstinis AE		
		AUROC	AUPRC	AUIoU	AUROC	AUPRC	AUIoU	AUROC	AUPRC	AUIoU
Medsriegis	Smaigalio deformacija	93,26	35,6	3,11	85,39	17,01	2,38	92,54	31,99	2,87
	Galvutės nulūžimas	95,92	34,16	2,91	94,18	13,74	2,46	95,27	31,75	2,85
	Kaklelio nuskilimas	95,53	40,74	3,68	94,38	23,86	3,27	95,33	38,07	3,64
	Sriegio deformacija profiliu	91,52	11,51	2,44	90,95	9,53	2,3	91,52	10,3	2,41
	Sriegio deformacija iš viršaus	95,63	39,64	5,21	94,6	29,55	4,8	95,81	39,9	5,28
Lazdyno riešutas	Kevalo skilimai	92,18	68,41	33,31	90,56	64,69	33,48	91,83	67,47	32,87
	Įpjovimai	91,07	35,27	6,63	90,89	33,55	6,57	91,29	38,12	6,85
	Skylės	93,34	56,72	10,83	92,93	51,75	10,58	93,38	57,48	10,9
	Dažai	95,2	63,18	18,04	95,6	65,99	18,54	95,49	66,53	18,39
Stiklo butelis	Dideli suskilimas	84,93	44,11	27,34	85,19	43,9	27,54	86,39	49,02	29,44
	Maži suskilimai	88,82	37,93	14,43	87,94	28,14	14,11	88,38	38,88	14,78
	Užteršimas	85,7	43,08	25,14	85,5	41,65	24,01	87,83	52,62	28,63

Iš lentelėje pateiktų rezultatų galima daryti prielaidą, jog aukščiausi rezultatai gauti segmentuojant aiškių struktūrinių ir spalvinių pokyčių anomalijas (žr. 15 pav.), nes būtent šie defektai savo šviesumo verte labiausiai išsiskirdavo iš konteksto ir būdavo ypač kontrastingi, lyginant su kokybiška objekto dalimi. Tuo tarpu dauguma prasčiausių rezultatų buvo gauti aptinkant iš šono nufotografuotas sriegio deformacijas medsriegiuose. Nors spalvine verte šis defektas gan ryškiai išsiskiria, tačiau užima

santykinai labai mažą dalį viso objekto bei nedaug varijuoja savo forma nuo kokybiško objekto atitikmens, ir dėl to dažniausiai rekonstrukcijos metu lieka nepastebimas.



15 pav. Trijų skirtingų AE modelių defektų atpažinimo rezultatai su pavyzdžiu iš palankiausiai aptinkamų defektų kategorijų

Išvados ir rezultatai

1. Šiame tiriamajame darbe buvo apžvelgti ir išanalizuoti neprižiūrimo mokymosi dirbtinio intelekto algoritmai anomalijoms vaizduose rasti, jų struktūros, privalumai ir trūkumai. Nustatyta, jog dauguma tokių algoritmų taiko enkoderio ir dekoderio architektūras, o defektą objekte segmentuoja pagal skirtumą tarp rekonstruoto ar sugeneruoto vaizdo ir įvesties. Šių struktūrų mokymui naudoja nežymėtus duomenis be defektų.
2. Tiriant autoenkoderių efektyvumą aptinkant anomalijas objektuose, eksperimentiškai buvo nustatyta, jog latentinės erdvės dydžio modifikavimas tikslumui didelės įtakos neturėjo, t.y. taikant siauresnės, platesnės ar erdvinės z latentinės matricas autoenkoderis rekonstravo anomalijas panašiai kaip bazinis AE, kokybės įvertis AUPRC svyravo apie 36%.
3. Išnagrinėjus įvairių architektūrų autoenkoderių rekonstravimo rezultatus galima daryti išvadą, kad filtrų skaičius didinimas konvoliuciniuose sluoksniuose yra efektyvi priemonė pagerinti AE algoritmo veikimo kokybę aptinkant vaizdų anomalijas. Modelio modifikacija su papildomu filtrų skaičiumi pasiekė 29,91%, 51,35% ir 36,16% AUPRC reikšmes su atitinkamai *medsriegio*, *lazdyno riešuto* ir *stiklo butelio* objektais. Tuo tarpu rinkinio normalizacijos ar pilnai sujungtų sluoksnių naudojimas autoenkoderiuose parodė prastesnius rezultatus nei bazinis modelis, per -7,09%, -2,89% ir -1.09% vertes.
4. Remiantis atliktų bandymų gautais kokybės įverčiais galima teigti, kad efektyviausia nuostolių funkcija buvo $L2$ – naudota baziniame modelyje. Lyginant su pataruoju modeliu, pagal ROC, PRC ir IoU kreivių vertes, ir suvaržytas AE ir variacinis AE, su papildoma reguliacijos dedamąja klaidos funkcijoje, ženkliai prasčiau aptiko defektus *medsraigčio* nuotraukose ir šiek tiek efektyviau stiklo butelyje.
5. Atsižvelgiant į tyrimo rezultatus kontekstinis AE, apmokytas su atsitiktinai užtušuotais duomenimis, parodė aukščiausią veiksmingumą aptinkant anomalijas stiklo buteliuose ir lazdyno riešutuose, AUPRC vidurkis – 41,6%. Šio neuroninio tinklo greitaveika, kaip ir daugelio kitų autoenkoderinių modelių modifikacijų, rekonstruojant nuotrauką, yra ~ 40 milisekundžių.
6. Iš tyrime naudojamų kokybės įverčių AUROC, AUPRC ir AUIoU rezultatų, galima pastebėti, jog šios trys metrikos ne visada atspindi algoritmo tikslumą. Lyginant skirtingus autoenkoderių modelius AUPRC ir AUIoU vertės parodydavo santykinai panašų tikslumą ir nesiekė 60%, kai tuo tarpu AUROC, aptinkant anomalijas, rodydavo skirtingą efektyvumą ir svyrudavo apie 90%. Kadangi defektai sudaro tik mažą dalį visos nuotraukos vaizdo, todėl *sumaišymo matricos* didžiąją dalį pripildo TN reikšmės, kas išbalansuoja FPR charakteristiką prie skirtingų slenkstinių verčių ir ROC kreivė tampa mažiau reprezentatyvi algoritmo tikslumui. Taip pat algoritmų prastam kokybiniam įvertinimui neigiamą įtaką darė netikslus (per daug platus) defektų žymėjimas testiniuose duomenyse.

Literatūros sąrašas

1. R. W. Conners, C. W. Mcmillin, K. Lin. 1983. *Identifying and Locating Surface Defects in Wood: Part of an Automated Lumber Processing System*. [žiūrėta 2022 gegužės 20d.] Prieiga per internetą: <https://ieeexplore.ieee.org/document/4767446>
2. Xianghua Xie. 2008. *A Review of Recent Advances in Surface Defect Detection using Texture analysis Techniques*. [žiūrėta 2022 gegužės 20d.] Prieiga per internetą: <https://pdfs.semanticscholar.org/b809/841f0f45fb91e70f9623cf70bd4514cf4f91.pdf>
3. Y. LeCun, B. Boser, H. S, Denker ir kt. 2019. *Handwritten digit recognition with a back-propagation network*. (1990). [žiūrėta 2022 gegužės 20d.] Prieiga per internetą: <https://papers.nips.cc/paper/1989/file/53c3bce66e43be4f209556518c2fcb54-Paper.pdf>
4. C. Baur, S. Denner, B. Wiestler ir kt. 2021. *Autoencoders for unsupervised anomaly segmentation in brain MR images: A comparative study*. [žiūrėta 2022 gegužės 20d.] Prieiga per internetą: <https://doi.org/10.1016/j.media.2020.101952>
5. G. E. Hinton, S. Osindero ir Y. Teh. 2006. *A fast learning algorithm for deep belief nets*. [žiūrėta 2022 gegužės 20d.] Prieiga per internetą: <https://www.cs.toronto.edu/~hinton/absps/fastnc.pdf>
6. E. Dragašius ir A. Lipnickas. *Intelektualios mechatroninės sistemos*. Vilnius, 2008.
7. J. Deng, W. Dong, R. Socher ir kt. 2009. *Imagenet: a large-scale hierarchical image database*. [žiūrėta 2022 gegužės 20d.] Prieiga per internetą: <https://ieeexplore.ieee.org/document/5206848>
8. K. Simonyan ir A. Zisserman. 2014. *Very deep convolutional networks for largescale image recognition*. [žiūrėta 2022 gegužės 20d.] Prieiga per internetą: <https://arxiv.org/pdf/1409.1556.pdf>
9. Y. LeCun, Y. Bengio ir G. Hinton. 2015. *Deep Learning*. Nature 521, 436-444.
10. S. Ioffe ir C. Szegedy. 2015. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. [žiūrėta 2022 gegužės 20d.] Prieiga per internetą: <https://arxiv.org/pdf/1502.03167.pdf>
11. P. Bergmann, M. Fauser, D. Sattlegger ir kt. 2020. *Uninformed Students: Student–Teacher Anomaly Detection with Discriminative Latent Embeddings*. [žiūrėta 2022 gegužės 20d.] Prieiga per internetą: <https://ieeexplore.ieee.org/document/9157778>
12. J. T. Springenberg, A. Dosovitskiy, T. Brox ir kt. 2015. *Striving for Simplicity: The All Convolutional Net*. [žiūrėta 2022 gegužės 20d.] Prieiga per internetą: <https://arxiv.org/abs/1412.6806>
13. T. He, Y. Liu, X. Zhou ir kt. 2019. *A Fully Convolutional Neural Network for Wood Defect Location and Identification*. [žiūrėta 2022 gegužės 20d.] Prieiga per internetą: <https://ieeexplore.ieee.org/document/8812894>
14. <https://arxiv.org/abs/1807.02011> P. Bergmann, S. Lowe, M. Fauser ir kt. 2019. *Improving Unsupervised Defect Segmentation by Applying Structural Similarity to Autoencoders*. [žiūrėta 2022 gegužės 20d.] Prieiga per internetą: <https://arxiv.org/abs/1807.02011>
15. P. Bergmann, M. Fauser, D. Sattlegger ir kt. 2019. *MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection*. [žiūrėta 2022 gegužės 20d.] Prieiga per internetą: <https://ieeexplore.ieee.org/document/8954181>
16. T. Schlegl, P. Seebock, S. M. Waldstein ir kt. 2017. *Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery*. [žiūrėta 2022 gegužės 20d.] Prieiga per internetą: <https://arxiv.org/abs/1703.05921>

17. D. Pathak, P. Krahenbuhl, J. Donahue ir kt. 2016. *Context Encoders: Feature Learning by Inpainting*. [žiūrėta 2022 gegužės 20d.] Prieiga per internetą: <https://arxiv.org/abs/1604.07379>
Context AE
18. D. Zimmer, S. A. A. Kohl J. Petersen ir kt. 2018. *Context-encoding Variational Autoencoder for Unsupervised Anomaly Detection*. [žiūrėta 2022 gegužės 20d.] Prieiga per internetą: <https://arxiv.org/abs/1812.05941>
19. X. Chen ir E. Konukoglu. 2018. *Unsupervised Detection of Lesions in Brain MRI using constrained adversarial auto-encoders*. [žiūrėta 2022 gegužės 20d.] Prieiga per internetą: <https://arxiv.org/abs/1806.04972>
20. P. Bergmann, K. Batzner, M. Fauser ir kt. 2021. *The MVTec Anomaly Detection Dataset: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection*. [žiūrėta 2022 gegužės 20d.] Prieiga per internetą: <https://link.springer.com/content/pdf/10.1007/s11263-020-01400-4.pdf>
21. J. Lehr, A. Sargsyan, M. Pape ir kt. 2020. *Automated Optical Inspection Using Anomaly Detection and Unsupervised Defect Clustering*. [žiūrėta 2022 gegužės 20d.] Prieiga per internetą: <https://ieeexplore.ieee.org/document/9212172>
22. H. Chung, J. Park, J. Keum ir kt. 2020. *Unsupervised Anomaly Detection Using Style Distillation*. [žiūrėta 2022 gegužės 20d.] Prieiga per internetą: <https://ieeexplore.ieee.org/document/9288772>
23. M. Salehi, A. Arya, B. Pajoum ir kt. 2020. *ARAE: Adversarially Robust Training of Autoencoders Improves Novelty Detection*. [žiūrėta 2022 gegužės 20d.] Prieiga per internetą: <https://arxiv.org/abs/2003.05669v1>
24. D. Kim, D. Jeong, H. Kim ir kt. 2022. *Spatial Contrastive Learning for Anomaly Detection and Localization*. [žiūrėta 2022 gegužės 20d.] Prieiga per internetą: <https://ieeexplore.ieee.org/document/9709224>
25. C. Baur, B. Wiestler, S. Albarqouni ir kt. 2018. *Deep Autoencoding Models for Unsupervised Anomaly Segmentation in Brain MR Images*. [žiūrėta 2022 gegužės 20d.] Prieiga per internetą: <https://arxiv.org/abs/1804.04488>
26. P. Napoletano, F. Piccoli ir R. Schettini. 2018. *Anomaly Detection in Nanofibrous Materials by CNN-Based Self-Similarity*. [žiūrėta 2022 gegužės 20d.] Prieiga per internetą: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5795842/>
27. Y. Huang, C. Qiu ir K. Yuan. 2020. Surface defect saliency of magnetic tile. [žiūrėta 2022 gegužės 20d.] Prieiga per internetą: <https://link.springer.com/article/10.1007/s00371-018-1588-5>
28. D. Bank, N. Koenigstein, R. Giryes. 2020. *Autoencoders*. [žiūrėta 2022 gegužės 20d.] Prieiga per internetą: <https://arxiv.org/abs/2003.05991>
29. D. P. Kingma ir M. Welling. 2013. *Auto-Encoding Variational Bayes*. [žiūrėta 2022 gegužės 20d.] Prieiga per internetą: <https://arxiv.org/abs/1312.6114>
30. A. Redford, L. Metz ir S. Chintala. 2015. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. [žiūrėta 2022 gegužės 20d.] Prieiga per internetą: <https://arxiv.org/abs/1511.06434>

Priedai

1 priedas. Autoenkoderio su 5×5 filtru architektūra

Sluoksnis						Sluoksnio išvesties dydis
x - įvestis						(256, 256, c)
Sluoksnio Nr.	Sluoksnio pavadinimas	Filtrų sk.	Filtro dydis	Žingsnis	Aktyvacijos funkcija	
1	Konvoliucinis	32	(5x5)	2	LeakyReLU	(128, 128, 32)
2	Konvoliucinis	32	(5x5)	2	LeakyReLU	(64, 64, 32)
3	Konvoliucinis	32	(5x5)	2	LeakyReLU	(32, 32, 32)
4	Konvoliucinis	32	(5x5)	1	LeakyReLU	(32,32, 32)
5	Konvoliucinis	64	(5x5)	2	LeakyReLU	(16, 16, 64)
6	Konvoliucinis	64	(5x5)	1	LeakyReLU	(16, 16, 64)
7	Konvoliucinis	128	(5x5)	2	LeakyReLU	(8, 8, 128)
8	Konvoliucinis	64	(5x5)	1	LeakyReLU	(8, 8, 64)
9	Konvoliucinis	32	(5x5)	1	LeakyReLU	(8, 8, 32)
10	z - Konvoliucinis	100	(8x8)	1	-	(1, 1, 100)
10'	Dekonvoliucinis	32	(8x8)	1	LeakyReLU	(8, 8, 32)
9'	Konvoliucinis	64	(5x5)	1	LeakyReLU	(8, 8, 64)
8'	Konvoliucinis	128	(5x5)	1	LeakyReLU	(8, 8, 128)
7'	Dekonvoliucinis	64	(5x5)	2	LeakyReLU	(16, 16, 64)
6'	Konvoliucinis	64	(5x5)	1	LeakyReLU	(16, 16, 64)
5'	Dekonvoliucinis	32	(5x5)	2	LeakyReLU	(32, 32, 32)
4'	Konvoliucinis	32	(5x5)	1	LeakyReLU	(32, 32, 32)
3'	Dekonvoliucinis	32	(5x5)	2	LeakyReLU	(64, 64, 32)
2'	Dekonvoliucinis	32	(5x5)	2	LeakyReLU	(128, 128, 32)
1'	\hat{x} - Dekonvoliucinis	c	(1x1)	2	sigmoid	(256, 256, c)

2 priedas. Autoenkoderio su papildintu filtrų skaičiumi architektūra

Sluoksnis						Sluoksnio išvesties dydis
x - įvestis						(256, 256, c)
Sluoksnio Nr.	Sluoksnio pavadinimas	Filtrų sk.	Filtro dydis	Žingsnis	Aktyvacijos funkcija	
1	Konvoliucinis	64	(4x4)	2	LeakyReLU	(128, 128, 32)
2	Konvoliucinis	64	(4x4)	2	LeakyReLU	(64, 64, 32)
3	Konvoliucinis	64	(4x4)	2	LeakyReLU	(32, 32, 32)
4	Konvoliucinis	64	(3x3)	1	LeakyReLU	(32,32, 32)
5	Konvoliucinis	128	(4x4)	2	LeakyReLU	(16, 16, 64)
6	Konvoliucinis	128	(3x3)	1	LeakyReLU	(16, 16, 64)
7	Konvoliucinis	256	(4x4)	2	LeakyReLU	(8, 8, 128)
8	Konvoliucinis	128	(3x3)	1	LeakyReLU	(8, 8, 64)
9	Konvoliucinis	64	(3x3)	1	LeakyReLU	(8, 8, 32)
10	z - Konvoliucinis	100	(8x8)	1	-	(1, 1, 100)
10'	Dekonvoliucinis	64	(8x8)	1	LeakyReLU	(8, 8, 32)
9'	Konvoliucinis	128	(3x3)	1	LeakyReLU	(8, 8, 64)
8'	Konvoliucinis	256	(3x3)	1	LeakyReLU	(8, 8, 128)
7'	Dekonvoliucinis	128	(4x4)	2	LeakyReLU	(16, 16, 64)
6'	Konvoliucinis	128	(3x3)	1	LeakyReLU	(16, 16, 64)
5'	Dekonvoliucinis	64	(4x4)	2	LeakyReLU	(32, 32, 32)
4'	Konvoliucinis	64	(3x3)	1	LeakyReLU	(32, 32, 32)

Sluoksnio Nr.	Sluoksnis pavadinimas	Filtrų sk.	Filtro dydis	Žingsnis	Aktyvacijos funkcija	Sluoksnio išvesties dydis
3'	Dekonvoliucinis	64	(4x4)	2	LeakyReLU	(64, 64, 32)
2'	Dekonvoliucinis	64	(4x4)	2	LeakyReLU	(128, 128, 32)
1'	\hat{x} - Dekonvoliucinis	c	(1x1)	2	sigmoid	(256, 256, c)

3 priedas. AE Denner'o pasiūlyta architektūra

Sluoksnis						Sluoksnio išvesties dydis
x - įvestis						(256, 256, c)
Sluoksnio Nr.	Sluoksnio pavadinimas	Filtrų sk.	Filtro dydis	Žingsnis	Aktyvacijos funkcija	
1	Konvoliucinis + BatchNormalization	32	(5x5)	2	LeakyReLU	(128, 128, 32)
2	Konvoliucinis + BatchNormalization	64	(5x5)	2	LeakyReLU	(64, 64, 64)
3	Konvoliucinis + BatchNormalization	128	(5x5)	2	LeakyReLU	(32, 32, 128)
4	Konvoliucinis + BatchNormalization	128	(5x5)	2	LeakyReLU	(16, 16, 128)
5	Konvoliucinis + BatchNormalization	128	(5x5)	2	LeakyReLU	(8, 8, 128)
6	Konvoliucinis	16	(1x1)	1	-	(8, 8, 16)
7	Flatten					(1024)
8	z - Dense	128			-	(1, 1, 128)
8'	Dense	1024			-	(1, 1, 1024)
7'	Reshape					(8, 8, 16)
6'	Konvoliucinis + Batchnormalization	16	(1x1)	1	LeakyReLU	(8, 8, 16)
5'	Dekonvoliucinis + Batchnormalization	128	(5x5)	2	LeakyReLU	(16, 16, 128)
4'	Dekonvoliucinis + Batchnormalization	128	(5x5)	2	LeakyReLU	(32, 32, 128)
3'	Dekonvoliucinis + Batchnormalization	128	(5x5)	2	LeakyReLU	(64, 64, 128)
2'	Dekonvoliucinis + Batchnormalization	64	(5x5)	2	LeakyReLU	(128, 128, 64)
1'	Dekonvoliucinis + Batchnormalization	32	(5x5)	2	LeakyReLU	(256, 256, 32)
	\hat{x} - Konvoliucinis	c	(1x1)	1	-	(256, 256, c)

4 priedas. AE modifikacijos, su padidintu filtrų skaičiumi, rezultatai

Įvestis



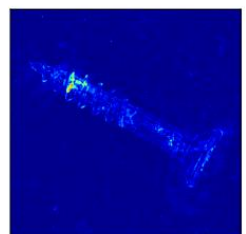
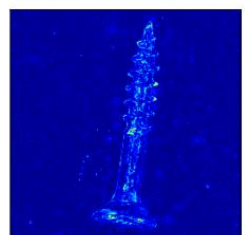
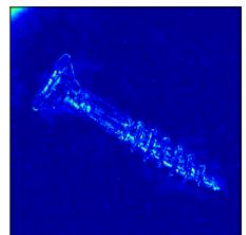
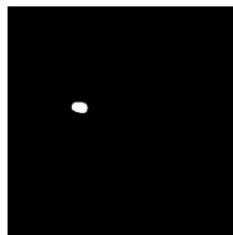
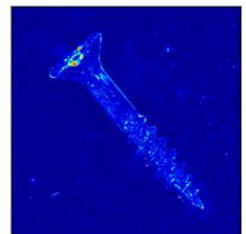
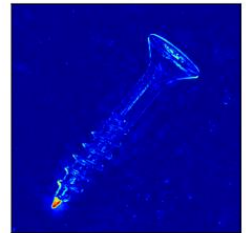
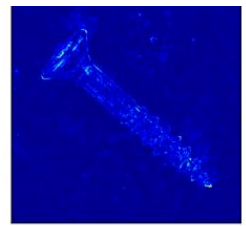
Rekonstrukcija


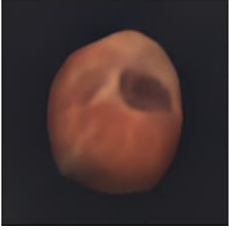

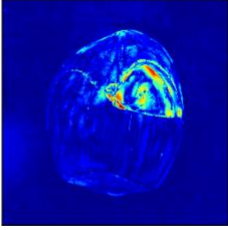

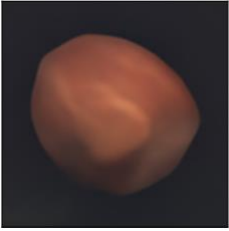
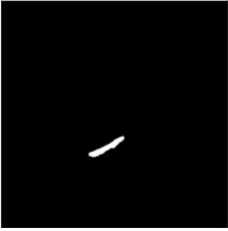
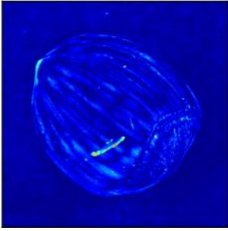

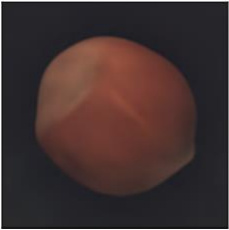
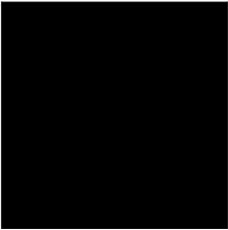
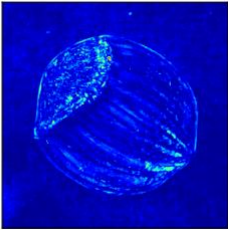

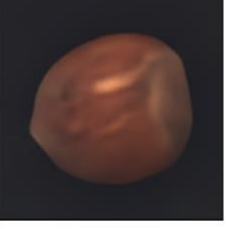
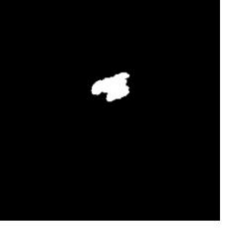
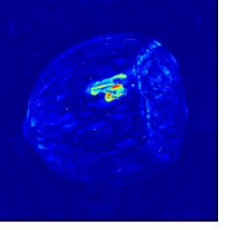

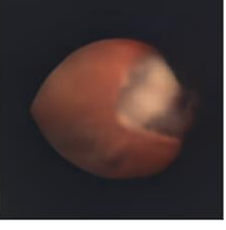

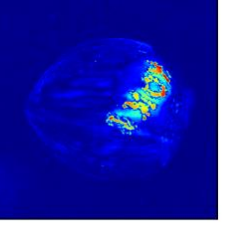


Neginčitina tiesa



Anomalių įvertinimas



Īvestis	Rekonstrukcija	Neginčītina tiesa	Anomaliju īvertinimas
			
			
			
			
			

Įvestis	Rekonstrukcija	Neginčytina tiesa	Anomalijų įvertinimas
