

Kauno technologijos universitetas

Informatikos fakultetas

GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimas

Baigiamasis magistro krypties studijų projektas

Eimantas Noreika

Projekto autorius

doc. Mantas Lukoševičius

Vadovas

Kaunas, 2022



Kauno technologijos universitetas

Informatikos fakultetas

GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimas

Baigiamasis magistro krypties studijų projektas

Informatika 6211BX007

Eimantas Noreika

Projekto autorius

doc. Mantas Lukoševičius

Vadovas

doc. Recenzentas Recenzaitis

Recenzentas

Kaunas, 2022



Kauno technologijos universitetas

Informatikos fakultetas

Eimantas Noreika

GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimas

Akademinio sąžiningumo deklaracija

Patvirtinu, kad mano, Eimanto Noreikos, baigiamasis projektas tema „GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimas“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

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

(vardą ir pavardę įrašyti ranka)

(parašas)

Autoriaus Eimantas, Noreika. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimas. Magistro krypties studijų baigiamasis projektas vadovas doc. dr. Mantas Lukoševičius; Kauno technologijos universitetas, informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Informatika, Informatikos mokslai.

Reikšminiai žodžiai: Giliojo mokymosi metodai, GAN, CycleGAN, Pix2Pix, neuroniniai tinklai, rentgenogramos, medicininių vaizdų analizė.

Kaunas, 2022. 94 p.

Santrauka

Radiologinės nuotraukos suteikia galimybę greitai ir tiksliai diagnozuoti ligas. Tačiau radiologai dažnai negali skirti daug laiko rentgeno nuotraukų analizei, dėl to ne visada atkreipia dėmesį į svarbias detales. Siekiant išspręsti šią problemą, buvo pasiūlyta daugybė kompiuterizuoto aptikimo ir diagnostikos metodų, iš kurių naujausias yra giliojo mokymosi metodas. Šiame tyrime siekiame identifikuoti naujas galimybes efektyviau apdoroti rentgeno nuotraukas, rasti efektyviausias modelių architektūras ir ištirti generatyvinių priešiško neuroninių modelių sintetinių nuotraukų generavimo galimybes siekiant pagerinti giliojo mokymosi modelių apmokymą ir atpažinimo tikslumą.

Author's Noreika, Eimantas. Applying GANs to improve arm x-ray image classification. Master's Final Degree Project abbreviation of the Assoc. Doc. Dr. Mantas Lukoševičius; Faculty of Informatics, Kaunas University of Technology.

Study field and area (study field group): Informatics, Information Sciences.

Keywords: deep learning, neural network, GAN, CycleGAN, Pix2Pix, CNN, x-ray, medical image analysis.

Kaunas, 2022. 94 p.

Summary

Radiological images allow for quick and accurate diagnosis of diseases. However, radiologists are often unable to devote much time to the analysis of X-rays, which can lead to the omission of important details. To alleviate this problem, a number of computer-aided detection and diagnostic methods have been proposed, of which the deep-learning method is the latest to be applied. The aim of this study is to identify new possibilities for more efficient X-ray image processing, to find the most efficient model architectures, and to explore the possibilities of synthetic image generation capabilities of generative adversarial network models to improve training and recognition accuracy of deep learning models.

Turinys

Lentelių sąrašas	8
Paveikslų sąrašas	9
Santrumpų ir terminų žodynas	10
Įvadas	11
1. Rentgeno nuotraukų analizės naudojant giliojo mokymosi metodus analizė	13
1.1. Klasikinė konvoliucinio neuroninio tinklo architektūra	14
1.1.1. Liekanų neuroninis tinklas (ResNet)	15
1.1.2. RoR architektūra	15
1.1.3. Platus liekanų neuroninis tinklas	16
1.1.4. Pilnai sujungtas konvoliucinis tinklas	17
1.1.5. Praretintų konvoliucinių savybių piramidinis tinklas (DCFPN)	18
1.1.6. Giliojo mokymosi modelių panaudojimo su MURA duomenų rinkiniu palyginimas	20
1.2. GAN architektūrų pritaikymo analizė	20
1.2.1. Generatyvinio priešiško neuroninio tinklo architektūra	22
1.2.2. Sąlyginio generatyvinio priešiško neuroninio tinklo architektūra	23
1.2.3. PixelCNN generatyvinio priešiško neuroninio tinklo architektūra	23
1.2.4. CycleGAN generatyvinio priešiško neuroninio tinklo architektūra	25
1.2.5. Generatyvinių priešiškų neuroninių tinklų apibendrinimas	26
1.3. Duomenų rinkiniai	26
1.3.1. MURA duomenų rinkinys	26
1.3.2. LERA duomenų rinkinys	27
1.3.3. Duomenų apdorojimo metodai	27
1.3.4. Triukšmo pašalinimas iš rentgeno nuotraukų	27
1.3.5. Rentgeno nuotraukos ryškumo didinimas	28
1.3.6. Stebimo srities išplėtimas rentgeno nuotraukose mažuose lūžiuose	28
1.3.7. Rentgeno nuotraukų apdorojimas	28
1.3.8. Duomenų apdorojimo rezultatai	28
1.4. Programinės įrangos analizė	29
1.4.1. Giliojo mokymosi karkasų analizė	30
1.4.2. Tensorflow karkaso apžvalga	30
1.4.3. Pytorch karkaso apžvalga	31
1.4.4. ML.NET karkaso apžvalga	32
1.4.5. Scikit learn karkaso apžvalga	32
1.4.6. Mašininio mokymosi karkasų apibendrinimas	33
2. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo projektas ..	34
2.1. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo projekto reikalavimų analizė	34
2.2. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo funkciniai reikalavimai	34
2.3. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo nefunkciniai reikalavimai	35

2.4. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo reikalavimai programinei įrangai	35
2.4.1. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo reikalavimai testavimui	35
2.5. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo sprendimo projektavimas	37
2.5.1. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo panaudos atvejų diagramos	37
2.5.2. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrėjo panaudos atvejai	37
2.5.3. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti sistemos panaudos atvejai	41
2.6. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo sprendimo realizacija	47
2.6.1. Giliojo mokymosi metodų realizacija	47
2.6.2. GAN modelių realizacija	48
2.6.3. Duomenų rinkinio apdorojimo realizacija	49
2.7. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo sprendimo testavimas	49
2.8. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo projektavimo išvados	51
3. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo eksperimentai ir rezultatai	52
3.1. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo eksperimentų planavimas	52
3.2. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo eksperimentų rezultatai	52
3.3. Giliojo mokymosi modelių tikslumo priklausomumas nuo išankstinio duomenų rinkinio apdorojimo	53
3.3.1. Giliojo mokymosi modelių tikslumo priklausomumas nuo išankstinio duomenų rinkinio apdorojimo	53
3.3.2. Giliojo mokymosi modelių, apmokytų su ImageNet duomenų rinkiniu, tikslumo priklausomumas nuo išankstinio duomenų rinkinio apdorojimo	55
3.3.3. Giliojo mokymosi modelių treniravimo laikas	57
3.4. Rankos rentgeno nuotraukų eskizų generavimo eksperimentai	57
3.4.1. Duomenų rinkinio išplėtimas naudojant GAN modelius	61
3.4.2. Duomenų rinkinio išplėtimas naudojant Pix2Pix modelį	62
3.4.3. Duomenų rinkinio išplėtimas naudojant CycleGAN modelį	65
3.4.4. Duomenų rinkinio išplėtimo naudojant GAN modelius eksperimentų rezultatai	65
Išvados	71
Literatūros sąrašas	72
4. Priedai	77

Lentelių sąrašas

1.1 lentelė.	Plataus residual tinklo struktūra.	17
1.2 lentelė.	DCFPN architektūros rezultatų palyginimas su kitais giliojo mokymosi metodais	19
1.3 lentelė.	Giliojo mokymosi modelių palyginimas naudojant MURA duomenų rinkinį	21
1.4 lentelė.	MURA duomenų rinkinio duomenų pasiskirstymas srityse	27
1.5 lentelė.	MURA duomenų rinkinio apdorojimo naudojimo literatūros apžvalga	30
2.1 lentelė.	Rentgeno nuotraukų filtravimo panaudos atvejis	37
2.2 lentelė.	Rentgeno nuotraukų duomenų rinkinių kūrimo panaudos atvejis	38
2.3 lentelė.	Giliojo mokymosi modelių tikslumo kitimo statistikos peržiūros panaudos atvejis	39
2.4 lentelė.	Išankstinio apdorojimo veiksnių pritaikymas duomenų rinkiniui panaudos atvejis	40
2.5 lentelė.	Giliojo mokymosi modelių treniravimo su pasirinktu duomenų rinkiniu panaudos atvejis	41
2.6 lentelė.	Giliojo mokymosi modelių eksperimentų su skirtingais duomenų rinkiniais testavimo panaudos atvejis	42
2.7 lentelė.	Giliojo mokymosi modelių veiksmingumo statistikos rinkimo panaudos atvejis	43
2.8 lentelė.	Išankstinių nuotraukų apdorojimo veiksnių įvykdymas duomenų rinkinio aibėje panaudos atvejis	44
2.9 lentelė.	Rentgeno nuotraukų generavimas naudojant GAN modelius panaudos atvejis	45
2.10 lentelė.	Tyrėjo panaudos atvejų testai	50
2.11 lentelė.	Sistemos panaudos atvejų testai	51
3.1 lentelė.	Giliojo mokymosi modelių klasifikacijos tikslumo rezultatai su alkūnės rentgeno nuotraukų duomenų rinkiniu be išankstinio apdorojimo	54
3.2 lentelė.	Giliojo mokymosi modelių klasifikavimo tikslumo rezultatai su alkūnės rentgeno nuotraukų duomenų rinkiniu su išankstiniu apdorojimu	55
3.3 lentelė.	Giliojo mokymosi modelių, apmokytų su ImageNet duomenų rinkiniu, tikslumo rezultatai su alkūnės rentgeno nuotraukų duomenų rinkiniu be išankstinio apdorojimo	56
3.4 lentelė.	Giliojo mokymosi modelių, apmokytų su ImageNet duomenų rinkiniu, tikslumo rezultatai su alkūnės rentgeno nuotraukų duomenų rinkiniu su išankstiniu apdorojimu	57
3.5 lentelė.	Giliojo mokymosi modelių tikslumo rezultatai su alkūnės rentgeno nuotraukų MURA duomenų rinkinyje su Pix2Pix sugeneruotomis nuotraukomis	67
3.6 lentelė.	Giliojo mokymosi modelių tikslumo rezultatai su alkūnės rentgeno nuotraukų MURA duomenų rinkinyje su CycleGAN sugeneruotomis nuotraukomis	69

Paveikslų sąrašas

1.1 pav.	LeNet-5 CNN architektūros pavyzdys [15]	14
1.2 pav.	DenseNet architektūra [29]	17
1.3 pav.	Magistralinis blokas su įprastomis konvoliucijomis pažymėtas kaip A ir magistralinis blokas su išsiplėtusiomis konvoliucijomis pažymėtas kaip B [30]. . .	20
1.4 pav.	Sąlyginio generatyvinio priešiško neuroninio tinklo struktūra	24
1.5 pav.	MURA duomenų rinkinio alkūnės rentgeno nuotraukų apdorojimo rezultatai [2] .	29
2.1 pav.	GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo panaudos atvejų diagrama	37
2.2 pav.	Duomenų rinkinio apdorojimo realizacijos operacijų rezultatai	50
3.1 pav.	Giliojo mokymosi modelių klasifikacijos tikslumas alkūnės rentgeno nuotraukų duomenų rinkinyje su išankstiniu apdorojimu ir be išankstinio apdorojimo	54
3.2 pav.	Giliojo mokymosi modelių, apmokytų su ImageNet duomenų rinkiniu, klasifikacijos tikslumas alkūnės rentgeno nuotraukų duomenų rinkinyje su išankstiniu apdorojimu ir be išankstinio apdorojimo	56
3.3 pav.	Giliojo mokymosi modelių mokymosi laikas per vieną epochą	58
3.4 pav.	Giliojo mokymosi modelių, apmokytų su ImageNet duomenų rinkiniu, klasifikacijos tikslumas alkūnės rentgeno nuotraukų duomenų rinkinyje su išankstiniu apdorojimu ir mokymosi laiko per vieną epochą koreliacija	58
3.5 pav.	Rankos rentgeno nuotraukų eskizų generavimo rezultatai	60
3.6 pav.	Rankos rentgeno nuotraukų eskizų generavimo rezultatai duomenų rinkinyje su išankstiniu apdorojimu	60
3.7 pav.	A originali rentgenograma, B rentgenograma po išankstinio apdorojimo ir C rentgenograma sukurta naudojant HED metodą, D rentgenograma sukurta naudojant morfologines operacijas	61
3.8 pav.	GAN modelio rezultatai generuojant 224×224 raiškos alkūnės rentgeno nuotraukas	62
3.9 pav.	Pix2Pix modelio treniravimo schema naudojant MURA duomenų rinkinio alkūnės rentgenogramas ir sukurtus eskizus	63
3.10 pav.	Pix2Pix modelio sugeneruotų alkūnės rentgeno nuotraukų pavyzdžiai su sukurtais eskizais	64
3.11 pav.	Pix2Pix modelio treniravimo efektyvumo <i>BCEWithLogitsLoss</i> ir <i>L1Loss</i> nuostolio funkcijų reikšmių pokytis epochose	64
3.12 pav.	CycleGAN modelio treniravimo schema naudojant MURA duomenų rinkinio alkūnės rentgenogramas ir sukurtus eskizus	66
3.13 pav.	CycleGAN modelio sugeneruotų alkūnės rentgeno nuotraukų pavyzdžiai su sukurtais eskizais	67
3.14 pav.	Giliojo mokymosi modelių tikslumo pokytis su alkūnės rentgeno nuotraukomis MURA duomenų rinkinyje ir MURA duomenų rinkinyje su Pix2Pix ir CycleGAN sugeneruotomis nuotraukomis	68
4.1 pav.	DCFPN giliojo mokymosi modelio architektūra [30].	77

Santrumpų ir terminų žodynas

GAN – Generatyvinis priešiškas neuroninis tinklas.

CNN – Konvoliucinis neuroninis tinklas.

KT – Kompiuterinė tomografija.

AP – Vidutinis tikslumas.

ResNet – Liekanų neuroninis tinklas.

FPN – Piramidinis požymių tinklas.

RPN – Regionų pasiūlymų tinklas.

Rentgenograma – rentgeno spinduliais padaryta nuotrauka.

Karkasas (angl. *framework*) – struktūra, kuri praplečia esamą struktūrą į naudingesnę, taip paspartinant programos kūrimą.

Ivadas

Radiologinės nuotraukos suteikia galimybę greitai ir tiksliai diagnozuoti ligas. Dėl didėjančio rentgeno nuotraukų naudojimo atliekant diagnostiką kyla problema, kad radiologai negali skirti pakankamai laiko kiekvienos nuotraukos analizei dėl didelio darbo krūvio [1, 2, 3, 4, 5, 6, 7, 8, 9]. Siekiant išspręsti šią problemą, buvo pasiūlyta daugybė kompiuterizuoto aptikimo ir diagnostikos metodų iš kurių naujausias yra giliojo mokymosi metodas. Ši technologija pritaikyta skirtingose srityse atliekant objektų aptikimą ir klasifikaciją suteikia galimybę perkelti sukauptas šios srities technologijas į medicinos sritį ir padėti radiologams lengviau aptikti ligas.

Mokslininkai ir gydytojai gerai supranta, kad giliojo mokymosi metodai įgauna svarbų vaidmenį radiologijoje. Vis daugiau mokslinių tyrimų orientuojasi į giliojo mokymosi metodų taikymą medicinoje, siekiant ištirti giliojo mokymosi metodų pritaikymą sprendžiant kaulų lūžių diagnozavimo uždavinį. Tačiau tyrimai daugiausiai sprendžia klasifikavimo uždavinį, o kaulų lūžio vietos nustatymui nėra skiriama dėmesio [10]. Radiologai darbo metu negali skirti daug laiko rentgeno nuotraukų analizei, todėl ne visada atkreipia dėmesį į mažai pastebimas detales. Tyrimuose teigiama, kad neteisingas ligos pažymėjimas atliekant rentgeno nuotraukos analizę nėra laikoma kaip klaida. Radiologas gali greitai nuspręsti ar tai yra sutrikimas jeigu yra nurodoma tiksli dėmesio sritis nuotraukoje, todėl analizės metu svarbiausia nukreipti radiologo dėmesį, o sprendimo priėmimą apie ligos egzistavimą palikti radiologui.

Kaulų lūžio vietos nustatymas yra labai svarbus norint palengvinti radiologų darbo procesą. Kadangi nustatyti lūžius rentgenogramose sunku, nes lūžiai yra išskirtiniai ir nevienalyčiai, jie gali atsirasti bet kuriame kaule, o jų išvaizda priklauso nuo kūno dalies anatomijos ir radiografinės projekcijos. Šiuo darbu bus siekiama ištirti giliojo mokymosi metodų pritaikymą, kurie gebėtų atpažinti anomalijas rentgenogramose ir atkreiptų į jas radiologų dėmesį. Siekiant pritaikyti giliojo mokymosi modelius, reikia išplėsti duomenų rinkinio įvairiapusiškumą ir ne visada yra galimybės naudoti daugiau duomenų, todėl šiame darbe atliekamas tyrimas ištirti generatyvinių priešiško tinklų (angl. *generative adversarial network*) pritaikymo galimybes išplėsti duomenų rinkinį.

Darbo problematika ir aktualumas

Medicininis rentgeno nuotraukų surinkimas yra labai brangus procesas, todėl kad išplėsti surinktų duomenų aibes taikomi įprastiniai duomenų apdorojimo metodai kaip apkarpymas, horizontalus arba vertikalus apvertimas, pasukimas ir kt. GAN pagrįstas duomenų apdorojimas suteikia daug didesnes galimybes išplečiant duomenų rinkinį, nes yra sukuriami nauji vaizdai, kurie yra panašūs į pradinius duomenis. GAN pagrįstų duomenų apdorojimo procedūrų yra atlikta itin mažai raumenų ir kaulų sistemos rentgeno nuotraukų srityje [11]. Tyrimai atlikti kitose srityse (pvz., kepenų pažeidimo sintezė iš ribotos kompiuterinės tomografijos (KT) duomenų rinkinio, kūno KT sintezė iš skaitmeninių kūno fantomų), parodė, kad yra įmanoma gauti didelį kiekį vaizdinių variantų, kurie gali būti taikomi raumenų ir kaulų sistemų srityje [11].

Darbo tikslas ir uždaviniai

Ištirti galimybes pagerinti skirtingų giliojo mokymosi architektūrų efektyvumą aptikti anomalijas rentgeno nuotraukose pritaikant duomenų rinkinio išankstinio apdorojimo metodus ir duomenų aibės išplėtimą pasitelkiant generatyvinius priešiškus tinklus.

1. Išnagrinėti mokslinę literatūrą apie giliojo mokymosi metodų pritaikymą rentgeno nuotraukų analizei;
2. Identifikuoti anomalijas rentgeno nuotraukose pritaikant skirtingas giliojo mokymosi architektūras;
3. Atlikti eksperimentus su giliojo mokymosi metodais ir nustatyti, kuris metodas tiksliausiai nustato anomalijas rentgeno nuotraukose;
4. Eksperimentiškai ištirti generatyvinių priešinių tinklų pritaikymą išplėsti duomenų rinkinį sintetinėmis nuotraukomis.

1. Rentgeno nuotraukų analizės naudojant giliojo mokymosi metodus analizė

Šiame skyriuje pirmiausia yra apžvelgiamos skirtingos konvoliucinių neuroninių tinklų (angl. *convolutional neural networks*) (CNN) architektūros. Apžvalga pradama nuo klasikinių CNN architektūrų ir tęsiama su pažengusiomis, kurios gali būti pritaikytos atlikti anomalijų aptikimą rentgeno nuotraukose. Analizuojant yra apžvelgiama privalumai ir trūkumai tarp architektūrų ir atliekami palyginimai. Analizuojant skirtingas architektūras stengiamasi atkreipti dėmesį į pagrindinius architektūros bruožus ir išskirtinumus.

Didžiausias tyrimas surinkęs daugiausiai raumenų ir kaulų sistemos rentgenogramų buvo atliktas Stanfordo universiteto mokslininkų, kurio metu buvo surinkta 40 561 nuotrauka [12]. Šis rinkinys buvo pateiktas kaip iššūkis pritaikyti giliojo mokymosi modelius išspręsti anomalijų aptikimą rentgeno nuotraukose uždavinį. Tyrimo metu jie parengė 169 sluoksnių DenseNet modelį, kad nustatytų ir lokalizuotų anomalijas. Modelis pasiekia 0,929 AUROC, 0,815 jautrumą ir 0,887 specifiškumą. Modelio rezultatai yra prastesni nei geriausi radiologo rezultatai nustatant anomalijas atliekant alkūnės, dilbio, plaštakos, žastikaulio ir pečių tyrimus [12]. Šis iššūkis iššaukė sekančius tyrimus, kurie siekė pritaikyti skirtingas architektūras pasiekti geresnius rezultatus. Tyrime buvo išplėstas MURA duomenų rinkinys stengiantis išspręsti ir lokalizavimo užduotį panaudojant tik 4 000 rentgeno nuotraukų 2019 metais buvo pasiūlytas pagerintas dviejų stadijų R-CNN modelis, kuris pasiekė 62.04% bendrą tikslumą [2]. Šio eksperimento rezultatai rodo, kad dviejų stadijų R-CNN metodas pasiekia aukštus rankų lūžių aptikimo vidutinio tikslumo (angl. *average precision*) (AP) rezultatus ir jis turi didelį potencialą būti pritaikytas realioje klinikinėje aplinkoje. Tyrime atliekančiame žastikaulio lūžių klasifikaciją turint tik 1891 rentgeno nuotraukas, CNN parodė aukštą 96% tikslumą, 1,00 AUC, 0,99 jautrumą ir 0,97 specifiškumą pralenkdamis specializuotus rentgenologus [13]. Kiti autoriai 2020 pristatė kelių tinklų sistemą (MSCNN-GCN), kurią sudaro kelių atšakų 2D CNN ir pilnai sujungtas GCN, kuris naudodamas MURA duomenų rinkinį įrodė, kad siūloma sistema ne tik pasiekė didžiausią tikslumą, bet ir parodė aukščiausius balus tiek pagal F1, tiek pagal kappa metriką [14]. Tokie rezultatai pasiekti su MURA duomenų rinkiniu yra išskirtiniai, nes yra susiduriama su duomenų rinkinio apdorojimo problemomis. Kiti autoriai panaudoję privatų duomenų rinkinį su 715 tūkstančių rentgeno nuotraukų ir sukūrę dešimtimi DenseNet modelių paremtą architektūrą sugebėjo pasiekti aukštą klasifikacijos efektyvumą ir lokalizuoti lūžius ir visiškai pralenkti rentgenologus tikslume pasiekiant bendrą 0.974 AUC, 0.952 jautrumą ir 0.813 specifiškumą [1]. Tyrėjai ne tik apjungė skirtingas giliojo mokymosi architektūras, bet ir pasiūlė pritaikyti GAN tinklus kaip naują metodą duomenų rinkinio papildymui. Atliktas tyrimas su CycleGAN perkeliant be kontrasto CT nuotraukas į nuotraukas su kontrastu, taip padidinat duomenų rinkinį [11]. Šio tyrimo metu jiems pavyko pasiekti didelį Dice įverčio pagerėjimą inkstų nuotraukose (nuo 0,09 iki 0,66, $p < 0,001$), bet kepenų (nuo 0.86 iki 0.89, $p < 0,001$) ir blužnies (nuo 0.65 iki 0.69, $p < 0,001$) Dice įverčio pagerėjimas buvo mažesnis. GAN modelių pritaikymas yra naudingas medicinos srities tyrėjams, siekiant sumažinti rankinio segmentavimo pastangas ir kompiuterinės tomografijos išlaidas.

Tyrimo metu didelis dėmesys yra skiriamas duomenų rinkinių analizei siekiant užtikrinti naujumą ir naujausių metodų pritaikymą. Duomenų rinkinių analizėje yra apžvelgiami prieinami

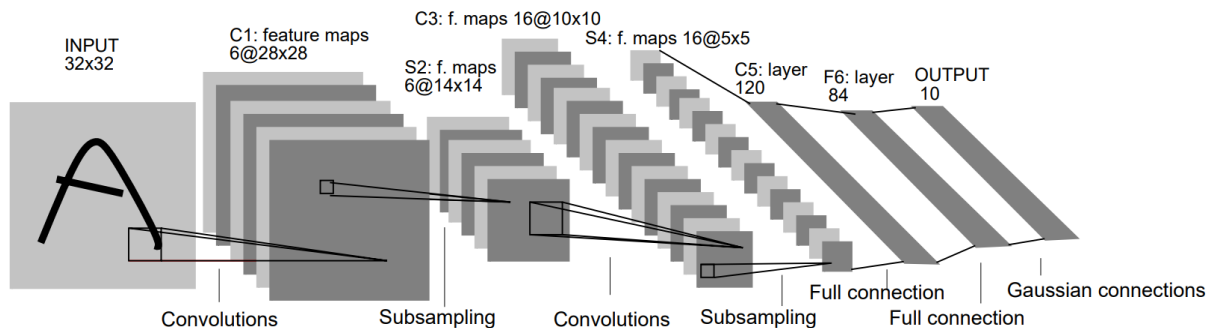
rentgeno nuotraukų duomenų rinkiniai ir jų ypatumai. Tai pat analizuojami dažniausiai naudojami duomenų rinkinio išankstinio apdorojimo ir praplėtimo metodai, kurie gali būti įtrauki eksperimentų atlikimo metu.

Analizė baigiama atliekant paprasto generatyvinio priešiško tinklo, sąlyginio GAN, Pix2Pix ir CycleGAN architektūrų analizę. Šių architektūrų ypatumai yra išanalizuojami stengiantis rasti jų pritaikymo galimybes generuojant papildomas rentgeno nuotraukas duomenų rinkiniui. Kiekviena iš šių architektūrų turi skirtingus ypatumus, kurie gali būti pritaikyti išplėsti duomenų aibę. Naudojant šiuos modelius galima papildyti esamą duomenų rinkinį sintetinėmis rentgenogramomis ir taip padidinti giliojo mokymosi modelių tikslumą nekeičiant pačių modelių.

1.1. Klasikinė konvoliucinio neuroninio tinklo architektūra

Klasikinė LeNet CNN architektūra pristatyta [15] autorių yra sudaryta iš 8 apmokomu sluoksnių. Iš kurių 5 yra konvoliuciniai ir trys pilnai sujungti sluoksniai. Šios architektūros pavyzdys pateiktas 1.1 paveiksle.

Šioje architektūroje buvo panaudotas *Rectified Linear Unit* (ReLU) netiesiškumas [16] – standartinis būdas modelio neuronui išreikšti f kaip funkciją su parametru x yra su $f(x) = \tanh(x)$ arba $f(x) = (1 + e^{-x})^{-1}$. Vertinant treniravimo greitį atliekant gradientinio nusileidimą paieška neužsiriboja (angl. *non-saturating*), todėl netiesinės funkcijos yra daug greitesnės $f(x) = \max(0, x)$. Treniravimo laikas sumažėjo keletą kartų CNN architektūroje panaudojus šią funkciją.



1.1 pav. LeNet-5 CNN architektūros pavyzdys [15]

ReLU pasižymi savybe, kad nereikia normalizuoti įvesties, kad išvengti saturacijos. Bet gauti rezultatai [17] straipsnyje rodo, kad lokalus atsako normalizavimas (angl. *local response normalization*) padeda apibendrinti duomenų įvestis. Neurono aktyvumo apskaičiavimas išreiškiamas žymint $\alpha_{x,y}^i$, kur filtro branduolys (angl. *kernel*) i yra pozicijoje (x, y) . Pritaikant ReLU nelinejinį aktyvumą gaunamas atsako normalizavimo aktyvumas $b_{x,y}^i$, kurio apibrėžimas pateiktas (1).

$$b_{x,y}^i = \alpha_{x,y}^i \left(k + \alpha \sum_{j=\max(0, i-\frac{n}{2})}^{\min(N-1, i+\frac{n}{2})} (\alpha_{x,y}^j)^2 \right)^\beta \quad (1)$$

Lygtyje suma yra skaičiuojama visuose gretimuose branduolio žemėlapiuose vienuose erdvinėse pozicijose. N žymimas sluoksnyje esančių branduolių kiekis. Kintamieji k , n , α ir β yra hiperparametrai (angl. *hyperparameter*), kurių reikšmės parenkamos atliekant validaciją.

Persidengiantis sutelkimas (angl. *overlapping pooling*) – tai būdas subendrinti šalia esančių sričių išvestis [18]. Autoriai analizės metu pastebėjo, kad naudojant persidengiančias apibendrinimo sritis galima sumažinti persimokymą 0.3% – 0.4% procentais, lyginant su nepersidengiančiu apibendrinimu. Darbe teigiama, kad naudojant persidengiantį išvesčių apibendrinimą galima gauti didesnę tikslumą su testavimo duomenų rinkiniu.

1.1.1. Liekanų neuroninis tinklas (ResNet)

Liekanų (angl. *residual*) neuroninis tinklas sukurtas 2015 metais, kurio pagrindinis bruožas – sutrumpinti keliai tarp sluoksnių, kurie išsprendė tinklo optimizavimo, treniravimo [19, 20], sumažėjusio srauto į priekį, nykstančių gradientų [21] ir persimokymo (angl. *overfitting*) problemas [22]. Tai leido sukurti didesnius ir gilesnius modelius, tačiau tikslumas tūkstančių sluoksnių gylio modeliuose vis tiek išliko prastesnis [23].

Vienas iš pagrindinių išskirtinumų ResNet šeimos tinkluose, kad norint išspręsti gilių tinklų lėto treniravimo problemą, jo metu naudojamas stochastinis gylis, o testavimo metu naudojamas pilnas tinklas. Tokiu būdu treniravimo metu yra išmetamos tinklo dalys naudojant tapatybės funkciją ir siunčiant mažas dalis duomenų tik į dalį tinklo. Stochastinis gylis tinkle veikia kaip regularizatorius kartu su duomenų dalies normalizacija [24]. Tai yra vienas iš svarbiausių veiksmų, leidžiančių liekanų tinklams pasiekti tokius gerus rezultatus treniruojant tūkstančių sluoksnių gylio tinklus [21].

1.1.2. RoR architektūra

Visi liekanų tinklai kuriami remiantis viena hipoteze: naudojant sutrumpintus kelius tarp jungčių, liekanų tinklai atlieka sujungimus sugrupuotuose netiesiniuose sluoksniuose, kurie leidžia lengviau optimizuoti negu originalūs sujungimai [25]. Autoriai iškėlė hipotezę, kad atliekant liekanų sujungimus liekanų sujungimuose gali būti dar lengviau optimizuoti tinklus ir gali pasiekti geresnį tikslumą [19].

RoR architektūros privalumai [19]:

1. Pagerino optimizavimo ResNet galimybę pridėdamas keletą tapatybės nuorodų. Ir RoR pasiekė geresnių rezultatų nei ResNet naudodamas tiek pat sluoksnių skirtinguose duomenų rinkiniuose;
2. Tinka ir kitiems liekanų tinklams, nes padidina jų našumą, todėl yra veiksmingas liekanų tinklų šeimos papildymas;
3. Sukurtos strategijos kaip naudoti RoR skirtinguose duomenų rinkiniuose nustatant skirtingą gylį, plotį, tinklo praleidimo tipą ir maksimalų epochų skaičių.

Autorių teigimu, RoR ne tik reikšmingai pagerino vaizdo klasifikavimo našumą, bet taip pat gali veiksmingai papildyti liekanų tinklų šeimą.

1.1.3. Platus liekanų neuroninis tinklas

Liekanų neuroninio tinklo autoriai kuriant architektūrą stengėsi pasiekti kuo didesnę gylį, todėl buvo aukojamas tinklo platumas ir parametrų kiekis. Atliekant tai, buvo galima ištrenuoti tūkstančių sluoksnių tinklą, panaudojant ir 1×1 konvoliucijas kurios sumažindavo tinklo apmokomų parametrų kiekį. Toks architektūrinis sprendimas leido pasiekti trumpą tinklo treniravimo laiką ir sklandų gradiento perdavimą, tačiau sukelia kitas problemas. Gradientui slenkant per tinklą nėra nieko, kas priverstų gradientą eiti per liekanų bloką svorius, todėl gali praeiti treniravimo sesijos ir tinklas nieko neišmoksta. Šios teorijos tikslumą patvirtina atsitiktinis liekanų bloką išjungimas treniravimo metu ir atsitiktinio praretinimo transformacijos (angl. *dropout*) panaudojimas.

ResNet bloką išplėtimas autorių teigimu [26] yra daug efektyvesnis būdas gerinti tinklo tikslumą už tinklo gylio didinimą. Platūs liekanų tinklai pasižymi geresniu tikslumu už liekanų giliuosius tinklus su tapatybės susiejimu (angl. *identity mapping*) ir turi 50 kartų mažiau sluoksnių ir yra daugiau negu 2 kartus greitesni [24]. Architektūrų palyginimas rodo kad pagrindinė giliųjų liekanų tinklų galia yra liekanų blokai ir kad gylio poveikis yra pridėtinis.

Plačių liekanų tinklų veikimą smarkiai padidino plačiai priimto ir naudojamo atsitiktinio išmetimo pritaikymas. Atliktame tyrime [23] pridėjus tapatybės dalyje atsitiktinio išmetimo bloką autoriai gavo neigiamus rezultatus, bet perkėlus atsitiktinio išmetimo bloką tarp konvoliucinių sluoksnių galima pasiekti nuoseklų tikslumo pagerėjimą [26].

Liekanų bloką su tapatybės jungtimis išreiškimas pateiktas (2).

$$x_{l+1} = x_l + F(x_l, W_l) \quad (2)$$

kur x_{l+1} ir x_l yra 1-ojo tinklo vieneto įvestis ir išvestis, F yra liekamoji funkcija ir W_l yra bloko parametrai. Likęs tinklas susideda iš eilės liekanų bloką. Blokai gali būti išskirti:

1. Pagrindinis – su dvejomis iš eilės 3×3 konvoliucijomis su duomenų dalies normalizavimu ir ReLU aktyvacija prieš konvoliucijas (*conv3 × 3 – conv3 × 3*);
2. Siaurėjanti magistralė – su viena 3×3 konvoliucija, kurią supa matmenis mažinantis ir plečiant 1×1 konvoliucijos (*conv1 × 1 – conv3 × 3 – conv1 × 1*).

Plataus liekanų tinklo architektūroje yra naudojama duomenų dalies normalizavimas iš *conv – BN – ReLU* iki *BN – ReLUconv*, kuris sumažino treniravimosi laiką ir pagerino modelio tikslumą. Siaurinančios 1×1 konvoliucijos pirmą kartą pasiūlytos Inception [27] architektūroje ir naudojamos liekanų siaurinančiuose blokuose, kad sumažintų parametrų kiekį perduodama į sekantį bloką nebuvo analizuojamos autorių [24], nes visas dėmesys buvo skirtas gerinti pagrindinio bloko veikimą. Autorių pasiūlytos liekanų bloką tobulinimo gairės:

1. Padidinti konvoliucinių sluoksnių kiekį bloke;
2. Išplėsti tinklą pridėdant daugiau funkcinių plokštumų;
3. Padidinti filtrų dydžius konvoliuciniuose sluoksniuose.

Bendrinė plataus liekanų tinklo struktūra pavaizduota 1.1 lentelėje. Struktūra sudaryta iš pradinio konvoliucinio sluoksnio *conv1*. Sekantys trys liekanų blokai, kurių dydis N , pasibaigė su vidutinių reikšmių sutraukimu ir galutinio klasifikavimo sluoksniu.

1.1 lentelė. Plataus residual tinklo struktūra.

Grupės pavadinimas	Išvesties dydis	Bloko tipas = B (3,3)
Conv1	32×32	$[3 \times 3, 16]$
Conv2	32×32	$\begin{matrix} 3 \times 3, 16 * k \\ 3 \times 3, 16 * k \end{matrix} * N$
Conv3	16×16	$\begin{matrix} 3 \times 3, 32 * k \\ 3 \times 3, 32 * k \end{matrix} * N$
Conv4	8×8	$\begin{matrix} 3 \times 3, 64 * k \\ 3 \times 3, 64 * k \end{matrix} * N$
Avg-pool	1×1	$[8 \times 8]$

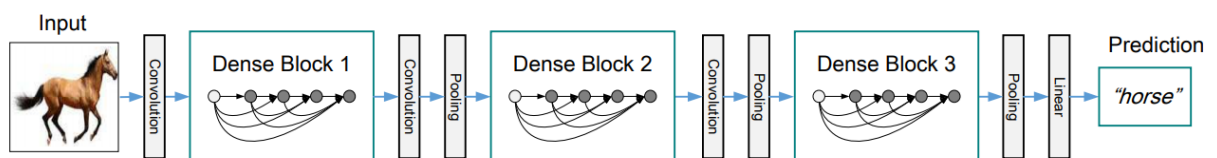
Keičiant skirtingus liekanų blokų kiekio, skaičiaus, gilumo ir dydžio parametrus smarkiai išskyla parametų kiekis. Autorių pasiūlytos gairės naudojant šią architektūrą:

1. Tinklo išplėtimas visada padidina skirtingo gylio liekanų tinklų veikimą;
2. Didinant gyli ir plotį didėja parametų skaičius, todėl reikia stipresnio duomenų reguliavimo;
3. Platūs liekanų tinklai gali treniruotis su daugiau parametų, o duomenų reguliavimas turi didesnę poveikį tinklo veikimui negu gilumu pasižyminčiuose liekanų tinkluose.

Platus liekanų tinklas su 16 sluoksnių gerokai pralenkė 1000 sluoksnių gilumo tinklą su CIFAR duomenų rinkiniu, taip pat 50 sluoksnių tinklas lenkia 152 sluoksnių gylio tinklą ImageNet duomenų rinkinyje [26]. Tai parodo, kad pagrindinė liekanų tinklų galia yra liekanų blokuose. Taip pat autoriai įrodė, kad platų liekanų tinklą galima kelis kartus greičiau apmokyti ir pasiekti geresnį tikslumą skirtinguose duomenų rinkiniuose.

1.1.4. Pilnai sujungtas konvoliucinis tinklas

Kuriant gilesnius CNN išskyla pranykstančios įvesties arba gradiento problema. Su šia problema susiduria ResNets, Highway Networks, Stochastic depth ir FractalNets [28]. Visos šios architektūros mėgina išspręsti šią problemą trumpinant įvesties ar gradiento kelią. DenseNet yra architektūra, kuri pasižymi ankstesnių sluoksnių sujungimu su vėliau esančiais, kurių požymių žemėlapių dydžiai sutampa. Architektūros scheminis atvaizdavimas pateiktas 1.2 paveiksle.



1.2 pav. DenseNet architektūra [29]

Skirtingai negu ResNet architektūroje DenseNet savybės iš kiekvieno sluoksnio nėra sumuojamos, o siunčiamos visiems tolimesniems sluoksniams juos sujungiant. Todėl gaunasi kad l^{th} sluoksnis, turi l įvesčių iš visų prieš jį buvusių sluoksnių požymių žemėlapių. Pačio sluoksnio požymių žemėlapių yra toliau siunčiama kaip įvestis sekantiems sluoksniams. Toks įvesčių perdavimas sukuria $\frac{L(L+1)}{2}$ jungčių L gylio tinkle, kai tradiciniuose architektūrose yra tik L jungčių tarp sluoksnių.

Pilnas sujungimas (angl. *dense connectivity*) – kai kiekvienas sluoksnis turi tiesioginį ryšį su visais sekančiais sluoksniais. Kiekvienas sluoksnis l^{th} gauna požymių žemėlapius įvesties x_0, \dots, x_{l-1} pavidalu iš prieš jį buvusių sluoksnių. Ryšys tarp sluoksnių pateiktas (3).

$$X_l = H_l([x_0, x_1, \dots, X_{l-1}]) \quad (3)$$

Architektūroje yra naudojami reikšmių sutelkimo sluoksniais, kurie yra vadinami pilnai sujungtais blokais. Šie blokai yra sujungiami perėjimo (angl. *transition*) sluoksniais, kurie atlieka konvoliuciją ir reikšmių sukaupimą. Perėjimo sluoksniai yra sujungiami su 1×1 konvoliucija, kuri sumažina požymių žemėlapius, remiantis inception architektūra ir panaudoja maksimalių reikšmių 2×2 sutelkimo sluoksnį išgauti didžiausią sluoksnių aktyvaciją.

Augimo tempas (angl. *growth rate*) - nusako kiek kiekvienas sluoksnis pilnai sujungtame bloke prideda papildomų savybių žemėlapių. Kadangi kiekvienas sluoksnis gali pasiekti visus anksčiau buvusius sluoksnius, todėl autorių teigimu nėra reikalo replikuoti informacijos. Dėl šios priežasties sekantys sluoksniai prideda tik nustatytą kiekį savybių žemėlapių į globalų pilnai sujungtą bloką.

Magistraliniai (angl. *bottleneck*) sluoksniai – Kadangi kiekvienas sluoksnis gauna informaciją iš prieš tai buvusių sluoksnių, siekiant sumažinti parametru kiekį, kuris pereina iš vieno pilnai sujungto bloko į kitą yra naudojamos 1×1 konvoliucijos, kurios drastiškai sumažina parametru kiekį, kuris kaip įvestis bus siunčiama sekančiam dense blokui.

Pilnai sujungtas konvoliucinis tinklas yra pirmasis, kuris pritaikė tiesioginį ryšį tarp dviejų sluoksnių su vienodu savybių žemėlapių dydžiu. Analizės metu buvo pateikta, kad taip sujungtas tinklas gali išsiplėsti į gylį šimtus sluoksnių nesukeliant optimizacijos apsunkinimų. Laikantis įprastų ryšio taisyklių tarp sluoksnių autoriams pavyko integruoti tapatybės, *deep supervision* ir *diversified depth*. Tai leidžia panaudoti išmoktas savybes ir sukurti mažesnius ir tikslesnius modelius. Dėl suspaustos vidinės duomenų reprezentacijos autorių teigimu DenseNet yra tinkami naudoti savybių ištraukimui [29].

1.1.5. Praretintų konvoliucinių savybių piramidinis tinklas (DCFPN)

DCFPN yra naujas giliojo mokymosi metodas pritaikytas šlaunies lūžių aptikimui, kuris pagrindinį piramidinį požymių (FPN) tinklo rėmą pakeičia su praretintomis konvoliucijomis. Praretintos konvoliucijos yra konvoliucijų tipas, kuris praretina branduolio elementų kiekį įterpiant tuščius tarpus. Gali būti naudojamas l parametras, kuris nurodo kiek bus praretintas branduolio plotis. Dažniausiai yra įterpiami tušti tarpai, bet tai gali būti keičiama [30]. Atliktas

tyrimas parodė, kad siūlomas algoritmas nepaprastai lenkia pažangiausią FPN ir pasižymi patenkinamais kaulų lūžių aptikimo rezultatais 1.3 lentelėje.

Šioje architektūroje pasiūlytas naujas magistralinis tinklas suprojektuotas naudojant išplėstą bloką, kuris naudojamas požymių žemėlapiams išgauti. Funkcinė piramidė su penkiomis skirtingomis skalėmis yra sukurta iš kiekvieno etapo paskutinės fazės magistralinio sluoksnio atliekant duomenų surinkimą ir šoninio ryšio išvesties integravimą. Požymių žemėlapiai ir požymių piramidė yra atskirai įterpiami į regionų pasiūlymų tinklą (RPN), kurie suteikia objektų pasiūlymus kiekvienoje pikselio vietoje. RPN suteikia informaciją Fast R-CNN klasifikatoriui, kurie naudojant inkarus gali turėti lūžių, generuodami 128 interesų sritys (RoI). RoI sujungiantis sluoksnis sugeneruoja mažus požymių žemėlapius su fiksuotu erdvinio dydžiu iš požymiuose esančių RoIs. Lūžio vietų nustatymui ir patikslinimui pasiūlytus regionus yra naudojamas R-CNN klasifikatorius su apkarpytais savybių žemėlapiais, kurių tinklo svoriai atnaujinami kartotinai.

Magistralinio tinklo architektūra - daugumoje objektų aptikimo algoritmų pagrindiniai tinklai naudojami funkcijų žemėlapiams išgauti iš originalių vaizdų. Šiame tinkle siekiant išgauti lūžio ypatybes, savybių išgavimas pagrįstas praretintomis konvoliucijomis ir giliu liekamuoju tinklu. Detalus konvoliucinių sluoksnių išdėstymas tinkle pavaizduotas 4.1. Praretinta konvoliucija gali būti apibūdinama kaip įprastas konvoliucinis sluoksnis su skylėmis. Tai gali padidinti konvoliucinio branduolio imlumo lauką ir gaunamos informacijos kiekį nedidinant parametru skaičiaus.

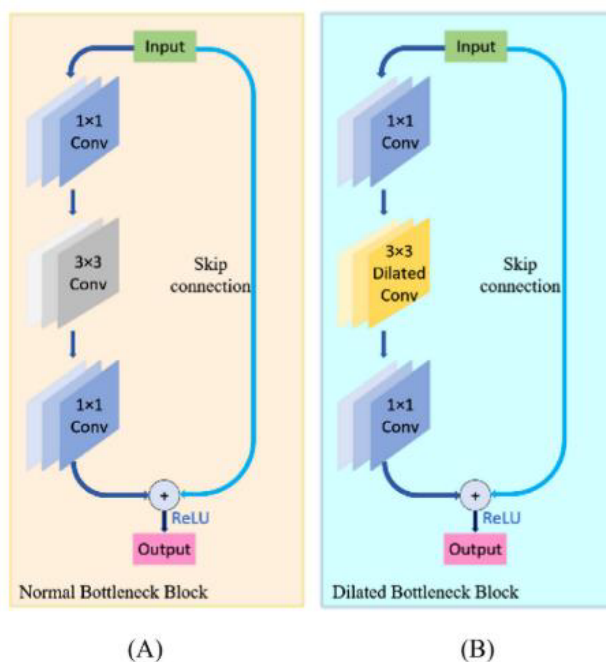
Originalioje FPN architektūroje magistralinis tinklas buvo paremtas ResNet, o praretintoje architektūroje 3×3 konvoliucijos sluoksnis yra pakeistas 3×3 praretinta konvoliucija. Šių tinklų struktūra pavaizduota 1.3 paveiksle. Abu tinklai yra panašūs nes yra sudaryti iš dviejų 1×1 konvoliucinių branduolių ir vieno 3×3 konvoliucinio branduolio ir abu turi praleidžiančias jungtis, kurių dėka gilūs tinklai gali būti ištreniruojami greičiau.

Šių blokų pagrindinis skirtumas yra 3×3 konvoliucijoje, kuri praretintame bloke padidina savybių žemėlapių imlumo lauką išgaunant daugiau informacijos.

Regionų pasiūlymų tinklas DCFPN architektūroje savybių piramidės yra sukurtos iš penkių savybių žemėlapių. Tokia savybių žemėlapių konstrukcija yra paremta šoninių požymių žemėlapių integravimu su išplėtimo (angl. *up-sampling*) operacija, sujungiant požymių žemėlapius su 1×1 konvoliucija, taip sumažinant perduodamų parametru dimensijas iš prieš tai buvusių šoninių jungčių.

1.2 lentelė. DCFPN architektūros rezultatų palyginimas su kitais giliojo mokymosi metodais

Algoritmas	Vidutinis tikslumas (AP)	Magistralinis tinklas
Faster R-CNN	72.1%	ResNet50
FPN	78.2%	ResNet101
Cascade R-CNN	74.7%	DenseNet
DCFPN	82.1%	



1.3 pav. Magistralinis blokas su įprastomis konvoliucijomis pažymėtas kaip A ir magistralinis blokas su išsiplėtusiomis konvoliucijomis pažymėtas kaip B [30].

Praretinto konvoliucinio tinklo pritaikymas lūžių nustatymui parodė didelį pranašumą 1.2 lentelė prieš jau esančius moderniausius metodus. Šis patobulintas kaulų lūžių nustatymo rentgeno nuotraukose metodas naudojant gilųjį mokymąsi gali pagerinti požymių žemėlapių išgavimą ir faktinės informacijos kiekį iš rentgeno nuotraukų.

1.1.6. Giliojo mokymosi modelių panaudojimo su MURA duomenų rinkiniu palyginimas

Konvoliucinių neuroninių tinklų panaudojimas nuotraukų klasifikacijoje, lokalizacijoje ir segmentacijoje yra labai populiarus. Tai pat šių tinklų naudojimas sprendžiant Stanfordo universiteto iškeltą užduotį su viršutinės galūnės rentgeno nuotraukomis MURA duomenų rinkinyje yra plačiai paplitęs literatūroje. Šio rinkinio iškeltas uždavinys yra suklasifikuoti rentgeno nuotraukas į normalias ir su anomalijomis. Bet yra ir mėginimų atlikti segmentaciją ir lokalizaciją su DenseNet modeliais, kadangi jie suteikia klasių aktyvavimo sritis (CAM). Analizuotą architektūrą tikslumo apžvalga yra pateikta 1.3 lentelėje.

MURA duomenų rinkinys dažniausiai naudojamas išspręsti klasifikacijos uždavinį ir patikrinti konvoliucinio neuroninio tinklo galimybes atlikti sudėtingų vaizdų analizę. Sprendžiant šį uždavinį geriausiai veikia sujungti keli modeliai į vieną (angl. *ensemble*), bet apžvelgiant tik architektūras jie į palyginimus nėra įtraukti. Iš 1.3 lentelės galima matyti, kad naujesni ir didesni apmokyti modeliai ant ImageNet sugeba surinkti virš 80% vidutinį tikslumą iš kurių geriausias RestNet-152, bet geriausia Kappa metrikos įvertinimą surinko liekanų konvoliucinis tinklas su praretinimais [10].

1.2. GAN architektūrų pritaikymo analizė

Sužymėtos medicininės vaizdinės medžiagos nėra daug ir jos generavimas reikalauja daug išteklių. Norint sukurti giliojo mokymosi modelius, kuriuos būtų galima pritaikyti reikia itin

1.3 lentelė. Giliojo mokymosi modelių palyginimas naudojant MURA duomenų rinkinį

Pritaikymas	Architektūra	Duomenų rinkinys	Naudotas mokymosi perkėlimas	Tikslumas	Literatūra
Klasifikacija	169-CNN ir DenseNet	MURA	Nenaudotas	AUROC 0.929 Jautrumas 0.815 Specifiškumas 0.887	[12]
Klasifikacija	VGG19	MURA	ImageNet	Vidutinis tikslumas (AP) - 80.45%	[31]
Segmentacija, Lokalizacija	ResNet	MURA	ImageNet	Vidutinis tikslumas (AP) - 78.76%	[31]
Klasifikacija	Faster R-CNN	MURA	Nenaudotas	Vidutinis tikslumas (AP) - 32.72%	[32]
Klasifikacija	RestNet-152	Peties rentgenogramos - 1891	ImageNet	Vidutinis tikslumas (AP) - 86%	[33]
Lokalizacija	DCFPN	MURA - Klubas	Nenaudotas	Vidutinis tikslumas (AP) - 82.1%	[10]
Klasifikacija, Lokalizacija, Segmentacija	Dilated Residual Networks	MURA	ImageNet	Kappa – 0.8	[14]
Lokalizacija	Dviejų stadijų R-CNN	MURA - Žastikaulis	Nenaudotas	Vidutinis tikslumas (AP) - 62.04%	[2]

daug duomenų. Kad eitų treniruoti giliojo mokymosi modelius duomenų rinkinio padidimui įprastai yra naudojamos standartinės duomenų praplėtimo operacijos. Generatyviniai priešiški tinklai siūlo naujas duomenų rinkinio papildymo galimybes [11].

Dėl naujausios pažangos vaizdų generavime naudojant neuroninius tinklus [34, 35, 32, 22, 21, 36, 18], tapo įmanoma sugeneruoti didelį kiekį natūralių vaizdų, kurie užfiksuotų aukšto lygio mokymosi duomenų struktūrą. Nors tokie absoliutiniai (besąlyginiai) modeliai patys savaime yra nepaprastai įdomūs, daugelis praktinių vaizdų modeliavimo pritaikymų reikalauja, kad modelis būtų sąlygojamas išankstine informacija: pavyzdžiui, vaizdo modelis, naudojamas sustiprinti mokymosi planavimą vaizdinėje aplinkoje, turėtų numatyti ateities scenas, atsižvelgiant į konkrečias būsenas ir veiksmus [30]. Taip pat, tokios vaizdo apdorojimo užduotys kaip triukšmo mažinimas, suliejimas, piešimas, raškos didinimas ir spalvinimas, pasikliauna pagerintų vaizdų generavimu, esant triukšmo ar duomenų nepakankamumo sąlygoms. Potencialus sąlyginio generavimo panaudojimas ateityje galėtų būti neuroninių meno kūrinų ir turinio generavimas [37].

Medicininėse rentgeno nuotraukose esančios skirtingos duomenų rinkinio charakteristikos iškelia sunkumą pritaikant giliojo mokymosi modelius, kurie yra apmokyti kituose duomenų rinkiniuose. Tai iškelia labai didelius sunkumus siekiant perkelti šių modelių panaudojimą [11]. Todėl buvo iškelta hipotezė, kad pritaikius generatyvinius priešiškus tinklus atlikti duomenų praplėtimą galėtų padėti lengviau pritaikyti giliojo mokymosi modelius visiškai skirtingų charakteristikų duomenų rinkiniuose.

1.2.1. Generatyvinio priešiško neuroninio tinklo architektūra

Generatyviniai priešiški tinklai yra sudaryti iš dviejų neuroninių tinklų, kurie yra supriešinti vienas su kitu tam, kad būtų generuojami nauji, dirbtiniai duomenų egzemplioriai, kurie gali būti priimti ir panaudoti kaip tikri duomenys. Šie modeliai yra plačiai naudojami generuojant vaizdus, vaizdo ir garso medžiagą [35].

Generatyvus priešiškas modelis yra supriešinamas su adversariu. Generuojantį modelį būtų galima prilyginti klastotojams, gaminantiems padirbtus pinigus, tuomet skiriamasis modelis būtų analogiškas policijai, kuri bando aptikti padirbtus pinigus. Konkurencija šiame žaidime skatina abi puses tobulinti savo metodus tol, kol klastotės nebesiskiria nuo originalių duomenų [35].

Norėdami sužinoti generatoriaus pasiskirstymą, p_g per duomenis x , apibrėžiame pirminius įvesties triukšmo kintamuosius $p_z(z)$, tada atvaizduojame duomenų erdvės susiejimą kaip $G(z; \theta_g)$, kur G yra diferencijuojama funkcija, vaizduojama daugiasluksniu perceptronu (angl. *perceptron*) su parametru θ_g . Taip pat apibrėžiame antrąjį daugiasluksnį perceptroną $D(x; \theta_d)$, kuris išveda vieną reikšmę. $D(x)$ reiškia tikimybę, kad x atsirado iš duomenų, o ne iš p_g . v yra mokomas maksimaliai padidinti tikimybę, kad tiek mokymo pavyzdžiams, tiek pavyzdžiams iš G bus priskirta tinkama etiketė. Tuo pačiu metu mokome G sumažinti $\log(1 - D(G(z)))$ [38].

Kitaip tariant, D ir G žaidžia šį dviejų žaidėjų *min-max* žaidimą su reikšmės funkcija $V(G; D)$,

kuri pateikta (4).

$$\min G \max D V(D, V) = E_{x \sim p_{data(x)}} [\log D(x)] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (4)$$

Praktikoje, (4) lygtis gali nesuteikti pakankamai gradiento, kad G galėtų gerai išmokti. Mokymosi pradžioje, kai G rodiklis yra prastas, D gali atmesti pavyzdžius su aukštu patikimumu, nes jie aiškiai skiriasi nuo mokymo duomenų. Šiuo atveju $\log(1 - D(G(z)))$ prisipildo. Užuoat mokę G sumažinti $\log D(1 - D(G(z)))$, galime išmokyti G padidinti $\log D(G(z))$. Ši tikslinė funkcija gauna tą patį fiksuotą G ir D dinamikos tašką, tačiau suteikia daug stipresnius gradientus mokymosi pradžioje.

1.2.2. Sąlyginio generatyvinio priešiško neuroninio tinklo architektūra

Priešpriešiniai tinklai GAN [35] puikiai sintezuoja realistiškus vaizdus. Jų sąlyginis išplėtimas, sąlyginiai GAN modeliai [39] leidžia valdyti vaizdo sintezę ir įgalina daugybę kompiuterinės vizijos ir grafikos pritaikymų, pavyzdžiui, galima interaktyviai sukurti vaizdą iš vartotojo piešinio [36], perkelti šokio judesį iš filmuotos medžiagos kitam žmogui ar sukurti VR veido animaciją nuotolinei sąveikai [34]. Sąlyginė vaizdų sintezė iškelia realistinių vaizdų generavimo užduotį, kuri yra sąlygojama įvesties duomenų. Pradiniame etape apskaičiuojamas išvesties vaizdas, sudedant dalis iš vieno vaizdo arba naudojant vaizdų rinkinį [36].

Generatyvinis priešiškas tinklas gali būti sąlygojamas papildomos informacijos Y [40]. Y gali būti bet kokia pagalbinė informacija, tokia kaip klasių pavadinimai ar duomenys iš kitų metodų. Ši sąlygiškumą galima įvykdyti paduodant Y diskriminatoriui ir generatoriui kaip papildomą įvesties sluoksnį [39].

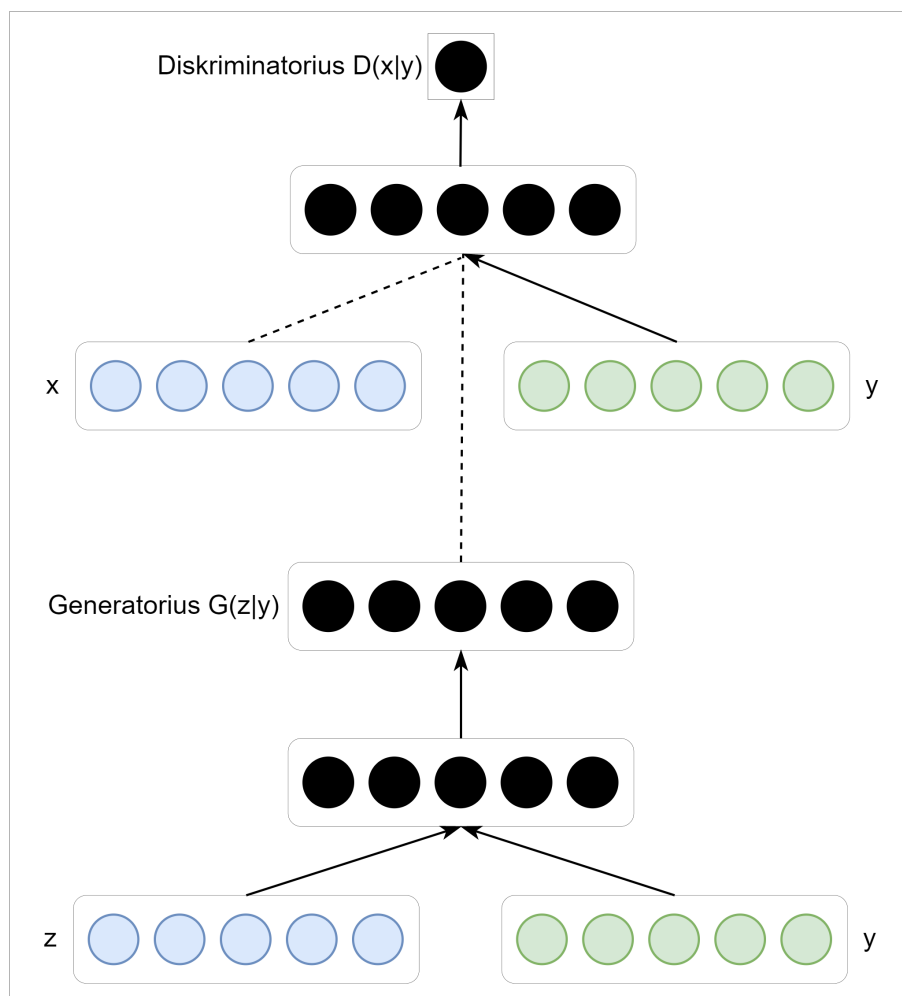
Generatoriuje, pirminis įvesties triukšmas $p_z(z)$ ir y yra kombinuojami į apjungtą paslėptą vaizdinį, o priešpriešinio mokymo metodas suteikia pakankamai lankstumo, kaip šis paslėptas vaizdinys yra sukomponuojamas. Diskriminatoriuje x ir y yra įvestys. Galutinė dviejų žaidėjų $\min - \max$ žaidimo funkcija yra pateikta (5) [39, 40].

$$\min G \max D V(D, G) = E_{x \sim p_{data(x)}} [\log D(x|y)] + E_{z \sim p_z(z)} [\log(1 - D(G(Z|y)))] \quad (5)$$

Treniravimo procese, GAN atnaujinama G ir D parametrai, naudojant gradientu grįstus optimizavimo metodus, tokius kaip stochastinis gradiento šaltinis SGD, Adam [41] ir RMSProp. Visas optimizacijos tikslas yra pasiekiamas, kai D negali atskirti sukurtos imties $x_0 = G(z)$ ir tikrosios imties x , t. y ., kai šioje būsenoje randama pusiausvyra. Praktikoje GAN mokymas dažnai įstringa ties būsenos žlugimu, todėl sunku pasiekti konvergenciją [40].

1.2.3. PixelCNN generatyvinio priešiško neuroninio tinklo architektūra

PixelCNN yra GAN architektūra su stebima tikimybe, kurią pristatė van den Oord ir kt. 2016 metais. Modelis visiškai faktorizuoja vaizdo x tikimybės tankio funkciją visuose jo daliniuose pikseliuose (spalvų kanalai pikseliulyje), kaip funkcija $p(x) = \prod_i p(x_i|x < 1)p(x)$. Sąlyginiai pasiskirstymai $p(x_i|x < 1)$ yra parametrizuojami su konvoliuciniais neuroniniais tinklais ir



1.4 pav. Sąlyginio generatyvinio priešiško neuroninio tinklo struktūra

jie visi naudoja tuos pačius parametrus. PixelCNN yra galingas modelis, kadangi šių sąlygų funkcinė išraiška yra labai lanksti. Taip pat, modelis yra našus, nes sąlygos gali būti įvertinamos lygiagrečiai vaizdai x grafiniėje plokštėje. Šių savybių dėka, PixelCNN atspindi dabartinę generatyvinių priešiško modelių pažangą, kai yra vertinama naudojant logaritminę tikimybę [42].

PixelCNN ir PixelRNN modeliai parodo bendrą pikselių pasiskirstymą vaizde x kaip tokią sąlyginių skirstinių sandaugą, kur x_i yra vienas pikselis. Pikselio reikšmės apskaičiavimas pateiktas (6).

$$p(x) = \prod_{i=1}^{n^2} p(x_i | x_1, \dots, x_{i-1}) \quad (6)$$

Pikselių priklausomybės išdėstomos rastrinio nuskaitymo tvarka: eilutė po eilutės ir pikselis po pikselio, kiekvienoje eilutėje. Todėl kiekvienas pikselis priklauso nuo visų aukščiau ir kairėje esančių pikselių, o ne nuo toliau esančių pikselių [43, 44].

Toliau pažengusi naujesnė uždaroji (angl. *gated*) pikselių CNN architektūra yra sudaryta iš automatiškai regresyvių (angl. *autoregressive*) jungčių, skirtų modeliuoti vaizdus pikselis po pikselio, išskaidant bendrą vaizdo pasiskirstymą pagal gautas sąlygas. Originaliame rašto darbe buvo pasiūlyti du variantai: PixelRNN [45], kur pikselių skirstiniai modeliuojami naudojant dvimatį LSTM [46], ir PixelCNN, kur jie modeliuojami naudojant konvoliucinius tinklus. PixelRNN modeliai įprastai yra našesni, tačiau PixelCNN yra žymiai greičiau treniruojami, nes konvoliucijas yra lengviau išlygiagretinti. Atsižvelgiant į didelį pikselių skaičių dideliuose vaizdo duomenų rinkiniuose, tai yra svarbus pranašumas. Tyrėjai sujungė abiejų modelių stipriąsias puses, ir taip pristatė uždara PixelCNN variantą, suderinantį tiek PixelRNN logaritminę tikimybę su CIFAR ir ImageNet. Taip pat, šis modelis pasižymi dvigubai greitesniu mokymosi greičiu [43].

1.2.4. CycleGAN generatyvinio priešiško neuroninio tinklo architektūra

CycleGAN yra vienas iš pirmųjų modelių, sulaukusių daug dėmesio verčiant vaizdą į vaizdą naudojant nesusietus (angl. *unpaired*) vaizdus. Kadangi Pix2Pix reikalauja visiškai suderintų vaizdų porų, kad vaizdas iš X domeno į Y domeną būtų paverstas naudojantis generatoriumi $G: X \rightarrow Y$, CycleGAN pristato papildomą generatorių $F: Y \rightarrow X$, kuris generuoja vaizdus iš domeno Y atgal į X domeną. CycleGAN idėja yra tokia: kai įvestis x perduodama per visą pakeitimų ciklą, $X \rightarrow Y \rightarrow X$, išvestis turi atitikti įvestį [38].

Kad tai būtų pasiekta, turi būti įvykdyti du ciklai: ciklas pirmyn, $X \rightarrow G(x) \rightarrow F(G(x)) \approx x$, ir atgalinis ciklas, $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$. Šis procesas yra proteguojamas sumažinus ciklo nuoseklumo praradimą (Lcyc), kuris matuoja neatitikimą tarp įvesties vaizdo ir cikle gauto išvesties vaizdo. Naudojant $T2$ svertinį ir $T1$ svertinio magnetinio rezonanso (MRI) vertimą kaip pavyzdį, generatorius G naudojamas įvesties $T2$ svertiniame vaizde, kad būtų sukurtas $T1$ svertinis vaizdas iš įvesties. Tada susintetintas $T1$ svertinis vaizdas vėl konvertuojamas į $T2$ svertinį vaizdą naudojant generatorių F . Originalo ir išversto $T2$ svertinio vaizdo neatitikimas

sumažinamas naudojant ciklo nuoseklumo praradimą. Ciklo nuoseklumo praradimas skatina generatorius susiaurinti galimų pasikeitimų aibę ir taip susintetinti vaizdus, kurie struktūriškai atitinka įvestis. Transformuotas $G(x)$ įvedamas į diskriminatorių DY , kur atskiria realius vaizdus y , o diskriminatorius DX siekia atskirti išverstus vaizdus $F(y)$ ir tikrus vaizdus x [38]. Realiose klinikiniuose pritaikymuose dauguma vaizdų yra nesusieti, tad CycleGAN gali būti plačiai panaudojamas radiologinių duomenų moksle.

1.2.5. Generatyvinių priešišku neuroninių tinklų apibendrinimas

GAN modeliai gali kompensuoti duomenų trūkumą arba klasių disbalansą generuodami patikimus vaizdus, kuriais gali būti treniruojamas giliojo mokymosi modelis. Lyginant su tradiciniais duomenų praplėtimo būdais, kaip paslinkimai, apkarpymai, horizontalūs ir vertikalūs apsukimai, rotacija, GAN modeliu grįstas duomenų praplėtimas suteikia naujus vaizdinius, panašius į originalius, kuriais galima papildyti mokymosi duomenų rinkinį. Pavyzdžiui, viename tyrime CycleGAN buvo naudojamas normalių šlaunų MRI vaizdams, siekiant padidinti riebalų infiltraciją, o tai palengvino šlaunies MRI segmentavimo tinklo sukūrimą su sunkia riebalų infiltracija, nes sintetiniai vaizdai buvo įtraukti į treniruočių duomenų rinkinį. Be to, siekiant automatizuoti B režimo vaizdų generavimą ir ženklumą, buvo pasiūlyta naudoti B režimo raumenų ir skeleto bei sintetines segmentavimo kaukes, treniruotas su CycleGAN. GAN pagrįstų duomenų pakeitimo procedūrų buvo atlikta itin mažai atliekant raumenų ir kaulų sistemos vaizdavimą. Tačiau tyrimai, atlikti kitose srityse (pvz., kepenų pažeidimo sintezė iš riboto KT duomenų rinkinio, kūno KT sintezė iš skaitmeninių kūno fantomų), parodė, kad yra įmanoma gauti didelį kiekį vaizdinių variantų, kurie gali būti taikomi raumenų ir kaulų sistemų srityje [38].

1.3. Duomenų rinkiniai

Šio tyrimo metu buvo naudotas Stanfordo mašininio mokymosi grupės sukurtas MURA duomenų rinkinys, skirtas atlikti tyrimus susijusius su rentgeno nuotraukų analize.

1.3.1. MURA duomenų rinkinys

MURA yra vienas didžiausių viešų rentgeno nuotraukų duomenų rinkinių, paskelbtų 2018 m. sausio mėn. Duomenų rinkinį sudaro 40 895 viršutinės galūnės daugialypiai rentgeno grafiniai vaizdai, įskaitant pečių, žastikaulio, alkūnės, dilbio, riešo, plaštakos ir piršto. Visi vaizdai yra 8 bitų PNG formatu. Atliekant tyrimą galima panaudoti žastikaulio, šlaunikaulio ir dilbio vaizdus kaip treniravimo ir tyrimo duomenis. Rentgeno nuotraukos be anomalijų bus naudojamos modelio testavimui, bet netinka modelio treniravimui. Tyrimo metu treniravimo ir validavimo duomenų rinkinio pasiskirstymas nebuvo išreikštas santykiu [12], bet nuspręsta naudoti MURA rinkinio pasiskirstymą siekiant išlaikyti galimybę rezultatus palyginti su jau esamų modelių tyrimų rezultatais. Treniravimo ir validavimo rentgeno nuotraukų pasiskirstymas pagal sritį pateiktas 1.4 lentelėje.

MURA duomenų rinkinį sudaro 9 045 normalūs ir 5 818 anomaliją turinčios rentgeno nuotraukos viršutinės galūnės kaulų rentgenografijos tyrimai, kuriuose yra peties, žastikaulio, alkūnės, dilbio, riešo, plaštakos ir piršto sritys. MURA yra pats didžiausias viešas viršutinės galūnės rentgenografijos vaizdų duomenų rinkinys [12].

1.4 lentelė. MURA duomenų rinkinio duomenų pasiskirstymas srityse

Sritis	Treniravimo be anomalijų	Treniravimo su anomalija	Tikrinimo be anomalijų	Tikrinimo su anomalija	Iš viso
Alkūnė	1 094	660	92	66	1 912
Rankos pirštai	1 280	655	92	83	2 110
Ranka	1 497	521	101	66	2 185
Žastikaulis	321	271	68	67	727
Dilbis	590	287	69	64	1 010
Peties sąnarys	1 364	1 457	99	95	3 015
Riešas	1 364	1 326	140	97	3 697
Iš viso	8 280	5 177	661	538	14 656

1.3.2. LERA duomenų rinkinys

HIPAA suderintame su IRB ir patvirtintame tyrime surinkti duomenys iš 182 pacientų, kuriems buvo atlikta rentgeno nuotrauka Stanfordo universiteto medicinos centre nuo 2003 iki 2014 metų. Duomenų rinkinį sudaro pėdos, kelio, kulkšnies ar klubo vaizdai susieti su kiekvienu pacientu. Etiketės priskiriamos pacientui, nurodant, kad ta pati klasifikacija taikoma visiems konkretaus paciento atvaizdams. Kadangi vaizdai buvo renkami per dvylika metų, duomenų rinkinyje yra labai įvairių rentgeno nuotraukų, kurios yra skirtingo dydžio, skiriamosios gebos ir spalvos [47].

1.3.3. Duomenų apdorojimo metodai

Rentgeno nuotraukos originaliame MURA duomenų rinkinyje turi dvi pagrindines problemas. Nuotraukose yra triukšmo ir tamsus vaizdų fonas apsunkinantis nuotraukų peržiūrą. Duomenų pasiruošimą kaip svarbų procesą išskiria daug autorių [2, 10, 12]. Atliekant rentgeno nuotraukų analizę tyrimuose yra aptinkamos dažniausios klaidos, dėl kurių yra prarandamas tikslumas. Nuotraukų apdorojimo metu yra stengiamasi padidinti esamų nuotraukų kokybę siekiant aptikti retus ir mažus lūžius ir išspręsti dažniausias modelių tikslumą mažinančias problemas:

1. Paslėpta lūžio linija [2, 10];
2. Paslėpti lūžiai skirtingose plokštumose [10];
3. Lūžiai yra per maži [2, 10];
4. Daug lūžių vienoje vietoje [2];
5. Retų lūžių formų aptikimas [10].

Tyrimų autorių teigimu, nuotraukų kokybės gerinimas ir didesnio nuotraukų kiekio užtikrinimas leidžia sumažinti išvardintas dažniausias problemas ir užtikrinti modelio tikslumą.

1.3.4. Triukšmo pašalinimas iš rentgeno nuotraukų

Norint sušvelninti triukšmo poveikį vaizdai yra apdorojami morfologiniu metodu. Panaudojus atidarymo operacija su 21×21 branduoliu yra išgaunamos geresnės kokybės pilkos spalvos vaizdas. Atliekant nuotraukos apdorojimo operaciją galima pašalinti pavienį vaizde esantį triukšmą ir nustatyti pagrindinę sritį. Atliekant šią operaciją yra išryškinamos sritys, kuriose yra kaulai ir lūžiai.

1.3.5. Rentgeno nuotraukos ryškumo didinimas

Rentgeno nuotraukos ryškumas yra padidinamas naudojant kumuliacinį įprasto paskirstymo funkciją, kuri atlieka pirminio vaizdo ištempimą. Čia pagrindinio ploto pikselių dispersija laikoma normalaus pasiskirstymo dispersija. Atsižvelgiant į tai, kad lūžio plotas dažnai yra ryškiausias pagrindinio ploto plotas, mes laikome maksimalią pagrindinio ploto pikselių vertę kaip normalaus pasiskirstymo vidurkį, kad transformacija būtų jautri lūžio sričiai. Atliekant aukščiau nurodytas dvi operacijas, galima pagerinti viso vaizdo kontrastą, o transformuotų vaizdų lūžių plotas tampa aiškesnis ir ryškesnis.

1.3.6. Stebimo srities išplėtimas rentgeno nuotraukose mažuose lūžiuose

Mažų objektų aptikimas išlieka iššūkis vaizdų analizavimo srityje. Mažų lūžių nustatymas MURA duomenų rinkinyje vienas iš pagrindinių sunkumų. Tiesą sakant, mažoje rentgeno nuotraukoje, kurios skalė yra mažesnė nei 40×40 pikselių, yra nepakankama informacija, kad tinklas galėtų diagnozuoti, ar joje yra lūžis. Norint išgauti naudingą informaciją iš šių mažų RoI, būtina išplėsti stebimos vietos lauką.

Imtino lauko išplėtimo įgyvendinimo procese aptikus sritį kurios plotis yra mažesnis nei 30 pikselių prie jo pločio pridedama 20 papildomų šalia esančių pikselių. Jei jo plotas yra mažesnis nei 40, prie jo pločio pridedama 10 pikselių. Šie veiksmai taip pat taikomi aptikto lūžio vietos nustatyme ilgio koregavimui. Ši taisyklė užtikrina, kad mažų lūžių imlus laukas yra padidinamas siekiant pagerinti tikslumą aptinkant lūžius.

1.3.7. Rentgeno nuotraukų apdorojimas

Rentgenologų atliktas nuotraukų apdorojimas buvo atliktas tyrime [33]. Šio tyrimo metu atliko nuotraukų apkarpyimą į kvadrato formą, kuriame žastikaulio galva ir kaklas buvo sutelkti ir sudarė maždaug 50% kvadratinio vaizdo, kuris pakeistas į 256×256 pikselių dydį ir saugomas kaip JPEG failas. Nuotraukų apdorojimo procese autoriai skatina nekeisti nuotraukų orientacijos, kadangi ji būna vienoda dėl nuotraukų darymo standarto, bet siekiant sumažinti per didelį atitikimą naudoti nuo -10 iki 10 laipsnių rotaciją visoms nuotraukoms. Toks duomenų apdorojimas padidino modelio tikslumą, bet naudotas nuostolingas JPEG suspaudimas galimai paveikė vaizdo kokybę, todėl autorių teigimu geriau naudoti kokybės išlaikantį glaudinimą, pvz., PNG ar TIFF.

Nuotraukos tai pat gali būti apdorojamos sukant, apkarpyant ir pritaikius kraštinių santykių, išsaugant keitimo mastelio operaciją, kad būtų gaunama fiksuota skiriamoji geba [12]. Taip pat yra atliekamas ir išankstinio apdorojimo etapas, kuriame vaizdai apdorojami naudojant tokias technologijas kaip Haar bangos transformacija ir masto nekintančių funkcijų transformacija SIFT. Haar bangų transformacija reikalinga norint iš anksto apdoroti vaizdus, siekiant juos suspausti ir sutaupyti vietos atmintyje. SIFT yra galingas metodas aptikti ypatybes, kurios turi didelį atsparumą sukimui, glaudinimui ir mastelio keitimui [48].

1.3.8. Duomenų apdorojimo rezultatai

Dėl rentgeno nuotraukų originaliame MURA duomenų rinkinyje susiduriame su dvejomis problemomis. Viena yra triukšmo egzistavimas, kita yra tamsus vaizdų fonas. Siekiant išspręsti

šias problemas reikia atlikti išankstinį rentgeno nuotraukų apdorojimą.

Norėdami sušvelninti triukšmo poveikį, mes morfologiniu metodu apdorojame vaizdus. Morfologinė atidarymo operacija $\times 21$ su 21 branduolys yra pritaikytas pilkos spalvos vaizdui apdoroti. Atidarymo operacija yra skirta izoliuoti vaizdo triukšmą, kad būtų galima jį pašalinti, o tuo tarpu tai pat galima nustatyti pagrindinę rentgeno nuotraukoje kaulų buvimo sritį.

Norėdami padidinti vaizdo ryškumą, tyrime galima pritaikyti normalaus pasiskirstymo kaupiamąją funkciją pilkame ruože ir pritaikyti originaliame vaizde. Čia pagrindinio ploto pikselių dispersija laikoma normalaus pasiskirstymo dispersija. Atsižvelgdami į tai, kad lūžio plotas dažnai yra ryškiausias plotas pagrindinėje srityje, mes imame didžiausią pikselių vertę pagrindinis plotas yra normalaus pasiskirstymo vidurkis, kad transformacija būtų jautri lūžio sričiai.

Atliekant aukščiau nurodytas dvi operacijas, galima pagerinti viso vaizdo kontrastą ir pertvarkyti lūžio plotą vaizdai tampa aiškesni ir ryškesni. Išankstinio apdorojimo procedūra parodyta 1.5 paveiksle.



1.5 pav. MURA duomenų rinkinio alkūnės rentgeno nuotraukų apdorojimo rezultatai [2]

Atlikus literatūros analizę pastebėta, kad vienas svarbiausių veiksnių dirbant su MURA duomenų rinkiniu yra rentgeno nuotraukų apdorojimas. Tai yra iššūkis nes šį duomenų rinkinį sudaro nuotraukos iš skirtingai atliktų tyrimų. Visas duomenų rinkinys nėra surinktas iš vienos gydymo įstaigos, bet iš skirtingų tyrimų kurie yra viešai prieinami. Todėl apdoroti nuotraukas yra vienas pirmųjų ir svarbiausių veiksnių prieš atliekant giliojo mokymosi modelių treniravimą. Literatūroje yra plačiai naudojamų įrankių ir metodų pasirinkimas kurie pateikta 1.5 lentelėje.

Iš lentelėje pateiktų metodų galima išskirti SIFT, kuris yra galingas metodas aptikti ypatybes, turinčias didelį atsparumą sukimui, glaudinimui ir masteliui. Taip pat Haar bangų transformacija reikalinga norint iš anksto apdoroti vaizdus, siekiant juos suspausti ir sutaupyti vietos atmintyje.

Apibendrinant lentelės duomenis, tyrimuose dažniausiai yra naudojama rentgeno nuotraukų apkarpymas, kraštinių santykio keitimas, horizontalus ir vertikalus apsuikimas.

1.4. Programinės įrangos analizė

Python programavimo kalba įsitvirtina kaip viena populiariausių mokslinio skaičiavimo kalbų. Dėl aukšto interaktyvumo lygio ir besiplečiančios mokslinių bibliotekų ekosistemos tai yra patrauklus pasirinkimas algoritminiam vystymuisi ir tiriamajai duomenų analizei. [56, 57].

1.5 lentelė. MURA duomenų rinkinio apdorojimo naudojimo literatūros apžvalga

Metodas	Literatūra
Nuotraukos šviesumo keitimas	[12, 2]
Taikomas kraštinių santykis (angl. <i>aspect ratio</i>)	[1, 49, 50, 51, 52]
Kaupiamasis paskirstymas (angl. <i>comulative distribution</i>)	[12, 2]
SIFT	[53, 54]
Morfologinis apdorojimas	[2, 12]
Horizontalus arba vertikalus apvertimas	[10, 31]
Nuotraukos pritraukimas (angl. <i>zooming</i>)	[1, 31]
Normalizacija	[1, 55]
HAAR bangų transformacija	[53, 54]
Nuotraukos rotacijos (angl. <i>rotating</i>)	[1, 49, 52]
Nuotraukos dydžio keitimas	[1, 55]
Apkarpymas (angl. <i>cropping</i>)	[12, 49, 50, 52, 13]
Lokalus kontrastas	[55]
Mastelio keitimas (angl. <i>rescaling</i>)	[1, 49, 13]

Tačiau, kaip bendrosios paskirties kalba, ji vis dažniau vartojama ne tik akademinėje aplinkoje, bet ir pramonėje [58].

1.4.1. Giliojo mokymosi karkasų analizė

Mašininis mokymasis remiasi algoritmais. Šiuos algoritmus yra labai sudėtinga suprasti ir su jais dirbti, nebent esate duomenų mokslininkas ar mašininio mokymosi ekspertas [59]. Mašininio mokymosi karkasai supaprastina mašininio mokymosi algoritmus. Karkasai yra bet koks įrankis, sąsaja ar biblioteka, leidžianti lengvai kurti mašininio mokymosi modelius, nesigilinant į pagrindinių algoritmų veikimą. Yra įvairių mašininio mokymosi karkasų, pritaikytų skirtingiems tikslams. Didžioje dalyje karkasų yra naudojama Python mašininio mokymosi programavimo kalba.

Galima teigti, kad Keras [60], TensorFlow [61, 62], PyTorch [63], ML.NET [64] ir scikit-learn [58] yra populiariausios mašininio mokymosi karkasai. Visi šie karkasai turi įrankių atlikti vaizdų apdorojimą, kurti giliojo mokymosi modelius, suteikia sąsaja sukurti ir išleisti modelį. Tačiau renkantis karkasą pirmiausia yra atsižvelgiama į jų galimybes įgyvendinti skirtingus giliojo mokymosi modelių architektūras. Šio darbo metu bus atliekami eksperimentai su asmenine įranga įgyvendinant modelius ir testuojant jų veikimo efektyvumą aptinkant anomalijas rentgeno nuotraukose.

1.4.2. Tensorflow karkaso apžvalga

Viena iš plačiausiai šiandien naudojamų karkasų yra TensorFlow [61]. Jį sukūrė „Google“ 2015 m., jis yra prižiūrimas ir nuolat atnaujinamas integruojant naujausių giliojo mokymosi tyrimų rezultatus. Todėl TensorFlow palaiko daugybę naujausių neuroninių tinklų sluoksnių, aktyvinimo funkcijų, optimizatorių ir įrankių, skirtų giliųjų neuronų tinklų analizei, profiliavimui ir derinimui (angl. *debugging*). Tam, kad būtų užtikrintas geras vieno mazgo našumas, sudėtingi skaičiavimo

branduoliai, tokie kaip konvoliucijos, tankios matricos daugybos ir kt., naudoja optimizuotas vidines bibliotekas [62].

Naudotojas gali sukurti ir įdiegti giliojo mokymosi modelius nuosekliojo programavimo metodu Python, tačiau vykdymo fazėje TensorFlow yra labai asinchroniškas, t. y., kai pradamas vadinamasis seansas (angl. *session*) yra sukonstruojamas ir vykdomas viso skaičiavimo matematinis grafikas asinchroniniu būdu, pvz., vykdant nepriklausomas grafo briaunas lygiagrečiai arba netvarkingai. Taikant paskirstytą TensorFlow skaičiavimo metodą, reikia į tai atsižvelgti ir nepasikliauti fiksuota tam tikrų kolektyvinių operacijų, pvz., visuotinių sumažinimų ar transliacijų tvarka [62].

1.4.3. Pytorch karkaso apžvalga

Giluminio mokymosi sistemos dažnai buvo sutelktos į patogumą arba greitį, bet ne į abu. PyTorch yra mašininio mokymosi biblioteka, parodanti, kad šie du uždaviniai iš tikrųjų yra suderinami: joje pateikiamas imperatyvus ir Python programavimo kalbos stilius, palaikantis kodą kaip modelį, palengvinantis derinimą ir suderinamas su kitomis populiariomis mokslinėmis kompiuterinėmis bibliotekomis, kartu išliekantis efektyvus ir palaikantis aparatūros greitintuvus, tokius kaip GPU. PyTorch – Python biblioteka, kuri iš karto atlieka dinaminį *tensor* skaičiavimą su automatiniu diferencijavimu ir GPU pagreitinimu, ir tai atlieka išlaikant našumą, kuris yra neatsiliekančias nuo dabartinių giluminio mokymosi bibliotekų. Ši giliojo mokymosi karkasas mokslininkų bendruomenėje yra labai populiarus ir plačiai naudojamas atliekant tyrimus.

Pagrindiniai Pytorch karkaso principai [63]:

Paremtas Python programavimo kalbos standartais - duomenų mokslininkai yra susipažinę su Python kalba, jos programavimo modeliu ir įrankiais. PyTorch turėtų būti pirmos klasės tos ekosistemos narys. Tai atitinka bendrai nustatytus dizaino uždavinius, kad sąsajos būtų paprastos ir nuoseklios, idealiu atveju taikant vieną idiomatinį veiksmų atlikimo būdą. Jis taip pat natūraliai integruojamas su standartiniais braižymo, derinimo ir duomenų apdorojimo įrankiais.

Sukurta tyrėjams – PyTorch siekia, kad rašymo modeliai, duomenų įkėlimo priemonės ir optimizavimo priemonės būtų kuo lengvesnės ir produktyvesnės. Mašininiam mokymuisi būdingą sudėtingumą turėtų tvarkyti PyTorch biblioteka ir paslėpti už intuityvių API (angl. *application programming interface*), neturinčių šalutinių poveikių ir netikėtų našumo šuolių.

Greitaveika – kad PyTorch būtų naudingas jis turi užtikrinti aukštą našumą, išlaikant paprastumą ir lengvą naudojimą. Pagal šio karkaso kūrimo principus mainyti 10% greičio už žymiai paprasčiau naudojamą modelį nėra 100% priimtina. Todėl siekiant padidinti karkaso našumą jo vidinis įgyvendinimas yra sudėtingas. Karkasas tai pat suteikia įrankius leidžiančius tyrėjams rankiniu būdu valdyti kodo vykdymą kas leidžia naudoti savo būdus našumui pagerinti, nepriklausomus nuo tų, kuriuos biblioteka teikia automatiškai.

Blogau yra geriau - atsižvelgiant į fiksuotą inžinerinių išteklių kiekį, laikas, sutaupytas išlaikant

paprastą PyTorch vidinį įgyvendinimą, gali būti naudojamas papildomoms funkcijoms įdiegti, prisitaikyti prie naujų situacijų ir neatsilikti nuo greitos pažangos dirbtinio intelekto srityje. Todėl geriau paprastas įgyvendinimas, nors ir nevisiškai užbaigtas, nei išsamus, bet sudėtingas ir sunkiai prižiūrimas dizainas.

PyTorch tapo populiariu įrankiu gilaus mokymosi tyrimų bendruomenėje, atsižvelgiant į naudojimo patogumą ir našumo aspektus. Šis karkasas remiasi naujausiomis gilaus mokymosi tendencijomis ir pažanga. Karkaso autoriai ateityje planuoja ir toliau gerinti PyTorch greitį ir pritaikymo galimybes.

1.4.4. ML.NET karkaso apžvalga

ML.NET yra mašininio mokymosi karkasas leidžiantis programuotojams kurti ir diegti programose sudėtingus ir naujausius dirbtinio intelekto modelius. Naudojant šį karkasą modelio apmokymas ir naudojimas atlikti prognozes yra visiškai vienodas procesas, todėl nereikia jaudintis dėl skirtingų veikimo aplinkų. Pačio modelio panaudojimas yra labai lengvas, nes reikia tik apjungti įvesties duomenis ir rezultatus į norimą formatą. Šis karkasas nuo pagrindų yra sukurtas suteikti C# programuotojams lengvą sąsają naudoti mašininio mokymosi modelius sukurtose programose. Tūkstantis Microsoft duomenų analitikos mokslininkų ir kūrėjų per pastarąjį dešimtmetį naudojo ML.NET integruojant mašininio mokymosi modelius į 100 produktų ir paslaugų integravimą, kurie yra naudojami 100 milijonų vartotojų pasaulyje [64].

ML.NET yra sąsaja tarp programuotojo ir mašininio mokymosi technologijų, todėl šis karkasas yra pritaikytas naudoti modelių integracijai. Šio tyrimo metu yra atliekami mašininio mokymosi modelių pakeitimai ir tobulinimai, todėl reikalinga pilna prieiga. Todėl šis karkasas nėra tinkamas, nes specializuojasi tik giliojo mokymosi modelių integravime.

1.4.5. Scikit learn karkaso apžvalga

Scikit-learn yra Python modulis, integruojantis platų pažangiausių mašininio mokymosi algoritmų spektrą vidutinio masto prižiūrimoms (angl. *supervised*) ir neprižiūrimoms (angl. *unsupervised*) problemoms spręsti. Šiame pakete pagrindinis dėmesys yra skiriamas mašininio mokymosi perteikimui ne specialistams, naudojantiems bendrosios paskirties aukšto lygio kalbą. Yra išskiriamas naudojimo paprastumas, našumas, dokumentacija ir API nuoseklumas.

Scikit-learn naudoja turtingą aplinką, kad pateiktų pažangiausių daugelio gerai žinomų mašininio mokymosi algoritmų įgyvendinimus, kartu išlaikant lengvai naudojamą sąsają kuri glaudžiai integruota su Python kalba. Pagrindiniai Scikit-learn skirtumai nuo kitų Python mašininio mokymosi įrankių rinkinių [58]:

Kodo kokybė – užuot suteikus kuo daugiau funkcijų, projekto tikslas yra užtikrinti patikimą įgyvendinimą. Kodo kokybė užtikrinama vienetų testais (angl. *unit tests*) ir naudojant statinės analizės priemones – nuo 0.8 versijos kodo padengimas testais siekia 81%.

Nesudėtingas dizainas ir API – siekiant palengvinti naudojimosi pradžia, yra vengiama karkasinio kodo ir įvairių objektų skaičius yra sumažinamas iki minimumo, tai pat naudojamas

NumPy, kuris Python kalbos programuotojams jau yra pažįstamas, todėl naudojimosi pradžia tampa lengvesnė.

Dokumentacija – scikit-learn pateikia 300 puslapių vartotojo vadovą, įskaitant klasių aprašymus, mokymo programą, diegimo instrukcijas, taip pat daugiau nei 60 pavyzdžių, kai kuriuose pateikiamos realios programos.

Scikit-learn suteikia prieigą prie daugybės mašininio mokymosi algoritmų, tiek prižiūrimų, tiek neprižiūrimų, naudodamas nuoseklią, į užduotis orientuotą sąsają, taip leidžiančią lengvai palyginti konkrečios programos metodus. Kadangi jis remiasi moksline Python ekosistema, jį galima lengvai integruoti į kitų sričių programas, kurios nėra statistinių duomenų analizės srityje [58]. Bet šis karkasas nesiorientuoja į giliojo mokymosi modelių kūrimą, bet yra neatskiriama dalis pagalbinių įrankių atliekant tyrimą ir analizuojant duomenis.

1.4.6. Mašininio mokymosi karkasų apibendrinimas

Atlikus giliojo mokymosi karkasų analizę buvo nuspręsta naudoti Pytorch biblioteką. Pytorch yra naujesnis ir labai populiarėjantis giliojo mokymosi įrankis, bet TensorFlow pasižymi didesniu funkcionalumu. Atsižvelgiant į eksperimentų reikalavimus ir kad modeliai bus laikomi Docker vaizduose karkasų pasirinkimas eksperimento metu nėra svarbus pasirinkimas. Skirtumas tarp karkasų greitaveikos nebus tyrimo metu vertinamas, todėl atliekant eksperimentus skirtingi giliojo mokymosi modeliai bus įgyvendinti Pytorch ar TensorFlow karkase.

2. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo projektas

Giliojo mokymosi metodai yra plačiai pritaikyti atlikti klasifikavimo užduotį medicinos srityje. Pagrindinė problema siekiant integruoti šiuos modelius yra duomenų rinkinių surinkimas ir jų paruošimas naudojimui su giliojo mokymosi modeliais. Ne visose medicinos srityse yra didelis rentgenogramų kiekis ir tai sukelia problema, kad naujausi modeliai negali būti apmokyti išspręsti klasifikavimo uždavinį priimtinam tikslumui. Norint išspręsti šią problemą šiame darbe atliekamas tyrimas norint sužinoti kokią įtaką skirtingų giliojo mokymosi architektūroms turi duomenų rinkinio augmentacija ir išankstinis rentgeno nuotraukų apdorojimas. Tai pat eksperimentiškai ištiriamas naujausių GAN modelių architektūrų pritaikymas išplėsti duomenų rinkinį sugeneruotomis nuotraukomis.

Šio darbo projektas yra suskirstytas į tris sistemos dalis. Pirmoji yra tyrėjo kuriami duomenų rinkiniai ir atliekami eksperimentai. O antroji dalis yra atsakinga už giliojo mokymosi modelių treniravimą ir statistinės informacijos rinkimą. Paskutiniai eksperimentai yra su GAN modeliais ir jų pritaikymu generuoti rentgeno nuotraukas specifinėse duomenų aibėse. Tiriamajame projekte siekiama surinkti informaciją apie duomenų rinkinių savybių daromą įtaką giliojo mokymosi modeliams.

2.1. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo projekto reikalavimų analizė

Tyrimo metu yra atliekami eksperimentai matuojant giliojo mokymosi modelius. Eksperimento metu yra stebimas tikslumas tarp skirtingų rentgeno nuotraukų duomenų rinkinių. Duomenų rinkiniai gali skirtis atliktu išankstiniu apdorojimu (angl. *preprocessing*), augmentacija (angl. *augmentation*) arba išplėstu duomenų rinkiniu, kur papildomos nuotraukos sugeneruotos GAN modelio. Giliojo mokymosi modelių tikslumo tyrimo rezultatai aptinkant anomalijas yra naudojami programos tobulinimo procese. Giliojo mokymosi modelių tikslumas aptinkant anomalijas skirtingose kūno dalių rentgeno nuotraukose yra fiksuojamas atliekant eksperimentinius modelių apmokymus su skirtingais duomenų rinkiniais ir skirtingais globaliais parametrais. Šių eksperimentų rezultatai suteikia išvalgų nustatant geriausius giliojo mokymosi modelius analizuojant skirtingų kūno dalių rentgeno nuotraukas ir surinkta informacija yra panaudojama priimant išvadą, kurie modeliai yra tinkamiausi skirtingų rentgeno nuotraukų anomalijų aptikimui.

Sistemos realizavimui iškelti reikalavimai, kuriuos reikia įgyvendinti norint atlikti tyrimą. Sistemos reikalavimai suskirstyti į funkcinius, nefunkcinius, programinės įrangos ir testavimo.

2.2. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo funkciniai reikalavimai

1. Tyrėjas turi galėti filtruoti rentgeno nuotraukas pagal sritį, anomaliją ir sukūrimo datą;
2. Tyrėjas turi galėti kurti rentgeno nuotraukų duomenų rinkinius;
3. Tyrėjas turi galėti matyti dirbtinio intelekto tikslumo kitimo statistiką;
4. Tyrėjas turi galėti pritaikyti išankstinio apdorojimo veiksmus duomenų rinkiniui;

5. Sistema turi treniruoti giliojo mokymosi modelius su pasirinktu duomenų rinkiniu;
6. Sistema turi atlikti giliojo mokymosi metodų testavimo eksperimentus;
7. Sistema turi generuoti giliojo mokymosi modelių tikslumo statistiką;
8. Sistema turi generuoti duomenų rinkinį su pasirinkta rentgeno nuotraukų aibe ir atlikti išankstinį nuotraukų apdorojimą;
9. Sistema turi generuoti naujas rentgeno nuotraukas panaudojant GAN modelius.

2.3. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo nefunkciniai reikalavimai

1. ASP.NET 6 karkaso versija;
2. MSSQL duomenų valdymo sistema;
3. Docker konteinerizacija;
4. Kubernetes atvirojo kodo sistema, skirta automatizuoti konteinerizuotų programų diegimą ir valdymą;
5. Pytorch giliojo mokymosi modelių kūrimo karkasas;
6. Sistemoje klaidos turi būti registruojamos į duomenų bazę.

2.4. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo reikalavimai programinei įrangai

1. Giliojo mokymosi modeliai ir sistema turi veikti Docker konteineriuose;
2. Vidinė sąsaja turi palaikyti C# programavimo kalbą;
3. Giliojo mokymosi modeliai sukurti naudojant Pytorch karkasą.

2.4.1. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo reikalavimai testavimui

1. Sistema renka statistinę informaciją apie modelių veikimą ir leidžia ją peržiūrėti;
2. Tyrėjas gali kurti duomenų rinkinius ir juos filtruoti ir peržiūrėti;
3. Eksperimento veikimas yra stebimas ir informacija išsaugoma į duomenų bazę.

Sistema glaudžiai bendrauja su vidine sistemos sąsaja, kuri yra atsakinga už nuolatinį giliojo mokymosi modelių treniravimą ir eksperimentų rezultatų saugojimą. Tai pat vidinėje sąsajoje atliekamas duomenų rinkinių kūrimas ir rentgeno nuotraukų apdorojimas.

MURA rentgeno nuotraukų duomenų rinkinys yra surinktas iš viešai prieinamų pacientų nuotraukų, kurios paskelbtos moksliniuose tyrimuose. Šis duomenų rinkinys pasižymi, kad nuotraukos suskirstytos pagal pacientus ir sužymėtos pagal anomalijų turėjimą. Kadangi nuotraukos nėra iš vieno šaltinio jos skiriasi dydžių, failo formatu, fotografavimo kampu, nuotraukų kiekiu vienam pacientui ir t.t. Literatūroje duomenų rinkinio apdorojimas įvertinamas kaip vienas iš svarbiausių kriterijų norint pasiekti aukštą tikslumą. Dėl šios priežasties visos nuotraukos prieš panaudojimą yra apdorojamos pritaikant šiuos veiksmus:

1. Rentgeno nuotraukų formato keitimas į jpg;
2. Nuotraukos dydžio keitimas į 224×224 ;
3. Stebimo srities išplėtimas;

4. Kaulų srities išdidinimas panaudojant OpenCv kontūrų (angl. *contour*) nustatymą ir apkarpymą.

Atliekant eksperimentus yra stebimas ir išankstinis nuotraukų apdorojimo poveikis modelių tikslumui, todėl skirtingiems eksperimentams gali būti pritaikyti ir šie veiksmai:

1. Ryškumo didinimas;
2. Rentgenogramos kaukės (angl. *mask*) radimas;
3. *Contrast Limited Adaptive Histogram Equalization* (CLAHE);
4. Nuotraukos dydžio keitimas;
5. Apkarpymas į kvadrato formą;
6. Rentgeno nuotraukų pasukimas;
7. Rentgeno nuotraukų apkarpymas;
8. Rentgeno nuotraukų mastelio keitimas;
9. HAAR bangos transformacija;
10. Rentgeno nuotraukos šviesumo, kontrasto, aštrumo (angl. *sharpness*) ir spalvos keitimas;
11. SIFT metodo pritaikymas.

Pagrindinė sistemos dalis yra giliojo mokymo modeliai atrinkti atlikus literatūros analizę. Vienas pagrindinių giliojo mokymosi metodo pritaikymo apribojimų yra didelio duomenų rinkinio surinkimas siekiant apmokyti CNN. Tai vienas didžiausių giliojo mokymosi metodo skirtumų palyginus su tradiciniais mašininio mokymosi metodais. Reikia tūkstančių nuotraukų, kad būtų galima giliojo mokymosi modeliui suprasti pasikartojančius šablonus nuotraukose. Nepakankamas mokymosi duomenų kiekis yra neišvengiama problema. Tai atsitinka srityse, kuriose duomenys yra brangūs, reikalauja daug laiko apdoroti. Šiose situacijose žinių perkėlimas (angl. *transfer learning*) yra tinkamas būdas apmokyti CNN [65]. Šiame darbe siekiant išvengti mažų skaičiavimo galimybių apribojimą pasirinkta apmokant tik paskutinius modelių sluoksnius su turimu duomenų rinkiniu. Modelių tikslumas yra matuojamas naudojant šiuos parametrus remiantis literatūroje esančiu panaudojimu [1, 9, 33, 53, 55]:

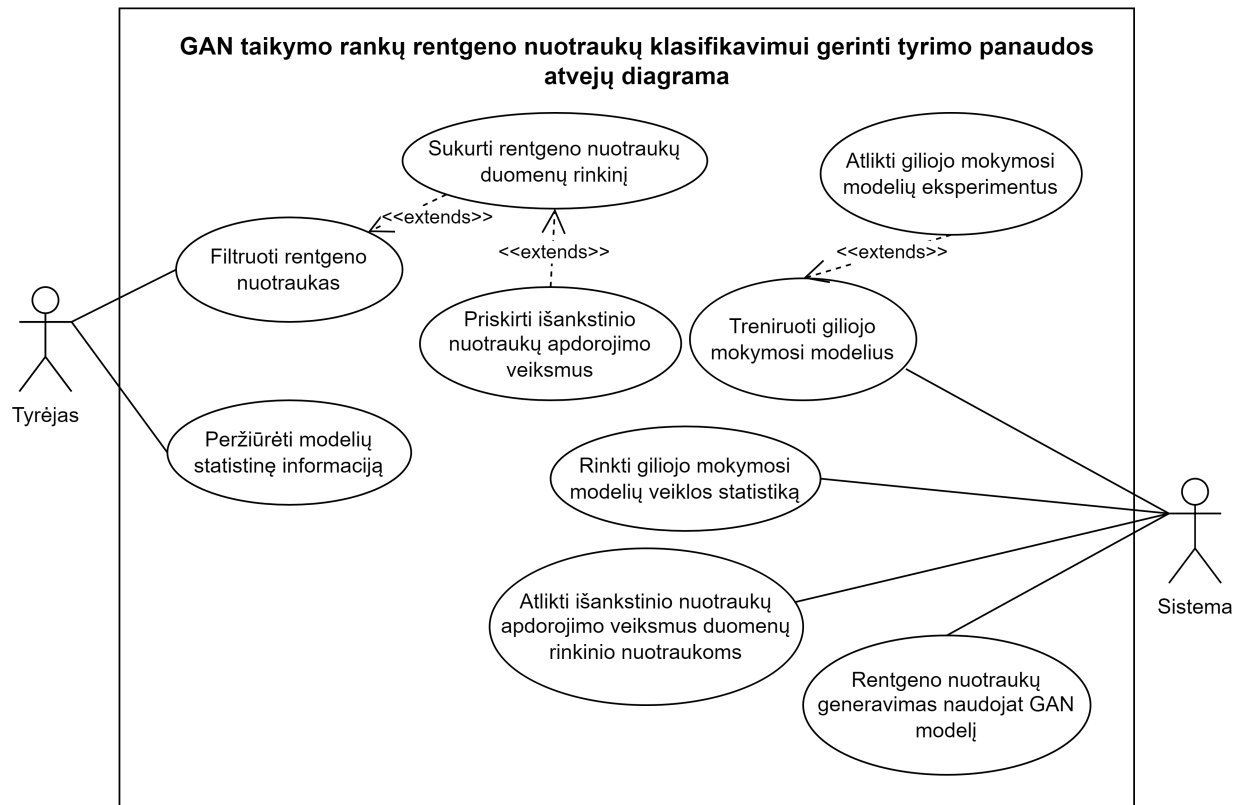
- Kappa – statistika naudojama vertinant kategorinių kokybinių elementų patikimumą tarp vertintojų;
- Jautrumas (angl. *sensitivity*);
- Specifiškumas (angl. *specificity*);
- AUC (angl. *area under the receiver operating characteristic curve*) [66].

Kadangi yra naudojami skirtingi giliojo mokymosi modeliai ir naudojami globalūs hiperparametrai ir aktyvavimo funkcijos (angl. *activation functions*) skiriasi buvo nuspręsta šių parametų keitimą iškelti į vidinę sąsają. Giliojo mokymosi modelių parametų saugojimas duomenų bazėje suteikia galimybę tyrėjui lengviau atlikti eksperimentus netik su skirtingais duomenų rinkiniais bet ir su skirtingais globaliais parametrais. Naudoti parametrai ir duomenų rinkinys visada bus išsaugoti su testavimo rezultatais siekiant palyginti skirtingų modelių versijų tikslumą.

2.5. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo sprendimo projektavimas

2.5.1. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo panaudos atvejų diagramos

GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo panaudos atvejų diagrama pateikta paveiksle 2.1. Atvaizduojamos pagrindinės programos sistemos ir tyrėjo atvejų diagramos.



2.1 pav. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo panaudos atvejų diagrama

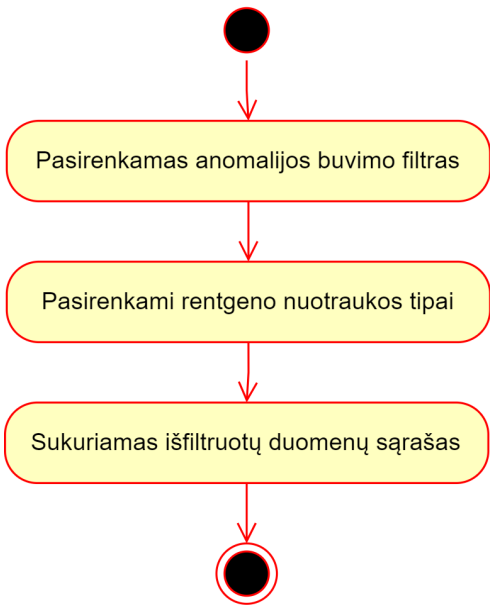
Tyrėjas – sistemos naudotojas galintis matyti visus rentgeno nuotraukas atlikti modelių globalių parametrų keitimą ir vykdyti eksperimentus su skirtingais giliojo mokymosi modeliais ir duomenų rinkiniais.

Sistema – vykdo giliojo mokymosi modelių treniravimą, generuoja sintetines rentgeno nuotraukas ir išsaugo statistinę informaciją apie modelių klasifikacijos tikslumą.

2.5.2. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrėjo panaudos atvejai

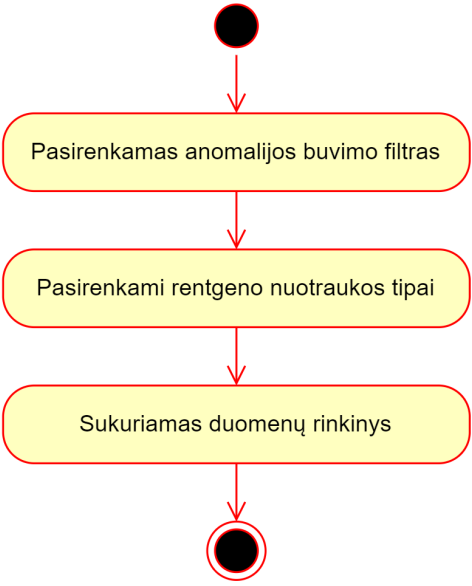
2.1 lentelė. Rentgeno nuotraukų filtravimo panaudos atvejais

Identifikatorius	1
Pavadinimas	Rentgeno nuotraukų filtravimas.
Aprašymas	Aktorius gali filtruoti rentgeno nuotraukas pagal pasirinktus parametrus.

Aktoriai	Tyrėjas.
Pradinės sąlygos	Yra įkelta rentgeno nuotraukų.
Pagrindiniai žingsniai	Aktorius šoniniame meniu pasirenka punktą <i>Datasets</i> kuris atveria naują puslapį su rentgeno nuotraukų tinkleliu. Pasirenka filtravimo galimybes (pavadinimą, anomalijos buvimą, nuotraukos tipą ir įkėlimo laiką).
Alternatyvūs žingsniai	Nėra.
Išskirtinės situacijos	Įvykus klaidai atliekant rentgeno nuotraukų filtravimą klaidos pranešimas yra įrašomas į sistemos veiklos žurnalą.
Galutinės sąlygos	Aktorius gali peržiūrėti išfiltruotas rentgeno nuotraukas.
	 <pre> graph TD Start(()) --> Step1(Pasirenkamas anomalijos buvimo filtras) Step1 --> Step2(Pasirenkami rentgeno nuotraukos tipai) Step2 --> Step3(Sukuriamas išfiltruotų duomenų sąrašas) Step3 --> End((())) </pre>

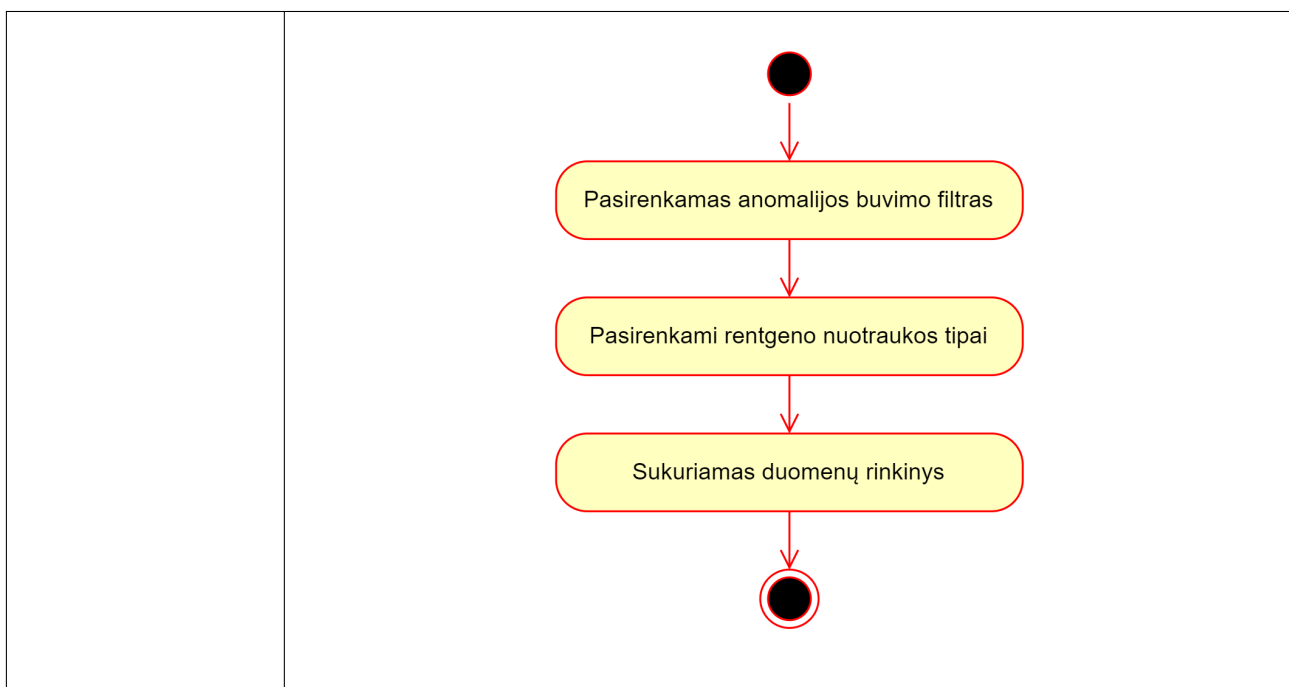
2.2 lentelė. Rentgeno nuotraukų duomenų rinkinių kūrimo panaudos atvejis

Identifikatorius	2
Pavadinimas	Rentgeno nuotraukų duomenų rinkinių kūrimas.
Aprašymas	Aktorius gali kurti rentgeno nuotraukų duomenų rinkinius su išfiltruotais duomenimis.
Aktoriai	Tyrėjas.
Pradinės sąlygos	Yra įkelta rentgeno nuotraukų.
Pagrindiniai žingsniai	Aktorius šoniniame meniu pasirenka punktą „Datasets“ kuris atveria naują puslapį su rentgeno nuotraukų tinkleliu. Atlikus duomenų filtravimą ir paspaudus mygtuką <i>Create dataset</i> sukuriamas duomenų rinkinys.
Alternatyvūs žingsniai	Nėra.

Išskirtinės situacijos	Įvykus klaidai sukuriant rentgeno nuotraukų duomenų rinkinį klaidos pranešimas yra įrašomas į sistemos veiklos žurnalą.
Galutinės sąlygos	Aktorius sukuria duomenų rinkinį su išfiltruotomis rentgeno nuotraukomis.
	 <pre> graph TD Start(()) --> Step1(Pasirenkamas anomalijos buvimo filtras) Step1 --> Step2(Pasirenkami rentgeno nuotraukos tipai) Step2 --> Step3(Sukuriamas duomenų rinkinys) Step3 --> End((())) </pre>

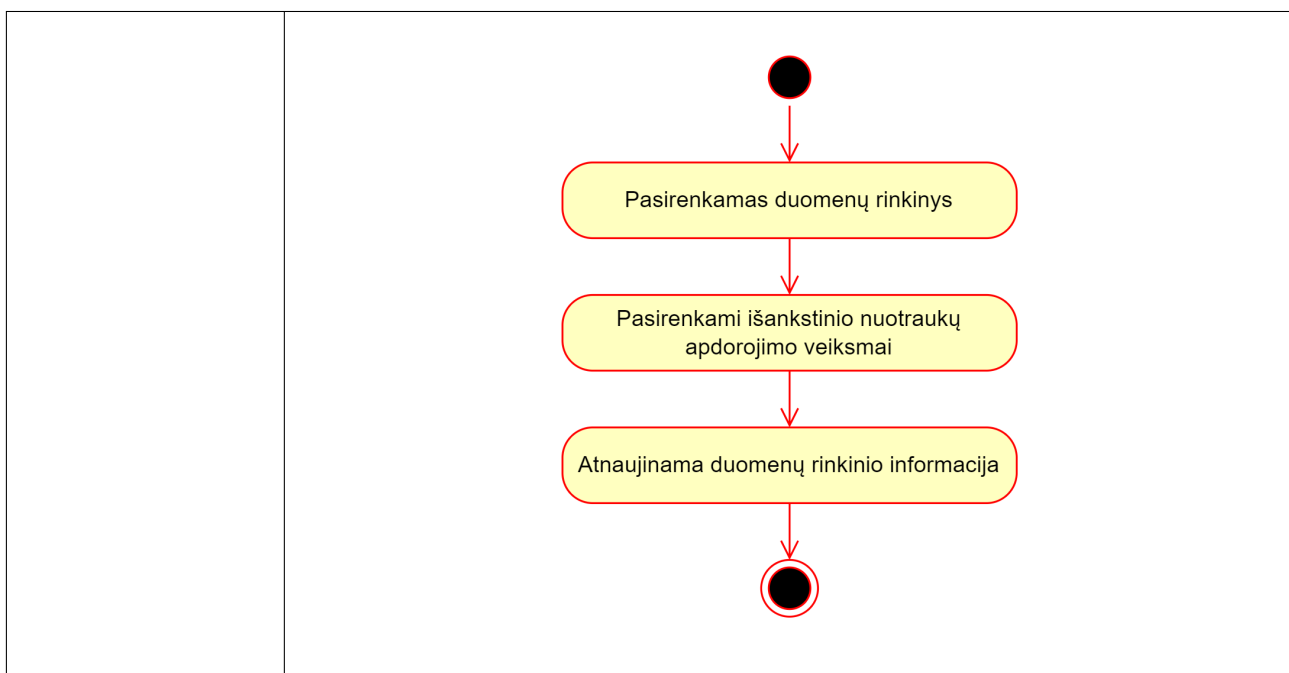
2.3 lentelė. Giliojo mokymosi modelių tikslumo kitimo statistikos peržiūros panaudos atvejis

Identifikatorius	3
Pavadinimas	Giliojo mokymosi modelių tikslumo kitimo statistikos peržiūra.
Aprašymas	Aktorius gali peržiūrėti modelio klasifikacijos tikslumo kitimą treniravimo metu ir testavimo eksperimentuose.
Aktoriai	Tyrėjas.
Pradinės sąlygos	<ul style="list-style-type: none"> • Įkelta rentgeno nuotraukų; • Pridėta giliojo mokymosi modelių; • Sukurti duomenų rinkiniai; • Atliktas modelių treniravimas ir eksperimentai.
Pagrindiniai žingsniai	Atliekamas eksperimentas su giliojo mokymosi modeliu. Gauti rezultatai yra išsaugomi duomenų bazėje.
Alternatyvūs žingsniai	Nėra.
Išskirtinės situacijos	Įvykus klaidai užkraunat statistinę informaciją klaidos pranešimas yra įrašomas į sistemos veiklos žurnalą.
Galutinės sąlygos	Įvykdyto giliojo mokymosi modelio rezultatai gali būti peržiūrėti.



2.4 lentelė. Išankstinio apdorojimo veiksmų pritaikymas duomenų rinkiniui panaudos atvejis

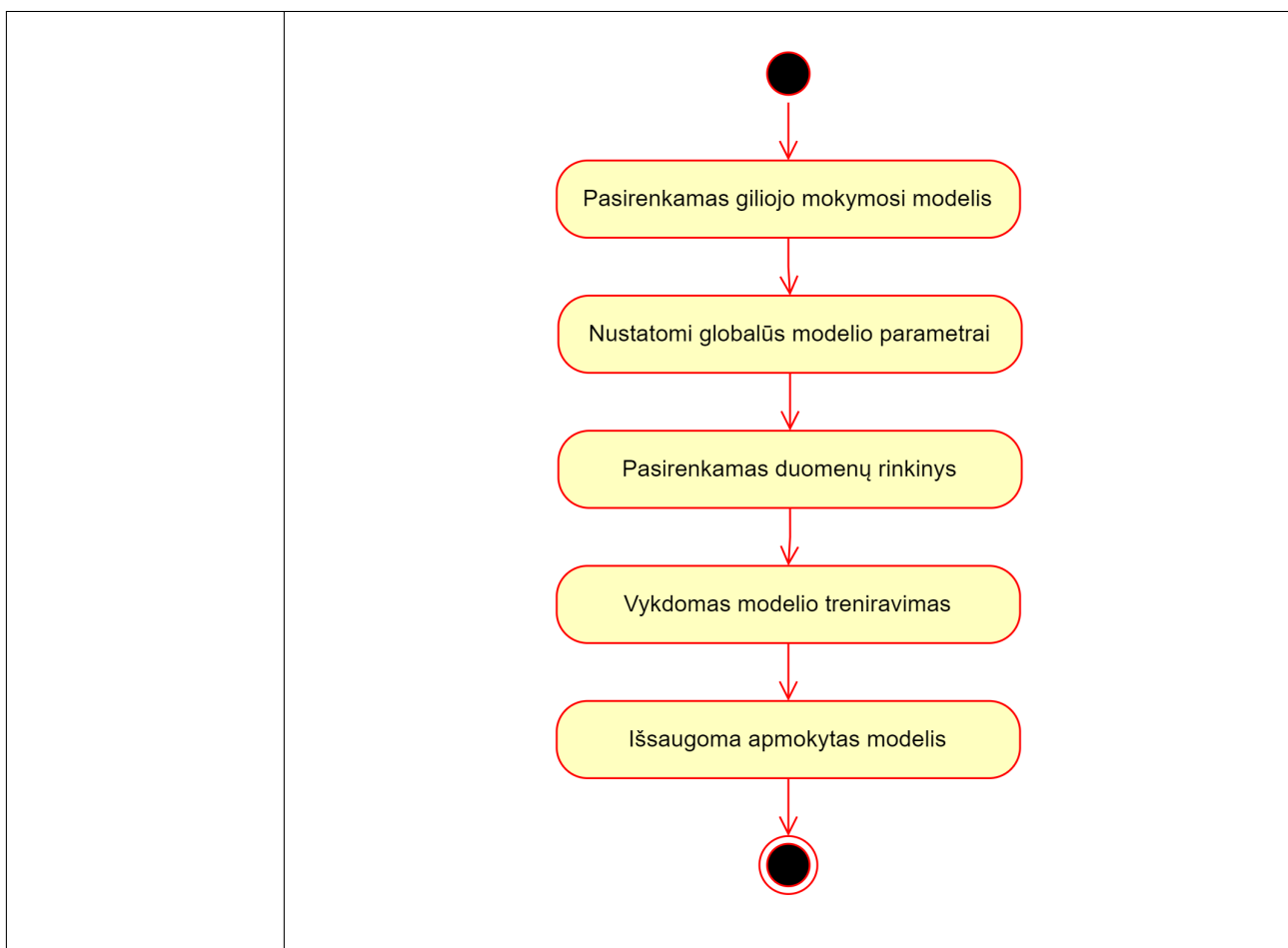
Identifikatorius	4
Pavadinimas	Išankstinio apdorojimo veiksmų pritaikymas duomenų rinkiniui.
Aprašymas	Aktorius turi galėti nustatyti pasirinktam duomenų rinkiniui papildomus išankstinius apdorojimo veiksmus.
Aktoriai	Tyrėjas.
Pradinės sąlygos	Yra įkelta rentgeno nuotraukų ir sukurta rentgeno nuotraukų duomenų rinkinys.
Pagrindiniai žingsniai	Aktorius nurodo duomenų rinkinį ir iš duotų išankstinio nuotraukų apdorojimo metodų nurodo bet vieną veiksmą. Nusiuntus užklausą yra atnaujinamas duomenų rinkinys.
Alternatyvūs žingsniai	Nėra.
Išskirtinės situacijos	Įvykus klaidai pritaikant išankstinio apdorojimo veiksmus klaidos pranešimas yra įrašomas į sistemos veiklos žurnalą.
Galutinės sąlygos	Atnaujinamas duomenų rinkinio išankstinio nuotraukų apdorojimo informacija.



2.5.3. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti sistemos panaudos atvejai

2.5 lentelė. Giliojo mokymosi modelių treniravimo su pasirinktu duomenų rinkiniu panaudos atvejais

Identifikatorius	5
Pavadinimas	Giliojo mokymosi modelių treniravimas su pasirinktu duomenų rinkiniu.
Aprašymas	Sistema atlieka giliojo mokymosi medelio treniravimą su pasirinktu duomenų rinkiniu.
Aktoriai	Sistema.
Pradinės sąlygos	<ul style="list-style-type: none"> Įkelta rentgeno nuotraukų; Pridėta giliojo mokymosi modelių; Sukurta rentgeno nuotraukų duomenų rinkinių.
Pagrindiniai žingsniai	Aktorius sukuria vykdymo aplinką ir paleidžia giliojo mokymosi modelį su nurodytais globaliniais parametrais ir priskirtu duomenų rinkiniu. Šis veiksmas yra atliekamas prieš kiekvieną eksperimentą užtikrinant, kad modelis turi izoliuotą treniravimo aplinką.
Alternatyvūs žingsniai	Nėra.
Išskirtinės situacijos	Įvykus klaidai eksperimento metu klaidos pranešimas yra įrašomas į sistemos veiklos žurnalą.
Galutinės sąlygos	Giliojo mokymosi modelis yra treniruojamas su pasirinktu duomenų rinkiniu.



2.6 lentelė. Giliojo mokymosi modelių eksperimentų su skirtingais duomenų rinkiniais testavimo panaudos atvejais

Identifikatorius	6
Pavadinimas	Giliojo mokymosi modelių eksperimentai su rentgeno nuotraukų duomenų rinkiniu.
Aprašymas	Aktorius atlieka eksperimentą su giliojo mokymosi modeliu atliekant rentgeno nuotraukų analizę ir išsaugo gautus rezultatus.
Aktoriai	Sistema.
Pradinės sąlygos	<ul style="list-style-type: none"> • Įkelta rentgeno nuotraukų; • Pridėta giliojo mokymosi modelių; • Sukurta rentgeno nuotraukų duomenų rinkinių; • Yra užregistruotų eksperimentų; • Nėra aktyvių eksperimentų.
Pagrindiniai žingsniai	Aktorius atlieka giliojo mokymosi modelio eksperimentą su priskirtu duomenų rinkiniu.

Alternatyvūs žingsniai	Nėra.
Išskirtinės situacijos	Įvykus klaidai eksperimento metu klaidos pranešimas yra įrašomas į sistemos veiklos žurnalą.
Galutinės sąlygos	Giliojo mokymosi modelio eksperimento įvykdymas ir rezultatų išsaugojimas duomenų bazėje.
	<pre> graph TD Start(()) --> A[Pasirenkamas giliojo mokymosi modelis] A --> B[Pasirenkamas duomenų rinkinys] B --> C[Vykdomas modelio testavimas] C --> D[Išsaugoma eksperimento statistinė informacija] D --> End((())) </pre>

2.7 lentelė. Giliojo mokymosi modelių veiksmingumo statistikos rinkimo panaudos atvejis

Identifikatorius	7
Pavadinimas	Giliojo mokymosi modelių veiksmingumo statistikos rinkimas.
Aprašymas	Sistema surenka informaciją apie modelių tikslumą.
Aktoriai	Sistema.
Pradinės sąlygos	<ul style="list-style-type: none"> • Įkelta rentgeno nuotraukų; • Pridėta giliojo mokymosi modelių; • Sukurta rentgeno nuotraukų duomenų rinkinių; • Atlikti eksperimentai su giliojo mokymosi modeliais.
Pagrindiniai žingsniai	Aktorius išsaugo informaciją apie giliojo mokymosi modelių treniravimą ir eksperimentų rezultatus.

Alternatyvūs žingsniai	Nėra.
Išskirtinės situacijos	Įvykus klaidai išsaugant duomenis klaidos pranešimas yra įrašomas į sistemos veiklos žurnalą.
Galutinės sąlygos	Aktorius išsaugo duomenis apie giliojo mokymosi modelių treniravimo ir eksperimentų rezultatus.
	<pre> graph TD Start(()) --> A(Vykdomas modelio treniravimas arba testavimas) A --> B(Kaupiama informacija (epocha, ciklas, globalūs parametrai, tikslumas, loss funkcijos reiksmė ir kt.)) B --> C(Išsaugoma kiekvienos epochos statistinė informacija) C --> End((())) </pre>

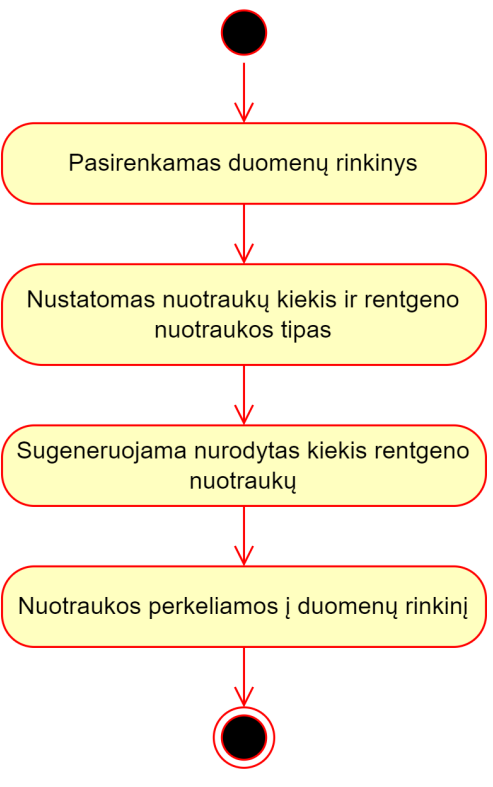
2.8 lentelė. Išankstinių nuotraukų apdorojimo veiksmų įvykdymas duomenų rinkinio aibėje panaudosis atvejis

Identifikatorius	8
Pavadinimas	Išankstinis nuotraukų apdorojimo veiksmų įvykdymas duomenų rinkinio aibėje.
Aprašymas	Aktorius atlieka išankstinį nuotraukų apdorojimą duomenų rinkinyje.
Aktoriai	Sistema.
Pradinės sąlygos	<ul style="list-style-type: none"> • Įkelta rentgeno nuotraukų; • Sukurta rentgeno nuotraukų duomenų rinkinių; • Duomenų rinkinys turi išankstinio apdorojimo veiksmų kurie dar nebuvo atlikti
Pagrindiniai žingsniai	Aktorius pagal nustatytas išankstinio nuotraukų generavimo veiksmų konfigūracijas sugeneruoja naujas rentgeno nuotraukas. Sukurtas naujas nuotraukas išsaugo į duomenų rinkinį.

Alternatyvūs žingsniai	Nėra.
Išskirtinės situacijos	Įvykus klaidai išankstinį nuotraukų apdorojimą arba nuotraukų saugojimą klaidos pranešimas yra įrašomas į sistemos veiklos žurnalą.
Galutinės sąlygos	Atnaujinamas duomenų rinkinys pritaikant nuotraukų išankstinio apdorojimo veiksmus.
	<pre> graph TD Start(()) --> Step1(Parenkamas duomenų rinkinys be sugeneruotų nuotraukų) Step1 --> Step2(Atliekamas nuotraukų išankstinis apdorojimas) Step2 --> Step3(Nuotraukos išsaugomos diskinėje atmintyje) Step3 --> End((())) </pre>

2.9 lentelė. Rentgeno nuotraukų generavimas naudojant GAN modelius panaudos atvejis

Identifikatorius	9
Pavadinimas	Rentgeno nuotraukų generavimas naudojant GAN modelį.
Aprašymas	Sistema išplečia rentgeno nuotraukų pridėdama nurodytą kiekį nuotraukų į duomenų rinkinį, kurios sugeneruotos GAN modelio.
Aktoriai	Sistema.
Pradinės sąlygos	<ul style="list-style-type: none"> • Sukurtas GAN modelis; • Įkelta rentgeno nuotraukų; • Sukurtas duomenų rinkinys.
Pagrindiniai žingsniai	Sistema pridėdama nurodytą kiekį rentgeno nuotraukų į duomenų rinkinį.
Alternatyvūs žingsniai	Nėra.
Išskirtinės situacijos	Įvykus klaidai plečiant duomenų rinkinį klaidos pranešimas yra įrašomas į sistemos veiklos žurnalą.

Galutinės sąlygos	Duomenų rinkinys yra papildomas sugeneruotomis rentgeno nuotraukomis.
	 <pre>graph TD; Start(()) --> Step1(Pasirenkamas duomenų rinkinys); Step1 --> Step2(Nustatomas nuotraukų kiekis ir rentgeno nuotraukos tipas); Step2 --> Step3(Sugeneruojama nurodytas kiekis rentgeno nuotraukų); Step3 --> Step4(Nuotraukos perkeliamos į duomenų rinkinį); Step4 --> End((()))</pre> <p>The flowchart illustrates a four-step process for adding X-ray images to a dataset. It begins with a solid black circle, followed by four yellow rounded rectangular boxes with red borders, each containing a step description. The steps are: 1. 'Pasirenkamas duomenų rinkinys' (Dataset is selected), 2. 'Nustatomas nuotraukų kiekis ir rentgeno nuotraukos tipas' (Number of images and X-ray image type are determined), 3. 'Sugeneruojama nurodytas kiekis rentgeno nuotraukų' (The specified number of X-ray images is generated), and 4. 'Nuotraukos perkeliamos į duomenų rinkinį' (Images are moved to the dataset). The process ends with a black circle with a red outline.</p>

2.6. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo sprendimo realizacija

Po atliktos literatūros analizės apie giliojo mokymosi modelius, karkasus ir išanalizavus duomenų rinkinius ir jų apdorojimo galimybes buvo projektuojamas eksperimentų realizavimas ir duomenų rinkinių integravimas į sukurtų giliojo mokymosi modelių treniravimo procesą.

Eksperimentų atlikimas susideda iš trijų pagrindinių dalių:

- Duomenų rinkinių sukūrimas – atliekant eksperimentus pagrindinis tikslas ištirti duomenų rinkinių pokyčio įtaką modelių tikslumui nustatyti anomaliją rentgeno nuotraukoje. Todėl didelis dėmesys yra skiriamas sukurti skirtingus duomenų rinkinius su skirtingu išankstiniu duomenų apdorojimu ir praplėtimu.
- Giliojo mokymosi modeliai – tyrimo tikslas ištirti giliojo mokymosi modelių tikslumo priklausomybę nuo duomenų rinkinio. Todėl tyrimui buvo panaudoti naujausi giliojo mokymosi modeliai sprendžiantys klasifikavimo uždavinį.
- GAN modelių pritaikymas – tyrimo metu siekiama sukurti ir nustatyti ar GAN modelių pritaikymas išplečiant duomenų rinkinį sugeneruotomis nuotraukomis gali išspręsti mažo duomenų rinkinio problemą.

Eksperimentų atlikimas susideda iš trijų dalių kurios veikia vieningai. Dėl šios priežasties atliekant realizaciją siekiant suvienodinti giliojo mokymosi modelių eksperimentų sąlygas buvo realizuota:

- Vidinė sąsaja – sąsaja eksperimentų metu yra atsakinga už modelių treniravimosi paleidimą, veikimo stebėjimą ir statistinės informacijos rinkimą.
- Python paketas – siekiant suvienodinti visų giliojo mokymosi modelių veikimą eksperimentų metu buvo sukurtas ir naudojamas Python paketas. Šis paketas suteikia abstrakciją Tensorflow ir Pytorch giliojo mokymosi karkasams. Paketas pagal nurodytus parametrus (modelį, įrenginį (CPU arba GPU), epochų skaičių, treniravimo, validavimo ir testavimo nuotraukų krovėją (angl. *dataloader*), optimizacijos funkciją, planuotojo (angl. *scheduler*) funkciją, modelio įvesties formuotoją, *loss* funkciją ir tikslumo (angl. *accuracy*) funkciją atlieka pilną eksperimento vykdymo ciklą. Šio paketo naudojimas užtikrina, kad modelių treniravimas ir testavimas yra atliekamas vienodose sąlygose.

2.6.1. Giliojo mokymosi metodų realizacija

Giliojo mokymosi modelių realizavimas atliekant eksperimentus iškelia reikalavimą turėti skirtingas vykdymo aplinkas skirtingiems giliojo mokymosi modeliams. Dėl šios priežasties realizacijos metu yra naudojamas Docker konteinerizavimo įrankis. Šio įrankio pagalba realizuojant giliojo mokymosi modelius yra galima panaudoti CPU ir GPU skaičiavimų galimybes. Realizavimo metu kūrimas buvo atliekamas naudojant CPU paremtus Docker atvaizdus (angl. *image*), kurie suteikia išsamius klaidos pranešimus leidžiančius lengviau diagnozuoti klaidas. Modelių mokymui buvo naudotas GPU sąsają turintis Docker atvaizdas paremtas NVIDIA Container Toolkit. Toks Docker vaizdų panaudojimas izoliuoja giliojo mokymosi modelius nuo vykdymo aplinkos ir leidžia atkartoti eksperimentų sąlygas.

Šiame tyrime siekiant sužinoti duomenų rinkinio pakeitimų daromą įtaką giliojo mokymosi modeliams buvo nuspręsta pasirinkti skirtingų architektūrų modelius. Pagrindinis modelis, kuris buvo pritaikytas MURA duomenų rinkiniui DenseNet 169 [29] su originale nuostolio funkcija ir kitais parametrais nurodytais straipsnyje [67] yra naudojamas kaip atskaitos taškas vertinant modelių tikslumą. Eksperimentų metu realizuoti giliojo mokymosi modeliai:

1. DenseNet 169;
2. ResNet 34, 101 ir 152 versijos;
3. Inception v3;
4. MobileNet v2;
5. EfficientNet B1 versija;
6. ResNext 101.

Atliekant giliojo mokymosi modelių realizacija panaudotas Pytorch karkasas su kuriuo sukurti devyni modeliai, kurie yra naudojami vykdant eksperimentus. Modelių veikimą buvo siekiama suvienodinti, naudojant vienodas funkcijas ir hiperparametrus. Modelių treniravimo naudotos funkcijos ir jų parametrai:

1. Modelis – kiekvienas modelis individualiai buvo pritaikytas atlikti dviejų klasių klasifikavimą pridodant papildomą pilnai sujungtą sluoksnį arba sigmoidinę funkciją kad galutinis rezultatas būtų binarinis;
2. Įvesties formuotojai – tai funkcijos pritaikančios paduodamus į modelis duomenis modeliui priimtinais. Tokiu būdu buvo išvengta kurti skirtingus duomenų rinkinio krovėjus;
3. Treniravimo ir validavimo nuostolio funkcijos - naudota originali nuostolio funkcija nurodyta straipsnio autorių pristačiusių MURA duomenų rinkinį [67]. Ši funkcija buvo pasirinkta siekiant turėti galimybę palyginti rezultatus su kitais tyrimais ir izoliuoti nuostolio funkcijos poveikį;
4. Tikslumo vertinimas – visuose modeliuose surinkti informaciją apie modelių tikslumą buvo naudota *sklearn* bibliotekos *precision_score*, *recall_score*, *f1_score*, *roc_auc_score* ir *cohen_kappa_score* funkcijos;
5. Optimizavimo funkcija – pasirinkta Adam funkcija su svorio nykimu (angl. *weight decay*) $1e^{-5}$, mokymosi žingsniu (0.0001) ir mokymosi žingsniu nykimu 0.5;
6. Planuotojo (angl. *scheduler*) funkcija - naudota Pytorch bibliotekos *ReduceLROnPlateau* funkcija minimaliame režime (min) su kantrumu (angl. *patience*) (1).

Modelių kūrimo procese buvo stengtasi juos suvienodinti, bet dėl vidinės modelių struktūros didžiausi pakeitimai tarp modelių turėjo būti atlikti įvesties formuotojuose ir pačių modelių pritaikyme atlikti klasifikaciją.

2.6.2. GAN modelių realizacija

GAN modelių panaudojimas siekiant išplėsti duomenų rinkinį yra pritaikytas kitose medicinos srityse [11]. Todėl šios architektūros pritaikymas išspręsti mažą MURA duomenų rinkinio nuotraukų pasiskirstymą yra daug žadanti hipotezė. Šiame tyrime buvo realizuoti keturi GAN modeliai:

- Paprastas GAN – tai modelis kuris yra treniruojamas kartu su diskriminatoriumi. Šis modelis treniravimosi metu stengiasi sugeneruoti rentgeno nuotraukas, kurių diskriminatoriaus modelis nesugebėtų atskirti nuo tikrų;
- Sąlyginis GAN – tai modelis veikiantis kaip paprastas GAN tik skirtumas, kad jo veikimas paremtas papildoma informacija apie generuojamą rentgeno nuotrauką. Šis GAN modelis pagal nurodytus parametrus gali sukurti nurodytos kūno dalies rentgeno nuotrauką su arba be anomalijos;
- PixelCNN – tai GAN modelio šaka kuri gali pridėti arba pašalinti anomaliją iš rentgeno nuotraukos taip padidinant duomenų rinkinio dydį. Šios architektūros trūkumas, kad reikia dviejų identiškų nuotraukų su ir be anomalijos norint apmokyti generuoti naujas nuotraukas.
- CycleGAN – tai GAN modelio šaka kuri gali generuoti rentgeno nuotraukas be apribojimų. Šios architektūros pagrindinis požymis, kad ji reikalauja papildomos nuotraukos ir papildomai kitos informacijos apie generuojamą nuotrauką.

GAN modelių realizacija reikalauja daug skaičiavimo pajėgumų ir laiko. GAN modelio treniravimas užtrunka 200 kartų ilgiau nei giliojo mokymosi klasifikatoriaus modelio. Todėl šių modelių pritaikymas nuotraukų generavimui yra sudėtingiausia tyrimo dalis, nes pritaikymas ir modelių treniravimas labai sudėtingas dėl ilgo modelių treniravimo laiko, duomenų rinkinio paruošimo, didelės laiko sąnaudos vertinant modelių tikslumą ir sugeneruotų nuotraukų įtraukimo į modelių mokymosi procesą.

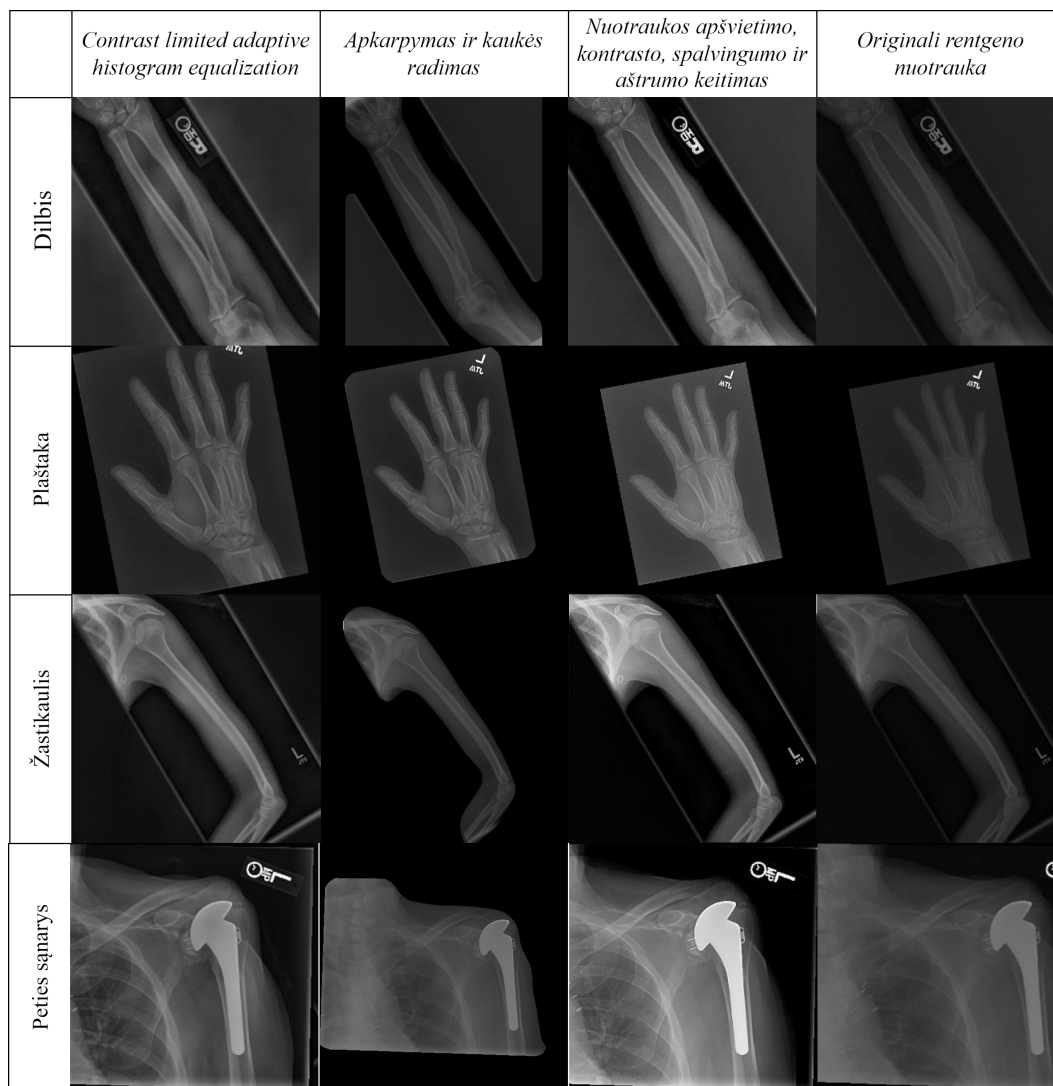
2.6.3. Duomenų rinkinio apdorojimo realizacija

MURA duomenų rinkinys yra surinktas iš daug skirtingų šaltinių, todėl dėl šios priežasties reikalingas duomenų išankstinis apdorojimas, siekiant pašalinti artefaktus ir suvienodinti nuotraukų dydį, formatą ir kt.. Atliekant eksperimentus su skirtingais giliojo mokymosi modeliais yra tiriama duomenų apdorojimo įtaka tikslumui. Kadangi eksperimentai yra pasikartojantys buvo realizuotas duomenų apdorojimo Python paketas. Šis paketas atlieka duomenų rinkinio išankstinį apdorojimą pagal konfigūracijas. Nuotraukų apdorojimo rezultatai pateikti 2.2 lentelėje.

Nuotraukų apdorojimas yra labai svarbus siekiant suvienodinti nuotraukų dydį, formatą, šviesumą ir kt. stengiantis pašalinti nuotraukoje esančius artefaktus ir išryškinti kaulų sritį. Didžiausias privalumas yra rankos nuotraukų apdorojime nes panaudojus kontūrų aptikimą galima rasti kvadrato formos objektą, kuris gali būti iškirptas iš nuotraukos taip pašalinant foną ir padidinant nuotraukos plotą. Kiti metodai yra skirti išryškinti nuotraukos savybėms ir kaulų kontūrams. Vizualiai stebint nuotraukas galima pastebėti, kad rentgenogramose kaulų kraštai yra aiškesni ir išsiskiria iš aplinkos labiau. Pritaikant šviesumo keitimą nuotraukoje ypač išryškėja esantys objektai, kurie signalizuoja apie nuotraukoje esančią anomaliją.

2.7. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo sprendimo testavimas

Siekiant užtikrinti iškeltus sistemos reikalavimus buvo paruoštas tyrėjo panaudos atvejais paremtas testavimo planas, kuris pateiktas 2.10 lentelėje.



2.2 pav. Duomenų rinkinio apdorojimo realizacijos operacijų rezultatai

2.10 lentelė. Tyrėjo panaudos atvejų testai

Panaudos atvejis	Testo aprašymas	Rezultatas
PA – 1	Tyrėjas gali filtruoti rentgeno nuotraukas pagal pavadinimą, anomalijos buvimą, nuotraukos tipą ir įkėlimo laiką.	Sėkmingas
PA – 2	Tyrėjas gali sukurti duomenų rinkinį nurodant nuotraukų tipą, anomalijos buvimą ir nustatant išankstinio apdorojimo veiksmus.	Sėkmingas
PA – 3	Tyrėjas gali peržiūrėti modelio klasifikacijos tikslumo kitimą eksperimento vykimo metu ir po eksperimento.	Sėkmingas
PA – 4	Tyrėjas gali sukurti duomenų rinkinį su išankstinio apdorojimo veiksmiais.	Sėkmingas

Siekiant užtikrinti iškeltus sistemos reikalavimus paruoštas sistemos aktoiaus testavimo planas paremtas panaudos atvejais, kuris pateiktas 2.11 lentelėje.

2.11 lentelė. Sistemos panaudos atvejų testai

Panaudos atvejis	Testo aprašymas	Rezultatas
PA – 5	Sistema įvykdo giliojo mokymosi modelio treniravimą su priskirtu duomenų rinkiniu.	Sėkmingas
PA – 6	Sistema įvykdo eksperimentus su giliojo mokymosi modeliais atliekant rentgeno nuotraukų analizę ir išsaugo gautus rezultatus.	Sėkmingas
PA – 7	Sistema išsaugo informaciją apie giliojo mokymosi modelių treniravimo ir eksperimentų rezultatus.	Sėkmingas
PA – 8	Duomenų rinkinys su išankstinio apdorojimo veiksmu turi nuotraukas, kurioms atlikti išankstinio apdorojimo veiksmu.	Sėkmingas
PA – 9	Duomenų rinkinys yra išplėstas sugeneruotomis rentgeno nuotraukomis.	Sėkmingas

Sukurta sistemos realizacija ištestuota iš tyrėjo ir sistemos perspektyvos pasinaudojus panaudos atvejais. Atliktas testavimas leido užtikrinti, kad realizuoti tyrimo įrankiai veikia kaip numatyta pagal iškeltus reikalavimus. Testavimo rezultatai sėkmingi ir visos sistemos dalys veikia korektiškai.

2.8. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo projektavimo išvados

Sistemos projektavimo planas įgyvendina visus iškeltus funkcinius, nefuncinius, programinės įrangos ir testavimo reikalavimus. Atliekant sistemos projektavimą buvo iškelti 4 tyrėjo ir 5 sistemos panaudos atvejai. Kuriuos siekiant įgyvendinti buvo sukurta:

1. Devyni giliojo mokymosi modeliai;
2. Keturi GAN modeliai;
3. Sukurta vidinė sąsaja;
4. Sukurtas Python paketas kontroliuojantis giliojo mokymosi modelių veikimą;
5. Sukurtas Python paketas generuoti skirtingus duomenų rinkinius.

Atliekant sistemos projektavimą buvo nuspręsta skirtingą funkcionalumą perkelti į Python paketus siekiant sumažinti sudėtingumą ir palengvinti eksperimentų vykdymą. Tyrimo metu tai suteikė didesnę lankstumą integruojant naujus giliojo mokymosi modelius ir pridėdant naujus išankstinio nuotraukų apdorojimo metodus. Giliojo mokymosi modelių perkėlimas į konteinerius buvo labai palankus, nes tai leidžia lengvai atkartoti eksperimentus išlaikant pradinės sąlygas. Konteineriai netik apsaugo eksperimentus nuo modelių pakeitimų, bet ir nuo giliojo mokymosi karkasų versijų pakeitimų, kurie gali sukelti klaidas, kurios neleidžia pakartoti eksperimentų.

3. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo eksperimentai ir rezultatai

Šiame skyriuje aprašomi eksperimentai atlikti testuojant skirtingus CNN konvoliucinius tinklus su MURA rentgeno nuotraukų duomenų rinkiniu. Taip pat aprašomi giliojo mokymosi metodų globalūs parametrai, optimizacijos funkcijos ir duomenų rinkiniai, kurie buvo naudoti atliekant eksperimentus ir jų įtaka giliojo mokymosi modelių tikslumui.

3.1. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo eksperimentų planavimas

Giliojo mokymosi modelių eksperimentai atlikti asmeniniame kompiuteryje. Siekiant pagreitinti modelių mokymąsi įdiegta NVIDIA vaizdo plokštės įrankiai ir Kubernetes su Docker konteinerių vaizdais. Kiekvienas modelis turi savo Docker atvaizdą, kuris yra naudojamas modelio testavimui ir treniravimui. Dėl šios priežasties eksperimentai yra atkartojami ir modelis nėra paveikiamas išorinių veiksnių. Eksperimentų vykdymo aplinkos parametrai:

1. Procesorius – Intel® Core™ i7-7700;
2. Procesoriaus dažnis – 2.5 GHz;
3. Procesoriaus stiprinimo dažnis – 3.5 GHz;
4. Diskretus grafikos plokštės modelis – NVIDIA GeForce GTX 1050;
5. Diskreti grafikos plokštės atmintis – 4 GB GDDR5;
6. Operatyvinė atmintis – 16 GB DDR4;
7. SSD disko talpa – 256 GB.

Eksperimentų atlikimo metu giliojo mokymosi modelių treniravimas užima nuo vienos dienos iki 5 dienų. Tai sukelia problemas vykdyti skirtingus eksperimentus vienu metu, kadangi rezultatų gavimas užima daug laiko. Diskrečios grafikos plokštės 4 gigabitų GDDR5 atminties neužtenka sutalpinti giliojo mokymosi modelio su duomenų rinkinio nuotraukomis, kurios naudojamos su 224×224 dydžiu, todėl kai kurie eksperimentai atlikti naudojant AMD Ryzen™ 7 3700X procesorių su 32 gigabitaish atminties. Šių modelių treniravimo laikai nėra lyginami laiko atžvilgiu, bet į pasiekto tikslumo vertinimą yra įtraukiami.

3.2. GAN taikymo rankų rentgeno nuotraukų klasifikavimui gerinti tyrimo eksperimentų rezultatai

Pirmojoje darbo dalyje yra atliekami eksperimentai patikrinti išankstinio duomenų rinkinio apdorojimo daromą įtaka tikslumui. Duomenų rikinimo išankstiniame apdorojime buvo pritaikytos šios rentgeno nuotraukų kokybės gerinimo operacijos.

1. Rentgenogramos kaukės (angl. *mask*) radimas;
2. *Contrast Limited Adaptive Histogram Equalization*;
3. Šviesumo didinimas su reikšme – 1.2;
4. Spalvingumo didinimas su reikšme – 1.4;
5. Kontrasto keitimas su reikšme – 1.2;
6. Ryškumo didinimas su reikšme – 1.8;
7. Kaulų srities išdidinimas panaudojant OpenCv kontūrų nustatymą ir apkarpymą;

8. Išdidinimas nuotraukos į kvadrato formą;
9. Nuotraukos dydžio keitimas į 224×224 .

Pateiktos reikšmės buvo naudotos apdoroti galutinį duomenų rinkinį, kuris buvo naudojamas galutiniuose tyrimo eksperimentuose. Darbo metu buvo testuojamos skirtingos nuotraukų apdorojimo metodų kombinacijos, bet šis metodų eiliškumas ir reikšmės suteikė geriausius rezultatus.

3.3. Giliojo mokymosi modelių tikslumo priklausomumas nuo išankstinio duomenų rinkinio apdorojimo

Pirmoji tiriamojo darbo dalis susidėjo iš giliojo mokymosi modelių priklausomumo nuo duomenų rinkinio išankstinio apdorojimo tyrimo. Šie eksperimentai buvo suskirstyti į modelius, kurie buvo treniruoti su ImageNet duomenų rinkiniu ir be. Treniruojant giliojo mokymosi modelius buvo pritaikyti skirtingi rentgeno nuotraukų apdorojimo būdai, kurių rezultatai pateikti 3.1 grafike. Atliekant išankstinį apdorojimą buvo kuriami skirtingi duomenų rinkiniai ir atliekami eksperimentai.

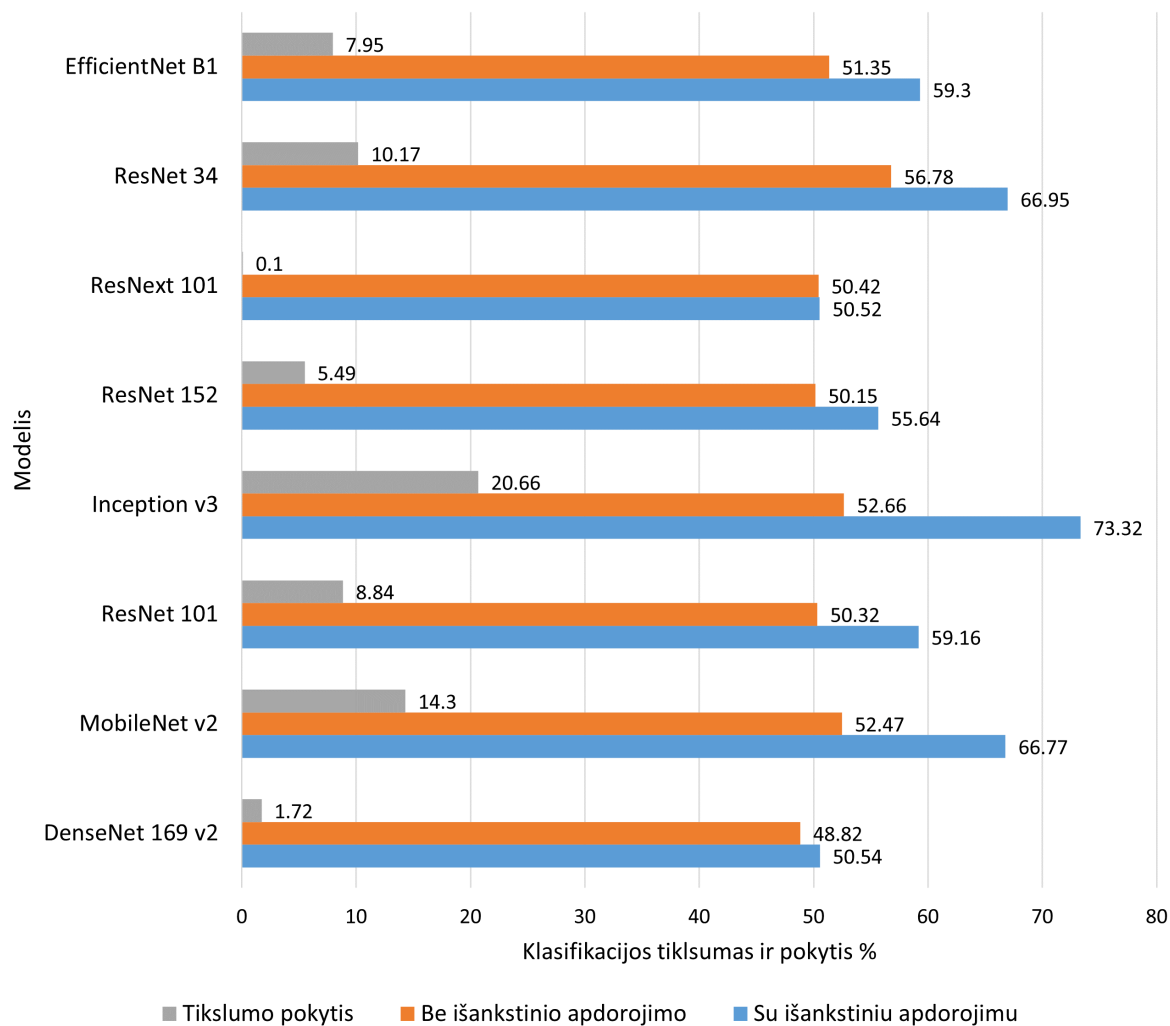
3.3.1. Giliojo mokymosi modelių tikslumo priklausomumas nuo išankstinio duomenų rinkinio apdorojimo

Šio eksperimento metu buvo naudojami giliojo mokymosi modeliai be išankstinio apmokymo ant kito duomenų rinkinio. Pateiktoje diagramoje galima matyti geriausia išankstinio apdorojimo metodų junginį ir jo poveikį giliojo mokymosi modelių efektyvumui.

Naudojant duomenų rinkinį su išankstiniu apdorojimu didžiausias giliojo mokymosi modelio tikslumo padidėjimas užfiksuotas Inception V3 20.66%, ResNet klasės modeliuose 0.1% iki 10.17% ir MobileNet v2 14.3%. Pateiktame 3.2 grafike galime pamatyti, kad Inception V3 giliojo mokymosi modelis pasižymėjo didžiausiu tikslumo pagerėjimu siekiančių nuo 52.66% iki 73.32%. Analizuojant rezultatus galima matyti, kad modeliai nesugeba aptikti anomalijų ir pasiekia vidutiniškai tik 50+% tikslumą, kuris reiškia, kad modeliai apmokyti duomenų rinkinyje be išankstinio apdorojimo nesugeba aptikti anomalijų. Tačiau panaudojus duomenų rinkinį su išankstiniu apdorojimu modelių tikslumas pakyla visuose modeliuose vidutiniškai per 8.6%.

Apžvelgiant giliojo mokymosi modelių rezultatus vertinant efektyvumą buvo stebimas netik klasifikacijos tikslumas %, bet ir kitos tikslumo vertinimo metrikos kaip kappa, preciziškumas, F1 ir AUC. Tyrime vertinant šias metrikas duomenų rinkinyje be išankstinio apdorojimo geriausias modelis buvo ResNet 34, kuris surinko aukščiausius rezultatus pagal klasifikacijos tikslumą 56.78%, kappa 0.22, F1 0.56 ir AUC 0.61. Šis modelis nusileido tik Inception V3 modeliui pasiekus preciziškumo 0.86 reikšmę ir aplenkus per 0.21. Eksperimentų rezultatai duomenų rinkinyje be išankstinio apdorojimo pateikti 3.1 lentelėje.

Duomenų rinkinyje su išankstiniu apdorojimu didžiausią efektyvumo padidėjimą pasiekė Inception V3 modelis, kurio klasifikacijos tikslumas padidėjo nuo 52.66% iki 73.32%. Šis efektyvumo padidėjimas leido šiam modeliui įgauti aukščiausią efektyvumą visose metrikose. Modelių tikslumo eksperimentų duomenų rinkinyje su išankstiniu apdorojimu rezultatai pateikti



3.1 pav. Giliojo mokymosi modelių klasifikacijos tikslumas alkūnės rentgeno nuotraukų duomenų rinkinyje su išankstiniu apdorojimu ir be išankstinio apdorojimo

3.1 lentelė. Giliojo mokymosi modelių klasifikacijos tikslumo rezultatai su alkūnės rentgeno nuotraukų duomenų rinkiniu be išankstinio apdorojimo

Modelis	Klasifikacijos tikslumas %	Kappa	Precizija	F1	AUC
MobileNet v2	52.47	0.1	0.73	0.22	0.55
ResNet 101	50.32	0	0.24	0.05	0.5
Inception v3	52.66	0.09	0.86	0.19	0.55
ResNet 152	50.15	-0.01	0.22	0.02	0.5
ResNext 101	50.42	0.01	0.29	0.02	0.5
ResNet 34	56.78	0.22	0.65	0.56	0.61
EfficientNet B1	51.35	0.01	0.53	0.21	0.51

3.5 lentelėje.

3.2 lentelė. Giliojo mokymosi modelių klasifikavimo tikslumo rezultatai su alkūnės rentgeno nuotraukų duomenų rinkiniu su išankstiniu apdorojimu

Modelis	Klasifikacijos tikslumas %	Kappa	Precizija	F1	AUC
MobileNet v2	66.77	0.33	0.73	0.61	0.67
ResNet 101	59.16	0.18	0.83	0.34	0.59
Inception v3	73.32	0.47	0.9	0.66	0.73
ResNet 152	55.64	0.11	0.77	0.26	0.56
ResNext 101	50.52	-0.01	0.12	0.07	0.49
ResNet 34	66.95	0.34	0.69	0.65	0.67
EfficientNet B1	59.3	0.18	0.67	0.46	0.59

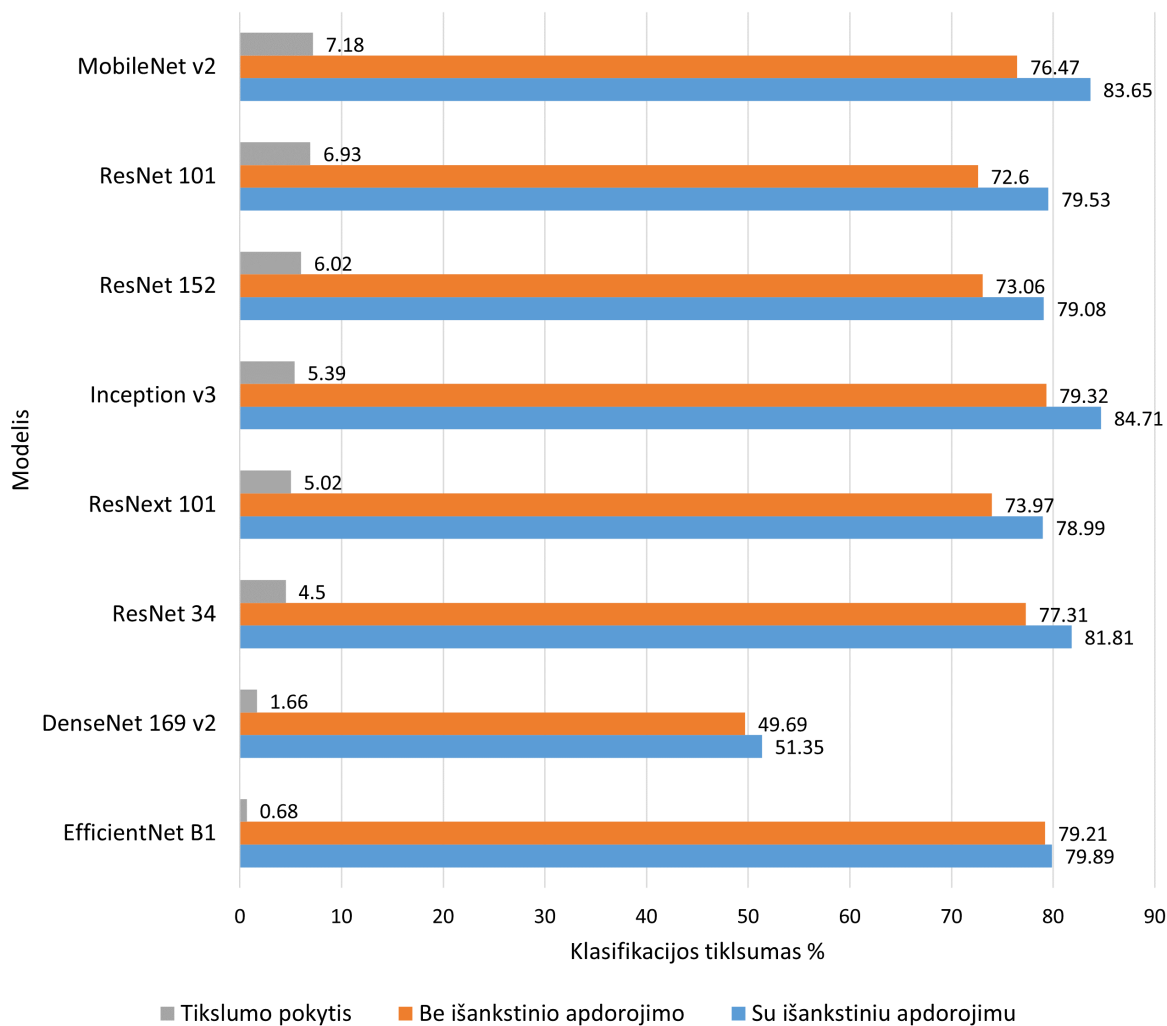
3.3.2. Giliojo mokymosi modelių, apmokyto su ImageNet duomenų rinkiniu, tikslumo priklausomumas nuo išankstinio duomenų rinkinio apdorojimo

Pateiktoje 3.2 diagramoje galima matyti geriausią išankstinio apdorojimo metodų junginį ir jo poveikį giliojo mokymosi modelių efektyvumui, kurie yra apmokyti su ImageNet.

Naudojant duomenų rinkinį su išankstiniu apdorojimu didžiausias giliojo mokymosi modeliuose apmokytuose su ImageNet duomenų rinkiniu tikslumo padidėjimas užfiksuotas MobileNet v2 7.18%, ResNet klasės modeliuose 4.5% iki 6.93% ir Inception v3 5.39%. Pateiktame 3.2 grafike galime pamatyti, kad ResNet šeimos giliojo mokymosi modeliai pasižymėjo didžiausiu tikslumo pagerėjimu siekiančių nuo 72.6% iki 81.81%. Analizuojant rezultatus galima pastebėti kad mažesnį gylį turintis ResNet 34 modelis pasiekia aukščiausią tikslumą be išankstinio duomenų rinkinio apdorojimo, bet gauna mažiausią tikslumo pagerėjimą su išankstinio duomenų rinkinio apdorojimu. Giliojo mokymosi modelis Inception V3 pasiekė aukščiausią preciziją be išankstinio apdorojimo 79.32% bet su 5.39% tikslumo pakilimu tapo geriausiu modeliu klasifikuojant alkūnės rentgeno nuotraukas.

Duomenų rinkinyje be išankstinio apdorojimo geriausias modelis buvo Inception V3, kuris surinko aukščiausius rezultatus pagal klasifikacijos tikslumą 79.32%, kappa 0.63 ir AUC 0.81. Šis modelis nusileido tik ResNet 152 modeliui pasiekus precizijos 0.97 reikšmę ir aplenkus per 0.06 ir EfficientNet modeliui F1 metrikoje pasiekus 0.79 reikšmę ir aplenkus per 0.01. DenseNet modelis nepasiekė didesnio nei 50% klasifikacijos tikslumo, todėl iš palyginimo lentelės pašalintas. Modelių tikslumo eksperimentų rezultatai duomenų rinkinyje be išankstinio apdorojimo pateikti 3.3 lentelėje.

Duomenų rinkinyje su išankstiniu apdorojimu didžiausią efektyvumo padidėjimą pasiekė MobileNet V2 modelis, kurio klasifikacijos tikslumas padidėjo nuo 76.47% iki 83.65%. Šis efektyvumo padidėjimas leido šiam modeliui įgauti aukščiausius rezultatus kappa (0.67), F1 (0.82) ir AUC (0.84) metrikose. Toks efektyvumo padidėjimas lėmė, kad MobileNet pralenkė Inception V3 modelį ir tapo geriausiu, bet dar nusileidžia ResNet 152 ir 101 precizijoje skirtumu per 0.04 ir klasifikacijos tikslume inception V3 modeliui skirtumu per 1.06%. Modelių tikslumo



3.2 pav. Giliojo mokymosi modelių, apmokyto su ImageNet duomenų rinkiniu, klasifikacijos tikslumas alkūnės rentgeno nuotraukų duomenų rinkinyje su išankstiniu apdorojimu ir be išankstinio apdorojimo

3.3 lentelė. Giliojo mokymosi modelių, apmokyto su ImageNet duomenų rinkiniu, tikslumo rezultatai su alkūnės rentgeno nuotraukų duomenų rinkiniu be išankstinio apdorojimo

Modelis	Klasifikacijos tikslumas %	Kappa	Precizija	F1	AUC
MobileNet v2	76.47	0.6	0.94	0.76	0.8
ResNet 101	72.6	0.49	0.96	0.67	0.75
Inception v3	79.32	0.63	0.93	0.78	0.81
ResNet 152	73.06	0.49	0.97	0.66	0.74
ResNext 101	73.97	0.53	0.96	0.7	0.76
ResNet 34	77.31	0.57	0.89	0.75	0.79
EfficientNet B1	79.21	0.6	0.82	0.79	0.8

eksperimentų duomenų rinkinyje su išankstiniu apdorojimu rezultatai pateikti 3.5 lentelėje.

3.4 lentelė. Giliojo mokymosi modelių, apmokytų su ImageNet duomenų rinkiniu, tikslumo rezultatai su alkūnės rentgeno nuotraukų duomenų rinkiniu su išankstiniu apdorojimu

Modelis	Klasifikacijos tikslumas %	Kappa	Precizija	F1	AUC
MobileNet v2	83.65	0.67	0.9	0.82	0.84
ResNet 101	79.53	0.59	0.91	0.76	0.79
Inception v3	84.71	0.64	0.92	0.79	0.82
ResNet 152	79.08	0.55	0.94	0.72	0.78
ResNext 101	78.99	0.58	0.94	0.74	0.79
ResNet 34	81.81	0.55	0.76	0.78	0.77
EfficientNet B1	79.89	0.6	0.76	0.81	0.8

Giliojo mokymosi modeliai naudodami duomenų rinkinį su išankstiniu apdorojimu pasiekia geresnius rezultatus apmokyti su ImageNet duomenų rinkiniu ir be. Apmokyti modeliai įgauna daug geresnius rezultatus, bet pasiekia mažesnę tikslumo pagerėjimą dėl duomenų rinkinio išankstinio apdorojimo. Tačiau modeliai be išankstinio apmokymo pasiekia klasifikacijos tikslumą vidutiniškai 51% be išankstinio apdorojimo duomenų rinkinyje ir su 60%, o modeliai su išankstiniu apmokymu pasiekia klasifikacijos tikslumą vidutiniškai 72% be išankstinio apdorojimo duomenų rinkinyje ir su 77%. Vertinant tyrimo rezultatus galima pasiekti maksimalų 26% klasifikacijos tikslumo pagerėjimą naudojant apmokytus modelius su ImageNet duomenų rinkiniu ir išankstinį duomenų rinkinio apdorojimą.

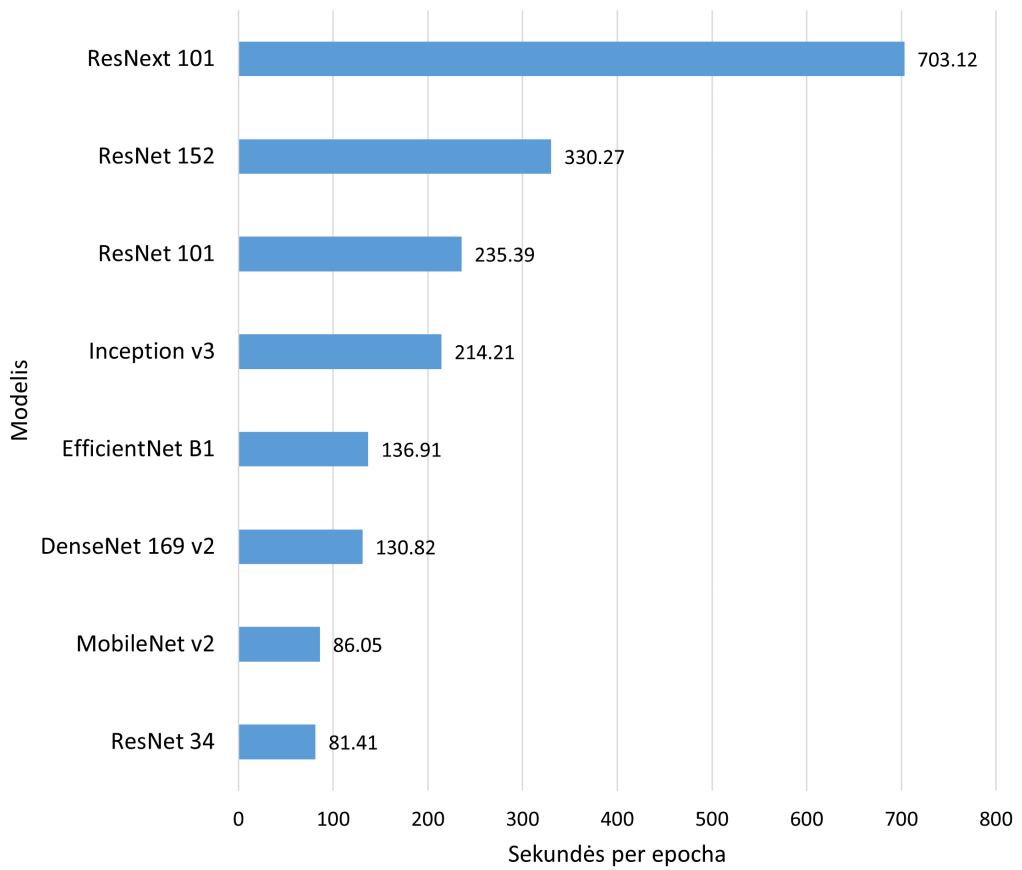
3.3.3. Giliojo mokymosi modelių treniravimo laikas

Tyrimo metu buvo stebimas ir laikas reikalingas ištrenuoti giliojo mokymosi modelius. Giliojo mokymosi modelių treniravimo laikai pateikti 3.3 paveiksle. Iš pateikto 3.4 paveikslo galime matyti, kad mokymosi laikas turi koreliaciją su klasifikacijos tikslumu. Greitesni modeliai atlieka daugiau epochų per 2 valandų mokymosi laiką ir pasiekia aukštesnę klasifikacijos tikslumą. DenseNet V2 modelis yra neatvaizduojamas grafike dėl per mažo klasifikacijos tikslumo.

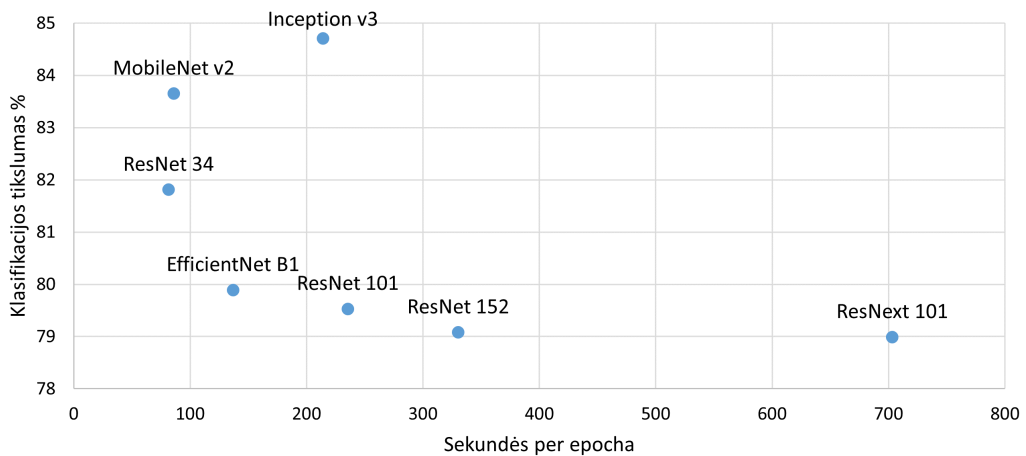
Pateiktame giliojo mokymosi modelių treniravimo laiko 3.3 grafike galima matyti, kad daugiausiai laiko užėmė ResNext 101 modelis užtrukęs net 703 sekundes vienai epochai. Vertinant modelių tikslumą galima pastebėti, kad modeliuose apmokytuose su ImageNet ir naudojant išankstinį apdorojimą turintį duomenų rinkinį ResNet 34 užtrukdamas 81 sekundę pasiekė 81.81% tikslumą ir pralenkė ResNext 101 2.81%. Tačiau vertinant kitas metrikas ResNext 101 pralenkė ResNet 34 Kappos metrikoje per 0.03 ir precizijoje per 0.18. Tai leidžia priimti išvadą, kad didelis modelio gylis ir parametrų skaičius suteikia tikslumo skirtingose metrikose, bet praranda bendrą pranašumą aptinkant anomalijas rentgeno nuotraukose.

3.4. Rankos rentgeno nuotraukų eskizų generavimo eksperimentai

MURA duomenų rinkinys neturi papildomos informacijos apie rentgeno nuotraukas, kurią būtų galima panaudoti pritaikant paveikslą į paveikslą generuojančius GAN modelius. Šiam GAN modelių tipui priklauso Pix2Pix ir CycleGAN modeliai. Tyrimo metu norint pritaikyti



3.3 pav. Giliojo mokymosi modelių mokymosi laikas per vieną epochą



3.4 pav. Giliojo mokymosi modelių, apmokytų su ImageNet duomenų rinkiniu, klasifikacijos tikslumas alkūnės rentgeno nuotraukų duomenų rinkinyje su išankstiniu apdorojimu ir mokymosi laiko per vieną epochą koreliacija

šiuos modelius buvo nuspręsta kurti rentgeno nuotraukų eskizus, kurie suteikia pakankamai informacijos apie kaulų išsidėstymą erdvėje ir formą. Eskizai gali būti naudojami GAN modelių treniravime remiantis [40] straipsnio autoriais.

Tyrimo metu siekiant sukurti eskizų nuotraukas buvo panaudoti trys kraštinių aptikimo metodai. Naudojant OpenCV bibliotekos kontūrų radimo algoritmą *findContours* su skirtingai slenksčio parametrais nepavyko rasti kraštinių rentgeno nuotraukose dėl mažo kontrasto ir nelygių kraštų. Šis algoritmas yra geras rasti tik lygius kraštus, todėl buvo naudojamas tik rentgeno nuotraukų apdorojimo procese išgaunant dėmesio sritį iškerpant kvadrato formą. Šis metodas gali būti pritaikytas atliekant eksperimentus su plaštakos nuotraukomis, kurių rezultatai yra pateikti 2.2.

Holistically-nested edge detection (HED) [68] sukurtas 2015 metais yra kraštų aptikimo metodas, sprendžiantis dvi svarbias ilgalaikės kraštų aptikimo problemas. Šis metodas pasižymi, kad atliekamas holistinis vaizdų treniravimas ir nustatymas, daugelio mastelio ir daugelio lygmenų savybių mokymasis. HED yra giliojo mokymosi modeliais paremtas sprendimas aptikti vaizde hierarchines reprezentacijas ir išsprendžia problemą kai yra sudėtingas kraštų ir objekto ribų aptikimo dviprasmiškumas. Šio modelio pritaikymas aptinkant kraštines suteikia gerus rezultatus, nes sukurti eskizai nėra tiesioginis rentgeno nuotraukos pikselių reprezentavimas.

Šio darbo metu buvo sukurtas ir kraštinių aptikimo algoritmas, kuris paremtas morfologinėmis operacijomis. Šio algoritmo metu pirmiausia buvo panaudota *Gaussian blur* funkcija sušvelninti kraštus. Funkcijos rezultatas naudojamas atliekant iteraciją per paveikslą pereinant per 12 pikselių kiekvienos iteracijos metu. Kiekvienoje iteracijoje panaudojamas vertikalus $[-2, -3, -2], [0, 0, 0], [2, 3, 2]$ ir horizontalus $[-2, 0, 2], [-3, 0, 3], [-2, 0, 2]$ filtras. Gautos reikšmės iš filtrų yra naudojamos apskaičiuoti kraštinės reikšmę $(V \cdot 2 + H \cdot 2) \cdot 0.5$, kur V vertikalaus ir H horizontalaus filtrų reikšmės. Rezultatai yra kaupiami ir sukuriamas naujas paveikslas, kuris gražinamas kaip rezultatas. Šio metodo privalumas - gaunama labai detali informacija apie kraštines, bet trūkumas - nesukuriama variacijų dėl tiesioginio pikselių konvertavimo.

Sukurtų algoritmų palyginimas rentgeno nuotraukose be išankstinio apdorojimo pateiktas 3.5 ir su išankstiniu apdorojimu pateiktas 3.6 paveiksle. Pagal gautus rezultatus galima matyti, kad OpenCV sugeba rasti tik ryškius ir lygius kraštus. HED ir morfologinėmis operacijomis paremtas metodas pasiekė daug geresnius rezultatus.

Stebint kraštinių aptikimo rezultatus rentgeno nuotraukose galima pastebėti, kad dėl išankstinio apdorojimo morfologinėmis operacijomis paremtas algoritmas daug ryškiau išskiria kraštines. Tai yra dėl pačių kaulų sričių išryškavimo ir didesnių šviesių pikselių reikšmių.

Apžvelgiant kraštinių aptikimo metodus, geriausiu laikomas HED metodas, dėl kaulų sričių neišryškavimo ir dėl netiesioginio pikselių reikšmių konvertavimo. Giliojo mokymosi modelių naudojimas aptinkant kraštines yra geresnis variantas lyginant su algoritmais, dėl platesnio pritaikymo ir netiesioginio konvertavimo.

	<i>AP oblique medial</i>	<i>AP oblique lateral</i>	<i>AP</i>	<i>Lateral lateromedial</i>
OpenCV Countours				
Kraštinių aptikimo algoritmas				
HED metodas				

3.5 pav. Rankos rentgeno nuotraukų eskizų generavimo rezultatai

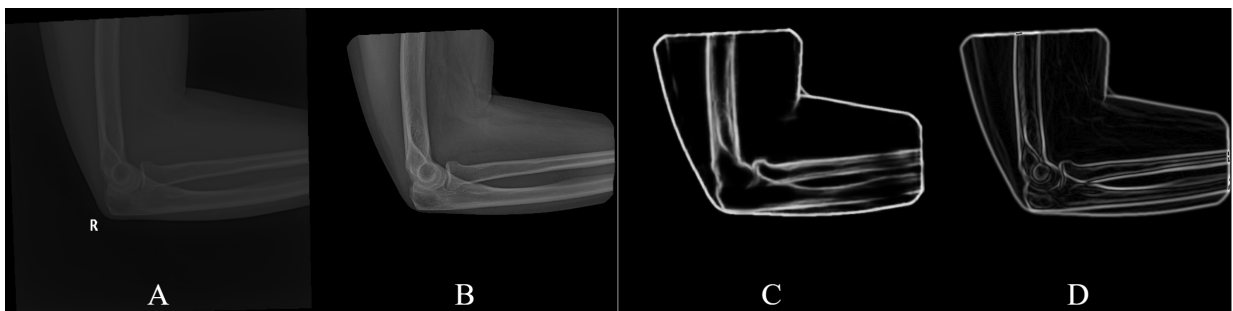
	<i>AP oblique medial</i>	<i>AP oblique lateral</i>	<i>AP</i>	<i>Lateral lateromedial</i>
OpenCV Countours				
Kraštinių aptikimo algoritmas				
HED metodas				

3.6 pav. Rankos rentgeno nuotraukų eskizų generavimo rezultatai duomenų rinkinyje su išankstiniu apdorojimu

Eskizų kūrimo metu susidurta su problema, kad nuotraukose nėra ryškių briaunų. Todėl kontūrų radimas naudojant OpenCV biblioteką nesuteikė jokių rezultatų. Siekiant išryškinti nuotraukų kontūrus ir briaunas atliktas duomenų rinkinio išankstinis apdorojimas. Rentgeno nuotraukoms po išankstinio apdorojimo buvo pritaikytas HED metodas nustatyti briaunas. Pasitelkus šį modelį gauti rezultatai pateikti 3.7 paveiksle yra pakankami, kad būtų galima generuoti alkūnės rentgeno nuotraukų eskizus. Kuo eskizas suteikia daugiau duomenų apie kaulų išsidėstymą ir poziciją tuo mažesnis generatoriaus nuotraukų kūrimo skirtumas. Todėl reikia ištirti ir eskizo išsamumo daromą įtaką nuotraukų generavimo įvairumui ir kokybei. Eskizų generavimui pritaikyti nuotraukų išankstinio apdorojimo metodai ir reikšmės:

1. Rentgenogramos kaukės radimas;
2. *Contrast limited adaptive histogram equalization*;
3. Ryškumo didinimas su reikšme – 1.6;
4. Kontrasto keitimas su reikšme – 3;
5. Šviesumo didinimas su reikšme – 1.2;
6. Ryškumo didinimas su reikšme – 3;
7. Kaulų srities išdidinimas panaudojant OpenCv kontūrų nustatymą ir apkarpyimą;
8. Išdidinimas nuotraukos į kvadrato formą;
9. HED;
10. Nuotraukos dydžio keitimas į 224×224 .

Atlikus nuotraukų apdorojimą rezultatus galima matyti 3.7 paveiksle.



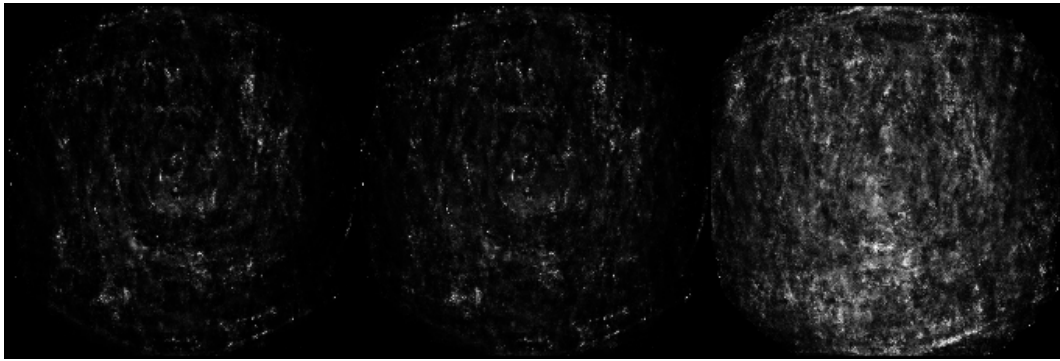
3.7 pav. A originali rentgenograma, B rentgenograma po išankstinio apdorojimo ir C rentgenograma sukurta naudojant HED metodą, D rentgenograma sukurta naudojant morfologines operacijas

Naudojant šiuos veiksmus sukurtas duomenų rinkinys, kurį sudaro originalios nuotraukos ir jų eskizai sukurti naudojant HED modelį, kuriam nebuvo atliktas mokinimas, nes MURA duomenų rinkinys neturi informacijos apie kraštines.

3.4.1. Duomenų rinkinio išplėtimas naudojant GAN modelius

GAN modelių pritaikymas išplečiant duomenų rinkinį yra antra tyrimo dalis. Atlikus klasikinio GAN modelio sudaryto iš generatoriaus ir diskriminatoriaus nepavyko sugeneruoti rentgeno nuotraukų, kurios atlikus vizualius patikrinimus atitiktų rentgenogramas. Dažniausiai pasitaikiusi problema, kad nepasiekiamas modelių konvergencija ir diskriminatorius greitai išmoksta atskirti sugeneruotas nuotraukas, todėl tolimesnis mokymasis nevyksta dėl gradiento perėjimo. Aukštos kokybės nuotraukų generavimas 224×224 yra sudėtingas, kadangi generatorius turi išmokti struktūrą, apšvietimą ir pasiskirstymą tarp pikselių, todėl dažniausiai atliekamas tik 24×24 ,

32×32 arba 64×64 nuotraukų generavimas naudojant Fashion-MNIST [69] duomenų rinkinį. GAN modelio treniravimo alkūnės rentgeno nuotraukų generavimo rezultatai pateikti 3.8 paveiksle.



3.8 pav. GAN modelio rezultatai generuojant 224×224 raiškos alkūnės rentgeno nuotraukas

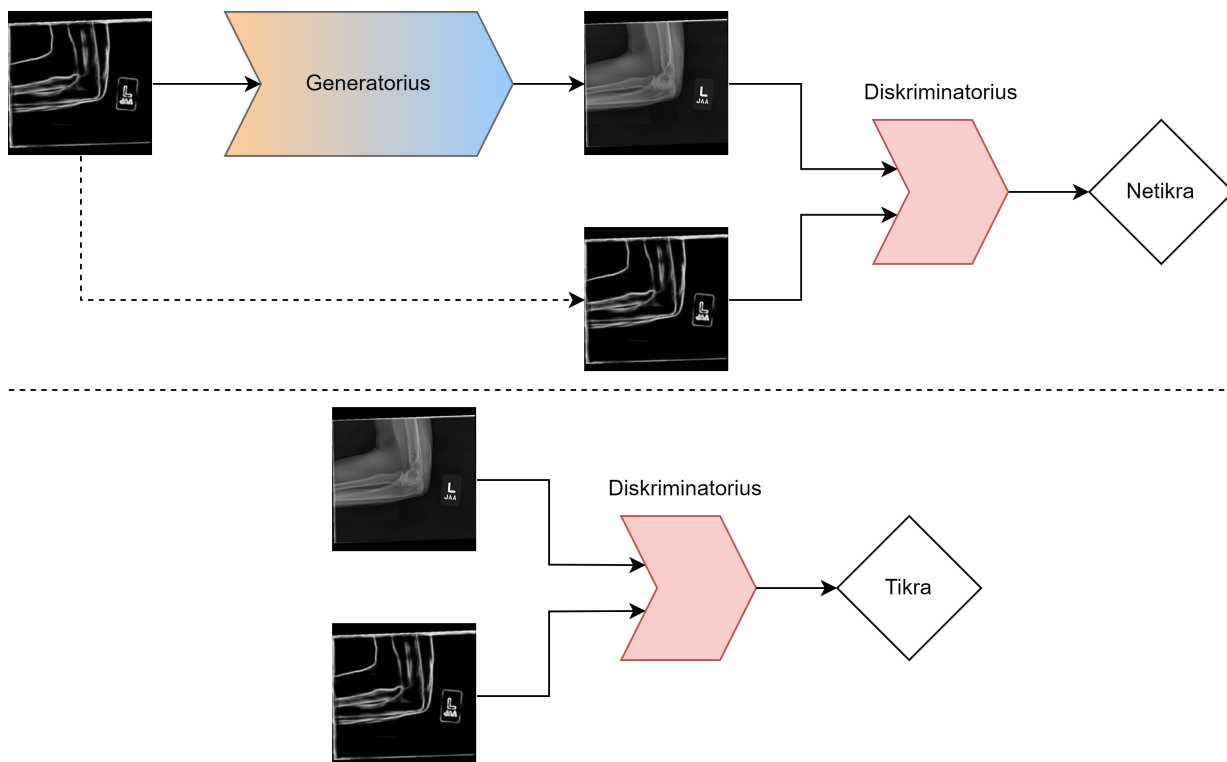
Gauti rezultatai pavaizduoti 3.8 paveiksle rodo, kad paprasto generatoriaus ir diskriminatoriaus modelio neužtenka pasiekti rentgeno nuotraukos sukūrimui. Po 100 epochų treniruojant tik su nuotraukomis be anomalijų kurių yra 2 935 nepavyko pasiekti net alkūnės rentgeno nuotraukos kontūrų sukūrimo. Naudojant sąlyginį GAN modelį gauti rezultatai yra tokie patys ir plokštuma kaip papildomas parametras generuojant nuotraukas nesuteikė pakankamai informacijos, kad sukurti sintetines alkūnės rentgeno nuotraukas.

3.4.2. Duomenų rinkinio išplėtimas naudojant Pix2Pix modelį

Treniravimo procesas atliekant sąlyginio GAN (Pix2Pix) treniravimą susieti eskizo kraštines su rentgeno nuotrauka pavaizduotas 3.9. Diskriminatorius D pirmiausia turi išmokti klasifikuoti su sintetinėmis rentgeno nuotraukomis, kurios yra sukurtos GAN generatoriaus ir tikromis nuotraukomis iš MURA duomenų rinkinio. Nuotraukos yra paduodamos sujungtos į vieną nuotrauką, kurią sudaro eskizas ir originali nuotrauka. Sekančiame etape generatyvinis priešiškas neuroninis tinklas yra treniruojamas sukurti sintetines rentgeno nuotraukas, kurios suklaidintų diskriminatorių. Skirtingai nei nesąlyginiuose GAN modeliuose tiek generatorius ir diskriminatorius mato eskizą ir originalią nuotrauką.

Pix2Pix modelis buvo sudarytas iš UNet 256 modelio generatoriaus, kuris geba generuoti 256×256 raiškos nuotraukas ir PatchGAN diskriminatoriaus. Šių modelių pasirinkimas buvo naudotas remiantis straipsniu [70]. Tai pat remiantis straipsnyje nurodytu pasiūlymu buvo naudotas jungčių praleidimas (angl. *skip connections*), kuris stipriai pagerina UNet modelio generuojamas nuotraukas.

Siekiant panaudoti Pix2Pix [70] architektūros generuojantį priešišką tinklą reikia nuotraukų iš skirtingų duomenų rinkinių, kurios atitiktų viena kitą vizualiai. Pavyzdžiui galima atlikti tiesioginį nuotraukų konvertavimą tarp MRI ir rentgenogramų. Šiame tyrime ši GAN ypatybė sukelia problemą, kadangi MURA duomenų rinkinys nesuteikia skirtingų tipų nuotraukų. Ir viešai prieinamų MRI nuotraukų taip pat nėra. Siekiant sugeneruoti nuotraukas reikia papildomo duomenų rinkinio, kurį būtų galima integruoti pritaikant Pix2Pix generatoriaus gebėjimą konvertuoti vieno tipo nuotraukas į kitą. Šio tyrimo metu buvo panaudotas HED metodas,



3.9 pav. Pix2Pix modelio treniravimo schema naudojant MURA duomenų rinkinio alkūnės rentgenogramas ir sukurtus eskizus

aptikti kraštines ir sukurti eskizą, kuris suteikia informaciją apie kaulų formą ir išsidėstymą erdvėje. Toks eskizų sukūrimas leidžia treniuoti GAN modelį generuoti naujas rentgenogramas tik iš eskizo kuriame nurodyta tik kaulų pozicija erdvėje. MURA duomenų rinkinyje alkūnės rentgeno nuotraukos vienam pacientui daugiausiai yra keturiose plokštumose. Kadangi nuotraukų yra mažai ir ne visuose atvejuose yra visų keturių plokštumų nuotraukų, tyrimo metu buvo išfiltruotos tik nuotraukos be anomalijų, kurios naudojamos modelio treniravimo metu neskirstant į skirtingas plokštumas. Atlikus 200 epochų treniravimą Pix2Pix modelio rezultatai pateikti 3.10 paveiksle.

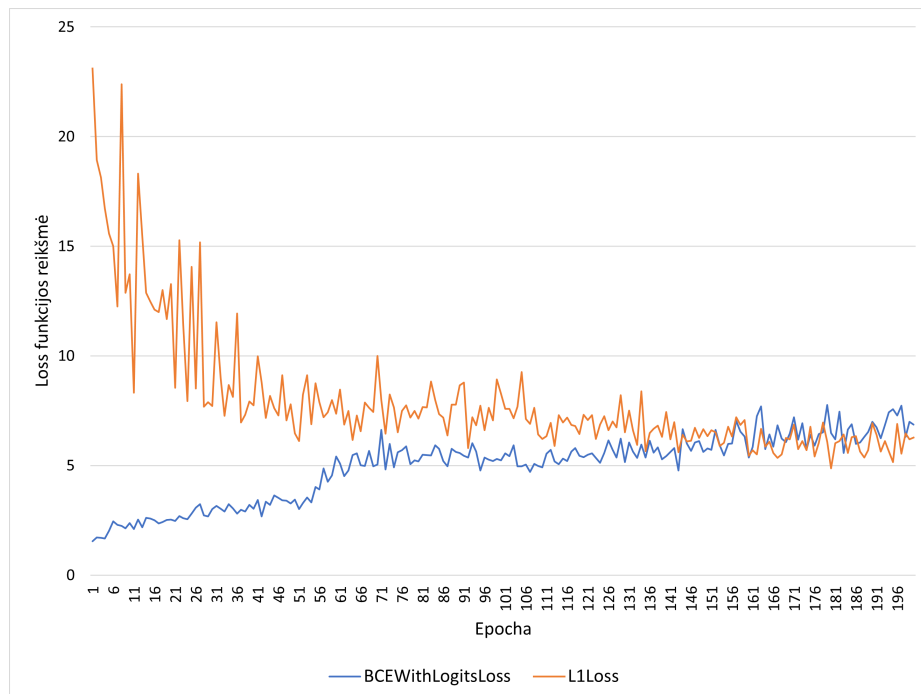
Pagrindiniai sunkumai pastebėti Pix2Pix generatoriaus buvo prasta nuotraukų kokybė. Vis skirtingi nuotraukų kampai sukuria efektą, kad modelis nenurodo aiškių alkūnės kaulų briaunų. Tai pat pastebėta kad yra rentgeno nuotraukų, kurios yra vienetinės kaip kūdikio alkūnės rentgenograma ar pusiau nukirptos alkūnės kaulų rentgenogramos. Tokiose atvejuose Pix2Pix sukurdavo rentgeno nuotrauką tik kaip šviesią sritį eskizo dalyje.

Treniruojant Pix2Pix buvo naudota dviejų skirtingų nuostolio funkcijų kombinacija norint pasiekti geresnius rezultatus. *L1Loss* yra sudarytas iš kriterijų, kurie matuoja vidutinę absoliučią paklaidą (angl. *mean absolute error*) (MAE) tarp kiekvieno įvesties elemento x ir tikslo y . *BCEWithLogitsLoss* yra *loss* funkcija, kuri sujungia sigmoidinį sluoksnį ir *BCELoss* vienoje klasėje. Pix2Pix modelio nuostolio funkcijų reikšmių kitimo kreivė pateikta 3.11 paveiksle. Grafike galime matyti kad Pix2Pix modelio tikslumas stabiliai didėja iki 145 epochos ir vėliau *BCEWithLogitsLoss* susilygina su *L1Loss*.

Gauti rezultatai generuojant rentgeno nuotraukas su Pix2Pix modeliu yra pakankami įtraukti



3.10 pav. Pix2Pix modelio sugeneruotų alkūnės rentgeno nuotraukų pavyzdžiai su sukurtais eskizais



3.11 pav. Pix2Pix modelio treniravimo efektyvumo *BCEWithLogitsLoss* ir *L1Loss* nuostolio funkcijų reikšmių pokytis epochose

į giliojo mokymosi modelių treniravimą. Tačiau atkreipiant dėmesį, kad nevisose atvejuose nuotraukos yra detalios ir turi blogai sugeneruotų sričių reikėtų jų neįtraukti į galutinį modelio treniravimo etapą. Iš gautų nuotraukų galima panaudoti Pix2Pix sintetines nuotraukas išmokyti giliojo mokymosi modelius apie rentgenogramų savybes ir išplėsti modelių pritaikymą generuojant nuotraukas su kuriomis giliojo mokymosi modeliai dažniausiai suklysta.

3.4.3. Duomenų rinkinio išplėtimas naudojant CycleGAN modelį

CycleGAN yra nauja GAN atmaina, kuri naudoja du generatorius ir du diskriminatorius, kurie sukuria nuotraukas atlikdami ciklinį nuotraukos generavimą. Pirmas generatorius sukuria nuotrauką perkeldamas duomenų rinkinio savybes ant eskizo nuotraukos, o antrasis generatorius iš sukurtos nuotraukos siekia atkurti eskizą. Tokiame treniravimo cikle vienas generatorius atsakingas už nuotraukos kūrimą, o kitas už erdvinės informacijos išlaikymą pagal eskizą. Abu generatoriai turi po vieną diskriminatorių, kurie įvertina sukurtas rentgeno nuotraukas ir eskizus. Siekiant pritaikyti šį modelį buvo nuspręsta pirmiausia apdoroti jau esamas nuotraukas sukuriant jų eskizus, kurie paverčiami į alkūnės rentgeno nuotraukas naudojant $B \rightarrow A$ GAN modelį. Kuriant CycleGAN specifinį duomenų rinkinį, kurį sudaro nuotraukos, kurios sudarytos iš originalios ir eskizo nuotraukos. Pagal eskizo tipą galima generuoti su ir be anomalijos rentgenogramas, tačiau dėl mažo kiekio nuotraukų, kuriose yra anomalijos pasirinkta generuoti tik nuotraukas be anomalijų. CycleGAN architektūra ir treniravimo procesas pateiktas 3.12 paveiksle.

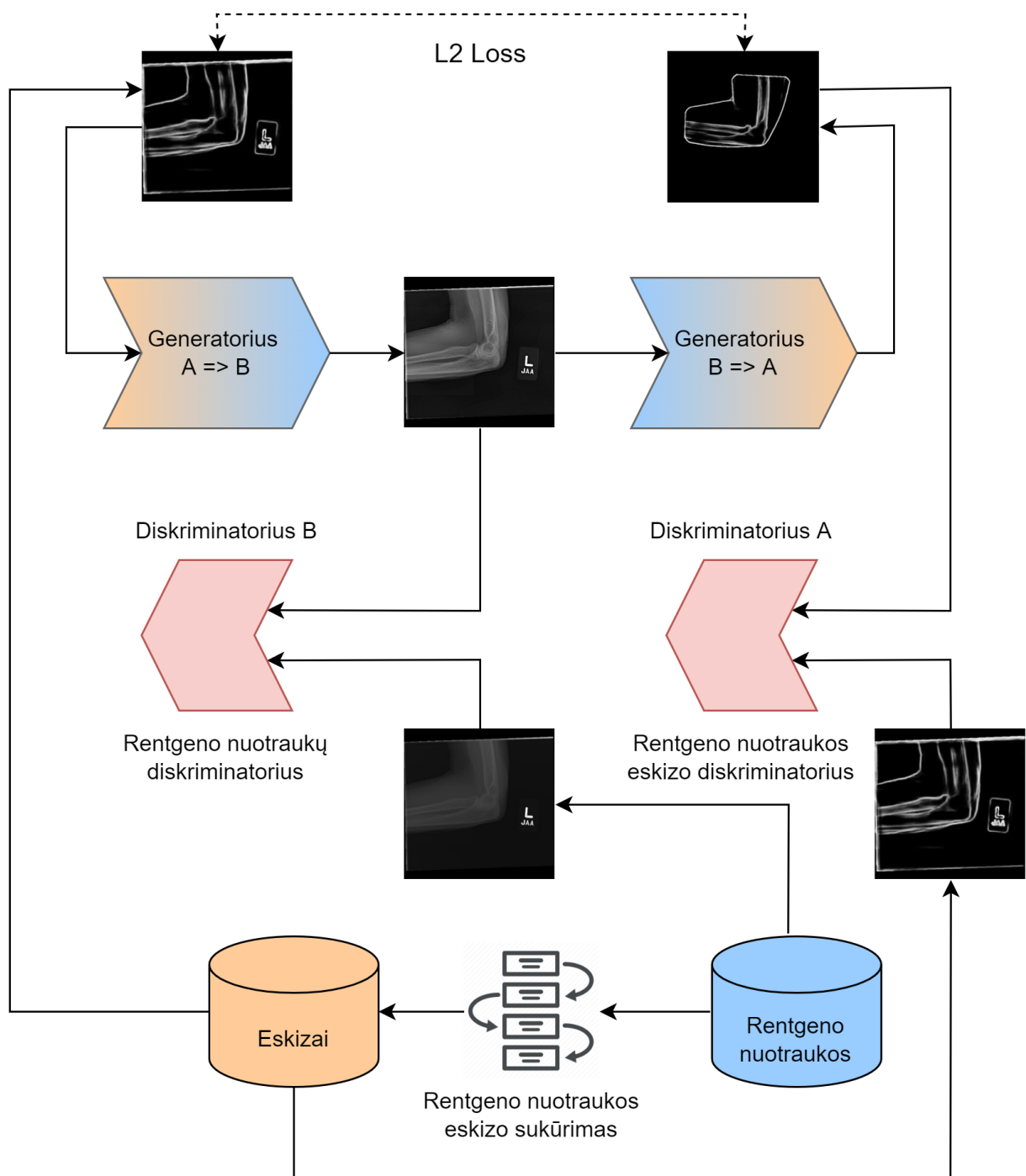
CycleGAN modelis buvo apmokytas 100 epochų. Šio modelio treniravimo laikas vienai epochai užėmė ± 3800 sekundžių ir užtruko 9-10 kartų ilgiau negu Pix2Pix modelio. Rentgeno nuotraukų generavimo rezultatai pateikti 3.13 paveiksle.

Iš gautų nuotraukų, galima daryti išvadą, kad sunkiausia yra generuoti alkūnės rentgeno nuotraukas *lateral lateromedial* plokštumoje. Tai pat skirtingose epochose CycleGAN generuoja skirtingo šviesumo nuotraukas, nes nuotraukos yra surinktos iš skirtingų šaltinių ir turi skirtingą šviesumą. Norint pasiekti geresnius rezultatus reikia identifikuoti skirtingą apšvietimą turinčias rentgeno nuotraukas ir suvienodinti. Pakeitus nuotraukų šviesumą CycleGAN modeliui būtų lengviau generuoti nuotraukas ir išspręstą skirtingo kontrasto nuotraukų problemą. Tai pat vietos, kuriose persidengia kaulai būtų negeneruojamos kaip šviesios sritys.

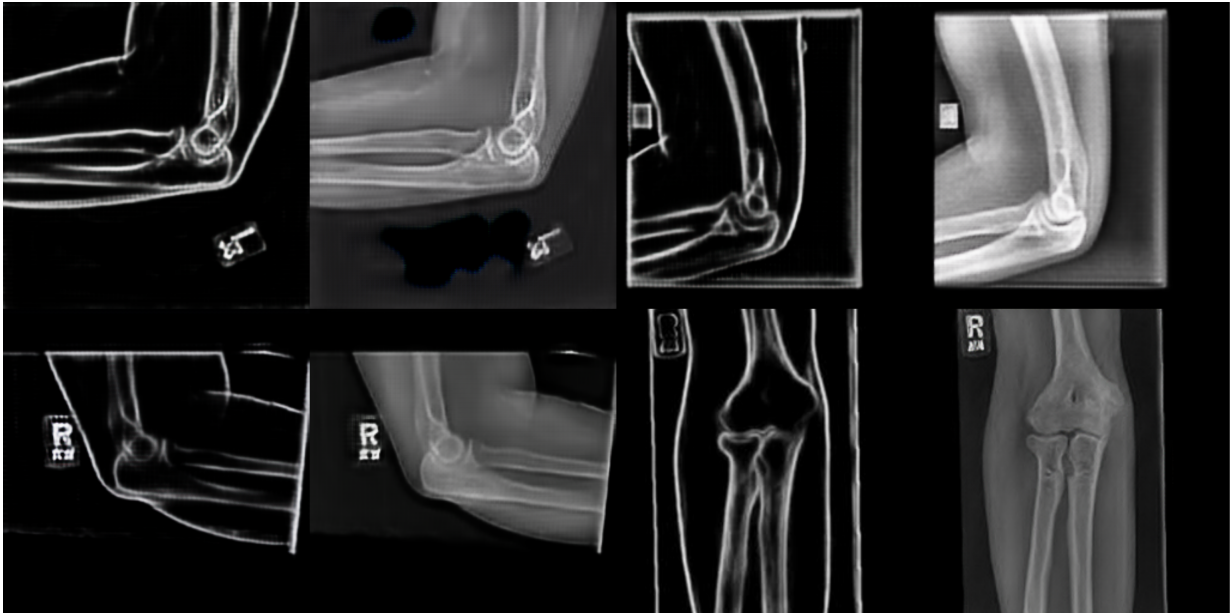
3.4.4. Duomenų rinkinio išplėtimo naudojant GAN modelius eksperimentų rezultatai

Atlikus giliojo mokymosi modelių treniravimą su duomenų rinkiniais, kurie papildyti 2424 sintetinėmis nuotraukomis sugeneruotomis CycleGAN ir Pix2Pix modelių. Tikslumo padidėjimas užfiksuotas visuose giliojo mokymosi modeliuose. Pix2Pix modelio išplėstame duomenų rinkinyje tikslumas vidutiniškai padidėjo per 4.27% ir geriausią tikslumą pasiekė Inception V3 modelis. Naudojant CycleGAN tikslumas vidutiniškai padidėjo per 4.48% ir geriausią tikslumą pasiekė MobileNet V2 modelis.

Didžiausias klasifikacijos tikslumo padidėjimas pastebėtas MobileNet v2 su CycleGAN 9.03% ir Pix2Pix 7.04%. ResNet klasės modeliuose su CycleGAN duomenų rinkiniu padidėjo nuo 5.28% iki 6.57%, o naudojant Pix2Pix nuo 3.85 % iki 5.94%. Pateiktame 3.14 grafike galime pamatyti,



3.12 pav. CycleGAN modelio treniravimo schema naudojant MURA duomenų rinkinio alkūnės rentgenogramas ir sukurtus eskizus



3.13 pav. CycleGAN modelio sugeneruotų alkūnės rentgeno nuotraukų pavyzdžiai su sukurtais eskizais

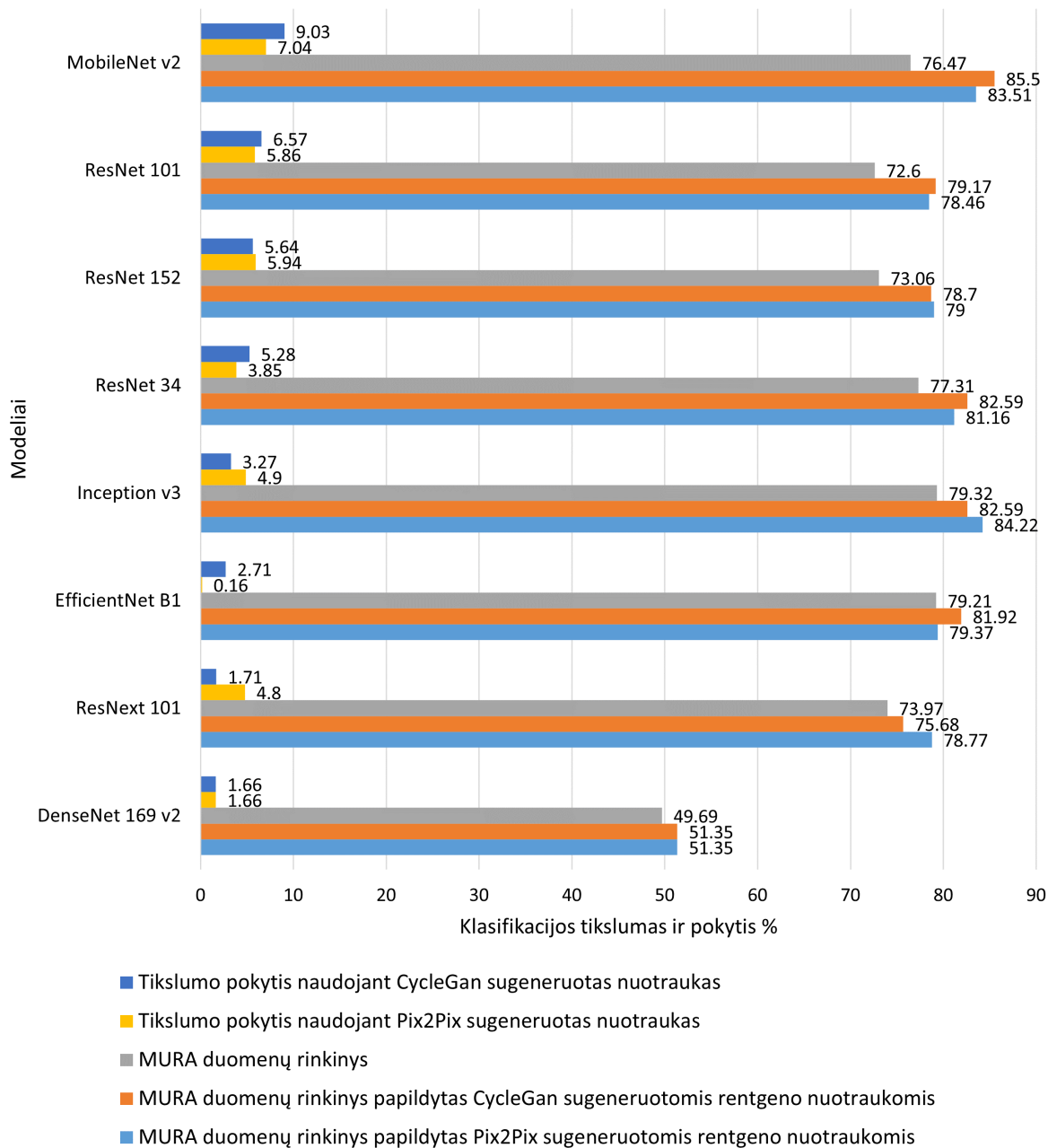
kad MobileNet V2 giliojo mokymosi modelis pasižymėjo didžiausiu tikslumo pagerėjimu nuo 52.66% iki 73.32% su CycleGAN sugeneruotomis nuotraukomis. Naudojant MURA duomenų rinkinį be pakeitimų modeliai pasiekia tik 50+% tikslumą, tačiau GAN tinklų sukurtos nuotraukos įtrauktos į mokymosi procesą gali pagerinti tikslumą vidutiniškai 4.38%.

Duomenų rinkinyje su Pix2Pix sugeneruotomis rentgenogramomis geriausias modelis buvo Inception V3, kuris surinko aukščiausius rezultatus pagal klasifikacijos tikslumą 84.22%, kappa 0.67 ir AUC 0.83. Inception V3 nusileido tik MobileNet v2 modeliui pasiekus F1 0.82 reikšmę ir aplenkus per 0.01 ir ResNext 101 modeliui pasiekusiam preciziškumo 0.95 reikšmę ir aplenkus per 0.03. DenseNet modelis nepasiekė didesnio nei 50% klasifikacijos tikslumo, todėl iš palyginimo lentelės pašalintas. Eksperimentų rezultatai duomenų rinkinyje be išankstinio apdorojimo pateikti 3.6 lentelėje.

3.5 lentelė. Giliojo mokymosi modelių tikslumo rezultatai su alkūnės rentgeno nuotraukų MURA duomenų rinkinyje su Pix2Pix sugeneruotomis nuotraukomis

Modelis	Klasifikacijos tikslumas %	Kappa	Precizija	F1	AUC
MobileNet v2	83.51	0.66	0.87	0.82	0.83
ResNet 101	78.46	0.57	0.9	0.74	0.78
Inception v3	84.22	0.67	0.92	0.81	0.83
ResNet 152	79	0.58	0.91	0.75	0.79
ResNext 101	78.77	0.56	0.95	0.72	0.78
ResNet 34	81.16	0.56	0.77	0.78	0.78
EfficientNet B1	79.18	0.59	0.76	0.8	0.79

Duomenų rinkinyje su CycleGAN sugeneruotomis rentgenogramomis geriausias modelis buvo MobileNet v2, kuris surinko aukščiausius rezultatus pagal klasifikacijos tikslumą 85.5%,



3.14 pav. Giliojo mokymosi modelių tikslumo pokytis su alkūnės rentgeno nuotraukomis MURA duomenų rinkinyje ir MURA duomenų rinkinyje su Pix2Pix ir CycleGAN sugeneruotomis nuotraukomis

kappa 0.68, F1 0.83 ir AUC 0.84. MobileNet v2 nusileido tik ResNext 101 modeliui pasiekus preciziškumo 0.96 reikšmę ir aplenkus per 0.1. Eksperimentų rezultatai duomenų rinkinyje be išankstinio apdorojimo pateikti 3.6 lentelėje.

3.6 lentelė. Giliojo mokymosi modelių tikslumo rezultatai su alkūnės rentgeno nuotraukų MURA duomenų rinkinyje su CycleGAN sugeneruotomis nuotraukomis

Modelis	Klasifikacijos tikslumas %	Kappa	Precizija	F1	AUC
MobileNet v2	85.5	0.68	0.86	0.83	0.84
ResNet 101	79.17	0.58	0.93	0.75	0.79
Inception v3	82.59	0.64	0.95	0.78	0.82
ResNet 152	78.7	0.57	0.95	0.73	0.78
ResNext 101	75.68	0.5	0.96	0.67	0.75
ResNet 34	83.63	0.48	0.71	0.75	0.74
EfficientNet B1	81.92	0.6	0.78	0.8	0.8

GAN modelių generuojamų nuotraukų eksperimentų rezultatai sekcijoje 3.4.1 nurodo, kad išplečiant duomenų rinkinį Pix2Pix ir CycleGAN modeliai gali sukurti patikimas sintetines alkūnės rentgeno nuotraukas ir padidinti giliojo mokymosi modelių klasifikacijos tikslumą vidutiniškai 4.38%. Tačiau šių modelių generuojamos nuotraukos nėra labai geros kokybės. Šias problemas iškelia nestandartizuotos nuotraukos, šviesumo skirtumai, eskizų generavimo sudėtingumas ir lėtas bei sudėtingas treniravimo procesas neleidžiantis atlikti daugiau eksperimentų rasti geriausiai kombinacijai. Blogai sugeneruotų nuotraukų įtraukimas į treniravimo procesą gali ir pabloginti giliojo mokymosi modelių rezultatus, todėl svarbu neįtraukti per daug nuotraukų ir reikia nuolatos stebėti kaip giliojo mokymosi modeliai reaguoja į naujai sugeneruotas ir į duomenų rinkinį pridėtas rentgenogramas. Blogas duomenų rinkinio išplėtimas pablogina rezultatus, nes pridėtos rentgenogramos yra klasifikuojamos kaip turinčios anomaliją. Todėl išspręsti blogai generuojamų nuotraukų problemą buvo apmokytas MobileNet V2 modelis klasifikuoti rentgeno nuotraukas į sugeneruotas ir ne. Šis modelis veikia kaip filtras atrinkti tik gerai sugeneruotas rentgeno nuotraukas. Naudojant papildomą giliojo mokymosi modelį pavyko pasiekti geresnius rezultatus išplečiant duomenų rinkinį.

Naudojant CycleGAN ir Pix2Pix buvo sugeneruotos papildomos nuotraukos duomenų rinkinyje ir įtrauktos į MobileNet V2 mokymosi procesą. MobileNet V2 buvo pasirinktas kaip pagrindinis modelis tikrinant duomenų rinkinio išplėtimo efektyvumą, nes pasižymėjo greitu mokymosi greičiu ir aukštu klasifikacijos tikslumu. Tik pasiekus gerus rezultatus su MobileNet V2 modeliu buvo atliekami eksperimentai su kitais modeliais. Toks procesas iškelia MobileNet V2 pranašumą prieš kitus modelius, todėl siekiant pagerinti kitą giliojo mokymosi modelį ir pritaikius generuojamas nuotraukas galima pasiekti dar geresnius rezultatus su kitais modeliais. Validavimo ir testavimo duomenų rinkiniai nebuvo pakeisti ir buvo naudojamos tik MURA duomenų rinkinio nuotraukos. Buvo atlikti eksperimentai keičiant generuojamų nuotraukų kiekį ir jų raišką. Testuotos 124×124 , 224×224 ir 330×330 raiškos nuotraukos. Tai pat atliktas testavimas įtraukiant 25%, 50%, 75% ir 100% sugeneruotų nuotraukų, kurios atrenkamos

atsitiktinai išskyrus 100% generavimo metu. Sugeneruotos rentgenogramos buvo naudojamos tik 66% modelio mokymosi laiko ir kitą laiką modelis buvo apmokomas tik su tikromis nuotraukomis. Toks modelių apnokinimas suteikia geresnius rezultatus validavimo duomenų rinkinyje.

Duomenų rinkinio išplėtimo su CycleGAN ir Pix2Pix sugeneruotomis rentgenogramomis eksperimentų rezultatai:

1. Įtraukus 25% generuotų nuotraukų modelio tikslumas padidėjo 2.39%. Įtraukiant 50% sintetinių nuotraukų buvo pasiektas aukščiausias vidutinis tikslumas 4.38%. Naudojant didesnę procentą sintetinių nuotraukų tikslumas sumažėja;
2. Naudojant 124×124 raiškos nuotraukas giliojo mokymosi modeliai nesugeba atskirti anomalijų. Su 330×330 raiškos nuotraukomis nesuteikia geresnių rezultatų galimai dėl sumažėjusio nuotraukų kiekio, kuris gali būti naudojamas vienos iteracijos metu;
3. CycleGAN ir Pix2Pix nesugeba sugeneruoti nuotraukų kur yra persidengiantys kaulai ir palieka šviesias sritis be aiškių kontūrų, tai lemia, kad giliojo mokymosi modeliai tokią sugeneruotą rentgenogramą klasifikuoja kaip anomaliją;
4. Siekiant padidinti rentgenogramų generavimo efektyvumą naudinga naudoti papildomą modelį, filtruojantį sintetines rentgeno nuotraukas. Tokiu būdu blogai sukurti eskizai ar GAN sukurtos nuotraukos nėra įtraukiamos į treniravimo duomenų rinkinį.

Atlikus tyrimą pastebėta, kad siekiant pritaikyti generatyvinius priešiškus tinklus būtina reikalingi šie veiksmai: duomenų rinkinio išankstinis apdorojimas, siekiant išryškinti nuotraukose kontūrus, giliojo mokymosi modelis generuojantis rentgenogramų eskizus, kontūrų aptikimas rentgeno nuotraukose ir dinamiškas duomenų rinkinių sukūrimas. Viso proceso apjungimas į vieną modelį suteiktų didesnę lankstumą, bet reikalauja daug vaizdo grafinės plokštės atminties dėl 9 skirtingų modelių naudojant CycleGAN ir dėl mokymo procese naudojamų aukštos raiškos nuotraukų.

Eksperimentų rezultatai rodo, kad CycleGAN ir Pix2Pix modeliai gali būti pritaikyti generuoti rentgeno nuotraukas. Giliojo mokymosi modeliai su duomenų rinkiniu su išankstiniu apdorojimu suteikė vidutinį 4.67% tikslumo pagerėjimą, Pix2Pix išplėstu duomenų rinkiniu suteikė vidutinį 4.27% pagerėjimą, o CycleGAN vidutinį 4.48% pagerėjimą. Generatyvinių priešišku tinklų pritaikymas išplėsti duomenų rinkinį nusileidžia nuo 0.19-0.4% lyginant su išankstiniu duomenų rinkinio apdorojimu.

Išvados

1. Mažas duomenų rinkinys ir nestandardizuotos rentgeno nuotraukos yra viena iš didžiausių problemų pritaikant giliojo mokymosi modelius MURA duomenų rinkinyje. Giliojo mokymosi modelių klasifikacijos tikslumas gali padidėti iki 20% pritaikant išankstinį rentgeno nuotraukų apdorojimą;
2. Pritaikant išankstinę rentgeno nuotraukų apdorojimo metodų kombinaciją galima padidinti giliojo mokymosi modelių klasifikacijos tikslumą vidutiniškai iki 4.6% iš anksto apmokytuose modeliuose su ImageNet ir 8.6% be išankstinio apmokymo;
3. Atlikus giliojo mokymosi modelių testavimą geriausius rezultatus pasiekė MobileNet v2 architektūros modelis iš anksto apmokytas su ImageNet pasiekęs 85.5% anomalijos klasifikavimo tikslumą su CycleGAN išplėstu duomenų rinkiniu;
4. Atlikus eksperimentus su generatyviniais priešiškais tinklais geriausius rezultatus pasiekė paveikslą į paveikslą konvertuojantys Pix2Pix ir CycleGAN modeliai, kurie yra tinkami generuojant sintetines alkūnės rentgeno nuotraukas. Naudojant sintetines rentgeno nuotraukas duomenų rinkinyje giliojo mokymosi modelių anomalijų klasifikavimo tikslumas padidėja vidutiniškai 4.38%.

Literatūros sarakšas

1. JONES, Rebecca M. ir kt. Assessment of a deep-learning system for fracture detection in musculoskeletal radiographs. *npj Digital Medicine*. 2020, t. 3, p. 1–6. ISSN 23986352. Pasiiekiamas per DOI: 10.1038/s41746-020-00352-w.
2. GUAN, Bin ir kt. Arm fracture detection in X-rays based on improved deep convolutional neural network. *Computers and Electrical Engineering*. 2020, t. 81, p. 106530. ISSN 00457906. Pasiiekiamas per DOI: 10.1016/j.compeleceng.2019.106530.
3. JONES, Rebecca M. ir kt. Assessment of a deep-learning system for fracture detection in musculoskeletal radiographs. *npj Digital Medicine*. 2020, t. 3, p. 1–6. ISSN 23986352. Pasiiekiamas per DOI: 10.1038/s41746-020-00352-w.
4. MA, Yangling; Yixin LUO. Bone fracture detection through the two-stage system of Crack-Sensitive Convolutional Neural Network. *Informatics in Medicine Unlocked*. 2021, t. 22, p. 100452. ISSN 23529148. Pasiiekiamas per DOI: 10.1016/j.imu.2020.100452.
5. THIAN, Yee Liang ir kt. Convolutional Neural Networks for Automated Fracture Detection and Localization on Wrist Radiographs. *Radiology: Artificial Intelligence*. 2019, t. 1, e180001. ISSN 2638-6100. Pasiiekiamas per DOI: 10.1148/ryai.2019180001.
6. YADAV, D. P.; Sandeep RATHOR. Bone Fracture Detection and Classification using Deep Learning Approach. *2020 International Conference on Power Electronics and IoT Applications in Renewable Energy and its Control, PARC 2020*. 2020, p. 282–285. ISBN 9781728165752. Pasiiekiamas per DOI: 10.1109/PARC49193.2020.236611.
7. KIM, D. H.; T. MACKINNON. Artificial intelligence in fracture detection: transfer learning from deep convolutional neural networks. *Clinical Radiology*. 2018, t. 73, p. 439–445. ISSN 1365229X. Pasiiekiamas per DOI: 10.1016/j.crad.2017.11.015.
8. PASA, F. ir kt. Efficient Deep Network Architectures for Fast Chest X-Ray Tuberculosis Screening and Visualization. *Scientific Reports*. 2019, t. 9, p. 2–10. ISSN 20452322. Pasiiekiamas per DOI: 10.1038/s41598-019-42557-4.
9. TOKIME, Roger Booto; Xavier MALDAGUE; Luc PERRON. Automatic Defect Detection for X-Ray inspection: Identifying defects with deep convolutional network. 2019, p. 1–8. Prieiga per: <http://www.ndt.net/?id=24703>.
10. GUAN, Bin; Jinkun YAO; Guoshan ZHANG; Xinbo WANG. Thigh fracture detection using deep learning method based on new dilated convolutional feature pyramid network. *Pattern Recognition Letters*. 2019, t. 125, p. 521–526. ISSN 01678655. Pasiiekiamas per DOI: 10.1016/j.patrec.2019.06.015.
11. SANDFORT, Veit; Ke YAN; Perry J. PICKHARDT; Ronald M. SUMMERS. Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks. *Scientific Reports*. 2019, t. 9. ISSN 20452322. Pasiiekiamas per DOI: 10.1038/s41598-019-52737-x.
12. RAJPURKAR, Pranav ir kt. MURA: Large dataset for abnormality detection in musculoskeletal radiographs. *arXiv*. 2017, p. 1–10. ISSN 23318422.
13. CHUNG, Seok Won ir kt. Automated detection and classification of the proximal humerus fracture by using deep learning algorithm. *Acta Orthopaedica*. 2018, t. 89, p. 468–473. ISSN 17453682. Pasiiekiamas per DOI: 10.1080/17453674.2018.1453714.

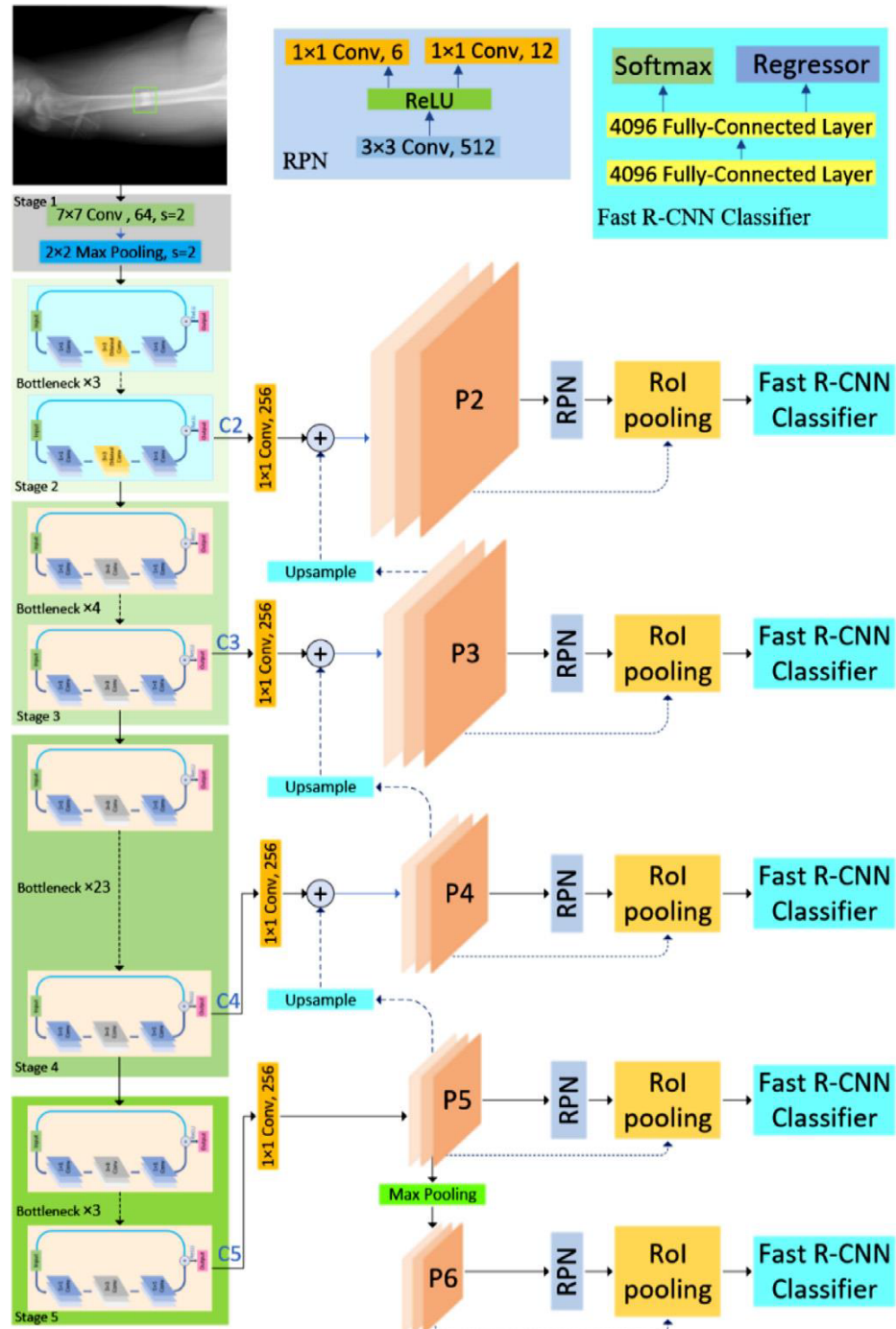
14. LIANG, Shuang; Yu GU. Towards robust and accurate detection of abnormalities in musculoskeletal radiographs with a multi-network model. *Sensors (Switzerland)*. 2020, t. 20. ISSN 14248220. Pasiiekiamas per DOI: 10.3390/s20113153.
15. LECUN, Yann; Léon BOTTOU; Yoshua BENGIO; Patrick HAFFNER. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*. 1998, t. 86, p. 2278–2323. ISSN 00189219. Pasiiekiamas per DOI: 10.1109/5.726791.
16. GLOROT, Xavier; Antoine BORDES; Yoshua BENGIO. Deep sparse rectifier neural networks. *Journal of Machine Learning Research*. 2011, t. 15, p. 315–323. ISSN 15324435.
17. KRIZHEVSKY, By Alex; Ilya SUTSKEVER; Geoffrey E HINTON. ImageNet Classification with Deep Convolutional Neural Networks. *Communications of the ACM*. 2012, t. 60, p. 84–90.
18. LECUN, Yann; Léon BOTTOU; Yoshua BENGIO; Patrick HAFFNER. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*. 1998, t. 86, p. 2278–2323. ISSN 00189219. Pasiiekiamas per DOI: 10.1109/5.726791.
19. LI, Sihan; Jiantao JIAO; Yanjun HAN; Tsachy WEISSMAN. Demystifying ResNet. 2016, p. 1–18. Prieiga per: <http://arxiv.org/abs/1611.01186>.
20. GEVERS, Theo; Arnold SMEULDERS. Deep Networks with Stochastic Depth. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2016, t. 9914 LNCS, p. V. ISBN 9783319488806. ISSN 16113349. Pasiiekiamas per DOI: 10.1007/978-3-319-46493-0.
21. GEVERS, Theo; Arnold SMEULDERS. Deep Networks with Stochastic Depth. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2016, t. 9914 LNCS, p. V. ISBN 9783319488806. ISSN 16113349. Pasiiekiamas per DOI: 10.1007/978-3-319-46493-0.
22. ZHANG, Ke ir kt. Residual Networks of Residual Networks: Multilevel Residual Networks. *IEEE Transactions on Circuits and Systems for Video Technology*. 2018, t. 28, p. 1303–1314. ISSN 10518215. Pasiiekiamas per DOI: 10.1109/TCSVT.2017.2654543.
23. HE, Kaiming; Xiangyu ZHANG; Shaoqing REN; Jian SUN. Identity mappings in deep residual networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2016, t. 9908 LNCS, p. 630–645. ISBN 9783319464923. ISSN 16113349. Pasiiekiamas per DOI: 10.1007/978-3-319-46493-0_38.
24. IOFFE, Sergey; Christian SZEGEDY. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *32nd International Conference on Machine Learning, ICML 2015*. 2015, t. 1, p. 448–456. ISBN 9781510810587.
25. HE, Kaiming; Xiangyu ZHANG; Shaoqing REN; Jian SUN. Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2016, t. 2016-Decem, p. 770–778. ISBN 9781467388504. ISSN 10636919. Pasiiekiamas per DOI: 10.1109/CVPR.2016.90.
26. ZAGORUYKO, Sergey; Nikos KOMODAKIS. Wide Residual Networks. *British Machine Vision Conference 2016, BMVC 2016*. 2016, t. 2016-Sept, p. 87.1–87.12. Pasiiekiamas per DOI: 10.5244/C.30.87.

27. SZEGEDY, Christian ir kt. Rethinking the Inception Architecture for Computer Vision. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2016, t. 2016-Decem, p. 2818–2826. ISBN 9781467388504. ISSN 10636919. Pasiiekiamas per DOI: 10.1109/CVPR.2016.308.
28. LARSSON, Gustav; Michael MAIRE; Gregory SHAKHAROVICH. FractalNet: Ultra-deep neural networks without residuals. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*. 2017, p. 1–11.
29. MOCSARI, E.; S. S. STONE. Densely Connected Convolutional Networks. *American Journal of Veterinary Research*. 1978, t. 39, p. 1442–1446. ISSN 00029645.
30. YU, Fisher; Vladlen KOLTUN. Multi-scale context aggregation by dilated convolutions. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*. 2016.
31. KANDEL, Ibrahim; Mauro CASTELLI; Aleš POPOVIČ. Musculoskeletal Images Classification for Detection of Fractures Using Transfer Learning. *Journal of Imaging*. 2020, t. 6, p. 127. ISSN 2313-433X. Pasiiekiamas per DOI: 10.3390/jimaging6110127.
32. REN, Shaoqing; Kaiming HE; Ross GIRSHICK; Jian SUN. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2017, t. 39, p. 1137–1149. ISSN 01628828. Pasiiekiamas per DOI: 10.1109/TPAMI.2016.2577031.
33. CHUNG, Seok Won ir kt. Automated detection and classification of the proximal humerus fracture by using deep learning algorithm. *Acta Orthopaedica*. 2018, t. 89, p. 468–473. ISSN 17453682. Pasiiekiamas per DOI: 10.1080/17453674.2018.1453714.
34. WEI, Shih En ir kt. VR facial animation via multiview image translation. *ACM Transactions on Graphics*. 2019, t. 38. ISSN 15577368. Pasiiekiamas per DOI: 10.1145/3306346.3323030.
35. MENICK, Jacob; Nal KALCHBRENNER; Google Brain AMSTERDAM. *GENERATING HIGH FIDELITY IMAGES WITH SUBSCALE PIXEL NETWORKS AND MULTIDIMENSIONAL UPSCALING*.
36. PARK, Taesung; Ming-Yu LIU; Ting-Chun WANG; Jun-Yan ZHU. Semantic Image Synthesis with Spatially-Adaptive Normalization. 2019. Prieiga per: <http://arxiv.org/abs/1903.07291>.
37. DEN, Aäron Van ir kt. *Conditional Image Generation with PixelCNN Decoders*.
38. SHIN, Yirang; Jaemoon YANG; Young Han LEE. Deep generative adversarial networks: Applications in musculoskeletal imaging. *Radiology: Artificial Intelligence*. 2021, t. 3. ISSN 26386100. Pasiiekiamas per DOI: 10.1148/ryai.2021200157.
39. MIRZA, Mehdi; Simon OSINDERO. Conditional Generative Adversarial Nets. 2014. Prieiga per: <http://arxiv.org/abs/1411.1784>.
40. PANG, Yingxue; Jianxin LIN; Tao QIN; Zhibo CHEN. Image-to-Image Translation: Methods and Applications. 2021. Prieiga per: <http://arxiv.org/abs/2101.08629>.
41. KINGMA, Diederik P.; Jimmy BA. Adam: A Method for Stochastic Optimization. 2014. Prieiga per: <http://arxiv.org/abs/1412.6980>.

42. SALIMANS, Tim; Andrej KARPATY; Xi CHEN; Diederik P KINGMA. *PixelCNN++: Improving the PixelCNN with discretized logistic mixture likelihood and other modifications*. Prieiga per: <https://github.com/openai/pixel-cnn>, .
43. DEN, Aäron Van ir kt. Conditional Image Generation with PixelCNN Decoders. [S. a.].
44. SALIMANS, Tim; Andrej KARPATY; Xi CHEN; Diederik P. KINGMA. *PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications*. 2017. Prieiga per: <http://arxiv.org/abs/1701.05517>.
45. OORD, Aaron van den; Nal KALCHBRENNER; Koray KAVUKCUOGLU. *Pixel Recurrent Neural Networks*. 2016. Prieiga per: <http://arxiv.org/abs/1601.06759>.
46. THEIS, Lucas; Matthias BETHGE. *Generative Image Modeling Using Spatial LSTMs*. 2015. Prieiga per: <http://arxiv.org/abs/1506.03478>.
47. CENTER, Stanford AIMI. *LERA- Lower Extremity RAdiographs*. [S. a.].
48. DIMILILER, Kamil. IBFDS: Intelligent bone fracture detection system. *Procedia Computer Science*. 2017, t. 120, p. 260–267. ISSN 18770509. Pasiokiamas per DOI: 10.1016/j.procs.2017.11.237.
49. LINDSEY, Robert ir kt. Deep neural network improves fracture detection by clinicians. *Proceedings of the National Academy of Sciences of the United States of America*. 2018, t. 115, p. 11591–11596. ISSN 10916490. Pasiokiamas per DOI: 10.1073/pnas.1806905115.
50. OLCZAK, Jakub ir kt. Artificial intelligence for analyzing orthopedic trauma radiographs: Deep learning algorithms—are they on par with humans for diagnosing fractures? *Acta Orthopaedica*. 2017, t. 88, p. 581–586. ISSN 17453682. Pasiokiamas per DOI: 10.1080/17453674.2017.1344459.
51. ENNOUAMANI, Soukaina; Zouhir MAHANI. *Towards Adaptive Learning Systems Based on Fuzzy-Logic*. 2019. ISBN 9783030228705. ISSN 21945365. Pasiokiamas per DOI: 10.1007/978-3-030-22871-2_42.
52. CHENG, Chi Tung ir kt. Application of a deep learning algorithm for detection and visualization of hip fractures on plain pelvic radiographs. *European Radiology*. 2019, t. 29, p. 5469–5477. ISSN 14321084. Pasiokiamas per DOI: 10.1007/s00330-019-06167-y.
53. TANZI, Leonardo; Enrico VEZZETTI; Rodrigo MORENO; Sandro MOOS. X-Ray bone fracture classification using deep learning: A baseline for designing a reliable approach. *Applied Sciences (Switzerland)*. 2020, t. 10. ISSN 20763417. Pasiokiamas per DOI: 10.3390/app10041507.
54. DIMILILER, Kamil. IBFDS: Intelligent bone fracture detection system. *Procedia Computer Science*. 2017, t. 120, p. 260–267. ISSN 18770509. Pasiokiamas per DOI: 10.1016/j.procs.2017.11.237.
55. JONES, Rebecca M. ir kt. Assessment of a deep-learning system for fracture detection in musculoskeletal radiographs. *npj Digital Medicine*. 2020, t. 3, p. 1–6. ISSN 23986352. Pasiokiamas per DOI: 10.1038/s41746-020-00352-w.
56. MICHEL, Vincent ir kt. A supervised clustering approach for fMRI-based inference of brain states. 2011. Pasiokiamas per DOI: 10.1016/j.patcog.2011.04.006.
57. DUBOIS, PAUL F. Guest Editor’s Introduction: Python: Batteries Included. [S. a.].

58. PEDREGOSA, Fabian ir kt. Scikit-learn: Machine Learning in Python. 2012. Prieiga per: <http://arxiv.org/abs/1201.0490>.
59. MEHLIG, B. Machine learning with neural networks. 2019. Pasiiekiamas per DOI: 10.1017/9781108860604.
60. HAGHIGHAT, Ehsan; Ruben JUANES. SciANN: A Keras/Tensorflow wrapper for scientific computations and physics-informed deep learning using artificial neural networks. 2020. Pasiiekiamas per DOI: 10.1016/j.cma.2020.113552.
61. ABADI, Martín ir kt. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. 2016. Prieiga per: <http://arxiv.org/abs/1603.04467>.
62. KURTH, Thorsten ir kt. TensorFlow at Scale: Performance and productivity analysis of distributed training with Horovod, ML SL, and Cray PE ML. John Wiley ir Sons Ltd, 2019, t. 31. ISSN 15320634. Pasiiekiamas per DOI: 10.1002/cpe.4989.
63. PASZKE, Adam ir kt. PyTorch: An Imperative Style, High-Performance Deep Learning Library. 2019. Prieiga per: <http://arxiv.org/abs/1912.01703>.
64. AHMED, Zeeshan ir kt. Machine learning at microsoft with ML.Net. Association for Computing Machinery, 2019, p. 2448–2458. ISBN 9781450362016. Pasiiekiamas per DOI: 10.1145/3292500.3330667.
65. PANIGRAHI, Santisudha; Anuja NANDA; Tripti SWARNKAR. A Survey on Transfer Learning. *Smart Innovation, Systems and Technologies*. 2021, t. 194, p. 781–789. ISBN 9789811559709. ISSN 21903026. Pasiiekiamas per DOI: 10.1007/978-981-15-5971-6_83.
66. MAJDI, Mohammad S. ir kt. Deep Learning Classification of Chest X-Ray Images. *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation*. 2020, t. 2020-March, p. 116–119. ISBN 9781728157450. Pasiiekiamas per DOI: 10.1109/SSIAI49293.2020.9094612.
67. RAJPURKAR, Pranav ir kt. MURA: Large dataset for abnormality detection in musculoskeletal radiographs. *arXiv*. 2017, p. 1–10. ISSN 23318422.
68. XIE, Saining; Zhuowen TU. Holistically-Nested Edge Detection. 2015. Prieiga per: <http://arxiv.org/abs/1504.06375>.
69. XIAO, Han; Kashif RASUL; Roland VOLLGRAF. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. 2017. Prieiga per: <http://arxiv.org/abs/1708.07747>.
70. ISOLA, Phillip; Jun-Yan ZHU; Tinghui ZHOU; Alexei A. EFROS. Image-to-Image Translation with Conditional Adversarial Networks. 2016. Prieiga per: <http://arxiv.org/abs/1611.07004>.



4. Priedai



4.1 pav. DCFPN gilijo mokymosi modelio architektūra [30].

Article

An Intelligent Solution for Automatic Garment Measurement Using Image Recognition Technologies

Agne Paulauskaite-Taraseviciene ^{1,*} , Eimantas Noreika ¹, Ramunas Purtokas ¹, Ingrida Lagzdinyte-Budnike ¹ , Vytautas Daniulaitis ¹ and Ruta Salickaite-Zukauskiene ²

¹ Faculty of Informatics, Kaunas University of Technology, Studentu 50, 51368 Kaunas, Lithuania; eimantas.noreika@ktu.edu (E.N.); ramunas.purtokas@ktu.edu (R.P.); ingrida.lagzdinyte-budnike@ktu.lt (I.L.-B.); vytautas.daniulaitis@ktu.lt (V.D.)

² Noselfish MB, Slaito 4, 59204 Birstonas, Lithuania; ruta@noselfish.com

* Correspondence: agne.paulauskaite-taraseviciene@ktu.lt

Abstract: Global digitization trends and the application of high technology in the garment market are still too slow to integrate, despite the increasing demand for automated solutions. The main challenge is related to the extraction of garment information-general clothing descriptions and automatic dimensional extraction. In this paper, we propose the garment measurement solution based on image processing technologies, which is divided into two phases, garment segmentation and key points extraction. UNet as a backbone network has been used for mask retrieval. Separate algorithms have been developed to identify both general and specific garment key points from which the dimensions of the garment can be calculated by determining the distances between them. Using this approach, we have resulted in an average 1.27 cm measurement error for the prediction of the basic measurements of blazers, 0.747 cm for dresses and 1.012 cm for skirts.

Keywords: segmentation; UNet; garment size detection; edge detection; key points



Citation: Paulauskaite-Taraseviciene, A.; Noreika, E.; Purtokas, R.; Lagzdinyte-Budnike, I.; Daniulaitis, V.; Salickaite-Zukauskiene, R. An Intelligent Solution for Automatic Garment Measurement Using Image Recognition Technologies. *Appl. Sci.* **2022**, *12*, 4470. <https://doi.org/10.3390/app12094470>

Academic Editors: Mirco Peron, Luca Fumagalli and Simone Arena

Received: 25 March 2022

Accepted: 26 April 2022

Published: 28 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

One of the most powerful and widely used types of artificial intelligence is computer vision, which aims to mimic some of the complexity of the human visual system and enable computers to detect and identify objects in images and videos. Computer vision techniques cover an increasing number of applications and engineering aspects of computing related to image recognition, including scientific work proposing innovative algorithms or solutions for commercial, industrial, military and biomedical applications. The increasing use of computer vision in everyday life contributes to the efficiency of various aspects of the field. Although the use of these technologies allows solving many complex tasks (automated object detection and identification, tracking), the detection of defects and anomalies is one of the most valuable investigations in medicine [1], bio-medicine [2], manufacturing [3] and agriculture [4]. For example, image recognition techniques based on deep learning can be used to enable advanced disease control in agriculture [5–7], to identify product defects and increase the quality control in manufacturing [8], automated assessment, prediction and assistance in medicine [9,10], increase the success rate of bio-medicine procedures [11], provide intelligent road safety solutions [12,13] and many others.

Online shopping is the most popular online activity worldwide, and the key value of intelligent image recognition solutions for e-commerce lies in the ability to identify products quickly and accurately. However, global digitization trends and the application of high technology in the garment market are still too slow to integrate, despite the increasing demand for automated solutions and the fact that the challenges are quite clear and already discussed in different researches. In principle, the main challenge is related to the extraction of garment information-general clothing descriptions, automatic dimensional extraction and textual information retrieval from the tags (size, brand, fabric composition, etc.). Currently, the accuracy and completeness of the information about garments on sales

platforms still relies on a significant amount of manual and tedious work. Measuring a garment is extremely time-consuming and often requires multiple measurements to reduce measurement error. Artificial intelligence (AI) technologies have the potential to meet the demand to adopt the automation technology in this sector by increasing the speed and accuracy of garment measurement [14–16]. Given CNN's success in a range of domains, the deep learning-based solution has also demonstrated its superiority in performing a variety of garment recognition tasks. The approach based RCNN has been proposed for the shirt attributes recognition task, including the Inception-ResNet V1 model with LSoftmax for images representation and identification of their categories [17]. The experimental results show an overall labelling rate of 87.77%, a precision of 73.59% and a recall of 83.84%. A fully convolutional network and SP-FEN architecture have been proposed to parse clothing in fashion images. The proposed model has shown accuracy in terms of the overall pixel-wise accuracy and clothes parsing performance (pixel accuracy of 92.67 and MIoU of 48.26) [18]. However, the main objective is to identify individual garments, which implies a semantic segmentation task by assigning a class label to each pixel of the image.

A review of relevant research has shown that it is much easier to measure clothes lying down than hanging (e.g., on a mannequin). In [19], special equipment to capture images of tiled garments has been proposed which enables automatic garment measurements. The shooting device consists of a digital camera, LED light, shooting stand and workbench. A garment template is employed to recognize garment types and feature points, which are used to calculate garment sizes. Experimental results show that the accuracy of the approach can meet the requirements of the apparel industry since the average relative error is $\sim 2\%$. Tolerable error in the fashion industry is ~ 2 cm. The authors in [20] present an idea and apps that allow measuring the lay-down of a garment placed on a marked board. The proposed app then shoots the garment using the top camera form above and automatically captures many of the garment's standard measurement points. The basic strategy is to first detect key points of interest in the clothing item and then use known measurements from demarcations on the backdrop to infer distances between those points. To measure lying-down clothes a fuzzy edge-detection algorithm can be used to detect the edge of garment image [21]. Then a corner-detection algorithm based on Freeman code is invoked to locate the corner points. The experiment results show that the proposed approach can measure t-shirts with the related error from 0.73% to 2.84% depending on the measured points. The smallest error has been obtained for the garment length (less than 1%).

Measurement solutions with requirements for a fixed position of the garment may limit their use and application, although the accuracy of such solutions is quite high (up to 0.5 cm error) [22], because, under real-world conditions (non-laboratory or industrial-oriented), the position of different garments in each image can vary. This means that it is quite complicated to use pre-designed templates to extract the essential dimensions of a garment. Adherence to certain equipment or templates is more semi-automatic solutions that require additional calibration, widespread interruptions from distinct angles and specific positions (mobile apps). All this process takes a lot of time and therefore the essential goal of optimizing time by measuring the garment is lost. How to make automatic garment measurement as versatile and accurate as possible is also one of the most important issues for the autonomous retrieval of garments information. In this work, we focus on the challenge of automatically measuring the hanging garments (in this particular study, on the mannequin), without being restricted by space, background requirements, shooting distances or additional tags needed for measurements. The aim of this research is to create a solution by implementing an automatic clothing segmentation and measuring algorithm that would let us not only separate clothing into different groups but also measure their basic measurements such as distance between shoulders, length of a sleeve, etc. Identifying the main problems and limitations of both objectives—accurate segmentation and measurement—is also an essential task, as it can provide insights and avenues for further research.

2. Materials and Methods

Initially, in this study, 683 images of clothing were collected including different types of garments. As one of the objectives of this study is to investigate the feasibility of automatic garment sizing with household photographs that can be uploaded to different platforms (e.g., second-hand clothing platforms) photos taken under different conditions and with different mannequins or hanging on a hanger have been included. There are solutions for overcoming the effects of lighting and occlusion, but they are usually developed for specific groups of objects [12,23] or noises [24].

Initial experiments have shown that the distance from the camera to the object is quite an important aspect in the calculation of the size of the garment and can lead to a measurement error of 10 to 15 cm. This problem can be solved by adding a standard-sized object (e.g., a bank card) to the scale, but some requirements arise here as well. These include reflections, edge identification problems when the tag blends with the garment, etc. Many problems are caused by wrinkled, ruffled clothes, such as those that are the same colour as the background. Depending on the pose and condition of the garment, and the angle of the camera, difficulties arise, e.g., measuring the width of the sleeves, as they can look much narrower than they really are. In addition, it has been observed that the quality of the photos and the context vary considerably, including differences in shooting and lighting conditions, camera resolutions and clothing shooting angle, the appearance of multiple garments in one photo, redundant objects in the photo, more than 20 different types of clothing, etc. Given all these challenges, several iterations of data cleaning were carried out to improve the quality of the dataset. First of all, the variety of garments has been reduced to 13 classes according to [25], resizing all photos to 224×336 resolution retains information about the boundaries of the garment and reduces the resource requirements for size prediction methods. Finally, in order to have a stratified dataset, 330 images of clothing were selected including the same amount of blazers, skirts and dresses. This dataset has been divided into three parts: 70% for training, 15% for validation and 15% for testing.

More advanced exploration of the dataset has revealed that the application of classical methods to the segmentation task has a lot of potential, so it is appropriate to test other algorithms before employing deep learning architectures. The simplest way is therefore to use image processing techniques to extract information about the edges of the garment so that the location of the garment in the image can be determined and the size of the garment can be measured using an additional algorithm. The second way is to use the deep learning architecture (e.g., UNet family [26] model) to create the mask of the garment in the photo that would extract the position of the garment. Then using a classifier to determine the type of garment, pass the collected information to the specific algorithm to perform the final garment measurements prediction. Instead of a specific algorithm, all essential garment points can be predicted using deep learning models, whose provided output results allow calculating the distances between points and determining garment measurements. However, the identification of the most appropriate solution must focus not only on the accuracy but also on the complexity of implementation, computational resources and robustness to different environments.

2.1. UNet Model-Based Extraction of Contours of the Garments' Shape

Various image segmentation algorithms have been developed, but more recently, the success of deep learning models in various vision applications has led to a large number of studies on the development of image segmentation methods using deep learning architectures. U-Net is a convolution neural network [27] originally proposed for medical imaging segmentation, but various research has shown its potential for other segmentation tasks as well [28–30]. The U-Net network is fast, can segment a 512×512 image without the need for multiple runs and allows for learning with very few labelled images. This is an important feature in our case because the dataset is relatively small. Moreover, in this research, a network and training strategy that relies on the strong use of data augmentation is required in order to use the available annotated samples more efficiently.

As UNet model segmentation involves a masking process, therefore all masks were created using Open Source VGG Image Annotator version 2.0.10. The exported annotations were used to create a black and white image by drawing polygons for which there was only one in this study dataset.

In order to improve the segmentation results, the initial dataset has been expanded including the DeepFashion2 dataset [25]. There are no accurate measurements in the dataset, but there is clothing segmentation, which can improve the accuracy of the UNet model by defining the segmentation area and removing artefacts due to different environments. DeepFashion2 is a large dataset of photos collected from various fields and it contains 491,000 images of 13 popular clothing categories from commercial stores and consumers. It contains more than 800 K photos enabling it to extract dense landmarks and masks. However, in this study we aim to determine the size of the garment; therefore, this dataset can be used for segmentation purposes only. A filtering procedure was carried out to select suitable photos of clothing. Poor quality photos where the garment is covered, the garment is worn by a person, the garment is taken from the side or the back, there are several garments in a single photo, etc. were removed. A total of 18,000 images with only one garment visible from the front were identified as appropriate. The DeepFashion2 dataset does not provide clothing masks so using landmarks data we have developed an algorithm that creates masks. Finally, we obtained data similar to the original dataset that could be used to train the UNet model.

Few experiments with UNet models have been carried out in order to increase the segmentation results. First, the pre-trained UNet model (see Figure 1) with the classical structure has been employed and supplementary experiments have been performed with additional datasets, namely DeepFashion2 and Carvana datasets. However, this approach did not work well and the results were poor. Next, modified UNet architectures with the increased number of layers (added additional encoding and decoding layers) have been employed. Different size models pretrained with our small dataset have shown superiority compared to the classical UNet structure pre-trained with additional datasets. The included UNet family architectures, which differ in depth and in the different datasets on which they have been trained, are listed in Table 1.

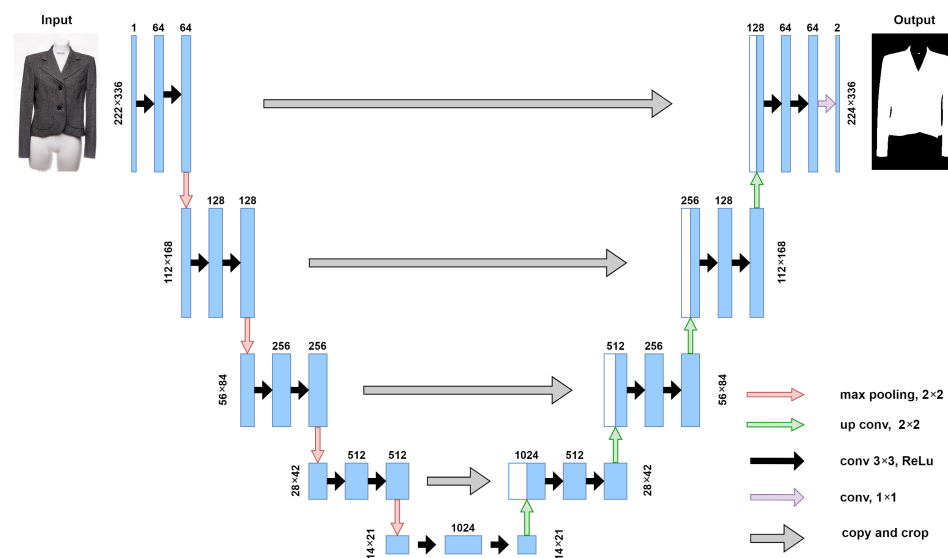


Figure 1. UNet model architecture used for clothes segmentation task.

Table 1. UNet models with different pre-trained datasets.

UNet Model	Pretraining Dataset
UNet	DeepFashion2+ our dataset
UNet	Carvana + our dataset
UNet	our dataset
UNet 128 × 128	our dataset
UNet 256 × 256	our dataset
UNet 512 × 512	our dataset

The different models were compared on the basis of segmentation results. Image segmentation aims to classify each pixel of an image as representing a certain class, e.g., could be a garment, a mannequin, or a background in our case. There may be more or fewer classes depending on the task. Specific segmentation metrics (usually Pixel accuracy, Dice and Jaccard coefficients) are used to measure the success of the model [31,32]. Experimental results in this study were compared using the Dice similarity coefficient. The Dice coefficient, also called the overlap index, is the most common metric evaluating segmentation results [33]. This coefficient was used in order to evaluate the overlap between the predicted mask and the manually-labelled ground truth mask. During model training, the coefficient was used to calculate the loss value. The Dice value was calculated after the model received a predicted mask. Dice indices are bounded between 0 (when there is no overlap) and 1 (when predicted and true masks match perfectly). The Dice coefficient is $2 \times$ the overlap area divided by the total number of pixels in both images. In terms of the confusion matrix, the metrics can be reformulated into true/false positives/negatives statements:

$$\text{Dice} = \frac{2|X \cap Y|}{|X| + |Y|} = \frac{2TP}{2TP + FP + FN} \quad (1)$$

where $|X|$ and $|Y|$ are the cardinalities of the two sets (i.e., the number of pixels in each area), X is the ground truth mask, while Y represents the predicted mask. The intersection ($X \cap Y$) is comprised of the pixels found in both the prediction mask and the ground truth mask. TP —true positives pixels that exactly match the annotated ground truth segmentation, FP —false positives pixels that are segmented incorrectly, FN —false negatives pixels that have been missed.

2.2. Garment Key Points Detection

The segmentation process is only the first step in determining the measurements of garments. Which measurements are relevant depends on the type of garment: for a skirt, for example, it is important to know the waist and length, but for a shirt or jacket you should also measure the length of the sleeves, the width of the shoulders, etc. Segmentation should then be followed by a classification task which allows identifying the necessary dimensions. Finally, once the type of garment has been identified, it is possible to identify the measurement key points which is the most challenging task and directly depends on segmentation results. An incorrect segmentation result can reduce the accuracy of key points detection or stop the algorithm altogether.

In this study different key points detection algorithms have been created. The principle of the developed algorithms is to identify the edge points of the garment in the image that are necessary to determine the relevant dimensions in certain areas. In Figure 2 the basic key points for blazers, skirts and dresses are provided. For the blazers, it is important to capture the left and right shoulders' fall bottom points as the distance between these points (1 and 8) is the measure of shoulders width. To measure the total length of the blazer we need to find the midpoint of the shoulder strap and the midpoint of the bottom of the blazer. These points are captured on both sides, left (2,11) and right (7,10). Finally, the average value of these distances is calculated. Points 3 and 6 are used to determine where the shoulder line starts. To measure the length of the left and right sleeves we have the

shoulders' fall bottom points (1,8) and the mid-points on the bottom of the sleeves (12,9). To measure the neck width, points (4) and (5) are included. It can be noticed from Figure 2 that for the dresses and skirts fewer key points are required. For the sleeveless dresses, 6 key points are included in order to measure the shoulder width, waist and total length. In addition, the width of the bottom of the dress can be calculated from points 5 and 4. To obtain a measurement for the skirt only four points are included: (1) top left point; (2) top right point; (3) bottom right point and (4) bottom left point. These points are sufficient to calculate the skirt waist, overall bottom length and width.

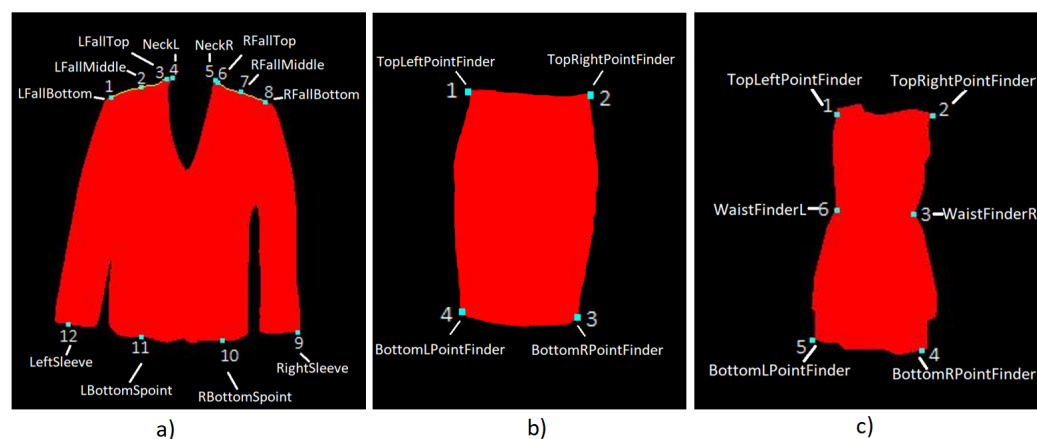


Figure 2. The basic key points of measurements for different type of garment: (a) blazer with 12 key points, (b) skirt with 4 key points and (c) dress with 6 key points.

Some general points that do not depend on the type of garment are included such as angle finder point, middle left and right points, bounding box, edge points, etc. The algorithms of such point detection can facilitate the finding process of the main key points. For example, the algorithm “ClothBoundariesCalculator” captures information on the position of the garment and the points of the bounding box. In this study we have performed experiments with the three types of garments, thus 26 key-point estimation algorithms have been created. The decisions in the algorithms are based on a threshold value applied for the pixel. For instance, Algorithm 1 represents the pseudo-code of the algorithm that finds the left and right side of the neckline with the generated mask and corresponds to the blazer’s key point (4).

This algorithm requires initial data on the position of the garment. All starting points have (x, y) coordinates indicating their position on the garment. The top point of the garment is defined as $T_{x,y}$. The middle point of the garment on the left side is annotated as $L_{x,y}$ and the right as $R_{x,y}$. The upper point of the collar on the left side is annotated as $CL_{x,y}$ and the right as $CR_{x,y}$. The middle of the garment is defined as point $M_{x,y}$. The set containing the garment mask is defined as $G_{x,y}$. These points are resolved using additional algorithms which must be created separately.

To find the midpoint between the extreme point of the shoulder and the highest point of the neck, the algorithm needs to find the highest point on the neck collar first. The algorithm begins the search of the neckline from the middle of the garment with the aim to find the smallest x and smallest y coordinates for the left neck collar and the largest x and smallest y for the right collar which are return from algorithm as r point. The algorithms for finding measurement points for garments consist of 4 parts:

- Identification of bounding box and the outermost points of the garment contour;
- Detection of garment’s angles, shapes, and changes in (x, y) coordinates;
- Key points prediction based on auxiliary algorithms that find the desired location;
- Final prediction that sets the sensitivity factor for developed algorithms to adapt to the type of the garment. Then the pixel to cm ratio is calculated and the dimensions of the garment are produced.

Algorithm 1: Pseudo-code for left and right neck line identification.**Left**

```

1.   $y = M_y$ 
2.  WHILE  $y > T_x$ 
3.     $x = M_x$ 
4.    WHILE  $x > L_x$ 
5.      IF  $(x, y) \in G$ 
6.        IF  $CL_x = 0$  OR  $y < CL_y$ 
7.           $r = (x, y)$ 
8.           $x-$ 
9.           $y-$ 
10. return  $r$ 

```

Right

```

1.   $y = M_y$ 
2.  WHILE  $y > T_x$ 
3.     $x = M_x$ 
4.    WHILE  $x < R_x$ 
5.      IF  $(x, y) \in G$ 
6.        IF  $CR_x = 0$  OR  $y < CR_y$ 
7.           $r = (x, y)$ 
8.           $x+$ 
9.           $y-$ 
10. return  $r$ 

```

3. Results*3.1. Extraction of the Contours of the Garment Shape**3.1.1. Edge Detection Techniques*

The experiments performed in this study aim to determine whether widely used methods for edge detection can be applied to determine the edges of a garment. The photos collected during the study have a clear edge with the environment, but the main drawback is that these edges may be on the garment itself, which causes a problem that will require processing of the results. There is also a mannequin in each photo and there may be other objects. The detection of the edges of extraneous artefacts is a side factor complicating the use of the resulting edges. The common image contour detection pipeline includes the conversion of an RGB image to a grayscale format, a binary threshold setting (which converts the image to black-and-white based on a threshold value and highlights objects of interest) and finally contours identification. The latter step uses a method that can set the boundaries of the uniform intensity form. To find contours, we can also use the Canny edge detection algorithm [34]. The Canny algorithm consists of five main steps. As the algorithm is based on greyscale images, it is necessary to convert the image to greyscale before performing all of the steps. The first step is to reduce the noise by performing a Gaussian blurring on the image. The second step is to determine the intensity gradients using edge detection operators. In our case, the Sobel filter has been applied to get the intensity and edge direction matrices. The third step involves non-maximum suppression to thin out the edges. This function works by finding the pixels with the highest value in the edge directions. If the pixels are not part of a local maximum, they are set to zero (converted to a black pixel), otherwise, they are not modified. Because the resulted image after non-maximum suppression is not perfect (there is some noise in the image) double thresholding is applied in a fourth step. All pixels with a value higher than the predefined high threshold value are considered to be a strong edge and are likely to be edges. All pixels with a value less than the predefined low threshold value are set to 0. Values between the low and high threshold values are considered “weak” edges, in other words, it is not clear whether they are real edges or not edges at all. Finally, the fifth step, based on the threshold results, invokes edge tracking by hysteresis, which performs a transformation of weak pixels. “Weak” edges connected to strong edges are treated as true edges and those not connected to strong edges are removed. The results of canny edge detection with a predefined threshold are provided in Figure 3. Different clothing types and colours were used in the experiment.

Another very popular edge detection technique is Sobel [35], which is a gradient-based algorithm including manipulations to the x and y derivatives. Sobel algorithm converts the image into grayscale and employs two 3×3 kernels which are convolved with the original image to calculate approximations of the derivatives for horizontal and vertical changes. The Gaussian filter is used for reducing noise that makes blurred images.

Figure 3 shows that edge detection using the Canny or Sobel algorithms is quite precise despite the type, background and type of the clothing, but the edges of the mannequin are detected along with the garment. Moreover, the shadows visible in the original images are depicted as a double contour line in the resulting image. Changing the algorithm parameters did not provide the required result either, since we need to find the edges of the outer shape of the garment.



Figure 3. Results of different contour detection techniques: Canny algorithm with predefined threshold values, Sobel algorithm and K-means based thresholding algorithm providing edge and mask images.

3.1.2. K-Means Clustering Approach

Many clustering methods have been developed for various purposes, usually unsupervised classification. K-means clustering is one of the instances of such type of algorithm that aims to divide N observations into K groups, with each observation belonging to the cluster with the closest mean. A cluster is a collection of data points that are clustered together due to similarities. For the image segmentation, including different colour spaces (RGB or L^*a^*b) clusters refers to different image colours [36,37]. The algorithm aims to minimize the Euclidean distance between observations and centroids. Single or few iteration thresholds can be used to segment the image adaptively and to filter the noise [38]. In general, the goal of the K-means approach is to find parameters that filter out the influence of the background on the image, so that the final segmentation result, the target object, is more accurately distinguished.

For segmentation of garment, K-means is used to identify the three most dominant colours in the image (e.g., background, mannequin and dominant colour of the garment) and calculate the thresholds in order to generate a binary image. The value of K should be specified in advance, and the correct selection of this value is not always straightforward. Values of K in the range from 2 to 6 have been experimentally tested. The best results were obtained when $K = 2$ or $K = 3$. However, with the given data the results are 6.8%

better (in terms of mask accuracy) when $K = 3$. Therefore, based on the three values obtained (centroid-based thresholds), all pixels in the image are converted to black and white. It is observed that the resulted image after clustering is still noisy. The noise is reduced using a median filter. To smooth the image a few iterations of morphological operations—dilation and erosion are applied. These techniques are used not only for noise reduction but also for identifying holes in the image (which is very relevant when we have mottled clothes), isolating individual elements and joining disparate elements in the image. In our case, we use structured element matrices of size 5 (kernel size = (5, 5)) and we performed three iterations of both operations. Increasing the number of iterations (up to 5) is relevant for multi-coloured garments (as it fills the holes and creates a continuous mask), but may have a negative effect on single-coloured garments. Contours are detected using the concept of Canny edge detection. An iteration process (“cleaning up”) of the remaining weak edges was performed setting them to zero. Finally, as a result, an image mask is provided (see Figure 3). Although the result with dark-coloured clothes looks really promising (the mannequin is excluded as well), the algorithm performs badly with multi-coloured fabrics, and especially with light-coloured garments where the garment is hardly distinguishable from both the background and the mannequin (Figure 3). Ambient shadows also strongly influence the resulting images of the K-means algorithm.

3.2. UNet-Based Segmentation

Figure 4 represents a few garment image segmentation results based on deep learning models described above. From the predicted masks we can see that UNet models pre-trained with DeepFashion2 and Carvana dataset have the lowest accuracy compared to other models. In the segmentation results, we can see that the clothing lines in the photos are not preserved and the entire shape of the garment is lost. However, as with all models, the segmentation of the skirt shows very good results due to the bright red colour. This indicates that the high contrast with the environment in the photos is an important factor. The best results in maintaining bright and smooth boundaries of clothing are obtained with UNet models including additional encoding and decoding layers.

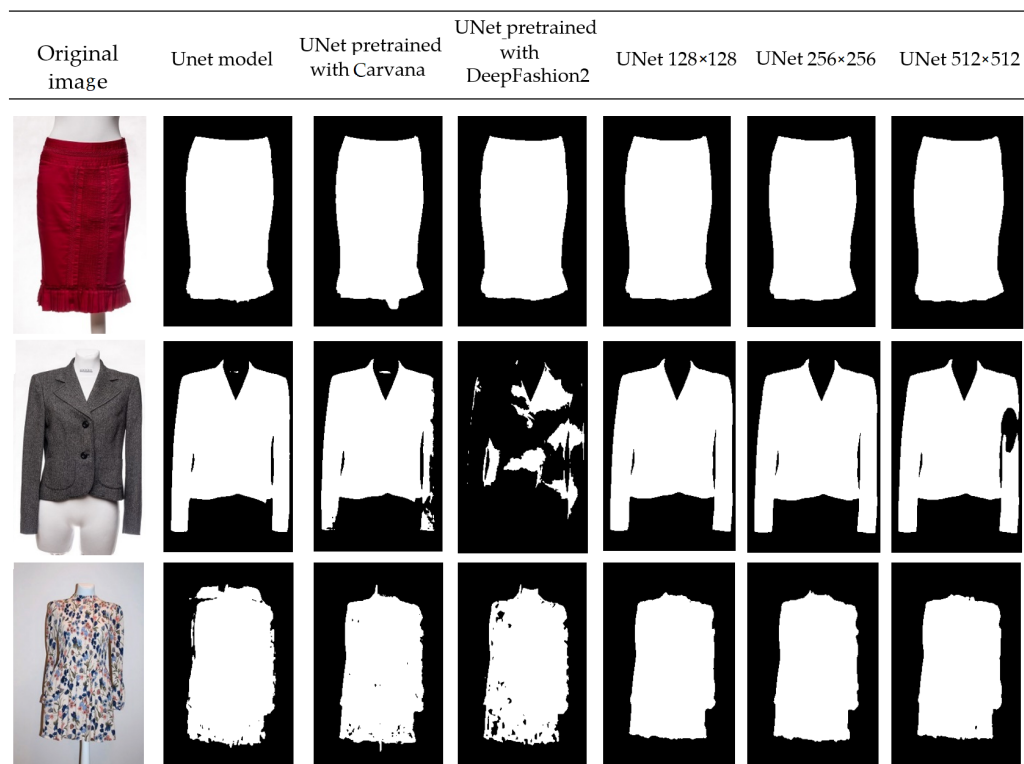


Figure 4. Different Unet models’ segmentation results for selected clothes: skirt, jacket and dress.

Dice values show that used deep learning models were highly volatile during the training phase and Dice coefficients ranged from 0.02 to 0.979. Average and maximum values of Dice coefficient were calculated estimating the results of five experimental runs (see Table 2). Models that were trained with the Carvana dataset or DeepFashion2 showed no significant improvement in accuracy. The UNet 128×128 model with a maximum Dice accuracy of 0.979 demonstrated the highest accuracy results obtained through all five runs including augmentation. The average value of the Dice coefficient reaches 0.917 with augmentation and 0.899 without augmentation. However, it can be concluded that in choosing a UNet model, the variation in model depth should be rationally evaluated, as the classical UNet uses much less computational resources than models with additional layers, but compared to UNet 128×128 its average accuracy according to the Dice value is 0.113 lower including augmentation and 0.047 without augmentation.

Table 2. Different UNet models' 5 run validation results including maximum and average values of Dice coefficients.

UNet Model	With Augmentation		Without Augmentation	
	MAX Dice	AVG Dice	MAX Dice	AVG Dice
UNet	0.918	0.804	0.919	0.852
UNet (DeepFashion2)	0.906	0.857	0.825	0.824
UNet (Carvana)	0.891	0.835	0.879	0.827
UNet 128×128	0.979	0.917	0.943	0.899
UNet 256×256	0.976	0.818	0.922	0.818
UNet 512×512	0.971	0.865	0.906	0.855

One of the five training processes of all included models during 50 epochs is shown in Figure 5 providing the variation of Dice coefficient value throughout the process. It can be noticed that more stable results are gained using deeper UNet models, observing more significant Dice value variations only until the 15th epoch, while others had larger fluctuations around 0.2. The classical UNet model achieved an average DICE value of 0.860 without augmentation and 0.800 with augmentation. However, it has only stabilized in the last four epochs of the training process (see Figure 5). DICE values were calculated by estimating epochs from 10 to 50 and excluding the "warming period" of the first 10 epochs.

3.3. Obtained Measurement Results

With accurate segmentation results, which means the garment is accurately separated from the background, we can predict measurements using proposed algorithms for the key point detection (see Figure 6). Table 3 shows the obtained results of garments' measurements providing mean absolute error (MAE). The best accuracy results have been achieved for the dresses with an average of 0.747 cm measurement error, when total length, waist and shoulders dimensions are considered. For dresses, the largest errors are observed in the measurements of the overall length of the dress. However, the predicted waist measurements are very similar to the actual ones, with an average error of only 0.3429 cm. Prediction results of waist dimensions are relatively precise for the skirts as well, because the MAE is 0.421 cm.

Table 3. Garment measurement errors given in centimetres.

	Total Length	Waist	Shoulders	Sleeves	Average Error
Dresses	1.113	0.343	0.783	-	0.747
Blazers	0.903	-	1.826	0.652	1.127
Skirts	1.650	0.421	-	-	1.012

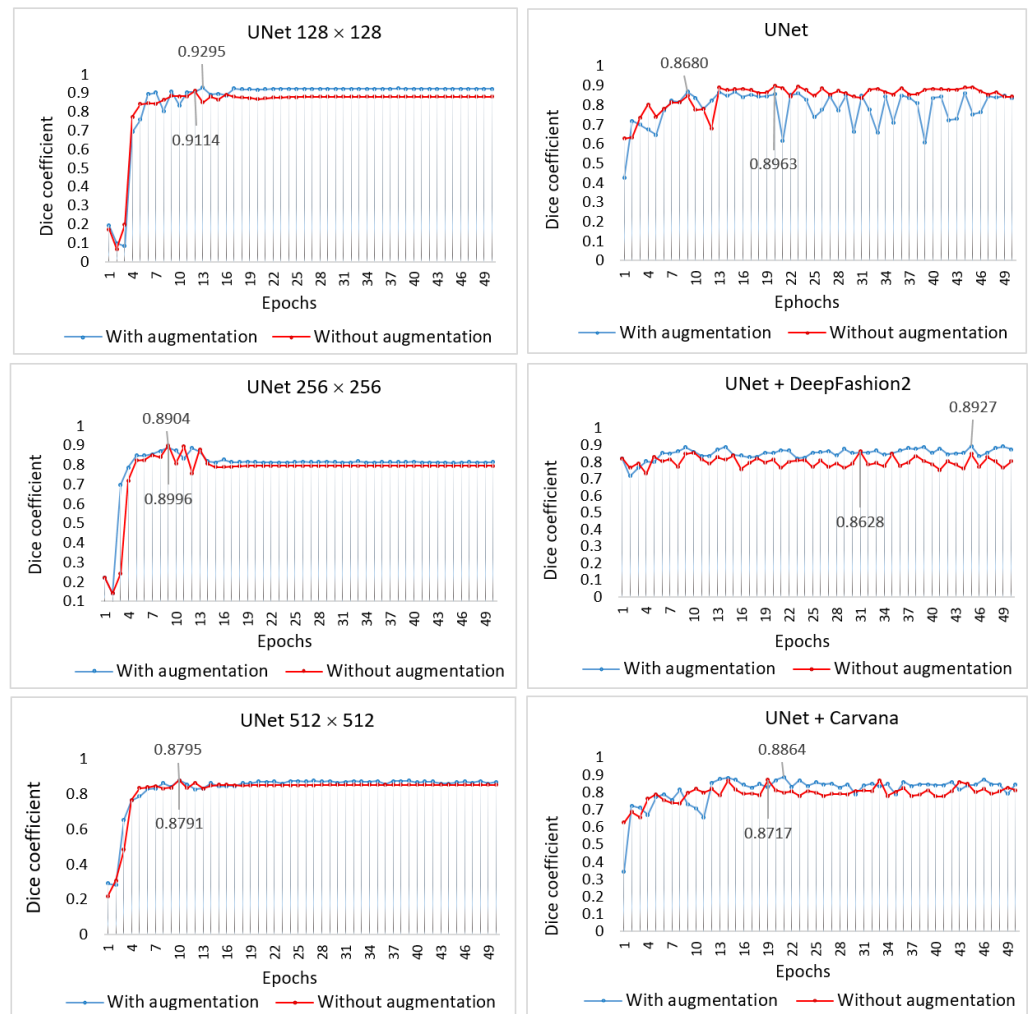


Figure 5. Dice coefficient value variation during the training process of different UNet architectures (with augmentation and without denoted maximum coefficient values).

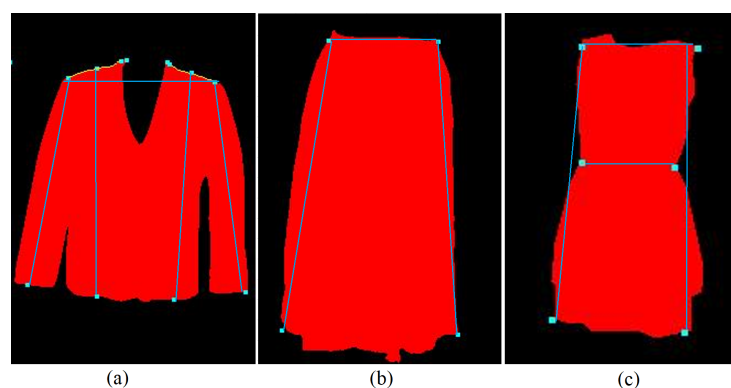


Figure 6. The instances of measurements predicted key points for different types of garments: (a) blazer, (b) skirt and (c) dress.

More difficulties were encountered in the measurement of blazers dimensions, with the largest errors (MAE = 1.826) predicting the width of shoulders. However, the sleeves are measured quite accurately (MAE = 0.652), even though the same blazers' key point (1) is used in the algorithm. However, it should also be considered that such errors may occur due to measurement inaccuracies, as the algorithm for converting pixels to centimetres with different coefficients gives more accurate results. The highest errors in the measurement of

the jacket shoulders can also be explained by the complexity of the jacket image set, which includes some cases where the shoulders are difficult to determine due to the small size of the mannequin, and the colour of the jacket, etc. A similar situation is seen with the skirts' length measurements, where about 15% of them have tassels, a translucent top layer, a crooked cut and other issues (see Figure 7).



Figure 7. Examples of specific cases: (a) in the blazers dataset regarding to shoulders line and (b) in skirts dataset regarding to the bottom line.

4. Discussion

For more extensive experimental purposes, three other convolutional neural network (CNN) models have been examined, namely MobileNetV2 [39], ResNet50 [40] and DeepPose [41] (see Figure 8). All models were trained with the same dataset and for the same time. Some models have employed pretrained weights while others used fixed, not trainable backbone part of neural network. Table 4 presents the experimental results including similarity metrics. Therefore, some widely accepted quantitative metrics are used in the study to measure the similarity between two images [42,43]. Mean squared error (MSE) is commonly used to estimate the difference between two images by directly computing the variation in pixel values. The smaller value of MSE represents better similarity. Its value is defined as:

$$\text{MSE} = \ell(x, y) = \text{sum}(L), \quad L = \{l_1, \dots, l_N\}^T, \quad l_n = (x_n - y_n)^2, \quad (2)$$

where N is the batch size, x and y are tensors of arbitrary shapes with a total of n elements each and the reduction is the sum operation.

From the results, we can notice that the MobileNetV2 fully trained model has provided better results (MSE loss = 0.009 and Dice = 0.985) compared to the fixed MobileNetV2 model using pretrained data. The worst results have been achieved using DeepPose with MSE loss = 0.039, Dice = 0.935, Dice loss = 0.065 and RMSE = 0.190.

Table 4. Values of accuracy metrics.

Model	MSE Loss	Dice	Dice Loss	RMSE
MobileNetV2 (fixed)	0.032	0.944	0.056	0.186
ResNet50 (fixed)	0.032	0.948	0.052	0.181
MobileNetV2	0.009	0.985	0.015	0.095
ResNet50	0.022	0.962	0.038	0.150
DeepPose	0.039	0.935	0.065	0.199

Clothing segmentation allows identifying the garment's location and distinguishing it from other objects in the photos. However the key points detection approach can be used not after the segmentation, but instead of segmentation, thus omitting one step and facilitating the calculation of dimensions [44]. Determining the coordinates of tens of points is an easier task than classifying all the pixels in a photo. This allows the use of a smaller artificial neural network model and an output layer with fewer neurons. A few instances of the results of predicted key points positions are provided in Figure 8.

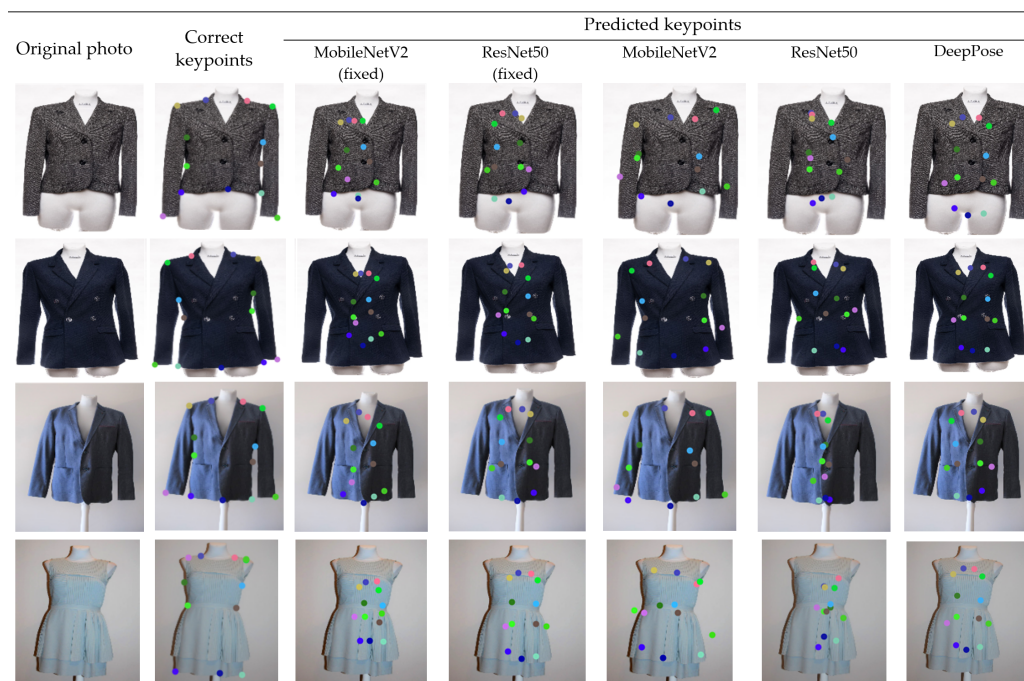


Figure 8. The comparison of different CNN architectures for keypoints detection.

4.1. Limitations

The main drawback of our solution is that the developed algorithm cannot adapt to the different conditions that occur on exceptional terms when a garment mask is created poorly. Uneven edges and unfilled cavities can corrupt the results or completely stop the operation of the algorithm, as finding some points is necessary to calculate the final results for all dimensions of the garment. For this reason, we believe that a multi-level prediction could be more appropriate. The first step in creating a mask is to set the points according to the type of clothing and a simple algorithm to determine the distance between the points. This allows calculating the dimensions of garments if garments are photographed at the same distance and with the same camera. Since the ratio of pixels to centimetres does not change in the photos, the only thing needed is to measure a constant or train a neural network model. However, experimentation has shown that it is difficult to ensure exactly the same experimental conditions, such as the distance, angle, lens and resolution of the camera. One possible solution is to capture the garment together with an object of a fixed size. This could be a credit card, a geometric shape of a certain size, for example, a square, etc. These objects should also be recognized in the image and used for scaling. However, even universally sized objects (e.g., bank cards) can have different colours, reflected in the photograph, blending in with the clothing, which causes additional problems. For these reasons, printed templates are often used for scaling. They are easy to recognize in photographs and can be used to calculate an accurate scale regardless of the camera, the shooting angle and distance.

The other method of estimating the scale of the object is with an algorithm that utilizes continuous frames to estimate the camera's pose [45]. This method has been implemented in smartphone apps (e.g., iPhone) thus users can determine object size and scale. However, this approach requires a video or some other references, therefore it is not suitable for scale estimation from clothing images.

The UNet-based solution developed in this study removes extraneous artefacts visible in the image and solves the problem of varying environmental conditions. UNet collects information about the garment position in the photo, which is used in the algorithm as a binary array. With this data, the algorithm can easily identify points on the garment that can be used to calculate the dimensions of the garment. The advantage of this solution is that it is possible to clearly identify the problems that cause the model to predict garment

dimensions poorly. Such a division between mask prediction and algorithm makes it possible to achieve high accuracy, expandability and wide applicability. It is a flexible solution that does not require strict environmental conditions, and measurements can be made using different mannequins, photographing clothes on a person, hanging on a hanger or lying down.

4.2. Human Measurement Error

Writing down detailed information about each garment element is manual and time-consuming labour. By correctly identifying main clothing parts, such automated segmentation and measuring system could even improve over human-made measurements. Our empirical experiment has shown that human measurement error can be up to 3 cm, depending on the specific areas of the garment being measured. The experiment involved 20 people aged 22–58 years. Each of them was asked to measure two types of clothes. For skirts, they were asked to provide two dimensions—waist and skirt length, and for men’s jackets—shoulder width, overall jacket length and sleeves length. Even with prior instruction on how and what to measure, errors still occur, for example, up to 3.02 cm in the case of a jacket length measurement (see Figure 9).

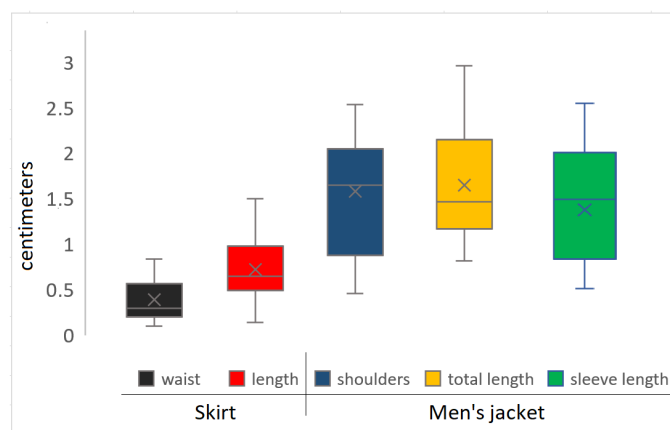


Figure 9. The error of manual measurement for two different types of clothes: skirts and men’s jackets.

5. Conclusions

This paper addresses the problem of automatic measurements of garments dimensions. The proposed solution consists of deep learning-based garment segmentation and the detection of key points needed to measure the main garment dimensions. Different UNet family architectures have been employed for segmentation tasks. The UNet 128×128 model with a Dice accuracy of 0.977 has demonstrated the highest accuracy results compared with other UNet models and showed the superiority over the UNet models pre-trained with the additional datasets. The key points detection process has been performed on the predicted masks obtained using the Unet 128×128 model. Separate algorithms (for the blazers, skirts and dresses) have been developed in this research to identify general and specific garment key points enabling us to measure the dimensions of the garment. Automatic measurements experiments including three types of garments (blazers, skirts and dresses) have resulted in an average 1.27 cm measurement error for the prediction of the basic measurements of blazers, 0.747 cm for dresses and 1.012 cm for the skirts. The results are promising, given that in the industry a measurement error of up to ~ 2 cm is acceptable, while human measurement error can be up to 3.02.

The comparison of existing solutions is quite difficult due to the purpose of the proposed solution itself, diverse environmental conditions, individual datasets or evaluation metrics. Commercial solutions currently available on the market aim to make the process of purchasing clothes easier while reducing the number of returns. Therefore, the first and most important step is to obtain accurate body measurement data by integrating deep learning algorithms and other artificial intelligence techniques. The second objective of

such systems (mobile apps) is to provide personalized clothing sizing recommendations to help eliminate sizing problems. Such systems can be called smart shopping assistants with quite clear objectives, including sustainability. The solution proposed in this study focuses on the automated extraction of garment information, i.e., the recognition of the type of garment and the accurate measurement of its dimensions, without being tied to the positioning of the garment (lying down, on a mannequin or on a hanger). One of the key goals of this research is to enable less-standardized garment photography, in contrast to current garment measurement systems which require fixed position setup, calibration, and/or dedicated infrastructure to ensure small error (~0.318 cm). Although the quality of the photos is certainly important, our solution does not require the highest quality professional photos, so it can be used both in the industry and on online platforms selling second-hand or new clothes (e.g., “Ebay”, “Vinted”). The results presented in this study are related to the most important and challenging aspect of garment information identification—automated garment dimension measurement. However, the potential for extending such a solution is significant. A further objective is to automate the manual entering of all the information about the garment, such as what is the type, colour, size, etc. Moreover, additional solutions could be added to retrieve information from the label, which includes information on fabric composition, garment size and brand. Colour identification is one of the simplest tasks, but it is possible to develop a more sophisticated solution based on unsupervised learning algorithms to automatically identify a few dominant colours (in the case of a multi-coloured or patterned garment), and incorporate an adaptive and broad palette of colours, rather than a fixed and narrowly defined range of colours.

Author Contributions: Conceptualization, A.P.-T.; data curation: R.S.-Z., A.P.-T., I.L.-B. and E.N.; investigation, A.P.-T., V.D., E.N. and R.P.; methodology, A.P.-T., I.L.-B., E.N. and R.P.; software, A.P.-T., E.N., R.P. and I.L.-B.; resources A.P.-T., R.S.-Z., I.L.-B. and E.N.; validation A.P.-T., E.N., V.D. and R.P.; writing—original draft, A.P.-T., R.S.-Z., I.L.-B., V.D., E.N. and R.P.; writing—review and editing, A.P.-T., E.N., R.P. and I.L.-B.; supervision, A.P.-T.; funding acquisition, A.P.-T. and R.S.-Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by MB (small partnership) NOSELFISH within the framework of the European Union-funded project “Technical Feasibility Study for the Adaptation of an Overlapping Object Detection and Classification System for the Identification of Clothing Characteristics” (No. 01.2.1-MITA-T-851-01-0201).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data are not publicly available due to commercial sensitivity and data privacy.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kumar, S.N.; Fred, A.L.; Varghese, P.S. An Overview of Segmentation Algorithms for the Analysis of Anomalies on Medical Images. *J. Intell. Syst.* **2020**, *29*, 612–625. [[CrossRef](#)]
2. Cao, C.; Liu, F.; Tan, H.; Song, D.; Shu, W.; Li, W.; Zhou, Y.; Bo, X.; Xie, Z. Deep Learning and Its Applications in Biomedicine. *Genom. Proteom. Bioinform.* **2018**, *16*, 17–32. [[CrossRef](#)] [[PubMed](#)]
3. Tang, Y.; Zhu, M.; Chen, Z.; Wu, C.; Chen, B.; Li, C.; Li, L. Seismic performance evaluation of recycled aggregate concrete-filled steel tubular columns with field strain detected via a novel mark-free vision method. *Structures* **2022**, *37*, 426–441. [[CrossRef](#)]
4. Wu, F.; Duan, J.; Chen, S.; Ye, Y.; Ai, P.; Yang, Z. Multi-Target Recognition of Bananas and Automatic Positioning for the Inflorescence Axis Cutting Point. *Front. Plant Sci.* **2021**, *12*, 705021. [[CrossRef](#)]
5. Gui, J.; Fei, J.; Wu, Z.; Fu, X.; Diakite, A. Grading method of soybean mosaic disease based on hyperspectral imaging technology. *Inf. Process. Agric.* **2021**, *8*, 380–385. [[CrossRef](#)]
6. Chen, Z.; Wu, R.; Lin, Y.; Li, C.; Chen, S.; Yuan, Z.; Chen, S.; Zou, X. Plant Disease Recognition Model Based on Improved YOLOv5. *Agronomy* **2022**, *12*, 365. [[CrossRef](#)]
7. Krishnamoorthy, N.; Prasad, L.N.; Kumar, C.P.; Subedi, B.; Abraha, H.B.; Sathishkumar, V.E. Rice leaf diseases prediction using deep neural networks with transfer learning. *Environ. Res.* **2021**, *198*, 111275. [[CrossRef](#)]

8. Yang, J.; Li, S.; Wang, Z.; Dong, H.; Wang, J.; Tang, S. Using Deep Learning to Detect Defects in Manufacturing: A Comprehensive Survey and Current Challenges. *Materials* **2020**, *13*, 5755. [CrossRef]
9. Pan, C.; Schoppe, O.; Parra-Damas, A.; Cai, R.; Todorov, M.I.; Gondi, G.; von Neubeck, B.; Bögürücü-Seidel, N.; Seidel, S.; Sleiman, K.; et al. Deep Learning Reveals Cancer Metastasis and Therapeutic Antibody Targeting in the Entire Body. *Cell* **2019**, *179*, 1661–1676.e19. [CrossRef]
10. Xu, Y.; Hosny, A.; Zeleznik, R.; Parmar, C.; Coroller, T.; Franco, I.; Mak, R.H.; Aerts, H.J. Deep Learning Predicts Lung Cancer Treatment Response from Serial Medical Imaging. *Clin. Cancer Res.* **2019**, *25*, 3266–3275. [CrossRef]
11. Vidas, R.; Agne, P.T.; Kristina, S.; Domas, J. Towards the automation of early-stage human embryo development detection. *Biomed. Eng.* **2019**, *18*, 1–21. [CrossRef]
12. Tan, F.; Xia, Z.; Ma, Y.; Feng, X. 3D Sensor Based Pedestrian Detection by Integrating Improved HHA Encoding and Two-Branch Feature Fusion. *Remote Sens.* **2022**, *14*, 645. [CrossRef]
13. Wang, J.; Yu, X.; Liu, Q.; Yang, Z. Research on key technologies of intelligent transportation based on image recognition and anti-fatigue driving. *EURASIP J. Image Video Process.* **2019**, *33*. [CrossRef]
14. Gabas, A.; Corona, E.; Alenya, G.; Torras, C. Robot-Aided Cloth Classification Using Depth Information and CNNs. In *Articulated Motion and Deformable*; Perales, F.J., Kittler, J., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 16–23. [CrossRef]
15. Nayak, R.; Padhye, R. *1-Introduction to Automation in Garment Manufacturing*; Automation in Garment Manufacturing; Woodhead Publishing: Boca Raton, FL, USA, 2018; pp. 1–27. ID: 317722. [CrossRef]
16. A Report: Study of the Automatic Garment Measurement, Robocoast, Leverage from EU 2014-2020, Aarila-Dots Oy. 2019; pp. 1–13. Available online: https://new.robocoast.eu/wp-content/uploads/2020/09/Feasibility-study-Automatic-garment-measurement_Aarila-Dots.pdf (accessed on 5 February 2022).
17. Xiang, J.; Dong, T.; Pan, R.; Gao, W. Clothing Attribute Recognition Based on RCNN Framework Using L-Softmax Loss. *IEEE Access* **2020**, *8*, 48299–48313. [CrossRef]
18. Ihsan, A.M.; Loo, C.K.; Naji, S.A.; Seera, M. Superpixels Features Extractor Network (SP-FEN) for Clothing Parsing Enhancement. *Neural Process. Lett.* **2020**, *51*, 2245–2263. [CrossRef]
19. Li, C.; Xu, Y.; Xiao, Y.; Liu, H.; Feng, M.; Zhang, D. Automatic Measurement of Garment Sizes Using Image Recognition. In *Proceedings of the International Conference on Graphics and Signal Processing*; Association for Computing Machinery: New York, NY, USA, 2017; ICGSP '17; pp. 30–34. [CrossRef]
20. Brian, C.; Tj, T. Photo Based Clothing Measurements | Stitch Fix Technology—Multithreaded. Available online: <https://multithreaded.stitchfix.com/blog/2016/09/30/photo-based-clothing-measurement/> (accessed on 10 February 2022).
21. Cao, L.; Jiang, Y.; Jiang, M. Automatic measurement of garment dimensions using machine vision. In *Proceedings of the 2010 International Conference on Computer Application and System Modeling (ICCA SM 2010)*, Taiyuan, China, 22–24 October 2010; Volume 9, pp. 9–33. [CrossRef]
22. Tailored-Garment Measuring App, 2022. Available online: <https://www.thetailoredco.com/> (accessed on 4 March 2022).
23. Zhou, S.; Nie, D.; Adeli, E.; Yin, J.; Lian, J.; Shen, D. High-Resolution Encoder–Decoder Networks for Low-Contrast Medical Image Segmentation. *IEEE Trans. Image Process.* **2020**, *29*, 461–475. [CrossRef]
24. Hu, C.; Sapkota, B.B.; Thomasson, J.A.; Bagavathiannan, M.V. Influence of Image Quality and Light Consistency on the Performance of Convolutional Neural Networks for Weed Mapping. *Remote Sens.* **2021**, *13*, 2140. [CrossRef]
25. Ge, Y.; Zhang, R.; Wu, L.; Wang, X.; Tang, X.; Luo, P. A Versatile Benchmark for Detection, Pose Estimation, Segmentation and Re-Identification of Clothing Images. *arXiv* **2019**, arXiv:1901.07973.
26. Adaloglouon, N. An Overview of Unet Architectures for Semantic Segmentation and Biomedical Image Segmentation. 2021. Available online: <https://theaisummer.com/unet-architectures/> (accessed on 9 January 2022).
27. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015*; Springer: Cham, Switzerland, 2015; pp. 234–241. [CrossRef]
28. Jing, J.; Wang, Z.; Ratsch, M.; Zhang, H. Mobile-Unet: An efficient convolutional neural network for fabric defect detection. *Text. Res. J.* **2020**, *92*, 30–42. [CrossRef]
29. Roy, K.; Chaudhuri, S.S.; Pramanik, S. Deep learning based real-time Industrial framework for rotten and fresh fruit detection using semantic segmentation. *Microsyst. Technol.* **2021**, *27*, 3365–3375. [CrossRef]
30. Wang, A.; Togo, R.; Ogawa, T.; Haseyama, M. Defect Detection of Subway Tunnels Using Advanced U-Net Network. *Sensors* **2022**, *22*, 2330. [CrossRef] [PubMed]
31. Wang, Z.; Wang, E.; Zhu, Y. Image segmentation evaluation: A survey of methods. *Artif. Intell. Rev.* **2020**, *53*, 5637–5674. [CrossRef]
32. Taha, A.A.; Hanbury, A. Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool. *BMC Med. Imaging* **2015**, *15*, 29. [CrossRef] [PubMed]
33. Yao, A.D.; Cheng, D.L.; Pan, I.; Kitamura, F. Deep Learning in Neuroradiology: A Systematic Review of Current Algorithms and Approaches for the New Wave of Imaging Technology. *Radiol. Artif. Intell.* **2020**, *2*, e190026. [CrossRef]
34. Ding, L.; Goshtasby, A. On the Canny edge detector. *Pattern Recognit.* **2001**, *34*, 721–725. [CrossRef]
35. Vincent, O.R.; Folorunso, O. A Descriptive Algorithm for Sobel Image Edge Detection. In *Proceedings of the Informing Science & IT Education Conference*, Macon, GA, USA, 12–15 June 2009; pp. 1–11.
36. Burney, S.M.A.; Tariq, H. K-Means Cluster Analysis for Image Segmentation. *Int. J. Comput. Appl.* **2014**, *96*, 1–8.

37. Dhanachandra, N.; Manglem, K.; JinaChanu, Y. Image Segmentation Using K-means Clustering Algorithm and Subtractive Clustering Algorithm. *Procedia Comput. Sci.* **2015**, *54*, 764–771. [[CrossRef](#)]
38. Zheng, X.; Lei, Q.; Yao, R.; Gong, Y.; Yin, Q. Image segmentation based on adaptive K-means algorithm. *EURASIP J. Image Video Process.* **2018**, *68*, 1–10. [[CrossRef](#)]
39. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520. [[CrossRef](#)]
40. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
41. Toshev, A.; Szegedy, C. DeepPose: Human Pose Estimation via Deep Neural Networks. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1653–1660. [[CrossRef](#)]
42. Jadon, S. A survey of loss functions for semantic segmentation. In Proceedings of the 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), Via del Mar, Chile, 27–29 October 2020; pp. 1–6. [[CrossRef](#)]
43. Ma, J.; Chen, J.; Ng, M.; Huang, R.; Li, Y.; Li, C.; Yang, X.; Martel, A.L. Loss odyssey in medical image segmentation. *Med. Image Anal.* **2021**, *71*, 102035. [[CrossRef](#)]
44. Qian, S.; Lian, D.; Zhao, B.; Liu, T.; Zhu, B.; Li, H.; Gao, S. KGDet: Keypoint-Guided Fashion Detection. *Proc. Aaai Conf. Artif. Intell.* **2021**, *35*, 2449–2457.
45. Lu, Y. Automatically Measure Your Clothes on a Smartphone with AR, Mercari Engineering. 2022. Available online: <https://engineering.mercari.com/en/blog/entry/2020-06-19-150222/> (accessed on 5 January 2022).