

Radoslav TITKO

**GAMINIO BROKO APTIKIMAS TAIKANT GILIOJO MOKYMOSI
METODUS**

DETECTING PRODUCT DEFECTS USING DEEP LEARNING METHODS

Magistro baigiamasis darbas

Informacinių elektroninių sistemų studijų programa, valstybinis kodas 6211BX015

Dirbtinio intelekto sistemų specializacija

Informatikos inžinerijos studijų kryptis

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
ELEKTRONIKOS FAKULTETAS
ELEKTRONINIŲ SISTEMŲ KATEDRA

TVIRTINU
Katedros vedėjas

(parašas)
Prof. dr. A. Serackis
20__ m. _____ mėn. ____ d.

Radoslav TITKO

**GAMINIO BROKO APTIKIMAS TAIKANT GILIOJO MOKYMOSI
METODUS**
DETECTING PRODUCT DEFECTS USING DEEP LEARNING METHODS

Magistro baigiamasis darbas

Informacinių elektroninių sistemų studijų programa, valstybinis kodas 6211BX015

Dirbtinio intelekto sistemų specializacija

Informatikos inžinerijos studijų kryptis

Vadovas

doc. dr. D. Matuzevičius

(pedag. vardas, moksl. laipsnis, vardas, pavardė) (parašas)

(data)

Lietuvių kalbos konsultantas dr. A. Žemienė

(pedag. vardas, moksl. laipsnis, vardas, pavardė) (parašas)

(data)

Vilnius, 2026

Vilniaus Gedimino technikos universitetas		ISBN	ISSN
Elektronikos fakultetas		Egz. sk.	
Elektroninių sistemų katedra		Data*	

Antrosios pakopos studijų Informacinių elektroninių sistemų programos magistro baigiamasis darbas	
Pavadinimas	Gaminio broko aptikimas taikant giliojo mokymosi metodus
Autorius	Radoslav Titko
Vadovas	Dalius Matuzevičius

	Kalba: lietuvių
--	-----------------

Anotacija
Šiame darbe tiriamas giliojo mokymosi metodų taikymas gaminio broko aptikimui stiklinių butelių taros nuotraukose ir vaizdų sekose. Tikslui pasiekti atlikta naujausių broko aptikimo metodų analitinė apžvalga, atrinkti perspektyvūs giliojo mokymosi metodai, paruoštas nuosavas eksperimentinis duomenų rinkinys, įgyvendinti keturi modeliai naudojant specializuotą programinę įrangą bei eksperimentiškai ištirti jų veikimas. Tyrimo metu nustatyta, kad fono pašalinimas yra kritiškai svarbus duomenų paruošimo žingsnis, turintis didesnę įtaką galutiniam modelių tikslumui nei architektūros pasirinkimas, o prižiūrimojo ir neprižiūrimojo mokymosi metodai gali būti sėkmingai derinami sprendžiant praktines gamybos kokybės kontrolės problemas. Tyrimas parodė, kad veiksmingą defektų aptikimo sistemą galima sukurti naudojant vartotojų klasės įrangą ir atviro kodo bibliotekas.

Prasminiai žodžiai: Gilusis mokymasis, gaminio broko aptikimas, stiklinių butelių inspekcija, objektų aptikimas, anomalijų aptikimas, YOLOv11, Faster R-CNN, SSD, PatchCore.

<table border="1"> <tr><td>Vilnius Gediminas Technical University</td></tr> <tr><td>Faculty of Electronics</td></tr> <tr><td>Department of Electronic Systems</td></tr> </table>		Vilnius Gediminas Technical University	Faculty of Electronics	Department of Electronic Systems	<table border="1"> <tr> <td>ISBN</td> <td>ISSN</td> </tr> <tr> <td>Copies No.</td> <td></td> </tr> <tr> <td>Date-.....-.....</td> <td></td> </tr> </table>		ISBN	ISSN	Copies No.		Date-.....-.....	
Vilnius Gediminas Technical University												
Faculty of Electronics												
Department of Electronic Systems												
ISBN	ISSN											
Copies No.												
Date-.....-.....												
Master Degree Studies Information Electronics Systems study programme Master Graduation Thesis												
Title	Detecting Product Defects Using Deep Learning Methods											
Author	Radoslav Titko											
Academic supervisor	Dalius Matuzevičius											
			Thesis language: Lithuanian									
Annotation This thesis investigates the application of deep learning methods for product defect detection in glass bottle images and image sequences. An analytical review of the latest defect detection methods was conducted, suitable deep learning methods were selected, a custom experimental dataset was prepared, four models were implemented using specialized software, and their performance was experimentally evaluated. The study found that background removal is a critically important data preparation step with a greater impact on final model accuracy than the choice of architecture, and that supervised and unsupervised learning methods can be successfully combined to address practical manufacturing quality control problems. The study demonstrated that an effective defect detection system can be built using consumer-grade hardware and open-source libraries.												
Keywords: Deep learning, product defect detection, glass bottle inspection, object detection, anomaly detection, YOLOv11, Faster R-CNN, SSD, PatchCore.												

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
ELEKTRONIKOS FAKULTETAS
ELEKTRONINIŲ SISTEMŲ KATEDRA

Studijų kryptis: Informatikos inžinerija
Studijų programa: Informacinės elektroninės sistemos, valstybinis kodas 6211BX015
Specializacija: Dirbtinio intelekto sistemos

TVIRTINU
Katedros vedėjas
Artūras Serackis
2026-04-13

MAGISTRO BAIGIAMOJO DARBO UŽDUOTIS

Nr. DISfm-24-12618

Vilnius

Studentas (-ė): Radoslav Titko

Baigiamojo darbo tema: Gaminio broko aptikimas taikant giliojo mokymosi metodus

Baigiamojo darbo užbaigimo terminas pagal numatytą studijų kalendorinį grafiką.

BAIGIAMOJO DARBO UŽDUOTIS:

Darbo apibūdinimas. Darbo tikslas: Ištirti giliojo mokymosi metodų taikymą gaminio broko aptikimui gaminio nuotraukose bei vaizdų sekose.

Darbo apibūdinimas. Darbo uždaviniai:

1. Atlikti naujausių gaminio broko aptikimo vaizduose ir vaizdų sekose metodų, grįstų giliuoju mokymusi, analitinę apžvalgą.
2. Atrinkti giliojo mokymosi metodus, perspektyvius taikyti įvairaus dydžio gaminio defektų aptikimui specifiniuose vaizdų srautuose.
3. Paruošti duomenų rinkinį stiklo butelių broko aptikimo metodų eksperimentiniams tyrimams.
4. Įgyvendinti atrinktus gaminio broko aptikimo metodus, prireikus taikant specializuotą aparatinę ir programinę įrangą.
5. Eksperimentiškai ištirti įgyvendintų metodų veikimą ir pateikti tyrimo rezultatus.

Aiškinamojo rašto turinys. II pakopa:

1. Įvadas
 - 1.1. Darbo aktualumas ir tikslas
 - 1.2. Darbo uždaviniai
 - 1.3. Naudoti tyrimo ir analizės metodai
 - 1.4. Darbo naujumas ir praktinė nauda
 - 1.5. Darbo struktūra
2. Giliojo mokymosi metodų, skirtų gaminio brokui aptikti vaizduose ir vaizdų sekose, analitinė apžvalga.
3. Atrinktų giliojo mokymosi metodų taikymas stiklinių butelių ir taros defektų aptikimui.
4. Duomenų paruošimas gaminio broko aptikimo metodų eksperimentiniam tyrimui.
5. Gaminio broko aptikimo metodų įgyvendinimas ir eksperimentinio tyrimo rezultatai.
6. Apibendrinimas. Išvados

Literatūra

Priedai:

A priedas. Gaminio broko aptikimo metodų eksperimentinio tyrimo rezultatai.

B priedas. Tyrimų rezultatams gauti naudotas programinis kodas ir sudarytas duomenų rinkinys.

Vadovas profesorius Dalius Matuzevičius

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Radoslav Titko, 20194254

(Studento vardas ir pavardė, studento pažymėjimo Nr.)

Elektronikos fakultetas

(Fakultetas)

Informacinės elektroninės sistemos, DISfm-24

(Studijų programa, akademinė grupė)

BAIGIAMOJO DARBO (PROJEKTO) SĄŽININGUMO DEKLARACIJA

2026 m. gegužės 16 d.

Patvirtinu, kad mano baigiamasis darbas tema „Gaminio broko aptikimas taikant giliojo mokymosi metodus“ yra savarankiškai parašytas. Šiame darbe pateikta medžiaga nėra plagijuota. Tiesiogiai ar netiesiogiai panaudotos kitų šaltinių citatos pažymėtos literatūros nuorodose.

Prenkant ir įvertinant medžiagą bei rengiant baigiamąjį darbą, mane konsultavo mokslininkai ir specialistai: profesorius daktaras Dalius Matuzevičius. Mano darbo vadovas profesorius daktaras Dalius Matuzevičius.

Kitų asmenų indėlio į parengtą baigiamąjį darbą nėra. Jokių įstatymų nenumatytų piniginių sumų už šį darbą niekam nesusmokėjęs (-usi).



(Parašas)

Radoslav Titko

(Vardas ir pavardė)

TURINYS

Įvadas	13
Darbo aktualumas ir tikslas.....	14
Darbo uždaviniai.....	15
Naudoti tyrimo ir analizės metodai.....	15
Darbo naujumas ir praktinė nauda	16
Darbo struktūra	16
1. Giliojo mokymosi metodų, skirtų gaminio brokui aptikti vaizduose ir vaizdų sekose, analitinė apžvalga	17
1.1. Duomenų paruošimas.....	17
1.1.1. Duomenų rinkimas	17
1.1.2. Vaizdų anotavimas	18
1.1.3. Duomenų augmentacija	18
1.1.4. Fono šalinimas.....	18
1.2. Objektų aptikimo algoritmai	21
1.2.1. SSD (Single Shot MultiBox Detector)	22
1.2.2. Greitesnis R–CNN (Faster Region–based Convolutional Neural Networks)	23
1.2.3. YOLO (You Only Look Once).....	25
1.3. Defektų aptikimas	27
1.3.1. Prižiūrimas mokymasis	27
1.3.2. Neprižiūrimas mokymasis	27
1.3.3. Pusiau prižiūrimas mokymasis	28
1.3.4. Anomalijų aptikimo reikšmė šiame tyrime	29
1.4. Trumpas skyriaus apibendrinimas	29
2. Atrinktų giliojo mokymosi metodų taikymas stiklinių butelių taros defektų aptikimui	30
2.1. Metodų taikymo įvadas.....	30
2.2. Modelių architektūros ir konfigūracija	30
2.2.1. Eksperimentinė aplinka ir aparatūra	30
2.2.2. Modelių pasirinkimo pagrindimas.....	31
2.2.3. SSD MobileNetV2 FPN konfigūracija.....	31
2.2.4. Faster R–CNN ResNet50 konfigūracija	32
2.2.5. YOLOv11m konfigūracija.....	34
2.2.6. PatchCore konfigūracija	36
2.3. Perkėlimo mokymasis ir mokymas nuo nulio.....	38
2.4. Anotavimo variantų tyrimas	39
2.5. Modelių vertinimo metodika.....	40
2.6. Trumpas skyriaus apibendrinimas	42
3. Duomenų paruošimas gaminio broko aptikimo metodų eksperimentiniams tyrimui	44
3.1. Viešųjų duomenų analizė ir apribojimai	44
3.2. Nuosavo duomenų rinkinio kūrimo sprendimo pagrindimas.....	45
3.3. Nuosavo rinkinio kūrimo metodika	46
3.3.1. Fotografavimo metodika ir aplinkos sąlygos	48
3.3.2. Fotografavimo įranga	48
3.3.3. Apšvietimo sąlygos	49

3.3.4. Kameros padėtis ir fotografavimo kampai	49
3.3.5. Fotografavimo aplinka ir fono sąlygos.....	50
3.3.6. Duomenų rinkinių struktūra	50
3.3.7. Duomenų anotavimas	53
3.3.8. Duomenų augmentacija ir paruošimas modeliams	55
3.4. Trumpas skyriaus apibendrinimas	57
4. Gaminio broko aptikimo metodo įgyvendinimas ir eksperimentinio tyrimo rezultatai	58
4.1. Perkėlimo mokymasis ir mokymas nuo nulio.....	58
4.2. Anotavimo variantų palyginimas.....	60
4.2.1. Fono šalinimo įtaka (V1 vs V2)	61
4.2.2. Butelio anotavimo defektas (V3 vs V2)	62
4.2.3. Modelių palyginimas pagal geriausius rezultatus.....	62
4.3. PatchCore anomalijų aptikimo rezultatai	63
4.3.1. Klasifikacijos rezultatai	64
4.3.2. Lokalizacijos rezultatai.....	65
4.3.3. PatchCore palyginimas su prižiūrimo mokymosi modeliais	66
4.4. Diskusija ir praktinio pritaikymo galimybės.....	68
4.4.1. Pagrindinių rezultatų aptarimas.....	68
4.4.2. Tyrimo apribojimai.....	68
4.4.3. Praktinio pritaikymo galimybės.....	69
4.5. Trumpas skyriaus apibendrinimas	70
Apibendrinimas. Išvados	72
Literatūra	74
Priedai	78
A priedas. 13-oji tarptautinė elektronikos ir informacinių mokslų konferencija „eStream 2026”.....	78
B priedas. Stiklinių butelių brokų detekcijų rezultatai naudojant skirtingus modelius.	81

SANTRUMPOS

AOI (angl. <i>Automated Optical Inspection</i>)	automatizuota optinė inspekcija
API (angl. <i>Application Programming Interface</i>)	programų sąsaja
AUROC (angl. <i>Area Under the Receiver Operating Characteristic Curve</i>)	plotas po imtuvo veikimo charakteristikos kreive
C2PSA (angl. <i>Convolutional block with Parallel Spatial Attention</i>)	konvoliucinis blokas su lygiagrečiuoju erdviniu dėmesiu
C3K2 (angl. <i>Cross Stage Partial with kernel size 2</i>)	kryžminio etapo dalinis blokas su 2 dydžio branduoliu
CNN (angl. <i>Convolutional Neural Network</i>)	konvoliucinis neuroninis tinklas
COCO (angl. <i>Common Objects in Context</i>)	bendri objektai kontekste
CUDA (angl. <i>Compute Unified Device Architecture</i>)	skaičiavimų vieningoji architektūra
CVAT (angl. <i>Computer Vision Annotation Tool</i>)	kompiuterinio regėjimo anotavimo įrankis
DSLR (angl. <i>Digital Single–Lens Reflex</i>)	skaitmeninis veidrodinis fotoaparatas
F1 (angl. <i>F1 Score</i>)	F1 balas
Faster R–CNN (angl. <i>Faster Region–based Convolutional Neural Network</i>)	spartesnis regionais pagrįstas konvoliucinis neuroninis tinklas
FCN (angl. <i>Fully Convolutional Network</i>)	visiškai konvoliucinis tinklas
FN (angl. <i>False Negative</i>)	klaidingai neigiamas
FP (angl. <i>False Positive</i>)	klaidingai teigiamas
FPN (angl. <i>Feature Pyramid Network</i>)	požymių piramidinis tinklas
FPS (angl. <i>Frames Per Second</i>)	kadrų skaičius per sekundę
GAN (angl. <i>Generative Adversarial Network</i>)	generatyvinis priešiškas tinklas
GELAN (angl. <i>Generalized Efficient Layer Aggregation Network</i>)	apibendrintas efektyvus sluoksnių jungimo tinklas
GPU (angl. <i>Graphics Processing Unit</i>)	grafikos procesorius
HSV (angl. <i>Hue, Saturation, Value</i>)	atspalvis, sodrumas, ryškumas
IoU (angl. <i>Intersection over Union</i>)	sankirtos su sąjunga santykis
LED (angl. <i>Light Emitting Diode</i>)	šviesos diodas
LSTM (angl. <i>Long Short–Term Memory</i>)	ilgalaikės trumpalaikės atminties tinklas
mAP (angl. <i>mean Average Precision</i>)	vidutinis tikslumo vidurkis
NMS (angl. <i>Non–Maximum Suppression</i>)	nemaksimalių reikšmių slopinimas
PGI (angl. <i>Programmable Gradient Information</i>)	programuojama gradiento informacija
RPN (angl. <i>Region Proposal Network</i>)	regionų pasiūlymų tinklas
RSU (angl. <i>Residual U–block</i>)	likutinis U formos blokas
SGD (angl. <i>Stochastic Gradient Descent</i>)	stochastinis gradiento nusileidimas
SSD (angl. <i>Single Shot MultiBox Detector</i>)	vieno kadro detektorius
TN (angl. <i>True Negative</i>)	teisingai neigiamas
TP (angl. <i>True Positive</i>)	teisingai teigiamas
VAE (angl. <i>Variational Autoencoder</i>)	variacinis autokoderis

VRAM (angl. *Video Random Access Memory*)

vaizdo operatyvioji atmintis

YOLO (angl. *You Only Look Once*)

žiūri tik vieną kartą

PAVEIKSLŲ SĄRAŠAS

1.1 pav. Fono šalinimo rezultatai naudojant FCN, Tiramisu ir U-net[34]	19
1.2 pav. Dviejų etapų ir vieno etapo detektorių schemos[16]	22
1.3 pav. SSD architektūra[28]	22
1.4 pav. Faster R-CNN architektūra[5]	24
1.5 pav. YOLO architektūra[4]	25
1.6 pav. Prižiūrimas mokymas	27
1.7 pav. Neprižiūrimas mokymas	28
3.1 pav. Viešųjų duomenų rinkinių ir nuosavo rinkinio vaizdų palyginimas[8]	44
3.2 pav. Trys lyginamosios nuotraukos iš skirtingų fotografavimo aplinkų	50
3.3 pav. Reprezentatyvios nuotraukos iš prižiūrimo mokymosi rinkinio, rodančios skirtingus defektų tipus	51
3.4 pav. Anotavimo pavyzdžiai: oro burbuliuko, įbrėžimo ir nudaužtos galvutės anotacijos Roboflow aplinkoje	54
3.5 pav. Duomenų paruošimo proceso schema	57
4.1 pav. Perkėlimo mokymo ir mokymo nuo nulio metrikų palyginimas (V3 variantas)	59
4.2 pav. Anotavimo variantų palyginimas mAP@0.5 ir AR@100/Recall	61
4.3 pav. Modelių profilis ir inferavimo greičio palyginimas (V1 variantas)	63
4.4 pav. Anomalijų balo pasiskirstymo histograma ir konfūzijos matrica	64
4.5 pav. PatchCore anomalijų žemėlapių vizualizacijos pavyzdžiai	66
4.6 pav. Siūloma praktinė stiklinių butelių defektų aptikimo sistemos architektūra	70

LENTELIŲ SĄRAŠAS

1.1 lentelė. Fono šalinimo modelių tikslumo rodiklių palyginimas	21
2.1 lentelė. Eksperimentinės aplinkos specifikacijos	30
2.2 lentelė. SSD MobileNetV2 FPN mokymo hiperparametrai	32
2.3 lentelė. Faster R-CNN ResNet50 mokymo hiperparametrai	33
2.4 lentelė. YOLOv11m mokymo hiperparametrai	35
2.5 lentelė. PatchCore konfigūracijos parametrai	37
2.6 lentelė. Anotavimo variantų metodologinis palyginimas	40
2.7 lentelė. Naudotų vertinimo metrikų suvestinė	42
3.1 lentelė. Tyrime naudotų butelių charakteristikos	47
3.2 lentelė. Defektų pasiskirstymas duomenų rinkinyje	48
3.3 lentelė. Nuotraukų pasiskirstymas pagal defektų skaičių	51
3.4 lentelė. Duomenų rinkinių palyginimas	53
3.5 lentelė. Išplėsto duomenų rinkinio padalijimas	56
4.1 lentelė. Perkėlimo mokymasis vs. mokymas nuo nulio palyginimas	58
4.2 lentelė. Anotavimo variantų palyginimas visų modelių (tik defect klasė)	60
4.3 lentelė. Fono pašalinimo efektas – mAP@0.5 pokytis (V1 vs V2)	61
4.4 lentelė. V3 vs V2 palyginimas – mAP@0.5 pokytis	62
4.5 lentelė. Geriausių rezultatų palyginimas (V1 variantas)	62
4.6 lentelė. PatchCore klasifikacijos rezultatai	64
4.7 lentelė. PatchCore lokalizacijos rezultatai	65
4.8 lentelė. PatchCore vs. prižiūrėjo mokymosi modelių dalinis palyginimas	67

IVADAS

Šiuolaikinėje gamyboje, vykstančioje itin sudėtinguose pramonės procesuose (pvz., automobilių gamyba, specifinių detalių liejimas, aukštųjų technologijų elektronikos gamyba bei maistui skirtų produktų pramonė), gaminio kokybės užtikrinimas yra vienas didžiausių iššūkių. Gamybos metu dažnai atsiranda įvairaus pobūdžio defektų (pvz., vidinės ertmės, paviršiaus įdubimai ar įvairūs įbrėžimai) dėl įvairių priežasčių tokių kaip: pirminės konstrukcijos klaidos ar gamybos įrangos techniniai gedimai, nepalankios ar kintančios darbo sąlygos gamybinėse patalpose ir kt.[47]. Tokie defektai tiesiogiai didina įmonių veiklos sąnaudas[21], sutrumpina galutinį gaminių tarnavimo laiką ir sukelia didelį žaliavų bei energetinių išteklių švaistymą. Efektyvus ir savalaikis defektų aptikimas yra laikomas pagrindiniu įgūdžiu bei būtina pramonės įmonių kompetencija[42], siekiant ne tik pagerinti galutinių gaminių kokybę, bet ir išvengti neigiamo poveikio visai gamybos grandinei.

Ši defektų kontrolės sritis evoliucionavo palaipsniui, pereidama kelis svarbius technologinius etapus. Kol automatizuotos technologijos dar nebuvo sukurtos, defektai būdavo aptinkami rankiniu būdu: darbuotojai vizualiai tikrindavo kiekvieną produktą ar jo komponentą, ieškodami matomų nuokrypių. Kadangi žmogaus regos galimybės dėl fiziologinio nuovargio, dėmesio koncentracijos mažėjimo bei vertinimo subjektyvumo yra ribotos, rankiniu būdu pavyksta aptikti tik apie 70–80 % visų esamų defektų[30]. Įdiegus pramoninės automatizacijos sprendimus, pradėtos taikyti pirmosios mechanizuotos defektų aptikimo sistemos ir specializuoti jutikliai, o vėliau pažangios automatinės optinės inspekcijos sistemos (AOI). Automatizuotos sistemos gerokai padidino gamybos linijų spartą, tačiau vis dar nebuvo išspręstos sudėtingų tekstūrų ir specifinių paviršiaus netolygumo klasifikavimo problemos[22]. Galiausiai pradėtos taikyti išmaniosios sistemos, kurios defektų paieškai naudoja dirbtinį intelektą bei gilųjį mokymąsi (angl. *deep learning*). Tokios sistemos pasižymėjo akivaizdžiais privalumais, palyginti su anksčiau minėtomis technologijomis: jos ne tik sėkmingai prisitaikė prie nepalankių aplinkos veiksnių, bet ir užtikrino itin aukštą tikslumą bei operacinį efektyvumą ilgalaikėje perspektyvoje[18]. Tyrimai defektų aptikimo srityje leidžia optimizuoti gamybos išlaidas, didinti bendrą gamybos našumą bei produktų kokybę, sudarant tvirtą pagrindą viso pramonės sektoriaus intelektualiam transformavimui.

Nustatant gaminio defektą naudojant dirbtinį intelektą, šiandienos praktikoje galima išskirti keletą esminių technologijų: konvoliucinius neuroninius tinklus (angl. *CNN*), mašininį mokymąsi taikant prižiūrimąjį mokymąsi (angl. *supervised learning*), atvaizdų segmentavimą (angl. *image segmentation*) bei anomalijų aptikimo algoritmus (angl. *anomaly detection*). Kiekvienas metodas gali būti sėkmingai taikomas defektų paieškai įvairiose pramonės srityse, tačiau svarbu įvertinti jų skirtingas charakteristikas bei technines pritaikymo galimybes. Konvoliuciniai neuroniniai tinklai yra pripažinti kaip ypač veiksmingi vizualiems defektų modeliams atpažinti (pvz., įbrėžimams ar

įtrūkimams), nes jie geba giliai analizuoti vaizdo tekstūras ir sudėtingas formas[14]. Mašininis mokymasis su prižiūrimuoju mokymusi leidžia sukurti modelius, gebančius nustatyti itin sudėtingus defektų tipus, mokantis iš didelių žymėtų duomenų rinkinių, kuriuose aiškiai nurodomas defekto buvimas arba nebuvimas. Atvaizdų segmentavimo technologija efektyviai naudojama tiksliai defektų vietai nustatyti, leidžiant lokalizuoti problemas tiesiogiai ant gaminio paviršiaus[9]. Anomalijų aptikimo algoritmai laikomi tinkamais nereguliariems ar retai pasitaikantiems defektams nustatyti, apie kurių rūšis iš anksto nėra žinoma, nes šie algoritmai orientuojasi į bet kokius nuokrypius nuo standartinių vaizdų[24]. Renkantis tinkamą metodą, būtina atsižvelgti ne tik į gamybos proceso ypatumus, bet ir į specifinius defektų tipus bei technologijų integravimo galimybes.

Viena iš sričių, kurioje šios technologijos tampa gyvybiškai svarbios, yra stiklo pramonė. Čia susiduriama su unikaliais iššūkiais, siekiant efektyviai aptikti ir įvertinti įvairius defektus stikliniuose buteliuose bei kituose stiklo gaminiuose. Dabartiniai kontrolės metodai dažnai yra riboti dėl nepakankamo gebėjimo tiksliai reaguoti į gamybos linijose pasitaikančią produktų formų įvairovę, skaidrumą bei defektų kompleksiskumą. Giliojo mokymosi technologijos šiame kontekste atveria naujas galimybes, suteikdamos sistemoms būtiną adaptaciją ir lankstumą mokantis iš didelių duomenų srautų. Šios technologijos ypač vertingos stiklo gamyboje, kur būtina greitai nustatyti tokius defektus kaip mikroįtrūkimai ar oro burbuliukai, kurie anksčiau būdavo sunkiai pastebimi įprastomis automatizuotomis priemonėmis arba žmogaus akimi. Tikslus tokių defektų identifikavimas būtinas siekiant užtikrinti stiklo taros kokybę, ilgaamžiškumą bei saugumą galutiniam vartotojui[44].

Darbo aktualumas ir tikslas

Darbo tikslas: Ištirti giliojo mokymosi metodų taikymą gaminio broko aptikimui gaminio nuotraukose bei vaizdų sekose.

Darbo aktualumas: Tradiciniai rankinio tikrinimo metodai leidžia aptikti tik apie 70–80 % visų esamų defektų[30], o automatinės optinės inspekcijos sistemos susiduria su sunkumais klasifikuojant sudėtingas tekstūras[22]. Dabartiniai kontrolės metodai stiklo pramonėje yra riboti dėl produktų formų įvairovės, skaidrumo ir defektų kompleksiskumo, nors ir veikia geriau negu žmogaus tikrinimo etapas. Pastaraisiais metais giliojo mokymosi metodai vis plačiau taikomi stiklinių butelių kokybės kontrolėje. Latina[17] parodė, kad CNN grįsti modeliai, taikant perkėlimo mokymąsi ir duomenų augmentaciją, gali pasiekti iki 98 % defektų aptikimo tikslumą kontroliuojamoje aplinkoje. Claypo[6] taip pat pademonstruoja, kad CNN LSTM architektūra pasiekia aukštą klasifikavimo tikslumą skirtingose butelio dalyse. Nepaisant šių rezultatų, daugelis tyrimų orientuoti į pramonines gamybos linijas su specializuota įranga, o giliojo mokymosi veiksmingumo tyrimas ribotais ištekliais ir nekomercinės aplinkos sąlygomis lieka mažai išnagrinėta sritis[20].

Šis tyrimas aktualus trimis aspektais. Moksliniu, kadangi giliojo mokymosi algoritmų lyginamieji tyrimai stiklo defektų aptikimo srityje kontroliuojamomis, bet nekomercinėmis sąlygomis lieka mažai tyrinėta sritis, praktiniu, nes namų sąlygomis sudaryto duomenų rinkinio naudojimas leidžia įvertinti, ar šios technologijos gali būti pritaikytos be specializuotos pramoninės infrastruktūros, o tai ypač naudinga smulkioms ir vidutinėms įmonėms, bei ekonominiu, kadangi defektų aptikimas ankstyvoje stadijoje tiesiogiai mažina gamybos sąnaudas ir švaistomų išteklių kiekį[21].

Darbo problemos: Nors giliojo mokymosi metodai, naudojami defektams aptikti, aktyviai tiriami, įprastai apsiribojama pavienių modelių nagrinėjimu ir nėra atliekama skirtingų architektūrų, vieno etapo (SSD, YOLO) ir dviejų etapų (Faster R–CNN), lyginamoji analizė, siekiant įvertinti jų efektyvumą stiklinių butelių defektams aptikti kontroliuojamomis, nekomercinėmis sąlygomis. Todėl lieka neatsakytas klausimas: kuris giliojo mokymosi algoritmas tradicinis SSD, Faster R–CNN ar šiuolaikiniai YOLO modeliai geriausiai tinka stiklinių butelių defektams aptikti, turint ribotą duomenų rinkinį ir nekomercinę įrangą?

Darbo uždaviniai

1. Atlikti naujausių gaminio broko aptikimo vaizdose ir vaizdų sekose metodų, grįstų giliuoju mokymusi, analitinę apžvalgą.
2. Atrinkti giliojo mokymosi metodus, perspektyvius taikyti įvairaus dydžio gaminio defektų aptikimui specifiniuose vaizdų srautuose.
3. Paruošti duomenų rinkinį stiklo butelių broko aptikimo metodų eksperimentiniams tyrimams.
4. Įgyvendinti atrinktus gaminio broko aptikimo metodus, prireikus taikant specializuotą aparatinę ir programinę įrangą.
5. Eksperimentiškai ištirti įgyvendintų metodų veikimą ir pateikti tyrimo rezultatus.

Naudoti tyrimo ir analizės metodai

Tyrimo metodai apima mokslinės literatūros analizę, eksperimentinius metodus bei kiekybinius rezultatų palyginimo metodus. Literatūros analizei naudota duomenų bazė IEEE Xplore, Google Scholar ir kitos mokslinės publikacijos giliojo mokymosi ir defektų aptikimo srityje. Eksperimentams naudoti keturi skirtingų architektūrų giliojo mokymosi modeliai: SSD MobileNetV2 FPN, Faster R–CNN ResNet50, YOLOv11m (prižiūrimasis mokymasis) ir PatchCore (neprižiūrimasis mokymasis). Modelių veikimas įvertintas naudojant mAP@0.5, Precision, Recall ir F1 metrikas.

Darbo naujumas ir praktinė nauda

Darbo naujumą įvertinti atliktas pirminis lyginamasis tyrimas, kuriame palygintos skirtingų architektūrų giliojo mokymosi sistemos (prižiūrimojo ir neprižiūrimojo mokymosi) stiklinių butelių defektams aptikti kontroliuojamomis nekomercinėmis sąlygomis, naudojant nuosavą vaizdų duomenų rinkinį. Iš to gaunama praktinė nauda, kur tyrimo rezultatai parodo, kad veiksmingą defektų aptikimo sistemą galima sukurti naudojant vartotojų klasės įrangą ir atviro kodo bibliotekas.

Darbo struktūra

Darbą sudaro šeši pagrindiniai skyriai. Pirmame skyriuje pateikiamas įvadas, darbo aktualumas, tikslas, uždaviniai, naudoti metodai ir darbo struktūra. Antrame skyriuje atlikta mokslinės literatūros analizė apie giliojo mokymosi metodus, skirtus gaminio brokui aptikti vaizduose ir vaizdų sekose. Trečiame skyriuje aprašytas atrinktų metodų taikymas stiklinių butelių defektų aptikimui. Ketvirtame skyriuje pateiktas duomenų paruošimas eksperimentiniams tyrimams. Penktame skyriuje pateikiami eksperimentinio tyrimo rezultatai ir diskusija. Šeštame skyriuje suformuluotos apibendrinančios išvados.

1. GILIOJO MOKYMOSI METODŲ, SKIRTŲ GAMINIO BROKUI APTIKTI VAIZDUOSE IR VAIZDŲ SEKOSE, ANALITINĖ APŽVALGA

Norint efektyviai taikyti dirbtinio intelekto sistemas gaminio brokui aptikti, tikslinga naudoti konvejerio tipo sprendimą. Tokiu atveju konvejeriu judantys gaminiai fotografuojami iš skirtingų kampų, sudarydami nuoseklias vaizdų sekas, o gauti vaizdai analizuojami dirbtinio intelekto modeliais. Remiantis šia analize, sistema gali automatiškai nustatyti, ar gaminys atitinka kokybės reikalavimus, ar turi vieną ar kelis defektus. Analizuojant nuotrauką, sistema pirmiausia nustatytų pagrindinį objektą, t. y. pirmasis etapas būtų objekto identifikavimas ir atskyrimas nuo fono, taikant pažangius kompiuterinės regos ir dirbtinio intelekto metodus. Po objekto atskyrimo sistema pašalintų foną, palikdama tik pagrindinį objektą, kas leistų daugiau dėmesio skirti pačiam objektui ir efektyviau nustatyti jo būklę. Galutinis etapas būtų defektų aptikimas, taikant giliojo mokymosi algoritmus, kurie analizuotų objekto savybes ir nustatytų bet kokius defektus. Šis procesas užtikrintų tikslesnį ir efektyvesnį defektų aptikimą, prisidedant prie gamybos procesų optimizavimo ir kokybės kontrolės. Taigi šiame skyriuje analizuojamos panašios sistemos, lyginami skirtingų sprendimų privalumai bei trūkumai ir siekiama nustatyti, kokie neuroninių tinklų modeliai atliktų šią užduotį optimaliai.

1.1. Duomenų paruošimas

1.1.1. Duomenų rinkimas

Aukštos kokybės vaizdų duomenų rinkinio sudarymas yra vienas svarbiausių ir daugiausiai laiko reikalaujančių etapų, kuriant giliojo mokymosi modelį defektams aptikti. Tyrimai rodo, kad mokslininkai vidutiniškai praleidžia apie 70–80 % laiko dirbdami su duomenų rinkiniais juos rinkdami, valydami ir apdorodami[2], o likęs laikas paprastai skiriamas modelio pasirinkimui, mokymui ir testavimui. Svarbų vaidmenį duomenų rinkimo procese taip pat atlieka fotografavimo sąlygos, tokios kaip apšvietimas, kameros padėtis, fonas ir objekto orientacija, nes jos tiesiogiai lemia modelio gebėjimą tiksliai atpažinti defektus. Netolygus ar silpnas apšvietimas gali sukelti žemą kontrastą, spalvų iškraipymą ir triukšmą vaizduose, o tai neigiamai veikia tokias užduotis kaip objektų atpažinimas, aptikimas ir klasifikavimas. Dėl šios priežasties rekomenduojama fotografuoti kontroliuojamomis sąlygomis, naudojant pastovų apšvietimą ir neutralų foną[17]. Šių veiksnių svarba dar labiau išryškėja dirbant su stikliniais buteliais, nes jų skaidrumas ir paviršiaus atspindžiai apsunkina vaizdų analizę[40]. Mašininio mokymosi metodai tokiais atvejais gali susidurti su sunkumais dėl kintančių aplinkos apšvietimo sąlygų ir nedidelių duomenų rinkinių, o tai riboja jų veiksmingumą aptinkant defektus[6]. Todėl kontroliuojamos fotografavimo sąlygos stiklo gaminių tyrimuose yra ypač svarbios. Surinktas duomenų rinkinys paprastai skirstomas į tris grupes: mokymo

(angl. training), validavimo (angl. validation) ir testavimo (angl. testing), o dažniausiai rekomenduojamas jų santykis yra 80 % mokymo, 10 % validavimo ir 10 % testavimo duomenų[3].

1.1.2. Vaizdų anotavimas

Vaizdų anotavimas tai procesas, kurio metu kiekvienoje nuotraukoje rankiniu būdu pažymimi objektai ar defektai, suteikiant modeliui informaciją apie tai, kur ir kas yra vaizde. Tai itin daug laiko reikalaujantis etapas. Anotavimo kokybė tiesiogiai veikia modelio tikslumą: netiksliai pažymėti defektai gali sukelti modelio persimokymą arba klaidingus aptikimus. Objektų aptikimo užduotims dažniausiai naudojamos stačiakampės anotacijos (angl. *bounding box*), kurios apibrėžia objekto vietą vaizde naudojant keturias koordinates. Šiandien yra keletas populiarių anotavimo įrankių pvz. LabelImg, CVAT ir Roboflow. Roboflow išsiskiria tuo, kad yra platforma, apjungianti anotavimą, duomenų valdymą, augmentaciją ir eksportą į įvairius formatus (pvz., YOLO, COCO, TFRecord), todėl ypač tinka projektams, kuriuose naudojami keli skirtingi modeliai[8].

1.1.3. Duomenų augmentacija

Duomenų augmentacija yra metodų visuma, leidžianti dirbtinai padidinti mokymo duomenų rinkinį generuojant modifikuotas originalių vaizdų kopijas. Tai ypač svarbu tada, kai turimas duomenų rinkinys yra ribotas, nes augmentacija padeda modeliui geriau apibendrinti ir išvengti persimokymo (angl. *overfitting*)[33]. Tyrimai rodo, kad tinkama augmentacija gali ženkliai pagerinti modelio tikslumą net ir mažuose duomenų rinkiniuose[14]. Dažniausiai naudojami augmentacijos metodai apima geometrinius pakeitimus kaip pasukimą (angl. *rotation*), veidrodinį atspindį (angl. *flip*), apkarpyimą (angl. *crop*) bei spalvinės informacijos keitimą, pavyzdžiui, ryškumo, kontrasto ar soties reguliavimą. Taip pat naudojamas triukšmo pridėjimas (angl. *noise injection*), kuris padeda modeliui būti atsparesniam vaizdo kokybės svyravimams[33]. YOLO modelių mokymo aplinkoje augmentacija dažnai yra integruota tiesiogiai į mokymo procesą ir vykdoma realiu laiku, o Roboflow platforma leidžia augmentaciją taikyti dar prieš mokymą, taip padidindama duomenų rinkinio įvairovę eksporto metu.

1.1.4. Fono šalinimas

Fono šalinimas yra svarbus vaizdų apdorojimo defektų aptikimo sistemose etapas, leidžiantis modeliui sutelkti dėmesį į patį objektą, o ne į aplinkinius vaizdo elementus. Fono šalinimui dažnai yra naudojamas semantinis segmentavimas, kuris leidžia tiksliai nustatyti kiekvieno vaizdo pikselio priklausomybę objektui arba fonui ir tai yra pagrindinis metodas, taikomas šiai užduočiai atlikti[9]. Tyrimai rodo, kad fono šalinimas gali gerokai pagerinti defektų aptikimo tikslumą, ypač kai defektai yra smulkūs ar subtilūs.

Visiškai konvoliucinis neuroninis tinklas (FCN) buvo vienas pirmųjų giliojo mokymosi modelių, pritaikytų pikselio lygio segmentavimui. FCN pašalina visiškai sujungtus sluoksnius, būdingus klasikiniam CNN, ir juos pakeičia konvoliuciniais, todėl tinklas gali apdoroti bet kokio dydžio vaizdus ir generuoti erdvinis segmentacijos žemėlapius[31]. Taikant FCN defektams aptikti, kiekvienas pikselis klasifikuojamas kaip priklausantis objektui arba fonui, o tai leidžia efektyviai pašalinti nereikalingą informaciją. FCN pasiekia didesnę nei 99 % tikslumą teisingai klasifikuotų pikselių. Tačiau FCN trūkumas yra tas, jog segmentacijos kraštai dažnai būna neapibrėžti, nes modelis praranda detalią erdvinę informaciją dėl nuosekliai kaupiamų operacijų.

U-Net yra galingas dirbtinio neuroninio tinklo architektūros pavyzdys, sukurtas medicininiam vaizdams segmentuoti, tačiau vėliau sėkmingai pritaikytas pramoniniams vaizdams apdoroti. Šio algoritmo pavadinimas kilęs iš struktūros panašumo į raidę „U“, o pati architektūra susideda iš dviejų pagrindinių dalių: koduotojo (angl. *encoder*) ir dekodutojo (angl. *decoder*). Koduotojas naudoja konvoliucinius ir kaupimo sluoksnius (angl. *pooling layers*), kad palaipsniui sumažintų vaizdo raišką ir išskirtų svarbiausias savybes. Dekoduotojas atkuria vaizdą naudodamas transponuojančius konvoliucinius sluoksnius (angl. *transpose convolution layers*) ir praleistus ryšius (angl. *skip connections*), kurie perduoda erdvinę informaciją tiesiogiai iš koduotojo. Tai leidžia gauti detalius segmentacijos rezultatus su tiksliais kraštais[29]. U-Net ypač vertingas, kai reikia tiksliai atskirti objektus nuo fono turint nedidelį duomenų kiekį, nes architektūra efektyviai mokosi iš nedidelių duomenų rinkinių.

Tiramisu (*The One Hundred Layers Tiramisu*) neuroninis tinklas, pagrįstas DenseNet architektūra, kurioje kiekvienas sluoksnis gauna įvestį iš visų ankstesnių sluoksnių. Šis tankus ryšių tinklas leidžia modeliui efektyviai naudoti požymius iš skirtingų abstrakcijos lygių, o tai ypač naudinga sudėtingiems objektų kontūrams segmentuoti[11]. Tiramisu pasižymi gebėjimu aptikti sudėtingas objektų struktūras ir tiksliai jas segmentuoti, paliekant tik objektus ir pašalinant foną. Nors modelis sudėtingesnis ir reikalauja daugiau skaičiavimo išteklių nei U-Net, jo tikslumas sudėtingose vaizdo situacijose yra geresnis.



1.1 pav. Fono šalinimo rezultatai naudojant FCN, Tiramisu ir U-net[34]

Paveiksle (žr. 1.1 pav.) pateikiami FCN, Tiramisu ir U–Net modelių rezultatai. Pirmoje vaizdo dalyje matomas originalus vaizdas su žmogumi, leidžiančiu aitvarą. Antroje dalyje pavaizduotas FCN modelio rezultatas, kuris gana tiksliai išskyrė žmogaus siluetą, tačiau nepateikė spalvų. Trečioje dalyje pavaizduotas U–Net modelio rezultatas, kur modelis pašalino didžiąją dalį fono, tačiau kai kuriose vietose, ypač prie rankų, įtraukė dangaus fragmentų arba pašalino žmogaus dalis. Ketvirtoje dalyje pateikiamas Tiramisu modelio rezultatas, kuris tikslesnis nei U–Net, nes žmogaus siluetas aiškesnis, kraštai tiksliau apibrėžti, mažiau klaidų aplink rankas bei aitvarą. Tiramisu modelis, pasižymintis U formos architektūra, šioje užduotyje pasirodė geriausiai, nes tiksliau atskyrė žmogų nuo fono.

Verta dar paminėti U²–Net kuris yra paprastas, tačiau galingas giliojo mokymosi semantinio segmentavimo modelis, gerinantis fono šalinimą naudojant liekamojo U (RSU) blokus. Skirtingai nuo ankstesnių architektūrų, U²–Net naudoja įterptinę U formos struktūrą, kur kiekvienas RSU blokas pats yra mini U–Net tipo tinklas, o tai leidžia vienu metu išgauti požymius tiek lokalaus, tiek globalaus konteksto lygiu[25]. U²–Net publikuotas žurnale *Pattern Recognition* 2020 metais gavo geriausio straipsnio apdovanojimą. Tuomet Pythone atsirado Rembg biblioteka, kuri yra pagrįsta U²–Net modeliu ir dabar šis įrankis yra paprastas panaudojamumui ir leidžia foną pašalinti automatiškai, nereikalaujant rankinio objekto ribų apibrėžimo.

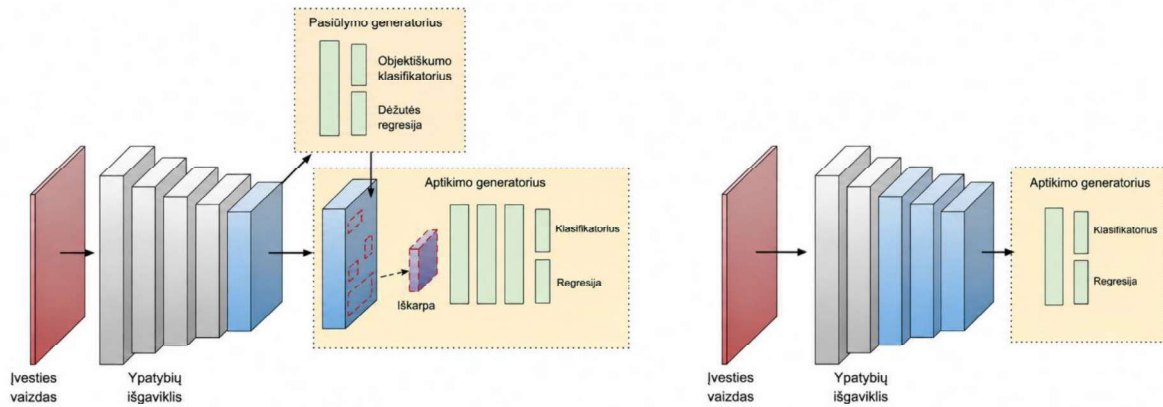
1.1 lentelė. Fono šalinimo modelių tikslumo rodiklių palyginimas

Modelis	Metrika	Rezultatas	Duomenų rinkinys	Šaltinis
FCN–8s	mIoU	67,2 %	PASCAL VOC 2012	Shelhamer[31]
U–Net	mIoU	65,57 % $\pm$ 0.15	Ultragarso medianos nervo vaizdai	Wang[38]
Tiramisu	IoU	66,9 %	CamVid	Jégou[11]
U ² –Net	mIoU	79,66 % $\pm$ 0.13	Ultragarso medianos nervo vaizdai	Wang[38]

Lentelėje (žr. 1.1 lent.) pateikiami fono šalinimo modelių tikslumo rodikliai. Kadangi U–Net ir U²–Net rezultatai gauti to paties tyrimo metu ir abu modeliai vertinti mIoU metrika tame pačiame ultragarso medianos nervo vaizdų duomenų rinkinyje, yra galimybė atlikti šių modelių lyginamąją analizę[38]. Nustatyta, kad U²–Net pasiekė 79,66 % mIoU, o standartinis U–Net 65,57 %, t. y. U²–Net pranoko U–Net 14,09 procentinių punktų. FCN–8s rezultatas gautas iš PASCAL VOC 2012 duomenų rinkinio[31], Tiramisu iš CamVid duomenų rinkinio[11]. Kadangi šie du modeliai vertinti skirtinguose duomenų rinkiniuose, jų tiesioginis palyginimas su U–Net ir U²–Net nėra galimas, todėl jų rezultatai pateikti tik iliustravimo tikslais. Nepaisant to, visuose tyrimuose pasiekti rezultatai yra panašaus diapazono (65–80 %), todėl galima teigti, kad visi aprašyti modeliai yra tinkami semantinės segmentacijos užduotims.

1.2. Objektų aptikimo algoritmai

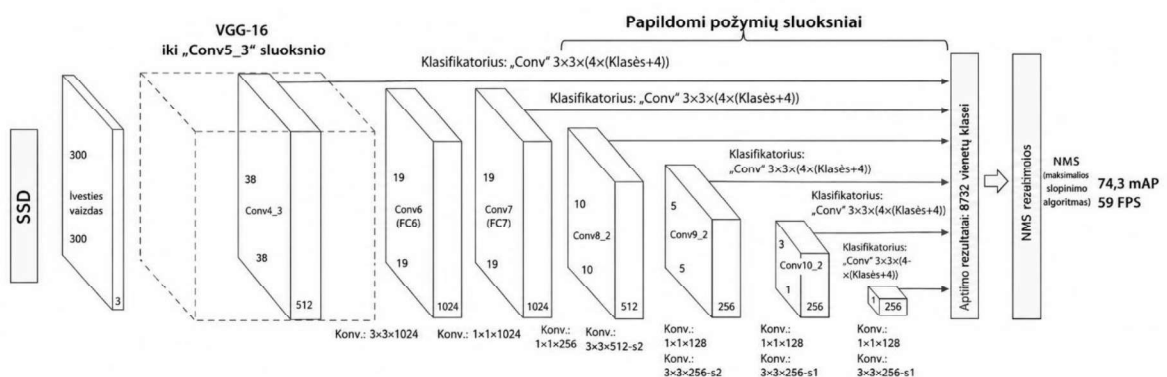
Objektų aptikimas mašininio mokymosi kontekste yra plačiai tyrinėjamas uždavinys, kurį galima suskirstyti į dvi pagrindines kategorijas: regresiją ir klasifikaciją[7]. Šis uždavinys apima tris svarbius komponentus: požymių ištraukimą, objektų klasifikavimą ir jų vietos lokalizavimą vaizduose. Svarbu atskirti dviejų etapų ir vieno etapo detektorius (žr. 1.2 pav.). R–CNN ir jo išvestiniai modeliai (pvz., Fast R–CNN ir Faster R–CNN) naudoja dviejų etapų metodą[32]: pirmajame etape generuojami regionų pasiūlymai, o antrajame vyksta objektų klasifikavimas ir lokalizavimas. Vieno etapo detektoriai (pvz., YOLO ir SSD) objektus aptinka vienu tinklo praėjimu, išvedant ribojančius langelius ir klasių tikimybes viso vaizdo erdvėje. Šie algoritmai pasižymi paprasta architektūra ir efektyvumu, tačiau dviejų etapų detektoriai, nors sudėtingesni, dažnai pasiekia didesnę tikslumą dėl atskirai siūlomų regionų[12].



1.2 pav. Dviejų etapų ir vieno etapo detektorių schemas[16]

1.2.1. SSD (Single Shot MultiBox Detector)

SSD (angl. *Single Shot MultiBox Detector*) yra vieno etapo objektų aptikimo algoritmas, pristatytas Liu[19] kaip greitesnė ir efektyvesnė alternatyva dviejų etapų detektoriams. SSD visiškai pašalina regionų pasiūlymų generavimą ir vėlesnio pikselių ar požymių atkūrimo etapus, visus skaičiavimus apjungdamas į vieną tinklą. Tai leidžia modeliui objektus lokalizuoti ir klasifikuoti vienu tinklo praėjimu gerokai didinant apdorojimo greitį. SSD veikimo principas grindžiamas numatytųjų rėmų (angl. *default boxes*) koncepcija. SSD diskretizuoja ribojančių langelių išvesties erdvę į numatytųjų rėmų rinkinį, paskirstytą pagal skirtingus kraštinių santykius ir mastelį kiekvienoje požymių žemėlapių vietoje. Prognozavimo metu tinklas generuoja kiekvienos objektų klasės buvimo kiekvienose numatytose rėmuose balus ir atlieka rėmų patikslinimus, kad geriau atitiktų objekto formą.



1.3 pav. SSD architektūra[28]

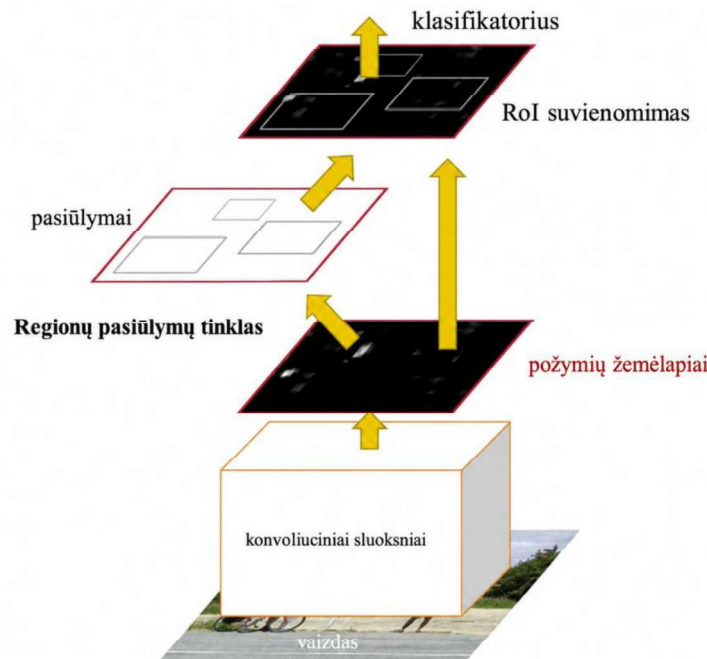
Esminis SSD architektūros bruožas yra požymių žemėlapių naudojimas. SSD architektūra apjungia prognozes iš kelių požymių žemėlapių skirtingomis raiškomis, natūraliai apdorojama įvairaus dydžio objektus ir gerindama aptikimo kokybę. Tai ypač svarbu defektų aptikimo kontekste, kur defektai gali būti labai skirtingų dydžių, nuo didelių įtrūkimų iki smulkių įbrėžimų.

Šiame tyrime naudotas SSD MobileNetV2 FPN variantas. MobileNetV2 SSD FPN–Lite apjungia lengvą MobileNetV2 stuburo tinklą su daugiamastelių požymių piramidės tinklo (angl. *Feature Pyramid Network*, FPN) požymių atvaizdavimu, leidžiančiu efektyviai ir tiksliai aptikti objektus mobiliuose ir įterptinėse platformose. MobileNetV2 pasirinktas kaip stuburo tinklas dėl lengvos architektūros, efektyvumo bei įrangos apribojimų. Jis naudoja gylio atžvilgiu atskiriamas konvoliucijas (angl. *depthwise separable convolutions*) ir atvirkštinius liekamuosius blokus (angl. *inverted residual blocks*), kurie gerokai sumažina parametrų skaičių išlaikant aukštą tikslumą.

SSD modelis sėkmingai pritaikytas pramoniniams defektams aptikti. Latina[17] pademonstruoja, kad SSD MobileNetV2 modelis, taikant perkėlimo mokymąsi ir duomenų augmentaciją, pasiekia iki 98,07 % bendrą sistemos tikslumą aptinkant stiklinių butelių defektus kontroliuojamoje aplinkoje. Analogiški rezultatai gauti ir kitose pramonės srityse. SSD MobileNetV2 architektūra, apmokytas naudojant TensorFlow Object Detection API, pasiekė 87,4 % mAP ir 0,89 F1 rodiklį aptinkant mūrinių tiltų defektus, tai patvirtina modelio universalumą skirtingose taikymo srityse, o pagrindinis SSD privalumas yra greitis: esant 300×300 įvesties vaizdui, SSD pasiekia 72,1 % mAP PASCAL VOC 2007 testavimo rinkinyje, veikdamas 58 kadrų per sekundę greičiu, o esant 500×500 įvesties vaizdui, gautas 75,1 % mAP, pranokdamas Faster R–CNN modelį. Tačiau SSD turi ir apribojimų, kadangi jis gali susidurti su sunkumais aptinkant labai mažus arba artimus objektus, o tai svarbus aspektas smulkių defektų aptikimo kontekste.

1.2.2. Greitesnis R–CNN (Faster Region–based Convolutional Neural Networks)

Faster R–CNN yra dviejų etapų objektų aptikimo modelis, pristatytas Ren[27] kaip esminis R–CNN šeimos evoliucijos žingsnis. Pagrindinė inovacija yra regionų pasiūlymų tinklas (angl. *Region Proposal Network*, RPN), leidžiantis generuoti regionų pasiūlymus beveik be papildomų skaičiavimo sąnaudų. RPN dalijasi viso vaizdo konvoliuciniais požymiais su aptikimo tinklu, todėl regionų pasiūlymų generavimas tampa beveik nemokamas. Tai visiškai konvoliucinis tinklas, kuris vienu metu numato objekto ribas ir objektyvumo balus kiekvienoje padėtyje. Visą Faster R–CNN sistemą sudaro du moduliai, gilus visiškai konvoliucinis tinklas, siūlantis regionų kandidatus ir Fast R–CNN detektorius, kuris šiuos pasiūlymus naudoja klasifikavimui ir tikslinimui. Naudojant VGG–16 modelį, sistema veikia 5 kadrų per sekundę greičiu ir pasiekia 73,2 % mAP PASCAL VOC 2007 duomenų rinkinyje bei 70,4 % mAP PASCAL VOC 2012 duomenų rinkinyje, naudojant tik 300 pasiūlymų vienam vaizdui.



1.4 pav. Faster R–CNN architektūra[5]

Duomenų srautas Faster R–CNN tinkle prasideda nuo įvesties vaizdo, kuris perduodamas per konvoliucinius sluoksnius požymiams išgauti. Gauti požymių žemėlapiai (angl. *feature maps*) tiekiami į RPN, kuris juos nuskaito slenkančio lango metodu ir generuoja objektų kandidatų regionus su atitinkamais objektyvumo balais. Šie regionai standartizuojami per RoI (angl. *Region of Interest*) telkimo sluoksnį, kuris įvairaus dydžio pasiūlymus paverčia vienodo dydžio požymių vektoriais. Galiausiai klasifikavimo ir regresijos sluoksniai nustato objekto klasę ir patikslina ribojančio langelio koordinatas (žr. 1.4 pav.).

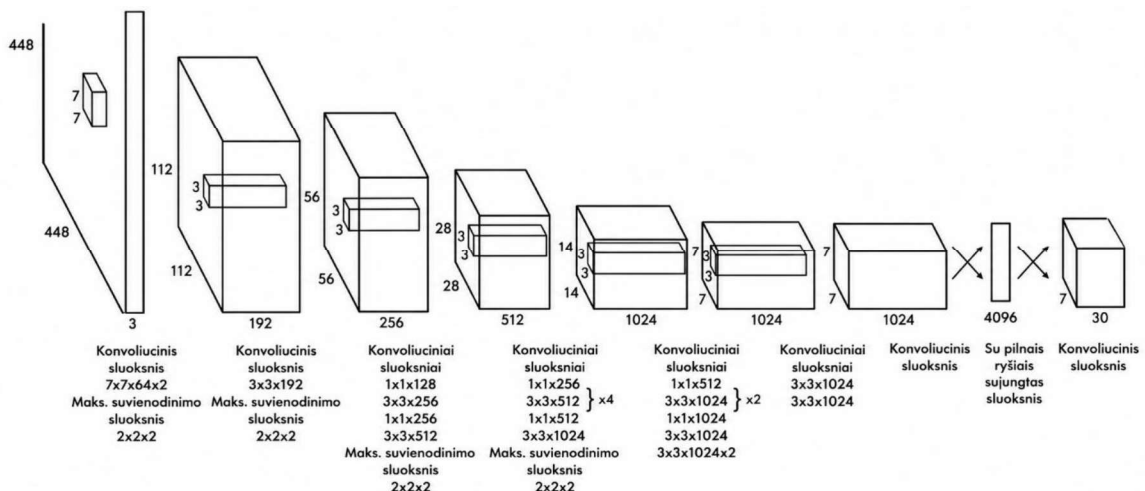
Šiame tyrime naudotas Faster R–CNN ResNet50 variantas, kuriame VGG–16 požymių ištraukimo tinklas pakeistas giliaja likutinio mokymosi architektūra ResNet50. Tai plačiai taikoma praktika defektų aptikimo tyrimuose. ResNet50 naudojamas kaip stuburo tinklas (angl. *backbone*) defektams aptikti, nes jo likutiniai blokai efektyviai sprendžia gilių tinklų degradacijos problemą ir leidžia išgauti turtingesnes požymių reprezentacijas[41]. Zhao[46] pademonstruoja, kad Faster R–CNN su ResNet50 ir požymių piramidės tinklu (FPN) ypač efektyvus įvairių mastelių defektams aptikti, kai defektų dydžiai būna skirtingi. Lyginant su SSD, Faster R–CNN pasižymi aukštesniu tikslumu dėl dviejų etapų architektūros. Pirmiausia generuojami regionų kandidatai, tuomet kiekvienas jų tikrinamas individualiai. Tačiau ši architektūra lėtesnė ir reikalauja daugiau skaičiavimo išteklių, todėl labiau tinkama situacijoms, kai tikslumas svarbesnis už greitį. Pramoniniuose defektų aptikimo tyrimuose Faster R–CNN su perkėlimo mokymusi nuosekliai pasiekia aukštesnius mAP rodiklius nei vieno etapo detektoriai, kai reikalingas tikslus vidutinių ir mažų defektų lokalizavimas.

1.2.3. YOLO (You Only Look Once)

Vienas plačiausiai naudojamų objektų aptikimo algoritmų, išsiskiriantis realaus laiko veikimo galimybėmis. Originalų YOLO modelį 2016 metais pristatė Redmon[26], pasiūlydami naują požiūrį į objektų aptikimą, kuris yra pagrįstas viso vaizdo analize vienu tinklo praėjimu (angl. *single pass*), eliminuojant atskiros regionų pasiūlymų generavimo etapą. Tai yra taip pat kaip SSD vieno etapo detektorius. Vieno praėjimo metu algoritmu siekiama numatyti objekto klasę ir ribojantį langelį, apibrėžiantį objekto vietą įvesties vaizde. Kiekvienas ribojantis langelis apibūdinamas keturiais parametrais:

- Ribojančio langelio centras (bx, by)
- Langelio plotis (b ω)
- Langelio aukštis (bh)

Be to, YOLO numato numatomos klasės indeksą (c) ir prognozės tikimybę (Pc)



1.5 pav. YOLO architektūra[4]

Pav 1.5 (žr. 1.5 pav) pirmoje dalyje pateikiamas įvesties vaizdas 448x448 pixelių, kurį algoritmas analizuoja, o toliau seka konvoliucinis tinklas (angl. *Convolutional Network*) sudarytas iš kelių konvoliucinių ir sumažinimo (maxpool) sluoksnių, kurie vis palaipsniui išgauna pagrindinius požymius ir mažina vaizdo matmenis. Konvoliuciniai sluoksniai naudojami požymiams aptikti, o *maxpool* sluoksniai sumažina duomenų kiekį, išlaikydamos svarbiausią informaciją. Paskutinėje dalyje pilnai sujungti sluoksniai (angl. *Connected layers*) atlieka du procesus, klasifikatorius nustato objekto klasę o ribojančio langelio regresorius apskaičiuoja tikslią objekto vietą.

Nuo pirmosios versijos YOLO algoritmas išgyveno reikšmingą evoliuciją. YOLOv3 ir YOLOv4 gerokai padidino tikslumą, įdiegę daugiamastelį požymių išgavimą ir patobulintus mokymo metodus. YOLOv5, išleistas 2020 metais, pristatė supaprastintą mokymo procesą ir patobulintą perkėlimo mokymosi palaikymą[35]. YOLOv8, išleistas 2023 metais, įvedė inkarų neturinčių (angl. 25

anchor-free) metodą su patobulinta CSPNet stuburo architektūra ir atskirta aptikimo galvute (angl. *decoupled head*), tapdamas vienu plačiausiai taikomų modelių pramoniniams uždaviniams. Paskutiniai YOLO kūrimo etapai atnešė esminių architektūrinių inovacijų. YOLOv9 (2024) pristatė dvi pagrindines naujoves: programuojamą gradientų informacijos mechanizmą (angl. *Programmable Gradient Information*, PGI) ir apibendrintą efektyvų sluoksnių agregavimo tinklą (angl. *Generalized Efficient Layer Aggregation Network*, GELAN), skirtus informacijos praradimo problemai spręsti giliuose tinkluose[36]. YOLOv10 (2024), sukurtas Tsinghua universiteto tyrėjų, pašalino nemaksimalios slopinimo (angl. *Non-Maximum Suppression*, NMS) žingsnį po aptikimo, taikydamas dvilypę etikečių priskyrimo strategiją, kas gerokai sumažino prognozavimo laiką[39]. YOLOv11, išleistas 2024 metų spalį, toliau tobulina architektūrą, orientuodamasis į geresnį požymių išgavimą ir efektyvesnį skaičiavimą[23]. YOLOv11 ankstesnį C2f bloką pakeičia kompaktiškesniu C3K2 bloku, kuris gerokai pagerina gradientų srautą ir skaičiavimo efektyvumą[13]. Taip pat įvedamas C2PSA modulis, integruojantis kanalus ir erdvinę informaciją su daugiagalvio dėmesio mechanizmais efektyvesniam požymių išgavimui. YOLOv11 turi 22 % mažiau parametrų nei palyginamas YOLOv8m modelis, tuo pat metu pasiekdamas aukštesnį mAP rodiklį[13].

Keletas empirinių tyrimų patvirtina YOLOv11 pranašumą lyginant su ankstesnėmis versijomis. Elektros įrangos objektų aptikimo tyrime, kuriame lyginami YOLOv5, YOLOv8, YOLOv9, YOLOv10 ir YOLOv11 modeliai, YOLOv11 pasiekė aukščiausią 57,2 % mAP, pranokdamas visus kitus modelius tiek aptikimo tikslumu, tiek klaidingų aptikimų mažinimo rodikliais[10]. Pramoninių defektų aptikimo srityje YOLOv11 su SA-GAN duomenų augmentacija NEU-DET plieninių paviršių defektų duomenų rinkinyje pasiekė 85,1 % mAP@0.5, tai 7,7 %, 3,1 %, 5,9 % ir 7,0 % daugiau nei atitinkamai YOLOv7, YOLOv8, YOLOv9 ir YOLOv10 modeliai. YOLOv11 modeliai vidutiniškai 31 % greitesni nei YOLOv9 ir 1,41 % greitesni nei YOLOv10, o tai daro juos tinkamais realaus laiko taikymui, reikalaujančiam greito apdorojimo neprarandant tikslumo. Wang[37] pademonstruoja, kad patobulintas YOLOv8 algoritmas efektyviai aptinka pramoninių aliuminio lakštų paviršiaus defektus, pasiekdamas aukštą tikslumą ir atitinkant pramoninio greičio reikalavimus. PCB defektų aptikimo tyrime YOLOv11 su papildomais dėmesio mechanizmais pasiekė aukštesnį tikslumą nei ankstesnės YOLO versijos, ypač aptinkant smulkius defektus[45]. YOLOv11 grįstas PSF-YOLO modelis, skirtas smulkiems plieninių paviršių defektams aptikti, efektyviai derina aukštą tikslumą su nedideliu parametrų skaičiumi, todėl gali būti diegiamas negalingų įrenginių aplinkose[23].

Šiame tyrime pasirinktas YOLOv11 modelis dėl to, jog jis demonstruoja aukščiausius tikslumo rodiklius lyginant su ankstesnėmis YOLO versijomis daugelyje pramoninių defektų aptikimo tyrimų, bei jo architektūra su C3K2 blokais ir C2PSA dėmesio mechanizmais yra ypač efektyvi smulkiems defektams aptikti, o tai svarbu stiklinių butelių paviršiaus defektų kontekste. Taip

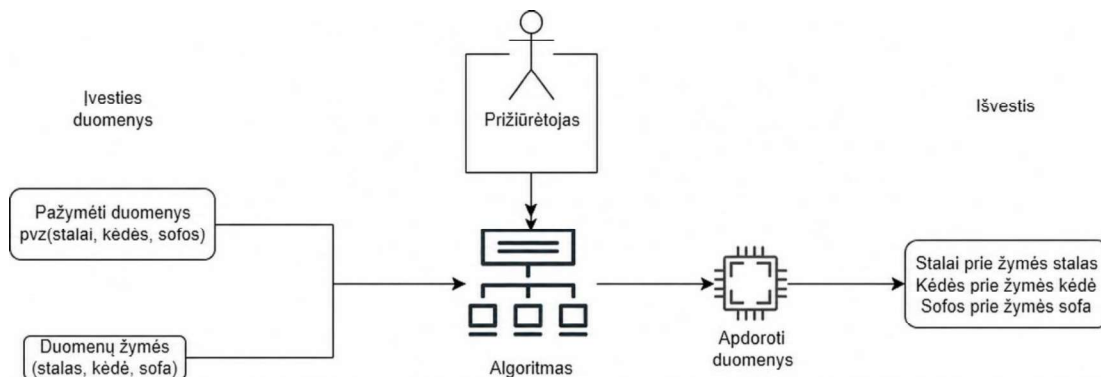
pat, YOLOv11 turi mažesnę parametrų skaičių lyginant su YOLOv8, kas leidžia efektyviau mokytis modelį naudojant ribotą įrangą bei duomenų rinkinį.

1.3. Defektų aptikimas

Defektų aptikimas giliojo mokymosi metodais gali būti realizuojamas trimis skirtingomis mokymosi paradigmomis, kurių pasirinkimas priklauso nuo turimo duomenų kiekio ir defektų tipo. Prižiūravimo mokymosi metodai reikalauja didelių žymėtų duomenų rinkinių su aiškiais defektų pavyzdžiais., pusiau prižiūravimo mokymosi metodai apjungia abi paradigmas, modeliai mokomi naudojant daug gerų vaizdų ir nedidelį kiekį pažymėtų brokuotų pavyzdžių ir neprižiūravimo mokymosi metodai leidžia anomalijas aptikti remiantis tik gerų gaminių vaizdais. Pramoniniuose taikymuose dažniausiai susiduriama su situacija, kai kokybiškų gaminių vaizdų yra daug, o brokuotų labai mažai, todėl neprižiūravimo ir pusiau prižiūravimo mokymosi metodai įgyja ypatingą reikšmę.

1.3.1. Prižiūrimasis mokymasis

Prižiūrimasis mokymasis (angl. *Supervised Learning*) reikalauja žymėtų duomenų rinkinių, kuriuose pateikiami tiek defektų turintys, tiek kokybiški gaminiai[15]. Ši paradigma leidžia pasiekti aukštą aptikimo tikslumą, tačiau kelia iššūkių, susijusių su duomenų disbalansu. Pramoniniuose taikymuose brokuotų gaminių paprastai yra gerokai mažiau nei kokybiškų[24]. Šiai problemai spręsti sukurti specialūs metodai, taikantys pažangias neuroninių tinklų struktūras efektyviam darbui sudėtingų fonų atvejais. Prižiūravimo mokymosi kontekste objektų aptikimo modeliai (pvz., YOLO, SSD ir Faster R–CNN) apmokyti konkretus defektų tipus atpažinti, naudojant anotuotus duomenis su tiksliai pažymėtomis defektų vietomis.



1.6 pav. Prižiūrimasis mokymasis

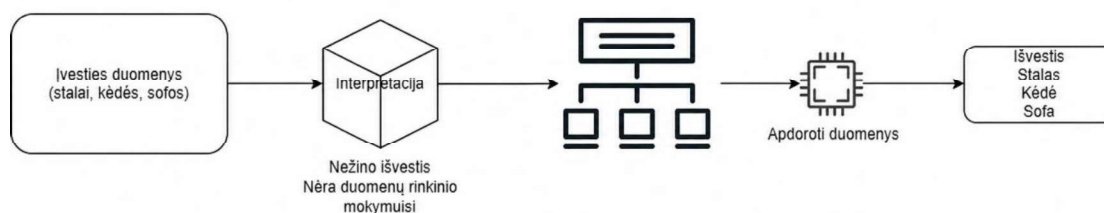
1.3.2. Neprižiūrimasis mokymasis

Neprižiūrimasis mokymasis (angl. *Unsupervised Learning*) remiasi tik nebrotuotų gaminių vaizdais ir mokosi atpažinti anomalijas kaip nukrypimus nuo išmuktos geros būklės reprezentacijos. Rekonstrukcija grįsti metodai remiasi prielaida, kad modelis, apmokytas tik su nebrotuotų gaminių

vaizdais, negali efektyviai rekonstruoti anomalijų, todėl didelė rekonstrukcijos paklaida signalizuoja apie defekto buvimą[43]. Pagrindiniai neprižiūrimojo mokymosi metodai defektams aptikti yra autoenkoderiai (angl. *autoencoders*), variaciniai autoenkoderiai (angl. *Variational Autoencoders*, VAE) ir generatyviniai priešingojo mokymosi tinklai (angl. *Generative Adversarial Networks*, GAN).

Autoenkoderiai susideda iš dviejų dalių: koduotojo (angl. *encoder*), kuris suspaudžia įvesties vaizdą į kompaktišką reprezentaciją, ir dekoduoja (angl. *decoder*), kuris rekonstruoja vaizdą. Modelis, apmokytas tik su gerais vaizdais, minimizuoja rekonstrukcijos paklaidą geriems vaizdams, tačiau anomalūs regionai rekonstruojami prasčiau, todėl jų rekonstrukcijos paklaida didesnė ir gali būti naudojama kaip anomalijos indikatorius[43]. Variaciniai autoenkoderiai (VAE) papildo klasikinių autoenkoderių architektūrą tikimybine latentine (angl. *latent*) erdve, leidžiančia modeliui geriau apibendrinti gerų vaizdų pasiskirstymą. VAE pranašumas prieš kitus neprižiūrimojo mokymosi metodus yra gebėjimas išmokyti sudėtingas latentines reprezentacijas ir rekonstruoti originalias įvestis taikant generatyvinio modeliavimo principus.

Generatyviniai priešingo mokymosi tinklai (GAN) naudoja dvi konkuruojančias neuronines architektūras: generatorių, kuris kuria sintetinius vaizdus, ir diskriminatorių, kuris skiria tikrus vaizdus nuo generuotų. GAN kontekste anomalijoms aptikti generatorius mokosi atkurti gerų vaizdų pasiskirstymą, o anomalijos identifikuojamos kaip vaizdai, kurių generatorius negali tiksliai atkurti[15].



1.7 pav. Neprižiūrimas mokymas

1.3.3. Pusiau prižiūrimas mokymasis

Pusiau prižiūrimasis mokymasis (angl. *Semi-Supervised Learning*) derina prižiūrimojo ir neprižiūrimojo mokymosi savybes, naudodamas tiek nebrotuotų objektų duomenis riboms nustatyti, tiek duomenis kur yra sužymėtos norimo objekto ribos[15]. Šis metodas ypač naudingas, kai turimas ribotas kiekis defektų turinčių gaminių, ir leidžia sumažinti rankinio duomenų žymėjimo poreikį.

1.3.4. Anomalijų aptikimo reikšmė šiame tyrime

Šiame tyrime anomalijų aptikimas naudojamas kaip papildomas tyrimo aspektas, lyginant su prižiūrimojo mokymosi metodais (SSD, Faster R–CNN, YOLO). Pramoniniuose taikymuose anomalijų aptikimas ypač vertingas, kai gamyboje gali atsirasti naujų, anksčiau nematytų defektų tipų, kurių prižiūrimojo mokymosi modelis neatpažintų, nes nebuvo apmokytas su tokiais pavyzdžiais. Tai yra esminis skirtumas tarp prižiūrimojo mokymosi ir anomalijų aptikimo, pastarasis metodas gali aptikti ir nežinomus defektus.

1.4. Trumpas skyriaus apibendrinimas

Apibendrinant, galima teigti, kad defektų aptikimo srityje egzistuoja dvi pagrindinės kryptys, prižiūrimojo ir neprižiūrimojo mokymosi metodai, kurie ne konkuruoja, o greičiau papildo vienas kitą. Prižiūrimojo mokymosi modeliai kaip Faster R–CNN ar YOLO reikalauja anototų defektų pavyzdžių, tačiau suteikia tikslią defekto vietą ir tipą, o neprižiūrimojo mokymosi metodai kaip PatchCore leidžia apsieiti be brokuotų gaminių pavyzdžių, todėl yra patrauklesni ten, kur defektų tiesiog nėra daug arba defektai nėra iš anksto žinomi. Be to, stiklinių butelių atveju prie viso prisideda dar vienas iššūkis – medžiagos skaidrumas ir atspindžiai, dėl kurių net gerai parinktas modelis gali prastai veikti, jei duomenys nebuvo tinkamai paruošti. Taigi literatūros analizė leidžia daryti išvadą, kad stiklo gaminių defektų aptikimui vien modelio pasirinkimas nėra pakankamas, svarbu ir tai, kaip duomenys renkami bei apdorojami prieš mokymą, o skirtingų architektūrų palyginimas nekomercinėmis sąlygomis lieka mažai išnagrinėta sritis, kurią šis tyrimas ir siekia užpildyti.

2. ATRINKTŲ GILIOJO MOKYMOSI METODŲ TAIKYMAS STIKLINIŲ BUTELIŲ TAROS DEFECTŲ APTIKIMUI

2.1. Metodų taikymo įvadas

Praktinėje dalyje siekiama empiriškai patikrinti teorinėje dalyje išanalizuotų giliojo mokymosi metodų tinkamumą stiklinių butelių defektams aptikti nekomercinėmis, ribotų išteklių sąlygomis. Tyrimui pasirinkti keturi modeliai, atstovaujantys skirtingoms paradigmoms: vieno etapo detektorius SSD MobileNetV2 FPN, dviejų etapų detektorius Faster R-CNN ResNet50, šiuolaikinis vieno etapo detektorius YOLOv11m ir neprižiūrimojo mokymosi anomalijų aptikimo metodas PatchCore. Modeliai vertinami pagal defektų aptikimo tikslumą ($mAP@0.5$, $mAP@0.5:0.95$) ir prognozavimo greitį. Praktinė dalis struktūruota trimis pagrindiniais tyrimų etapais. Pirmajame etape sudaromas nuosavas stiklinių butelių vaizdų duomenų rinkinys ir atliekama trijų skirtingų anotavimo strategijų sisteminė lyginamoji analizė. Antrajame etape lyginamas perkėlimo mokymasis su mokymu nuo nulio, siekiant įvertinti, ar COCO duomenimis nustatyti svoriai būtini ribotų duomenų sąlygomis. Trečiajame etape visi modeliai lyginami tarpusavyje pagal tikslumo bei greičio rodiklius, siekiant nustatyti efektyviausią algoritmą praktiniam stiklinių butelių kokybės kontrolės taikymui.

2.2. Modelių architektūros ir konfigūracija

2.2.1. Eksperimentinė aplinka ir aparatūra

Visi eksperimentai atlikti asmeniniame nešiojamajame kompiuteryje su Windows operacine sistema, nenaudojant specializuotos pramoninės ar debesų kompiuterijos infrastruktūros. Toks aparatūros pasirinkimas atitinka tyrimo tikslą, kuris yra įvertinti giliojo mokymosi modelių veiksmingumą ribotų išteklių nekomercinėje aplinkoje.

2.1 lentelė. Eksperimentinės aplinkos specifikacijos

Komponentas	Specifikacija
GPU	NVIDIA GeForce RTX 3060 Laptop GPU
GPU atmintis	8,2 GB (bendra), 7,7 GB (<i>shared</i>), 496 MB (<i>dedicated</i>)
RAM	16 GB
Operacinė sistema	Windows
Python (TF modeliai)	3.9
Python (PatchCore)	3.12

Prižiūravimo mokymosi modeliams buvo naudota TensorFlow Object Detection API su Python 3.9 aplinka. YOLOv11m modelis treniruotas naudojant Ultralytics biblioteką toje pačioje Python 3.9 aplinkoje. Anomalijų aptikimo modeliui PatchCore naudota anomalib 2.3.3 biblioteka (Intel OpenVINO), veikianti atskiroje Python 3.12 conda aplinkoje, atskira aplinka būtina dėl anomalib bibliotekos priklausomybių suderinamumo reikalavimų.

Verta atkreipti dėmesį į GPU konfigūracijos ypatumus. RTX 3060 Laptop GPU turi ribotą dedikuotą VRAM (496 MB), tačiau naudoja bendrosios atminties (angl. *shared memory*) mechanizmą, leidžiantį pasiekti iki 7,7 GB bendros GPU atminties. Dėl šios priežasties Faster R–CNN su ResNet50 stuburu (~41M parametrų) teko mažinti paketų dydį (angl. *batch size*) iki 2, kad būtų išvengta CUDA atminties perpildymo klaidų. SSD ir Faster R–CNN modeliams įvesties raiška apribota iki 640×640 pikselių, o YOLOv11m galėjo naudoti didesnę 896×896 pikselių raišką dėl efektyvesnės Ultralytics atminties valdymo implementacijos.

2.2.2. Modelių pasirinkimo pagrindimas

Remiantis teorinėje dalyje atlikta giliojo mokymosi metodų analize, šiam tyrimui pasirinkti keturi modeliai, atstovaujantys skirtingoms paradigmoms: SSD MobileNetV2 FPN, kaip lengvos architektūros vieno etapo detektorius, tinkamas ribotų išteklių aplinkai; Faster R–CNN ResNet50, kaip dviejų etapų detektorius, pasižymintis aukštu lokalizacijos tikslumu, YOLOv11m, šiuolaikinės YOLO architektūros su geriausiais tikslumo rodikliais pramoninių defektų aptikimo tyrimuose, bei PatchCore, kaip neprižiūravimo mokymosi atstovas, leidžiantis palyginti abi mokymosi paradigmas tame pačiame duomenų rinkinyje. Visi iš anksto apmokyti modeliai gauti iš oficialių šaltinių su COCO duomenų rinkiniu iš anksto apmokytais svoriais.

2.2.3. SSD MobileNetV2 FPN konfigūracija

SSD MobileNetV2 FPN modelis paruoštas naudojant TensorFlow Object Detection API, standartizuotą objektų aptikimo modelių treniravimo ir vertinimo biblioteką. Iš anksto apmokyto modelio svoriai gauti iš TensorFlow 2 Detection Model Zoo, kur modelis apmokytas su COCO 2017 duomenų rinkiniu, apimančiu 80 objektų klasių ir apie 100 000 anotacijų. Perkėlimo mokymosi atveju šie svoriai naudoti kaip pradinis taškas, o modelio išvestis pritaikyta dviejų klasių uždaviniui „bottle“ ir „defect“. Mokymo nuo nulio atveju konfigūracijos faile užkomentuotos *fine_tune_checkpoint* eilutės, taip inicializuojant visus modelio parametrus atsitiktinėmis reikšmėmis. Duomenys modeliui pateikti TFRecord formatu. Klasių žemėlapis apibrėžtas atskirame *label_map.pbtxt* faile su dviem klasėmis. Modelis treniruotas 10 000 žingsnių su paketų dydžiu (angl. *batch size*) 4. Optimizavimui naudotas SGD su momento metodu (momento koeficientas = 0,9), o tai rekomenduojama praktika SSD modelių treniravimui[19]. Mokymosi greičio valdymui taikytas kosinusinio atšaldymo (angl.

cosine decay) grafikas su šildymo (angl. *warmup*) faze, bazinis mokymosi greitis 0,008, šildymo mokymosi greitis 0,0008, šildymo žingsnių skaičius 2 000. Toks mokymosi greičio valdymas leidžia modeliui stabiliai pradėti mokymąsi ir palaipsniui mažinti mokymosi greitį artėjant prie optimalių parametrų. Pagrindiniai SSD MobileNetV2 FPN hiperparametrai pateikiami 2.2 lentelėje (žr. 2.2 lent.).

2.2 lentelė. SSD MobileNetV2 FPN mokymo hiperparametrai

Hiperparametras	Reikšmė
Mokymo žingsniai	10000
Batch dydis	4
Optimizatorius	SGD su momentu
Momento koeficientas	0,9
Bazinis mokymosi greitis	0,008
Šildymo mokymosi greitis	0,0008
Šildymo žingsnių skaičius	2000
Mokymosi greičio grafikas	Kosinusinis atšaldymas
Įvesties raiška	640×640 px
NMS IoU slenkstis	0,6
Maksimalus aptikimų skaičius	100
Klasių skaičius	2

Architektūriniu požiūriu SSD MobileNetV2 FPN naudoja giliąsias atskirtines konvoliucijas (angl. *depthwise separable convolutions*) ir atvirkštinius liekamuosius blokus (angl. *inverted residual blocks*) MobileNetV2 pagrindiniame tinkle, sumažinant parametrų skaičių iki maždaug 6,8 mln., išlaikant pakankamą tikslumą. Požymių piramidės tinklas (FPN) apima 3–7 lygius, o papildomo sluoksnio gylis siekia 128, kas leidžia efektyviai aptikti įvairaus dydžio defektus. Klasifikavimo nuostoliai apskaičiuojami taikant Focal Loss funkciją ($\gamma = 2,0$; $\alpha = 0,25$), padedančią spręsti klasių disbalanso problemą, ypač aktualią šiame tyrime, kuriame be defektų vaizdai sudaro apie 30 % duomenų rinkinio.

2.2.4. Faster R–CNN ResNet50 konfigūracija

Faster R–CNN ResNet50 modelis sukonfigūruotas ir apmokytas naudojant tą pačią TensorFlow Object Detection API aplinką kaip ir SSD modelis. Iš anksto apmokyto modelio svoriai gauti iš TensorFlow 2 Detection Model Zoo ir naudotas *faster_rcnn_resnet50_v1_640x640_coco17_tpu-8* kontrolinis taškas (angl. *checkpoint*), apmokytas

su COCO 2017 duomenų rinkiniu. Perkėlimo mokymosi ir mokymo nuo nulio įgyvendinimas identiškas SSD modeliui. Mokymo nuo nulio atveju *fine_tune_checkpoint* parametras konfigūracijos faile išjungtas (užkomentuotas). Duomenys modeliui pateikti taip pat kaip ir SSD konfigūracijoje TFRecord formatu, naudojant tą patį *label_map.pbtxt* klasių žemėlapi. Tai užtikrina vienodas duomenų paruošimo sąlygas abiem TensorFlow modeliams, pašalinant duomenų apdorojimo skirtumus rezultatų variacijai šaltinį. Faster R–CNN modelis treniruotas 10 000 žingsnių kur paketų dydis sumažintas iki 2, kas buvo būtina dėl įrangos apribojimų, kadangi Faster R–CNN ResNet50 su ~41M parametrų reikalauja daugiau GPU atminties nei SSD MobileNetV2 FPN (~6,8M parametrų). Optimizavimui taip pat naudotas SGD su momento metodu (momento koeficientas = 0,9), tačiau bazinis mokymosi greitis sumažintas iki 0,003, žemesnis mokymosi greitis rekomenduojamas dviejų etapų detektoriams dėl sudėtingesnės RPN ir aptikimo tinklo sąveikos[27]. Mokymosi greičiui valdyti taikytas kosinusinio atšaldymo grafikas su trumpesne šildymo faze, šildymo žingsnių skaičius 500 (palyginti su SSD modelio 2 000), tai leido greičiau pasiekti pradinį mokymosi greitį. Įvesties vaizdo dydis išlaikytas 640×640 pikselių, naudojant proporcijas išlaikantį dydžio keitimo metodą (angl. *keep aspect ratio resizer*) su užpildymu iki nustatyto dydžio. Pagrindiniai Faster R–CNN ResNet50 hiperparametrai pateikiami 2.3 lentelėje (žr. 2.3 lent.).

2.3 lentelė. Faster R–CNN ResNet50 mokymo hiperparametrai

Hiperparametras	Reikšmė
Mokymo žingsniai	10000
Batch dydis	2
Optimizatorius	SGD su momentu
Momento koeficientas	0,9
Bazinis mokymosi greitis	0,003
Šildymo mokymosi greitis	0,0008
Šildymo žingsnių skaičius	500
Mokymosi greičio grafikas	Kosinusinis atšaldymas
Įvesties raiška	640×640 px
Maksimalus regionų pasiūlymų skaičius	300
NMS IoU slenkstis	0,6
RPN NMS IoU slenkstis	0,5
Klasių skaičius	2

Architektūriniu požiūriu Faster R–CNN ResNet50 naudoja ResNet50 kaip pagrindinį požymių ištraukimo tinklą su aktyvuotu paketų normalizacijos mokymu (*batch_norm_trainable: true*), leidžiančiu modeliui efektyviau prisitaikyti prie stiklinių butelių domeno perkėlimo mokymosi

metu. Regionų pasiūlymų tinklas (RPN) naudoja keturis mastelio lygius (0,125; 0,25; 0,5; 1,0) ir tris kraštinių santykius (0,5; 1,0; 2,0), užtikrinant platų skirtingų defektų dydžių ir formų aprėptį. Maksimalus regionų pasiūlymų skaičius nustatytas į 300, leidžiantis išlaikyti balansą tarp aptikimo tikslumo ir skaičiavimo efektyvumo[27]. Lokalizacijos nuostolių svoriai abiem etapams nustatyti į 2,0, o klasifikacijos nuostolių svoriai į 1,0, pabrėžiant tikslios defektų lokalizacijos svarbą.

Lyginant su SSD konfigūracija, Faster R–CNN pasižymi keliais skirtumais: dviejų etapų architektūra leidžia pasiekti didesnę defektų lokalizacijos tikslumą, tačiau reikalauja mažesnio paketų dydžio dėl didesnių atminties sąnaudų. Be to, mažesnis mokymosi greitis atspindi sudėtingesnę optimizavimo proceso struktūrą.

2.2.5. YOLOv11m konfigūracija

YOLOv11m modelis treniruotas naudojant Ultralytics biblioteką, skirtingai nei SSD ir Faster R–CNN modeliai, kurie naudojo TensorFlow Object Detection API su protocol buffer konfigūracijos failais, Ultralytics biblioteka leidžia modelio parametrus apibrėžti tiesiogiai Python skripte, o tai gerokai supaprastina eksperimentų vykdymą. Iš anksto apmokyto modelio svoriai gauti iš oficialios Ultralytics saugyklos. Naudotas *yolo11m* variantas, apmokytas su COCO duomenų rinkiniu. Duomenys modeliui pateikti YOLO formatu kiekvienai nuotraukai priskiriamas atskiras tekstinis failas su normalizuotomis ribojančių stačiakampių (angl. *bounding boxes*) koordinatėmis bei *data.yaml* konfigūracijos failas, apibrėžiantis duomenų rinkinių kelius ir klasių pavadinimus. Šis formatas tiesiogiai eksportuojamas iš Roboflow platformos, o tai leido efektyviai integruoti duomenų paruošimo ir modelio treniravimo procesus. Perkėlimo mokymosi ir mokymo nuo nulio skirtumas YOLOv11 atveju įgyvendintas paprastai: perkėlimo mokymosi atveju naudotas *yolo11m.pt* (iš anksto apmokyti svoriai), o mokymo nuo nulio atveju *yolo11m.yaml* (tik architektūros aprašas be svorių). Tai standartinis Ultralytics bibliotekos mechanizmas šiems dviem mokymo režimams atskirti. YOLOv11m modelis treniruotas iki 150 epochų, naudojant ankstyvo sustabdymo (angl. *early stopping*) mechanizmą, kai mokymas automatiškai nutraukiamas, jei validavimo metrikos nepagerėja per 20 epochų (*patience* = 20). Paketų dydis nustatytas į 6, o įvesties vaizdų raiška 896×896 pikselių, kas yra gerokai didesnė nei SSD ir Faster R–CNN naudota 640×640 pikselių raiška. Didesnis įvesties dydis pasirinktas siekiant geriau išsaugoti smulkių defektų (siaurų įbrėžimų ir mažų oro burbuliukų) vizualines charakteristikas. Optimizavimui naudotas AdamW optimizatorius su pradine mokymosi sparta 0,001, skirtingai nei SSD ir Faster R–CNN modeliuose, kuriuose taikytas SGD optimizatorius su momentu. AdamW pasirinkimas grindžiamas jo gebėjimu adaptuoti mokymosi spartą kiekvienam parametru atskirai, o tai ypač naudinga dirbant su smulkiais ir retai pasitaikančiais defektais. Persidengimo su tikruoju langeliu santykio (angl. *Intersection over Union*, IoU) slenkstis nustatytas į 0,6, siekiant padidinti aptikimo jautrumą (angl. *recall*) ir sumažinti praleistų defektų skaičių.

Duomenų augmentacijos parametrai sureguliuoti siekiant išvengti pernelyg agresyvių transformacijų, galinčių neigiamai paveikti smulkių defektų aptikimą. Atspalvio (HSV–H: 0,015), sotumo (HSV–S: 0,5) ir ryškumo (HSV–V: 0,3) pokyčiai bei horizontalus apvertimas (*fliplr*: 0,5) palikti kaip pagrindinės augmentacijos technikos. *Auto_augment* funkcija išjungta, nes automatinė augmentacija gali taikyti pernelyg stiprias transformacijas, kurios maskuoja smulkius defektus arba iškraipo jų vizualines savybes. Pagrindiniai YOLOv11m hiperparametrai pateikiami 2.4 lentelėje (žr. 2.4 lent.).

2.4 lentelė. YOLOv11m mokymo hiperparametrai

Hiperparametras	Reikšmė
Maksimalus epochų skaičius	150
Early stopping patience	20
Batch dydis	6
Įvesties raiška	896×896 px
Optimizatorius	AdamW
Bazinis mokymosi greitis	0,001
Momento koeficientas	0,937
IoU slenkstis	0,6
HSV atspalvio augmentacija	0,015
HSV soties augmentacija	0,5
HSV ryškumo augmentacija	0,3
Horizontalus atspindys	0,5
Auto augmentacija	Išjungta
Klasių skaičius	2
Parametrų skaičius	~20–25M

Architektūriniu požiūriu YOLOv11m naudoja C3K2 blokus vietoj ankstesnių C2f blokų, o tai pagerina gradientų sklaidą ir skaičiavimo efektyvumą. C2PSA modulis integruoja kanalų ir erdvinę informaciją su daugiagalvio dėmesio mechanizmais, kas ypač naudinga smulkių defektų aptikimui, nes modelis gali koncentruotis į specifines vaizdo sritis nepriklausomai nuo jų padėties. Be inkarų (angl. *anchor-free*) aptikimo metodas, pristatytas YOLOv8 ir išlaikytas YOLOv11, eliminuoja rankinio inkarų parametrų derinimo poreikį, supaprastinant modelio pritaikymą skirtingų formų ir dydžių defektams[13].

Lyginant su SSD ir Faster R–CNN konfigūracijomis, YOLOv11m naudoja AdamW optimizatorius vietoj SGD, taikoma gerokai didesnė įvesties raiška, lankstesnis mokymo proceso

valdymas per ankstyvo sustabdymo mechanizmą bei tiesioginė duomenų augmentacijos kontrolė Python skripte.

2.2.6. PatchCore konfigūracija

PatchCore modelis įgyvendintas naudojant „Intel OpenVINO“ ekosistemos biblioteką „Anomalib“ (v2.3.3). Šis įrankis suteikia standartizuotas įvairių anomalijų aptikimo algoritmų implementacijas ir užtikrina GPU aparatinį palaikymą. Skirtingai nei prižiūrimojo mokymosi modeliai, PatchCore nereikalauja anotuotų defektų pavyzdžių, modelis mokosi tik iš nedefektuotų butelių vaizdų, sudarydamas nedefektuoto paviršiaus savybių atminties banką (angl. *memory bank*). Dėl anomalib bibliotekos priklausomybių suderinamumo reikalavimų PatchCore treniruotas atskiroje Python 3.12 conda aplinkoje, kuri skyrėsi nuo prižiūrimojo mokymosi modeliams naudotos Python 3.9 aplinkos. GPU spartinimui naudotas PyTorch su CUDA palaikymu, o atminties valdymui taikytas *expandable_segments* mechanizmas, leidžiantis efektyviau išnaudoti ribotą GPU atmintį. Duomenų struktūra PatchCore modeliui parengta naudojant anomalib Folder duomenų modulį. Kadangi anomalib reikalauja specifinės katalogų struktūros, sukurtas atskiras Python skriptas (*roboflow_to_anomalib.py*), konvertuojantis Roboflow COCO formato eksportą į anomalib suderinamą struktūrą. Konversijos metu defektuotų butelių vaizdai paskirstyti į atskirus katalogus, o iš bounding box anotacijų automatiškai sugeneruotos binarinės kaukės (angl. *binary masks*) – baltos stačiakampio formos sritys defektų vietose ant juodo fono. Šios kaukės naudojamos pikselių lygio lokalizacijos vertinimui, tačiau nenaudojamos modelio mokymo procesui. PatchCore modelis sukonfigūruotas naudojant ResNet18 kaip požymių ištraukimo tinklą, išgaunant požymius iš *layer2* sluoksnio. *Layer2* pasirinkimas pagrįstas aparatūros apribojimais – gilesni sluoksniai (*layer3* ir *layer4*) generuoja gerokai didesnės dimensijos požymių vektorius (512–1 536 matmenų), o tai galėtų sukelti CUDA atminties perpildymo klaidas turimoje įrangoje. *Layer2* generuoja kompaktiškus 128 matmenų požymių vektorius, leidžiančius efektyviai sudaryti atminties banką (angl. *memory bank*) iš 2 147 mokymo nuotraukų. Atrankos poaibio (angl. *coreset*) koeficientas nustatytas į 10 %, t. y. iš visų ištrauktų požymių fragmentų išlaikoma tik reprezentatyviausia 10 % imtis. Toks sumažinimas būtinas dėl dviejų priežasčių: pirma, 2 147 nuotraukos su *layer2* požymiais generuoja milijonus požymių fragmentų, kurių pilnas saugojimas viršytų turimos operatyviosios atminties ribas; antra, atrankos poaibio algoritmas išlaiko reprezentatyviausius požymius, todėl tikslumo nuostolis minimalus. Anomalijoms aptikti naudotas 9 artimiausių kaimynų metodas (*num_neighbors* = 9) – kiekvienas testinis požymių fragmentas lyginamas su 9 artimiausiais atminties banko elementais, o didžiausia atstumų reikšmė naudojama kaip anomalijos įvertis. Pagrindiniai PatchCore konfigūracijos parametrai pateikiami 2.5 lentelėje (žr. 2.5 lent.).

2.5 lentelė. PatchCore konfigūracijos parametrai

Parametras	Reikšmė
Biblioteka	anomalib 2.3.3
Stuburo tinklas	ResNet18
Požymių sluoksnis	layer2
Požymių dimensija	128
Coreset sampling	10 %
Artimiausių kaimynų skaičius	9
Mokymo epochos	1
Batch dydis (mokymas)	4
Batch dydis (vertinimas)	4
Mokymo nuotraukų skaičius	2147
Butelių tipai	2 (giros ir Coca-Cola)
Fono pašalinimas	Taip

Vienas reikšmingų PatchCore aspektų šiame tyrime yra atminties banko (angl. *memory bank*) sudarymo trukmė. Nors formaliai PatchCore treniruojamas vieną epochą, kadangi mokymosi procesas apsiriboja požymių išgavimu ir atminties banko sudarymu be iteratyvaus parametru optimizavimo, faktinė proceso trukmė siekė apie 6 valandas. Tokį laiką lėmė keli veiksniai: 2 147 mokymo nuotraukų apdorojimas naudojant ResNet18 reikalauja reikšmingų skaičiavimo išteklių, o atrankos poaibio algoritmas turi apdoroti milijonus požymių fragmentų prieš suformuodamas galutinį 10 % atminties banką. Tai vienas praktinių PatchCore apribojimų lyginant su prižiūrimojo mokymosi modeliais, nors inferavimo fazė greitai, pradinio atminties banko sudarymas gali užtrukti ilgai esant dideliems mokymo duomenų rinkiniams[15].

PatchCore modelio vertinimui naudotas atskiras testavimo duomenų rinkinys, apimantis tiek gerus, tiek brokuotus butelių vaizdus. Gerų butelių vaizdai (22 nuotraukos) naudoti klaidingai teigiamų aptikimų (angl. *false positives*) vertinimui, o brokuotų butelių vaizdai (91 nuotrauka) naudoti defektų aptikimo jautrumui (angl. *recall*) įvertinti. Vertinimui taikytos šios metrikos: vaizdo lygmens AUROC (anomalijų klasifikavimo tikslumui įvertinti), pikselių lygmens AUROC (lokalizacijos tikslumui įvertinti), taip pat standartiniai tikslumo (angl. *precision*), jautrumo (angl. *recall*) ir F1 rodikliai. Iš ribojančių stačiakampių anotacijų sugeneruotos binarinės kaukės naudotos kaip tikrosios reikšmės (angl. *ground truth*) pikselių lygmens vertinimui.

2.3. Perkėlimo mokymasis ir mokymas nuo nulio

Perkėlimo mokymasis (angl. *transfer learning*), tai metodas, kurio metu modelis, iš anksto apmokytas su dideliu duomenų rinkiniu, pritaikomas naujam uždaviniui naudojant gerokai mažesnę specializuotą duomenų rinkinį[46]. Šiame tyrime visi trys prižiūrimojo mokymosi modeliai (SSD MobileNetV2 FPN, Faster R–CNN ResNet50 ir YOLOv11m) apmokyti perkėlimo mokymosi būdu, naudojant COCO duomenų rinkiniu iš anksto apmokytus svorius. Perkėlimo mokymosi privalumai stiklinių butelių defektų aptikimo kontekste yra dvejopi. Pirma, naudojant COCO duomenų rinkinyje iš anksto apmokytus modelius, jau išvystyti žemesnio lygio požymių išgavimo gebėjimai. Antra, perkėlimo mokymasis leidžia pasiekti aukštą tikslumą naudojant gerokai mažesnius duomenų rinkinius nei mokant modelį nuo nulio, o tai ypač svarbu šiame tyrime, kuriame turimas duomenų rinkinys apima tik 708 nuotraukas[20]. Praktinis perkėlimo mokymosi įgyvendinimas skyrėsi priklausomai nuo modelio. SSD MobileNetV2 FPN ir Faster R–CNN atvejais perkėlimo mokymasis buvo aktyvuojamas nurodant iš anksto apmokyto modelio kontrolinio taško (angl. *checkpoint*) kelią konfigūracijos failo *fine_tune_checkpoint* parametre ir nustatant *fine_tune_checkpoint_type*: „*detection*“. YOLOv11m atveju perkėlimo mokymasis įgyvendintas naudojant *yolo11m.pt*, iš anksto apmokytų svorių failą, vietoj architektūros aprašo *.yaml* failo.

Mokymas nuo nulio (angl. *training from scratch*) reiškia modelio treniravimą su atsitiktinai inicializuotais parametrais, be jokių iš anksto apmokytų svorių. Šis metodas reikalauja gerokai daugiau duomenų ir mokymo laiko, tačiau leidžia modeliui išmokti požymius tiesiogiai iš tikslinės srities duomenų be jokių iš anksto suformuotų prielaidų[14]. Šiame tyrime mokymas nuo nulio taikytas siekiant empiriškai įvertinti perkėlimo mokymosi naudą riboto duomenų kiekio sąlygomis. Visi trys modeliai (SSD, Faster R–CNN ir YOLOv11) apmokyti nuo nulio, naudojant tuos pačius hiperparametrus kaip ir perkėlimo mokymosi atveju, sąmoningai išlaikant kitus veiksnus pastovius. Toks metodologinis sprendimas užtikrina, kad rezultatų skirtumai tarp perkėlimo mokymosi ir mokymo nuo nulio gali būti priskirti būtent iš anksto apmokytų svorių naudojimo poveikiui, o ne hiperparametrų skirtumams. SSD ir Faster R–CNN modeliams mokymas nuo nulio įgyvendintas užkomentuojant šias konfigūracijos failo eilutes:

fine_tune_checkpoint: „...*checkpoint/ckpt-0*“

fine_tune_checkpoint_type: „*detection*“

YOLOv11m atveju mokymas nuo nulio aktyvuotas pakeičiant *yolo11m.pt* į *yolo11m.yaml*, taip modelis inicializuojamas su atsitiktiniais svoriais išlaikant tą pačią architektūrą. Svarbu paminėti, kad mokymo nuo nulio eksperimentas atliktas tik su V3 anotavimo variantu (su fonu, bottle ir defect anotacijos) kaip baziniu scenarijumi. Perkėlimo mokymosi naudos įvertinimui pakanka vieno reprezentatyvaus anotavimo varianto, o visų trijų variantų kartojimas su mokymu nuo nulio reikštų neproporcingas laiko sąnaudas nesuteikiant papildomos mokslinės vertės.

2.4. Anotavimo variantų tyrimas

Vienas pagrindinių šio tyrimo metodologinių aspektų yra sisteminga trijų skirtingų anotavimo strategijų lyginamoji analizė, siekiant nustatyti optimalų duomenų paruošimo būdą stiklinių butelių defektams aptikti. Visi trys variantai testuoti su visais prižiūrimojo mokymosi modeliais naudojant identišką perkėlimo mokymosi konfigūraciją, kad rezultatų skirtumai galėtų būti priskirti išskirtinai anotavimo strategijos efektui.

V1 Fono pašalinimas su defektų anotacijomis apėmė visų 708 mokymo rinkinio nuotraukų fono pašalinimą naudojant rembg biblioteką, pagrįstą U^2 -Net modeliu[25]. Po fono pašalinimo nuotraukose liko tik butelio silueta neutraliame fone, o anotacijos žymėtos išskirtinai defektams. Manoma, jog pašalinus foną, modelis gali sutelkti visą dėmesį į butelio paviršių, eliminuojant fono triukšmą, kurį sukelia atspindžiai, šešėliai ir skirtingos aplinkos spalvos. Fono pašalinimas atliktas prieš anotavimo etapą, siekiant sudaryti geresnes sąlygas tiksliai identifikuoti defektų ribas. Pašalinus foną, galima aiškiau įvertinti defektų kontūrus ant homogeniško butelio paviršiaus. Tai ypač svarbu skaidriems, bespalviams buteliams, kurių defektai originaliuose vaizduose dažnai sunkiai atskiriami nuo šviesos lūžių ir atspindžių. Šio varianto trūkumas yra fono pašalinimo proceso integravimas į realią gamybinę liniją reikalauja papildomo skaičiavimo žingsnio kiekvienai nuotraukai prieš defektų aptikimą. Tačiau rembg biblioteka su U^2 -Net modeliu užtikrina pakankamą apdorojimo greitį realiojo laiko taikymams[25].

V2 Originalios nuotraukos su defektų anotacijomis variantas naudojo nuotraukas su fonu, žymint išskirtinai defektus be butelio klasės. Tai metodologiškai paprasčiausias ir literatūroje dažniausiai aprašomas požiūris[17]. Modelis šiuo atveju mokosi tiesiogiai aptikti defektus visame vaizde, nepriklausomai nuo butelio padėties ar ribų. Šio varianto privalumas yra minimalus duomenų paruošimo sudėtingumas, nereikalaujantis papildomo fono pašalinimo žingsnio. Tačiau tikėtinas trūkumas, kad modelis turi mokytis ignoruoti fono triukšmą, kurį sukelia skirtingos fotografavimo aplinkos, atspindžiai ant stiklo paviršiaus ir šviesos lūžiai, o tai gali apsunkinti smulkių defektų aptikimą.

V3 Originalios nuotraukos su butelio ir defektų anotacijomis variantas naudojo originalias nuotraukas su fonu, papildomai žymint ir patį butelį kaip atskirą „bottle“ klasę, sukuriant dviejų klasių anotacijų struktūrą. Šio varianto loginė prielaida, jog jei modelis tiksliai lokalizuoja butelį, defektų paieška gali būti efektyviai apribota butelio ribojančio langelio erdve, o tai turėtų sumažinti klaidingai teigiamų aptikimų skaičių už butelio ribų. Praktiškai šis variantas reikalauja papildomo anotavimo darbo, kiekviename vaizde reikia pažymėti ne tik defektus, bet ir patį butelį. Tačiau butelio anotavimas greitas procesas dėl aiškiai apibrėžtų butelio ribų, todėl papildomos laiko sąnaudos minimalios. Svarbu paminėti metodologinį iššūkį vertinant šio varianto rezultatus, kadangi modelis aptinka dvi klases (butelis ir defektas), bendras mAP rodiklis dirbtinai išaukštinamas butelio klasės

įtakos. Butelio aptikimas yra trivialiai paprastas uždavinys modeliui, nes butelis yra didelis, aiškiai matomas objektas kiekviename vaizde, todėl butelio klasės mAP artimas 100 %. Dėl šios priežasties šio varianto rezultatų analizėje naudojami išskirtinai defektų klasės rodikliai, atskirai apskaičiuoti iš mokymo .log failų, eliminuojant butelio klasės įtaką bendram mAP vidurkiui. Trijų variantų metodologinis palyginimas pateikiamas 2.6 lentelėje (žr. 2.6 lent.).

2.6 lentelė. Anotavimo variantų metodologinis palyginimas

Charakteristika	V1 Be fono	V2 Su fonu	V3 Butelis+defektas
Fono pašalinimas	Taip	Ne	Ne
Klasių skaičius	1 (defect)	1 (defect)	2 (bottle, defect)
Papildomas apdorojimas	Taip (rembg)	Ne	Ne
Anotavimo sudėtingumas	Vidutinis	Žemas	Vidutinis
Tinkamumas realiam laikui	Reikia fono pašalinimo	Tiesioginis	Tiesioginis
Bazinis scenarijus	Ne	Taip	Ne

Lentelėje matomi esminiai trijų variantų skirtumai. V1 variantas reikalauja papildomo fono pašalinimo žingsnio, tačiau suteikia modeliui švaresnį įvesties vaizdą. V2 yra paprasčiausias paruošimo požiūriu ir todėl pasirinktas baziniu scenarijumi. V3 papildomai anotuoja butelį kaip atskirą klasę, kas teoriškai turėtų pagerinti defektų lokalizacijos tikslumą, tačiau apsunkina rezultatų interpretavimą dėl dviejų klasių mAP vidurkio. Kiekvieno varianto eksperimentiniai rezultatai ir išsamus palyginimas pateikiami eksperimentų skyriuje.

2.5. Modelių vertinimo metodika

Modelių veiksmingumo įvertinimui naudotos metrikos, atspindinčios skirtingų vertinimo protokolų galimybes. Metrikų pasirinkimas grindžiamas objektų aptikimo srities standartine praktika bei naudotų bibliotekų (TensorFlow Object Detection API ir Ultralytics) teikiamų metrikų skirtumais.

mAP@0.5 (*mean Average Precision at IoU 0.5*) yra pagrindinė ir visų modelių bendra tikslumo metrika, skaičiuojama kaip vidutinis tikslumas esant 0,5 IoU (*Intersection over Union*) slenksčiui. IoU matuoja aptikto ribojančio langelio ir tikrojo anotacijos langelio persidengimo santykį, reikšmė 0,5 reiškia, kad aptiktas objektas laikomas teisingai aptiktu, jei jo ribojantis langelis persidengė su tikruoju bent 50 %. Ši metrika plačiausiai naudojama objektų aptikimo tyrimuose ir leidžia tiesiogiai palyginti rezultatus su literatūroje paskelbtais duomenimis[7].

AR@100 (*Average Recall at 100 detections*), antrinė metrika, naudojama SSD MobileNetV2 FPN ir Faster R–CNN ResNet50 modeliams, pateikiama TensorFlow Object Detection API COCO metrics protokolo. AR@100 matuoja vidutinį aptikimo jautrumą esant maksimaliam 100 aptikimų per vaizdą. Aukšta reikšmė rodo, kad modelis praleidžia mažai tikrų defektų.

Recall, tai aptiktų defektų jautrumas, skaičiuojamas kaip teisingai teigiamų aptikimų (TP) santykis su visais tikraisiais defektais (TP + FN). Ši metrika tiesiogiai pateikiama YOLOv11m modelio Ultralytics validavimo protokole. Aukšta Recall reikšmė rodo, kad modelis praleidžia mažai tikrų defektų, o tai svarbu kokybės kontrolės kontekste, kur praleistas defektas gali turėti rimtų pasekmių[24].

Svarbu paminėti, kad dėl skirtingų vertinimo protokolų AR@100 ir Recall nėra tiesiogiai palyginamos metrikos. AR@100 naudojama TensorFlow modeliams, recall – YOLOv11m. Abiejų metrių esmė yra įvertinti modelio gebėjimą aptikti kuo daugiau tikrų defektų, tačiau jų skaičiavimo metodika skiriasi. Šio tyrimo kontekste abu rodikliai laikomi svarbesniais nei tikslumas (*Precision*), kadangi praleistas defektas (FN) žalingesnis nei klaidingas broko signalas (FP), pastarasis gali būti patikrintas pakartotinai, tuo tarpu praleistas defektas pasiekia galutinį vartotoją.

SSD MobileNetV2 FPN ir Faster R–CNN ResNet50 modelių vertinimui naudoti COCO *detection metrics*, integruoti į TensorFlow Object Detection API. Vertinimas atliekamas periodiškai mokymo metu, kas 1 000 žingsnių, naudojant validavimo rinkinį (104 nuotraukos). Galutiniai rezultatai paimti ties 10 000 mokymo žingsnių, kai modelių veiksmingumas stabilizavosi. TensorBoard įrankis naudotas mokymo proceso vizualizavimui ir stebėjimui realiuoju laiku, loss kreivės, mAP dinamika ir Recall rodikliai stebėti visų treniravimų metu, leidžiant anksti pastebėti persimokymą ar konvergavimo problemas.

YOLOv11m modelio vertinimas atliekamas automatiškai Ultralytics bibliotekos. Validavimo metrikos skaičiuojamos po kiekvienos epochos, o galutiniai rezultatai paimti iš geriausių svorių failo (*best.pt*), kurį Ultralytics automatiškai išsaugo pagal aukščiausią validavimo mAP@0.5:0.95 reikšmę.

PatchCore vertinimui naudotos anomalijų aptikimui būdingos metrikos, skirtingos nuo objektų aptikimo metrių. Vaizdo lygmens AUROC (*Area Under the ROC Curve*) matuoja modelio gebėjimą atskirti gerus ir brokuotus butelius vaizdo lygyje, reikšmė 1,0 reiškia tobulą klasifikaciją. Pikselių lygmens AUROC matuoja lokalizacijos tikslumą pikselių lygyje, lyginant modelio generuojamą anomalijų šilumos žemėlapi su binarinėmis kaukėmis iš bounding box anotacijų. Papildomai skaičiuoti standartiniai Precision, Recall ir F1 rodikliai vaizdo klasifikacijos lygyje leidžia tiesiogiai palyginti PatchCore su prižiūrimojo mokymosi modeliais klasifikacijos (brokuotas/nebrokuotas) aspektu, nors tiesioginis mAP palyginimas neįmanomas dėl skirtingų vertinimo paradigmos.

Modelių inferavimo laikas matuotas ant tos pačios aparatinės įrangos (RTX 3060 Laptop GPU) visiems modeliams, užtikrinant palyginamumą. Kiekvienam modeliui apdorota pakankama nuotraukų imtis stabiliam vidurkio ir standartinio nuokrypio įvertinimui. Inferavimo laikas apima visą apdorojimo grandinę ir išreiškiamas milisekundėmis vienam vaizdui bei kadrais per sekundę (FPS).

2.7 lentelė. Naudotų vertinimo metrikų suvestinė

Metrika	Modeliai	Paskirtis
mAP@0.5	SSD, Faster R-CNN, YOLOv11	Pagrindinis tikslumo rodiklis
mAP@0.5:0.95	SSD, Faster R-CNN, YOLOv11	Lokalizacijos tikslumas
Precision	Visi	Klaidingų teigiamų kontrolė
Recall	Visi	Praleistų defektų kontrolė
Image AUROC	PatchCore	Klasifikacijos tikslumas
Pixel AUROC	PatchCore	Lokalizacijos tikslumas
F1 Score	PatchCore	Balansuotas tikslumas
Inference laikas (ms)	Visi	Greičio vertinimas
FPS	Visi	Realaus laiko tinkamumas

Lentelėje (žr. 2.7 lent.) apibendrinti visi šiame tyrime naudoti vertinimo rodikliai. mAP@0.5 yra vieninga metrika, leidžianti palyginti visus prižiūrimojo mokymosi modelius tarpusavyje. AR@100 ir Recall panašūs aptikimo jautrumo rodikliai, tačiau gaunami iš skirtingų vertinimo protokolų, todėl nėra tiesiogiai palyginami tarpusavyje. PatchCore vertinamas atskiromis anomalijų aptikimui būdingomis metrikomis, o inferavimo laikas ir FPS rodikliai matuoti visiems modeliams vienodomis aparatūros sąlygomis.

2.6. Trumpas skyriaus apibendrinimas

Šiame skyriuje aprašyti keturi atirinkti giliojo mokymosi modeliai (SSD MobileNetV2 FPN, Faster R-CNN ResNet50, YOLOv11m ir PatchCore) bei jų konfigūravimo principai stiklinių butelių defektų aptikimo uždaviniui. Prižiūrimojo mokymosi modeliai sukonfigūruoti naudojant TensorFlow Object Detection API ir Ultralytics aplinkas. Kiekvienam modeliui parinkti tinkami hiperparametrai, mokymo strategijos ir anotavimo variantai. Prižiūrimojo mokymosi modeliams išbandytas tiek perkėlimo mokymasis iš ImageNet svorių, tiek mokymasis nuo nulio, siekiant įvertinti perkėlimo mokymosi naudą riboto duomenų kiekio sąlygomis. PatchCore modelis konfigūruotas neprižiūrimojo

mokymosi paradigma, mokymas atliktas išskirtinai su nebrokuotų butelių vaizdais, iš kurių suformuotas savybių bankas anomalijoms aptikti. Taip pat aprašytos programinės bibliotekos bei jų integracija į bendrą eksperimentinę aplinką. Kiekvieno modelio architektūra ir mokymo aplinka parinkta atsižvelgiant į turimus kompiuterinius išteklius, duomenų kiekį ir tyrimo tikslus, siekiant užtikrinti sąžiningą modelių palyginimą.

3. DUOMENŲ PARUOŠIMAS GAMINIO BROKO APTIKIMO METODŲ EKSPERIMENTINIAMS TYRIMUI

3.1. Viešųjų duomenų analizė ir apribojimai

Duomenų rinkinio kokybė ir atitikimas tiriamajam domenui yra vienas svarbiausių veiksnių, lemiančių giliojo mokymosi modelio efektyvumą. Prieš sudarant nuosavą duomenų rinkinį, atlikta išsami viešai prieinamų stiklo defektų duomenų rinkinių analizė, siekiant įvertinti jų tinkamumą stiklinių butelių defektams aptikti. Analizės metu išnagrinėti keli Roboflow platformoje prieinami duomenų rinkiniai. „Glass Defects“ (Özkan, Roboflow Universe) apėmė įvairių stiklo gaminių defektų vaizdus, tačiau objektai buvo plokšti stiklo paviršiai, o ne trimatės stiklinių butelių formos. Antrasis nagrinėtas rinkinys buvo „Glass Defect Detection“ (Roboflow Universe), sudarytas iš juodai baltų stiklo oro burbuliukų vaizdų, fotografuotų kontroliuojamomis pramoninėmis sąlygomis su specializuota apšvietimo įranga. Trečiasis rinkinys „Glass Defects“ (SWQC, Roboflow Universe) bandytas praturtinti nuosavomis stiklinių butelių nuotraukomis ir naudotas pradiniais modelių apmokymo bandymams.

Visi trys bandymai parodė nepatenkinamus rezultatus dėl kelių esminių priežasčių. Pirma, yra reikšmingas domenų neatitikimas (angl. *domain mismatch*) tarp viešai prieinamų duomenų rinkinių ir tiriamojo objekto. Stiklo lapo ar pramoninio stiklo plokštės plyšys vizualiai skiriasi nuo stiklinio butelio paviršiaus įbrėžimo dėl skirtingo objekto kreivumo, skaidrumo, atspindžių pasiskirstymo ir defekto tekstūros ir tas lemia, kad modelis, apmokytas su vienu gaminių vaizdais, negali sėkmingai apibendrinti žinių kitų gaminių defektams atpažinti[20]. Antra, kai kuriuose viešuosiuose rinkiniuose naudotos fotografavimo sąlygos su specializuotu pramoniniu apšvietimu, kontroliuojami fonu, didele vaizdo raiška, kas gerokai skyrėsi nuo eksperimento sąlygų, o tai dar labiau padidino domenų atotrūkį. Atlikti bandymai tai patvirtino: nė vienas iš tirtų modelių, apmokytas su viešaisiais duomenimis, nesugebėjo aptikti defektų namų sąlygomis nufotografuotose buteliuose, o vertinimo rodikliai buvo statistiškai nereikšmingi. Domenų neatitikimas vizualiai iliustruojamas 3.1 paveikslėlyje (žr. 3.1 pav.).



3.1 pav. Viešųjų duomenų rinkinių ir nuosavo rinkinio vaizdų palyginimas[8]

Viešųjų duomenų rinkinių ir nuosavo rinkinio vaizdų palyginimas: automobilio stiklo lūžis iš „Glass Defects“ (Özkan, Roboflow Universe) duomenų rinkinio – plokščio stiklo paviršiaus defektas; stiklainio defekto aptikimas iš „Glass Defect Detection“ (Roboflow Universe) duomenų rinkinio – juodai baltas vaizdas su oro burbuliuku, gautas specializuotomis pramoninėmis sąlygomis; nuosavo duomenų rinkinio pavyzdys – rudas stiklinis butelis, fotografuotas kontroliuojamomis nekomercinėmis sąlygomis. Paveiksle aiškiai matomas domenų neatitikimas tarp viešųjų duomenų rinkinių ir tyrimo objekto, automobilio stiklo lūžio vizualinės charakteristikos skiriasi nuo stiklinio butelio paviršiaus defektų. Analogiškai, pramoninėmis sąlygomis gauti juodai balti vaizdai yra nepakankamas vizualinis atitikimas spalvotiems, kontroliuojamomis natūraliomis sąlygomis fotografuotiems buteliams.

Remiantis šia analize, priimtas sprendimas sudaryti nuosavą duomenų rinkinį kontroliuojamomis, nekomercinėmis sąlygomis. Toks sprendimas atitinka mokslinėje literatūroje pateikiamas rekomendacijas: Latina[17] pabrėžia, kad fotografavimo sąlygų kontroliavimas ir duomenų rinkinio sudarymas konkrečiam taikymo domenui yra būtina sąlyga aukštam tikslumui pasiekti. Kartu nuosavo rinkinio sudarymas leido įgyti praktinės patirties visais duomenų paruošimo etapais, nuo fotografavimo metodikos iki anotavimo.

3.2. Nuosavo duomenų rinkinio kūrimo sprendimo pagrindimas

Nustačius, kad viešai prieinami duomenų rinkiniai neatitinka tyrimo reikalavimų, svarstyty dvi alternatyvios duomenų gavimo strategijos: sintetinio duomenų rinkinio generavimas naudojant kompiuterinės grafikos įrankius arba nuosavo rinkinio sudarymas realiai fotografuojant fizinius objektus. Abiejų strategijų privalumai ir trūkumai įvertinti prieš priimant galutinį sprendimą.

Sintetinių duomenų generavimas yra metodas, kurio metu defektai dirbtinai pridedami prie esamų objektų nuotraukų naudojant vaizdo apdorojimo ar trimačio modeliavimo įrankius. Šiuolaikinės platformos (pvz., „Blender“ ar specializuoti duomenų augmentacijos įrankiai) leidžia ant realaus objekto nuotraukos uždėti sintetinį defektą ir sugeneruoti šimtus skirtingų variantų, keičiant kampo perspektyvą, apšvietimo intensyvumą, defekto dydį ir vietą. Šis metodas ypač populiarus, kai realių brokuotų gaminių kiekis yra labai nedidelis arba kai defektų kūrimas fiziškai pavojingas ar brangus[33]. Tačiau sintetiniai defektai turi esminį trūkumą, jie ne visada tiksliai atspindi realių defektų vizualines savybes. Stiklinio butelio įbrėžimo šviesos atspindys, oro burbuliuko skaidrumo charakteristikos ar mechaninio lūžio krašto tekstūra yra itin sudėtingi fiziniai reiškiniai, kuriuos sunku tiksliai atkurti sintetiškai. Dėl šios priežasties modeliai, apmokyti su sintetiniais defektais, gali patirti domenų neatitikimą testuojant su realiais objektais, tai yra problema, analogiška tai, su kuria susidurta naudojant viešuosius duomenų rinkinius.

Atsižvelgiant į šiuos apribojimus, priimtas sprendimas sudaryti nuosavą duomenų rinkinį fotografuojant fizinius stiklinius butelius su dirbtinai sukurtais, tačiau fiziškai autentiškais defektais. Šis pasirinkimas grindžiamas keliais argumentais. Fiziškai sukurti defektai kaip įbrėžimai, oro burbuliukų imitacijos, nudaužtos galvutės, vizualiai atitinka realius gamybos defektus, todėl modelis mokosi atpažinti autentiškas defektų charakteristikas, o ne sintetines aproksimacijas. Fotografavimas kontroliuojamomis, tačiau nekomercinėmis sąlygomis atspindi tikrąjį tyrimo tikslą, įvertinti, ar giliojo mokymosi metodai gali būti efektyviai taikomi be specializuotos pramoninės infrastruktūros[17]. Svarbu dar tai, jog nuosavo rinkinio sudarymas suteikė galimybę įgyti praktinės patirties visais duomenų paruošimo etapais, nuo fotografavimo metodikos pasirinkimo iki anotavimo strategijų, o tai yra svarbi šio tyrimo praktinė vertė, tiesiogiai prisidedanti prie tyrimo rezultatų interpretavimo.

Verta paminėti, kad pasirinktas metodas turi ir apribojimų. Nuosavas duomenų rinkinys mažesnis nei pramoniniai rinkiniai, o fotografavimo sąlygos negali visiškai atkartoti gamybinės linijos aplinkos. Šie apribojimai sąmoningai priimti kaip tyrimo ribos, atitinkančios tikslą įvertinti giliojo mokymosi metodų veiksmingumą ribotų išteklių sąlygomis, o ne maksimaliai optimizuoti modelio tikslumą pramoninėms sąlygoms.

3.3. Nuosavo rinkinio kūrimo metodika

Kadangi viešai prieinami duomenų rinkiniai pasirodė netinkami dėl domenų neatitikimo, nuspręsta sudaryti nuosavą vaizdų duomenų rinkinį. Šis sprendimas leido tiksliai kontroliuoti fotografavimo sąlygas, užtikrinti defektų reprezentatyvumą ir įgyti praktinės patirties visais duomenų paruošimo etapais. Analogišką metodologinį pasirinkimą aprašo Latina[17], pabrėždami, kad kontroliuojamomis sąlygomis sudarytas nuosavas rinkinys leidžia pasiekti aukštesnį modelio tikslumą nei nevienalyčiai viešieji šaltiniai.

Tyrimui pasirinkti aštuonių skirtingų tipų stikliniai buteliai, sąmoningai apimantys kuo platesnę paviršiaus, spalvos ir formos įvairovę. Toks pasirinkimas grindžiamas siekiu užtikrinti modelio apibendrinimo gebėjimą, kad apmokytas modelis sugebėtų aptikti defektus ne tik vieno tipo buteliuose, bet ir skirtingomis vizualinėmis sąlygomis. Butelių charakteristikos ir jų kiekiai duomenų rinkinyje pateikiami 3.1 lentelėje (žr. 3.1 lent.).

3.1 lentelė. Tyrime naudotų butelių charakteristikos

Nr.	Aprašymas	Spalva	Paviršius	Kiekis
1	Maisto produktų mažo stiklainio tipo buteliukas	Bespalvis / skaidrus	Lygus	65
2	Vyno butelis	Tamsiai žalias	Lygus	59
3	Giros buteliukas	Rudas	Nelygus (išlietos raidės ir ornamentai)	207
4	Giros butelis	Juodas	Nelygus (daugiausiai nelygi forma)	49
5	Coca–Cola buteliukas	Bespalvis / skaidrus	Nelygus (daug įdubimų, ryškus šviesos lūžis)	55
6	Skaidrus buteliukas	Bespalvis / skaidrus	Lygus	79
7	Tamsus rudas buteliukas	Rudas	Lygus	122
8	Skaidrus buteliukas (kita forma)	Bespalvis / skaidrus	Lygus	72
Iš viso				708

Butelių vizualinės savybės turėjo tiesioginę įtaką defektų matomumui nuotraukose. Rudos (Nr. 3, Nr. 7) ir tamsiai žalios (Nr. 2) spalvos buteliai pasižymėjo geriausiu defektų kontrastiškumu, defektai ant jų paviršiaus matėsi aiškiai net ir esant neoptimaliam apšvietimui. Juodos spalvos butelis (Nr. 4) taip pat užtikrino gerą kontrastą, tačiau jo nelygi forma sukėlė papildomų iššūkių anotavimo metu. Skaidrūs bespalviai buteliai (Nr. 1, Nr. 5, Nr. 6, Nr. 8) buvo vizualiai sudėtingiausi: dėl medžiagos skaidrumo šviesa lūžta per visą butelio tūrį, sukurdamą atspindžius ir šviesos žaismą, kuriuos modelis gali painioti su defektais. Ypač ryškiai ši problema pasireiškė su Coca–Cola buteliuku (Nr. 5), kurio paviršiuje yra daug gamyklinių įdubimų, sukuriančių intensyvių šviesos lūžių, vizualiai panašų į įbrėžimo defektą.

Tyrimui sukurti trys defektų tipai, atspindintys realiai gamyboje ir eksploatacijos metu pasitaikančius pažeidimus. **Įbrėžimų** tipo defektai sukurti naudojant šlifluoklį (dildelę), imituoiant gamybinių ar eksploatacinių įbrėžimų įvairovę. Defektai daromi skirtingo intensyvumo, nuo lengvų, vos pastebimų paviršiaus pažeidimų iki gilesnių, aiškiai matomų įbrėžimų. Taip pat varijuotas defekto plotas: vienos linijos įbrėžimai ir platesnės pažeistos zonos. Šis defektų tipas sudaro didžiausią dalį duomenų rinkinyje iš viso anotuotos 728 įbrėžimų atvejų, o tai atspindi realią gamybinę situaciją, kurioje paviršiaus įbrėžimai yra dažniausiai pasitaikantis defektų tipas. **Oro burbuliukų** tipo defektai imituoja gamybos proceso metu stiklo susiformuojančius oro įtraukimus, kurie yra viena klasikinių

stiklo gamybos defektų kategorijų[44]. Eksperimento sąlygomis oro burbuliukai sukurti naudojant permatomą laką, kuris užteptas ant butelio paviršiaus suformuodavo vizualiai autentišką burbulo efektą. Iš viso duomenų rinkinyje anotuotos 280 oro burbuliukų instancijos. **Nudaužtų galvučių** defektų tipas reprezentuoja mechaninį butelio pažeidimą, butelio kaklelio ar angos krašto atlaužimą. Defektai sukurti fiziškai pažeidžiant butelius. Kadangi tai vizualiai didžiausias ir lengviausiai pastebimas defektas, duomenų rinkinyje jo atvejų skaičius mažiausias tik 62 atvejai, remiantis prielaida, kad dėl ryškaus vizualinio skirtumo modeliui šio tipo defektą atpažinti pakaks mažesnio mokymo pavyzdžių kiekio.

3.2 lentelė. Defektų pasiskirstymas duomenų rinkinyje

Defekto tipas	Defekto kiekis	Dalis (%)
Įbrėžimai (scratches)	728	68,0 %
Oro burbuliukai (air bubbles)	280	26,2 %
Nudaužtos galvutės (broken neck)	62	5,8 %
Iš viso	1 070	100 %

Duomenų rinkinyje pastebimas klasių disbalansas, atspindintis realią gamybinę situaciją kur įbrėžimai yra gerokai dažnesnis defektų tipas nei mechaniniu poveikiu sukelti pažeidimai. Klasių disbalansas yra žinomas iššūkis prižiūrimojo mokymosi modeliams[24], todėl šis aspektas aptariamas duomenų augmentacijos skyriuje. Atsižvelgiant į defektų buvimą, duomenų rinkinio 708 nuotraukos pasiskirstė taip: 215 butelių nuotraukų neturi jokio defekto, 337 nuotraukose matomas vienas arba du defektai, 123 nuotraukose trys arba keturi defektai, o 33 nuotraukose penki ir daugiau defektų.

3.3.1. Fotografavimo metodika ir aplinkos sąlygos

Duomenų rinkinio fotografavimo proceso standartizavimas svarbus siekiant užtikrinti vaizdų kokybę ir modelio mokymosi stabilumą. Netolygus apšvietimas, nekontroliuojamas fonas ar atsitiktinė kameros padėtis gali įnešti nereikalingą vaizdinį triukšmą, kuris neigiamai veikia modelio gebėjimą apibendrinti[17]. Šiame tyrime fotografavimo metodika apibrėžta siekiant išlaikyti sąlygų kontroliavimo ir natūralumo pusiausvyrą.

3.3.2. Fotografavimo įranga

Nuotraukoms daryti naudotos dvi kameros: Canon EOS 6D Mark II DSLR fotoaparatas ir iPhone 14 išmanusis telefonas. Abiejų kamerų vaizdo raiška viršija tyrimo metu naudotą mokymo raišką, vaizdai sumažinti iki 640×640 pikselių SSD ir Faster R-CNN modeliams bei iki 896×896 pikselių YOLOv11 modeliui. Dėl šios priežasties skirtinga abiejų kamerų raiška neturėjo esminės

įtakos galutinei duomenų rinkinio kokybei, abu šaltiniai buvo laikomi lygiaverčiais ir naudoti bendrame rinkinyje be atskiro žymėjimo.

3.3.3. Apšvietimo sąlygos

Visos nuotraukos darytos dienos metu, esant natūraliai dienos šviesai, papildomai įjungus LED lubines lempas. Kombinuotas natūralios ir dirbtinės šviesos naudojimas leido pasiekti tolygų ir pakankamą apšvietimo lygį, sumažinant šešėlių atsiradimo tikimybę, galinčią užmaskuoti smulkius paviršiaus defektus. Apšvietimo sąlygos išliko santykinai pastovios visų fotografavimo sesijų metu, kadangi visos nuotraukos darytos toje pačioje patalpoje. Toks apšvietimo modelis atitinka Latina[17] rekomendacijas dėl pastovaus apšvietimo svarbos defektų aptikimo duomenų rinkiniuose.

3.3.4. Kameros padėtis ir fotografavimo kampai

Fotografavimo metu buteliai visą laiką stovėjo ant grindų, kurios priklausomai nuo fotografavimo vietos buvo skirtingos spalvos ir tekstūros. Kamera laikyta rankose, be trikojo, o tai leido laisvai ir greitai keisti fotografavimo kampą. Buteliai fotografuoti iš keturių pagrindinių krypčių: iš priekio, iš šono (abiejų pusių) bei su nedideliais kampo nuokrypiais, šiek tiek iš viršaus ir šiek tiek iš apačios. Toks kampų įvairovės principas rekomenduojamas praktikoje siekiant padidinti modelio atsparumą skirtingoms objekto orientacijoms. Atstumas nuo kameros iki butelio nebuvo griežtai fiksuotas, tačiau palaikomas toks, kad butelis užimtų didžiąją dalį kadro ir būtų pagrindinis vaizdo objektas. Atstumas variavo apytiksliai 20 cm ribose, priklausomai nuo butelio dydžio ir fotografavimo kampo. Kiekvienas butelis nufotografuotas vidutiniškai 30–40 kartų iš skirtingų kampų ir pozicijų, sudarant nuoseklias vaizdų sekas. Viso rinkinio 708 nuotraukos gautos fotografuojant apie 20 fizinių butelių, priklausančių aštuoniems skirtingiems tipams. Svarbu paminėti, kad dalis nuotraukų daryta prieš defektų sukūrimą, taip gautos kokybiškų, nebrokuotų butelių nuotraukos, sudariusios svarbią rinkinio dalį (215 nuotraukų be defektų). Be to, kadangi defektai dažnai buvo tik vienoje butelio pusėje, fotografuojant iš priešingos pusės defektas į kadra nepatekdavo, tokiu būdu papildomai gauta kokybiškų vaizdų su tuo pačiu fiziniu buteliu. Tais atvejais, kai vienas butelis turėjo kelis skirtingų tipų defektus, visi matomi defektai anotuoti vienoje nuotraukoje.

3.3.5. Fotografavimo aplinka ir fono sąlygos

Siekiant įvertinti fono ir aplinkos įtaką defektų matomumui bei modelio veikimui, nuotraukos darytos trijose skirtingose aplinkose (žr. 3.2 pav.). **Aplinka Nr. 1** Rudos spalvos grindys, balta siena fone. Ši aplinka sukūrė vidutinį kontrastą tarp butelio ir fono. Pastebėta, kad šviesių, skaidrių butelių defektai šioje aplinkoje matėsi sunkiau dėl šviesos atspindžių, kurių intensyvumas artimas kai kurių defektų vizualinėms charakteristikoms. **Aplinka Nr. 2** Baltos grindys, balta siena fone. Ši aplinka sukūrė mažiausią kontrastą tarp šviesių, skaidrių butelių ir fono. Skaidrių butelių defektai šioje aplinkoje matėsi sunkiausiai, nes šviesos lūžiai ir atspindžiai susiliejo su baltu fonu, o kai kuriais atvejais juos buvo praktiškai neįmanoma vizualiai atskirti nuo tikrų defektų. Tamsių butelių defektai šioje aplinkoje, priešingai, matėsi aiškiai dėl ryškaus kontrasto tarp tamsaus butelio ir šviesaus fono. **Aplinka Nr. 3** Juodas padas (juodas paviršius), juodas fonas. Ši aplinka pasirodė optimaliausia skaidriems bespalviams buteliams fotografuoti. Juodas fonas eliminavo šviesos prasiskverbimo efektą ir leido defektams išsiskirti gerokai aiškiau. Tamsių butelių fotografavimui ši aplinka buvo mažiau tinkama dėl mažo kontrasto tarp butelio ir fono.



3.2 pav. Trys lyginamosios nuotraukos iš skirtingų fotografavimo aplinkų

Paveiksle matomas fono spalvos poveikis defektų matomumui: (1) aplinkoje skaidrių butelių defektai matomi vidutiniškai, (2) aplinkoje sunkiausiai dėl mažo kontrasto su baltu fonu, o (3) aplinkoje juodas fonas užtikrina geriausią skaidrių butelių defektų kontrastą. Skirtingos aplinkos sudaro skirtingą iššūkių lygį modeliui, o jų derinimas viename rinkinyje padeda modeliui išmokti atpažinti defektus įvairiomis apšvietimo ir fono sąlygomis. Analogišką išvadą pateikia Claypo ir kt.[6], pabrėždami, kad kintančios aplinkos sąlygos yra vienas pagrindinių iššūkių stiklo gaminių defektų aptikimo sistemoms.

3.3.6. Duomenų rinkinių struktūra

Šiame tyrime naudoti du atskiri duomenų rinkiniai, kurių kiekvienas skirtas skirtingai mokymosi paradigmai. Prižiūravimo mokymosi modeliai (SSD, Faster R-CNN, YOLOv11) ir anomalijų aptikimo modelis (PatchCore) turi skirtingus duomenų reikalavimus, todėl bendro rinkinio naudojimas abiem tikslams būtų neoptimalus[24].

Pagrindinis duomenų rinkinys, skirtas prižiūrimojo mokymosi modeliams apmokyti, sudarytas iš 708 nuotraukų, kuriose iš viso anotuoti 1 070 defektų atvejai. Rinkinys apima tiek defektų turinčius, tiek kokybiškus butelius, o tai būtina sąlyga prižiūrimojo mokymosi modeliams, nes jie turi mokytis atskirti brokuoto butelio vaizdą nuo gerojo[15]. Didžiąją dalį rinkinio sudaro nuotraukos su vienu arba dviem defektais (47,6 %), o tai atspindi realistišką gamybinę situaciją, kurioje dauguma defektų turinčių gaminių turi ribotą pažeidimų skaičių. Nuotraukos su penkiais ir daugiau defektų sudaro mažiausią dalį (4,7 %), tai nuotraukos, kuriose butelis tyčia pažeistas keliais skirtingais būdais vienu metu, siekiant padidinti rinkinio įvairovę. Nuotraukų pasiskirstymas pagal defektų skaičių pateikiamas 3.3 lentelėje (žr. 3.3 lent.).

3.3 lentelė. Nuotraukų pasiskirstymas pagal defektų skaičių

Defektų skaičius nuotraukoje	Nuotraukų kiekis	Dalis (%)
0 (be defektų)	215	30,4 %
1–2 defektai	337	47,6 %
3–4 defektai	123	17,4 %
5 ir daugiau defektų	33	4,7 %
Iš viso	708	100 %

Rinkinys suskirstytas į tris dalis pagal rekomenduojamą santykį[3]: mokymo rinkinys (~75 %), validavimo rinkinys (~20 %) ir testavimo rinkinys (~5 %). Po duomenų augmentacijos, išsamiau aprašytos vėlesniame skyriuje, mokymo rinkinio dydis išaugo iki 1 710 nuotraukų, validavimo rinkinyje liko 104 nuotraukos, o testavimo rinkinyje 34 nuotraukos. Visos nuotraukos sumažintos iki 640×640 pikselių SSD ir Faster R–CNN modeliams bei iki 896×896 pikselių YOLOv11 modeliui. Didesnis įvesties dydis YOLOv11 modeliui pasirinktas siekiant geriau išsaugoti smulkių defektų (siaurų įbrėžimų ir mažų oro burbuliukų) vizualines charakteristikas, kurios gali būti prarastos mažesnės raiškos vaizduose.



3.3 pav. Reprezentatyvios nuotraukos iš prižiūrimo mokymosi rinkinio, rodančios skirtingus defektų tipus

Anomalijų aptikimo modelio PatchCore apmokymas reikalauja kitokios duomenų rinkinio struktūros. Skirtingai nuo prižiūrimojo mokymosi, anomalijų aptikimo paradigma remiasi išskirtinai kokybiškų gaminių vaizdais. Modelis mokosi butelio išvaizdos reprezentacijos ir vėliau identifikuoja bet kokius nukrypimus nuo jos kaip anomalijas[43]. Dėl šios priežasties anomalijų aptikimo rinkinyje mokymo fazėje negali būti gaminių nuotraukų su defektais. Anomalijų aptikimo rinkinį sudaro 2147 nuotraukos, kuriose vaizduojami išskirtinai kokybiški stikliniai buteliai. Skirtingai nuo prižiūrimojo mokymosi rinkinio, anomalijų aptikimo eksperimentui pasirinkti du butelių tipai, rudas giros buteliukas (Nr. 3) ir skaidrus Coca-Cola buteliukas (Nr. 5). Šie du buteliai pasirinkti sąmoningai kaip kontrastingi pavyzdžiai: rudas butelis pasižymi geru defektų kontrastiškumu ir nelygiu paviršiumi su išlietomis raidėmis, o skaidrus Coca-Cola buteliukas pasižymi sudėtingesne vizualine aplinka dėl ryškaus šviesos lūžio ir gamyklinių paviršiaus įdubimų. Toks pasirinkimas leido įvertinti anomalijų aptikimo metodo veikimą tiek palankiomis, tiek sudėtingomis vizualinėmis sąlygomis. Svarbus anomalijų aptikimo rinkinio metodologinis sprendimas yra visų nuotraukų fono pašalinimas naudojant rembg biblioteką, pagrįstą U²-Net modeliu[25]. Šis sprendimas priimtas remiantis prižiūrimojo mokymosi modelių testavimo rezultatais, kurie parodė, kad fono šalinimas teigiamai veikia defektų aptikimo tikslumą, nes modelis gali sutelkti dėmesį išskirtinai į butelio paviršių, eliminuodamas fono triukšmą. Anomalijų aptikimo kontekste fono šalinimas ypač svarbus, nes PatchCore modelis kuria savybių banką iš viso vaizdo, įskaitant foną, todėl fono variacija tarp mokymo ir testavimo nuotraukų galėtų būti klaidingai interpretuojama kaip anomalija[15]. Fono šalinimo proceso įtaka modelių tikslumui išsamiau aptarta duomenų anotavimo bei eksperimentų skyriuose.

Abiejų duomenų rinkinių pagrindinės charakteristikos lyginamos 3.4 lentelėje (žr. 3.4 lent.).

3.4 lentelė. Duomenų rinkinių palyginimas

Charakteristika	Prižiūrimo mokymosi rinkinys	Anomalijų aptikimo rinkinys
Paskirtis	SSD, Faster R–CNN, YOLOv11	PatchCore
Nuotraukų skaičius	708	2147
Brokuotų nuotraukų dalis	69,6 %	0 %
Defektų tipai	Įbrėžimai, oro burbuliukai, nudaužtos galvutės	–
Butelių tipai	8 (visi tipai)	2 (Nr. 3 ir Nr. 5)
Anotacijos	Bounding box	Nereikalingos

Lentelėje matomi esminiai dviejų rinkinių skirtumai: prižiūrimojo mokymosi rinkinys apima visus aštuonis butelių tipus su trimis defektų kategorijomis ir bounding box anotacijomis, o anomalijų aptikimo rinkinys sudarytas išskirtinai iš dviejų butelių tipų kokybiškų nuotraukų be anotacijų. Didesnis anomalijų aptikimo rinkinio dydis (2 147 nuotraukos) lyginant su prižiūrimojo mokymosi rinkiniu (708 nuotraukos) pagrįstas PatchCore modelio savybių banko sudarymo reikalavimais.

3.3.7. Duomenų anotavimas

Vaizdų anotavimas yra vienas daugiausiai laiko reikalaujančių duomenų paruošimo etapų: tyrimai rodo, kad dirbtinio intelekto mokslininkai vidutiniškai praleidžia apie 70 % laiko dirbdami su duomenų rinkiniais, įskaitant anotavimą[2]. Anotavimo kokybė tiesiogiai veikia modelio tikslumą: netiksliai pažymėti defektai gali sukelti modelio persimokymą arba klaidingus aptikimus. Šiame tyrime anotavimui naudota Roboflow platforma, apjungianti anotavimo įrankius, duomenų valdymą, augmentaciją ir eksportą į įvairius formatus, įskaitant YOLO, COCO ir TFRecord[8].

Objektų aptikimui naudotos stačiakampės anotacijos (angl. *bounding box*), apibrėžiančios objekto vietą vaizde keturiomis koordinatėmis. Visi trys defektų tipai buvo žymėti kaip viena bendra klasė „defect“. Toks sprendimas priimtas sąmoningai: tyrimo tikslas yra įvertinti modelių gebėjimą aptikti defektus apskritai, o ne klasifikuoti jų tipus. Atskiras defektų tipų klasifikavimas reikalautų gerokai didesnio kiekvieno tipo anotacijų skaičiaus, kad modelis galėtų patikimai išmokti kiekvienos klasės ypatumus, o turimo rinkinio dydžio greičiausiai tam nepakaktų. Be defektų klasės, tyrime naudota ir antroji klasė „bottle“, žyminti patį butelį kaip objektą.

Praktinio anotavimo metu susiduriama su keliais metodologiniais iššūkiais, būdingais stiklo gaminių defektų žymėjimui. Oro burbuliukų ir nudaužtų galvučių anotavimas buvo paprastas dėl

aiškiai apibrėžtų defektų ribų. Sudėtingesnis buvo įbrėžimų anotavimas dėl kelių priežasčių. Įbrėžimų intensyvumas labai variavo, nuo gilių, aiškiai matomų pažeidimų iki labai smulkių, vos pastebimų paviršiaus įbrėžimų. Kiekvieno smulkaus įbrėžimo žymėjimas būtų sukūręs pernelyg daug triukšmingų anotacijų, galinčių neigiamai paveikti modelio mokymąsi, todėl taikytas minimalaus reikšmingumo principas, žymėti tik vizualiai aiškiai matomi defektai. Taip pat kai kuriais atvejais keli skirtingo intensyvumo įbrėžimai buvo vienoje vietoje, tada reikėjo spręsti, ar juos žymėti kaip vieną bendrą defekto zoną, ar kaip atskirus atvejus. Tokiais atvejais vadovautasi erdvinio artumo principu: jei keli įbrėžimai buvo labai arti vienas kito ir sudarė vizualiai vientisą pažeidimo zoną, jie žymėti kaip vienas defektas, jei tarp jų buvo aiški vizualinė atskirtis.

Vienas svarbių šio tyrimo metodologinių aspektų – sisteminga trijų skirtingų anotavimo strategijų lyginamoji analizė, siekiant nustatyti, kuris būdas leidžia pasiekti geriausius defektų aptikimo rezultatus. Visi trys variantai testuoti su visais prižiūrimojo mokymosi modeliais – SSD, Faster R–CNN ir YOLOv11.



3.4 pav. Anotavimo pavyzdžiai: oro burbuliuko, įbrėžimo ir nudaužtos galvutės anotacijos RoboFlow aplinkoje

1 variantas tik defektų anotacijos. Šiame variante žymėti išskirtinai defektai, neįtraukiant butelio kaip atskiro objekto. Tai standartinis ir dažniausiai literatūroje aprašomas požiūris[17]. Modelis šiuo atveju mokosi tiesiogiai aptikti defektus visame vaizde, nepriklausomai nuo butelio padėties ar ribų.

2 variantas defektų ir butelio anotacijos. Šiame variante papildomai žymėtas ir pats butelis kaip atskira „bottle“ klasė, sukuriant dviejų klasių anotacijų struktūrą. Šio varianto loginė prielaida, jei modelis tiksliai lokalizuoja butelį, defektų paieška gali būti efektyviai apribota butelio ribojančio langelio erdve. Teoriškai tai turėtų sumažinti klaidingai teigiamų aptikimų skaičių, nes modelis galėtų atmeti defektų kandidatus, esančius už butelio ribų. Analogišką dviejų klasių strategiją aprašo Claypo[6], tiriant stiklinių butelių defektų klasifikavimą.

3 variantas fono šalinimas su defektų anotacijomis. Šiame variante prieš anotavimą pašalintas visų 708 nuotraukų fonas naudojant rembg biblioteką, pagrįstą U²-Net modeliu[25]. Po fono pašalinimo nuotraukose liko tik butelio siluetas su defektais neutraliame fone, o anotacijos žymėtos tik defektams. Pašalinus foną, modelis gali visą dėmesį sutelkti į butelio paviršių, eliminuojant fono triukšmą, galintį apsunkinti smulkių defektų aptikimą. Fono pašalinimas atliktas prieš anotavimą.

3.3.8. Duomenų augmentacija ir paruošimas modeliams

Duomenų augmentacija yra svarbus metodas dirbant su ribotos apimties duomenų rinkiniais: ji leidžia dirbtinai padidinti mokymo duomenų įvairovę, generuojant modifikuotas pradinių vaizdų versijas. Tai padeda sumažinti perdėto prisitaikymo prie mokymo duomenų (angl. *overfitting*) riziką ir gerina modelio gebėjimą apibendrinti[33]. Šiame tyrime augmentacija taikyta prižiūrimojo mokymosi duomenų rinkiniui, naudojant Roboflow platformą. Anomalijų aptikimo duomenų rinkiniui augmentacija netaikyta, nes 2 147 nuotraukos laikytos pakankamu kiekiu PatchCore modelio savybių bankui sudaryti.

Augmentacija vykdyta Roboflow platformoje prieš duomenų eksportavimą, o tai leido padidinti duomenų rinkinio dydį prieš modelių mokymą. Kiekviena originali nuotrauka pakeista kelis kartus, išplečiant bendrą rinkinio dydį nuo 708 iki 1 848 nuotraukų. Taikytos šios augmentacijos technikos:

- **Horizontalus ir vertikalus veidrodinis atspindys** leidžia modeliui atpažinti defektus nepriklausomai nuo jų orientacijos vaizde
- **Pasukimas nuo -15° iki $+15^\circ$ ribose**, imituojant nedidelius kameros padėties nukrypimus fotografavimo metu
- **Sotumo reguliavimas nuo -20% iki $+20\%$ ribose**, didinant modelio atsparumą spalvų variacijoms
- **Ekspozicijos reguliavimas nuo -10% iki $+10\%$ ribose**, imituojant nedidelius apšvietimo sąlygų svyravimus

Pasirinktos augmentacijos technikos atspindi realias fotografavimo sąlygų variacijas kur nedideli pasukimai imituoja rankinio fotografavimo netikslumus, o spalviniai pakeitimai modeliuoja skirtingus apšvietimo intensyvumo lygius. Analogišką augmentacijos techniką rinkinį taiko Latina[17], pasiekę aukštus tikslumo rodiklius stiklinių butelių defektų aptikimo tyrimuose.

Išplėstas duomenų rinkinys padalintas į tris dalis vadovaujantis rekomenduojama praktika[3] (žr. 3.5 lent.).

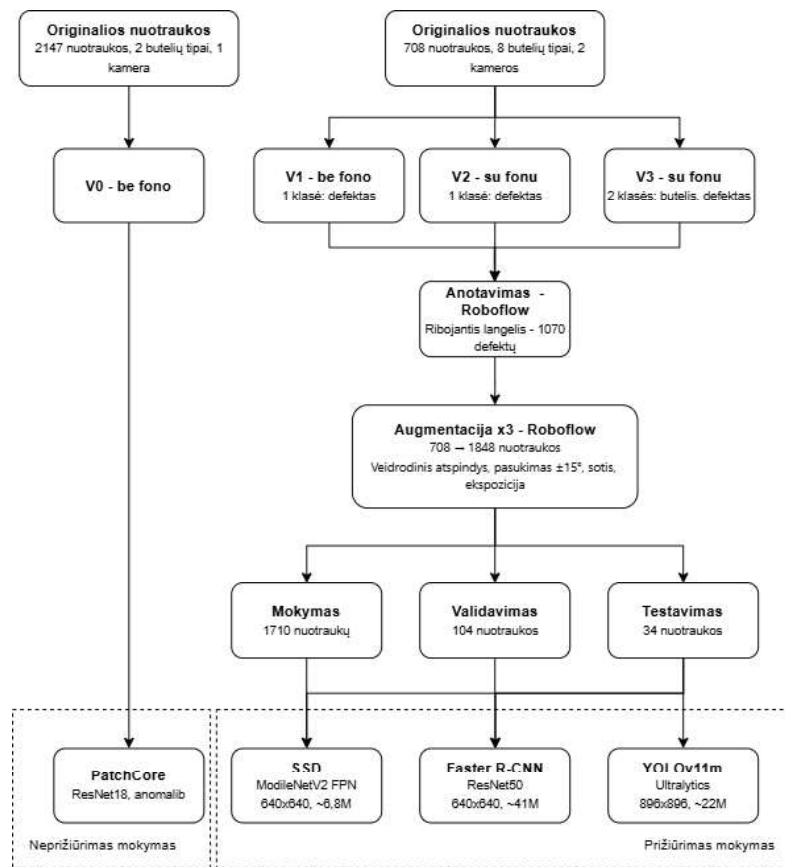
3.5 lentelė. Išplėsto duomenų rinkinio padalijimas

Rinkinys	Nuotraukų skaičius	Dalis (%)	Paskirtis
Mokymo (angl. <i>training</i>)	1710	92,5 %	Modelio parametrų optimizavimas
Validavimo (angl. <i>validation</i>)	104	5,6 %	Mokymosi proceso stebėjimas
Testavimo (angl. <i>testing</i>)	34	1,8 %	Galutinis modelio įvertinimas
Iš viso	1848	100 %	

Verta atkreipti dėmesį, kad duomenų rinkinio išplėtimas taikytas išskirtinai mokymo rinkiniui, validavimo ir testavimo rinkiniai liko nepakitę, kad galutinis modelio įvertinimas atspindėtų realias, nemodifikuotas sąlygas[33].

Skirtingi modeliai reikalauja skirtingos įvesties vaizdo raiškos. SSD ir Faster R–CNN modeliams visi vaizdai sumažinti iki 640×640 pikselių tai standartinė šių modelių įvesties raiška, užtikrinanti balansą tarp apdorojimo greičio ir tikslumo[19][27]. YOLOv11 modeliui vaizdai sumažinti iki 896×896 pikselių, didesnė įvesties raiška pasirinkta siekiant geriau išsaugoti smulkių defektų vizualines charakteristikas. Siaurų įbrėžimų ir mažų oro burbuliukų detalės gali būti prarastos mažesnės raiškos vaizduose, todėl didesnis įvesties dydis laikytas svarbiu YOLOv11 modelio tikslumo optimizavimo veiksniu. SSD ir Faster R–CNN modeliams didesnė raiška netaikyta dėl įrangos apribojimų.

Visas duomenų paruošimo procesas, nuo originalių nuotraukų iki galutinių mokymo rinkinių, schematiškai pavaizduotas 3.5 pav. (žr. 3.5 pav.).



3.5 pav. Duomenų paruošimo proceso schema

3.4. Trumpas skyriaus apibendrinimas

Šiame skyriuje pristatytas nuosavas stiklinių butelių duomenų rinkinys, sudarytas kontroliuojamomis sąlygomis, bei išsamiai aprašytas jo kūrimo procesas. Iš pradžių atlikta viešai prieinamų duomenų rinkinių analizė, kurios rezultatai parodė, kad esami rinkiniai netinkami dėl reikšmingo domenų neatitikimo, jų vaizdai gauti pramoninėmis sąlygomis ir neatitinka tyrimo aplinkos. Todėl priimtas sprendimas kurti nuosavą rinkinį: 708 nuotraukos su 1 070 anotuotų defektų atvejų, apimančios aštuonių tipų stiklinius butelius, fotografuotus trijose skirtingose aplinkose ir sudarant nuoseklias vaizdų sekas iš skirtingų kampų. Papildomai PatchCore modeliui sudarytas 2 147 nebrotuotų butelių vaizdų rinkinys. Aprašytas vaizdų anotavimo procesas naudojant Roboflow platformą, išbandyti trys skirtingi anotavimo variantai, vien defektų žymėjimas, defektų ir butelio žymėjimas bei fono pašalinimas, siekiant nustatyti optimaliausią duomenų paruošimo strategiją. Duomenų augmentacijos etape taikytos geometrinės ir spalvinės transformacijos, didinančios rinkinio įvairovę.

4. GAMINIO BROKO APTIKIMO METODO ĮGYVENDINIMAS IR EKSPERIMENTINIO TYRIMO REZULTATAI

4.1. Perkėlimo mokymasis ir mokymas nuo nulio

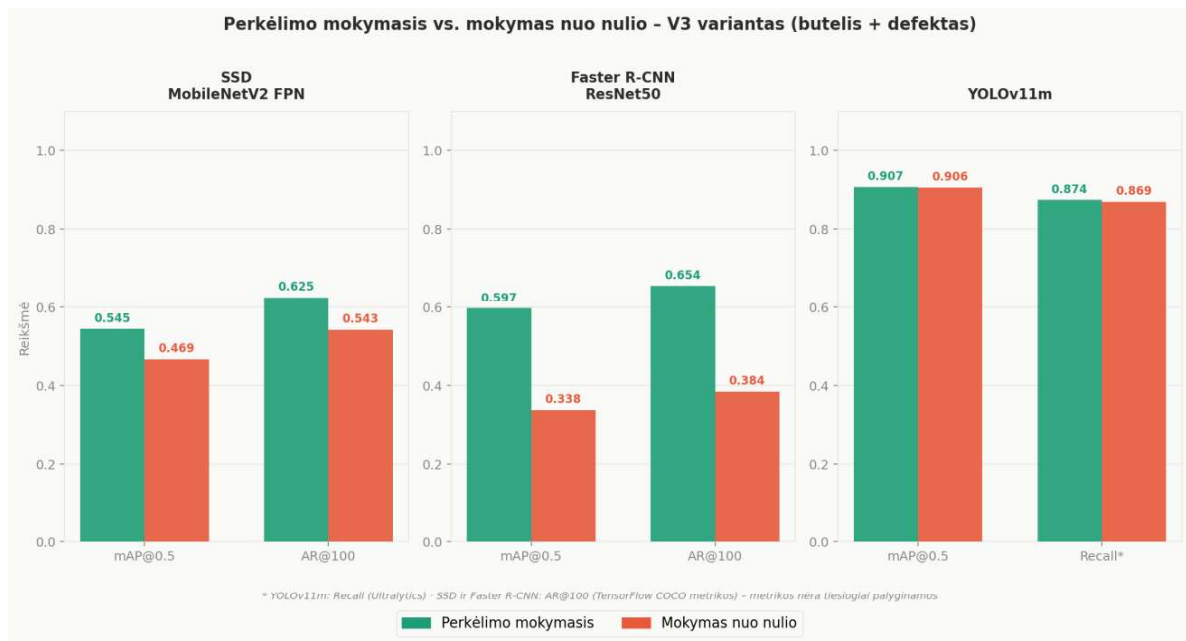
Pirmasis eksperimentų etapas skirtas įvertinti perkėlimo mokymosi naudą lyginant su mokymu nuo nulio ribotų duomenų sąlygomis. Visi trys modeliai apmokyti abiem būdais su V3 anotavimo variantu (butelis + defektas klasės), naudojant identišką hiperparametrų konfigūraciją. Rezultatai pateikiami 4.1 lentelėje (žr. 4.1 lent.).

4.1 lentelė. Perkėlimo mokymasis vs. mokymas nuo nulio palyginimas

Modelis	Metodas	mAP@0.5	AR@100	Laikas
SSD MobileNetV2 FPN	Transfer learning	0,545	0,625	~60 min
SSD MobileNetV2 FPN	Nuo nulio	0,469	0,543	~58 min
Faster R-CNN ResNet50	Transfer learning	0,598	0,654	~49 min
Faster R-CNN ResNet50	Nuo nulio	0,338	0,384	~50 min
YOLOv11m	Transfer learning	0,907	0,874*	~130 min
YOLOv11m	Nuo nulio	0,906	0,869*	~130 min

*YOLOv11m atveju standartinis Recall iš Ultralytics

SSD ir Faster R-CNN modelių vertinimui naudotas TensorFlow Object Detection API COCO metrics protokolas, pateikiantis mAP@0.5 ir AR@100 rodiklius. YOLOv11m vertinimui naudota Ultralytics biblioteka, pateikianti mAP@0.5 ir Recall rodiklius. Kadangi TensorFlow Object Detection API tiesiogiai nepateikia Precision rodiklio, o Ultralytics AR@100 nematuoja, šių dviejų metrikų tiesioginis palyginimas tarp TensorFlow ir Ultralytics modelių nėra galimas, lentelėje ir grafike kiekvieno modelio antrosios metrikos pavadinimas atitinka jo vertinimo protokolą.



4.1 pav. Perkėlimo mokymo ir mokymo nuo nulio metrikų palyginimas (V3 variantas)

Paveiksle (žr. 4.1 pav.) vizualiai pavaizduotas perkėlimo mokymosi ir mokymo nuo nulio metrikų palyginimas visiems trims modeliams V3 variante. Rezultatai atskleidžia tris skirtingus modelių elgsenos šablonus. SSD MobileNetV2 FPN perkėlimo mokymasis davė nuosaikų, bet pastebimą pranašumą (+0,073 mAP@0.5), o tai patvirtina, kad lengva MobileNetV2 architektūra su ~6,8M parametrų gali pasinaudoti COCO duomenų rinkiniu išmokytais požymiais, tačiau jos apibendrinimo gebėjimai riboti net ir be perkėlimo mokymosi.

Faster R-CNN ResNet50 perkėlimo mokymasis davė gerokai didesnę pranašumą (+0,260 mAP@0.5). Mokymas nuo nulio su ~41M parametrų modeliu ir tik 1 710 mokymo nuotraukomis aiškiai nepakankamas, modelis negali efektyviai išmokti tiek parametrų iš tokio riboto duomenų kiekio. Tai patvirtina literatūroje aprašomą išvadą, kad dideli dviejų etapų detektoriai ypač priklauso nuo perkėlimo mokymosi esant mažiems duomenų rinkiniams[20].

YOLOv11m perkėlimo mokymasis ir mokymas nuo nulio davė praktiškai identišką tikslumą (skirtumas tik 0,001 mAP@0.5). Šis radinys įdomus ir gali būti paaiškintas keliais veiksniais. Pirma, YOLOv11m architektūra su C3K2 blokais ir C2PSA dėmesio mechanizmais gerokai efektyvesnė duomenų naudojimo požiūriu nei SSD ar Faster R-CNN. Antra, AdamW optimizatorius su adaptyviu mokymosi greičiu leidžia modeliui greičiau konverguoti net ir be iš anksto apmokyty svorių. Trečia, 150 epochų mokymas su išankstiniu stabdymu leido modeliui visapusiškai išnaudoti mokymo duomenų galimybes, nepriklausomai nuo pradinių svorių. Šis rezultatas rodo, kad YOLOv11m pakankamai galingas išmokti iš 1 710 nuotraukų net be iš anksto apmokyty svorių, o tai svarbus praktinis radinys smulkioms ir vidutinėms įmonėms, kurios gali neturėti galimybių naudoti iš anksto apmokyty modelių dėl licencijų ar kitų apribojimų.

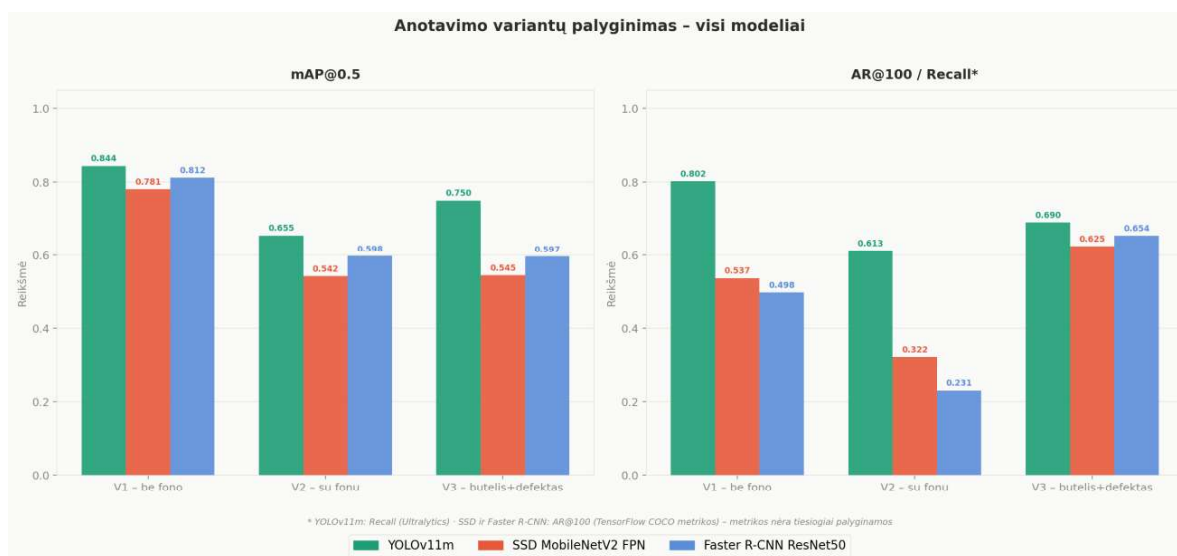
TensorBoard loss kreivių analizė atskleidžia papildomų įžvalgų. SSD perkėlimo mokymosi atveju klasifikacijos nuostoliai pradiniam etape buvo gerokai mažesni nei mokymo nuo nulio atveju, o tai rodo greitesnę konvergavimą dėl iš anksto apmokytų požymių. Faster R–CNN mokymo nuo nulio atveju pastebimi dideli lokalizacijos nuostolių svyravimai pirmuosiuose 3 000–4 000 žingsniuose, o tai rodo nestabilią konvergavimą, tipinis reiškinys mokant didelį dviejų etapų detektorių su ribotu duomenų kiekiu be iš anksto apmokytų svorių[27].

4.2. Anotavimo variantų palyginimas

Antrasis eksperimentų etapas skirtas sistemingai įvertinti trijų skirtingų anotavimo strategijų įtaką modelių defektų aptikimo tikslumui. Visi trys modeliai apmokyti perkėlimo mokymosi metodu kiekvienam anotavimo variantui, išlaikant identišką hiperparametrų konfigūraciją. Tai leidžia izoliuoti anotavimo strategijos efektą nuo kitų veiksnių. Rezultatai pateikiami 4.2 lentelėje (žr. 4.2 lent.).

4.2 lentelė. Anotavimo variantų palyginimas visų modelių (tik defect klasė)

Modelis	Variantas	mAP@0.5	AR@100
YOLOv11m	V1 be fonu	0,844	0,802
YOLOv11m	V2 su fonu	0,655	0,613
YOLOv11m	V3 bottle+defect	0,750	0,690
SSD	V1 be fonu	0,781	0,537
SSD	V2 su fonu	0,542	0,322
SSD	V3 bottle+defect	0,615	0,625
Faster R–CNN	V1 be fonu	0,812	0,498
Faster R–CNN	V2 su fonu	0,598	0,231
Faster R–CNN	V3 bottle+defect	0,662	0,654



4.2 pav. Anotavimo variantų palyginimas mAP@0.5 ir AR@100/Recall

Paveiksle (žr. 4.2 pav.) vizualiai pavaizduotas visų trijų modelių metrikų palyginimas pagal anotavimo variantus.

4.2.1. Fono šalinimo įtaka (V1 vs V2)

Ryškiausia ir nuosekliausia tendencija visuose trijuose modeliuose yra, jog fono pašalinimas (V1) gerokai pagerino defektų aptikimo tikslumą lyginant su originalių nuotraukų naudojimu (V2). Tikslūs skirtumai pateikiami 4.3 lentelėje (žr. 4.3 lent.).

4.3 lentelė. Fono pašalinimo efektas – mAP@0.5 pokytis (V1 vs V2)

Modelis	V2 (su fonu)	V1 (be fono)	Pokytis
YOLOv11m	0,655	0,844	+0,189
SSD	0,542	0,781	+0,239
Faster R-CNN	0,598	0,812	+0,214

Fono šalinimas pagerino visų modelių tikslumą vidutiniškai +0,214 mAP@0.5. Šis rezultatas patvirtina teorinėje dalyje aprašytą prielaidą, kad stiklinių butelių fotografavimas sukuria specifinių vizualinių iššūkių (atspindžių, šviesos lūžių ir kintančio fono), kurie apsunkina defektų aptikimą[6]. Pašalinus foną, modelis gali sutelkti visą savo gebėjimų potencialą į butelio paviršiaus analizę, eliminuojant fono triukšmą kaip klaidingų aptikimų šaltinį. Ypač ryškus efektas pastebimas SSD modelyje (+0,239), tai paaiškinama SSD architektūros ypatumu: vieno etapo detektorius visą vaizdą analizuoja vienu praėjimu, todėl fono triukšmas tiesiogiai konkuruoja su defektų signalais visuose požymių žemėlapių lygiuose. Faster R-CNN, nors turi dviejų etapų architektūrą, kuri teoriškai turėtų geriau filtruoti foną per RPN mechanizmą, taip pat gavo reikšmingą naudą iš fono pašalinimo (+0,214).

4.2.2. Butelio anotavimo defektas (V3 vs V2)

Butelio ir defekto anotavimas kartu davė nuosekliai geresnius rezultatus nei vien tik defektų žymėjimas (V2), tačiau nepasiekė fono šalinimo varianto (V1) lygio. Tai atskleidžia įdomų kompromisą: butelio bounding box informacija padeda modeliui sumažinti klaidingai teigiamus aptikimus už butelio ribų, tačiau fono triukšmas vis dar išlieka ir riboja tikslumą. Tikslūs skirtumai pateikiami 4.4 lentelėje (žr. 4.4 lent.).

4.4 lentelė. V3 vs V2 palyginimas – mAP@0.5 pokytis

Modelis	V2	V3	Pokytis
YOLOv11m	0,655	0,750	+0,095
SSD	0,542	0,615	+0,073
Faster R-CNN	0,598	0,662	+0,064

V3 varianto nauda yra nuosekli YOLOv11m atveju (+0,095), tačiau SSD ir Faster R-CNN modeliams mažiau pastebima. Tai rodo, kad butelio lokalizavimas padeda YOLOv11m geriau apibrėžti paieškos sritį, tačiau nekompensuoja fono triukšmo problemos TensorFlow modeliams. Praktiškai tai reiškia, kad jei fono pašalinimas negalimas (pvz., realiojo laiko sistemose su ribotais ištekliais), V3 strategija yra geresnė alternatyva nei V2 tik YOLOv11m atveju.

4.2.3. Modelių palyginimas pagal geriausius rezultatus

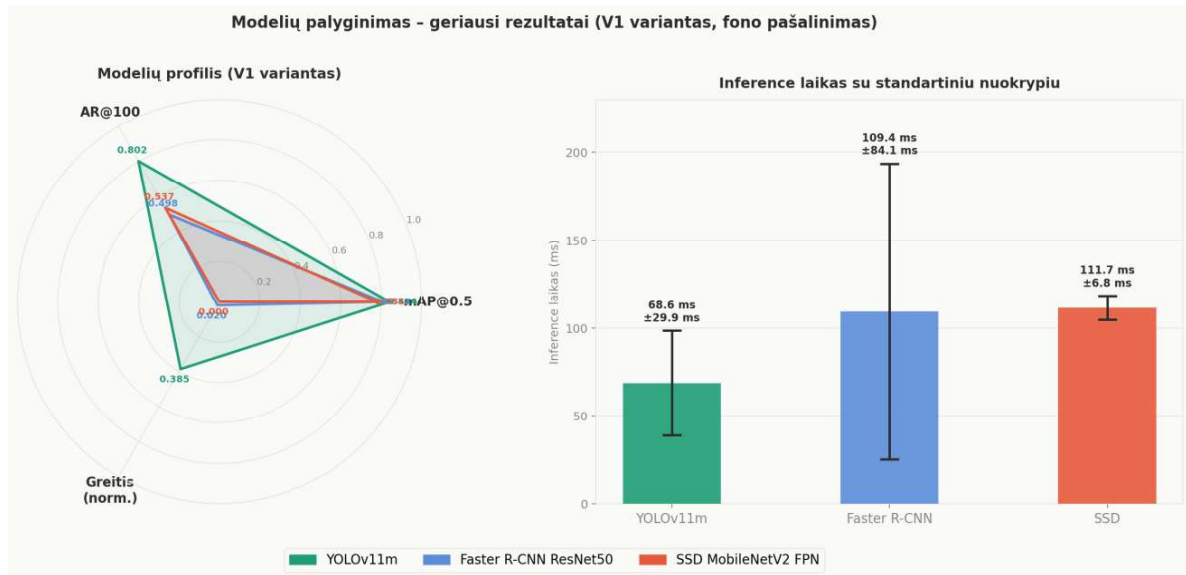
Lyginant geriausius kiekvieno modelio rezultatus – visi gauti su V1 (fondo pašalinimo) variantu – matoma aiški hierarchija. Rezultatai pateikiami 4.5 lentelėje (žr. 4.5 lent.).

4.5 lentelė. Geriausių rezultatų palyginimas (V1 variantas)

Modelis	mAP@0.5	AR@100	Parametrai	Inference	Std
YOLOv11m	0,844	0,802	~20–25M	68,63 ms	29,87 ms
Faster R-CNN	0,812	0,498	~41M	109,44 ms	84,09 ms
SSD	0,781	0,537	~6,8M	111,67 ms	6,85 ms

Lentelėje pateikiami V1 anotavimo varianto (fondo pašalinimas) rezultatai, geriausi kiekvieno modelio rodikliai. YOLOv11m Recall gautas iš Ultralytics validavimo metrikos, SSD ir Faster R-CNN – AR@100 iš TensorFlow Object Detection API COCO metrics protokolo. Inferavimo laikas matuotas ant RTX 3060 Laptop GPU; pateikiamas vidurkis ir standartinis nuokrypis. **YOLOv11m** pasiekė aukščiausią mAP@0.5 (0,844) ir Recall (0,802), kartu būdamas greičiausias (68,63 ms). Tai daro jį aiškiu lyderiu bendro efektyvumo požiūriu. **Faster R-CNN ResNet50** pasiekė aukštą

AR@100 (0,498) rodiklį, tačiau tai atperka didesnis parametų skaičius (~41M) ir lėtesnis inferavimas (109,44 ms). Didelis standartinis nuokrypis (84,09 ms) rodo nestabilų inferavimo laiką, o tai gali būti problema realiojo laiko sistemuose. **SSD MobileNetV2 FPN** pasiekė mažiausią tikslumą tarp trijų modelių, tačiau jo privalumas yra mažiausias parametų skaičius (~6,8M) ir stabiliausias inferavimo laikas (std: 6,85 ms). Tai daro jį tinkamu pasirinkimu itin ribotų išteklių aplinkose, kur modelio dydis ir inferavimo stabilumas svarbesni už maksimalų tikslumą.



4.3 pav. Modelių profilis ir inferavimo greičio palyginimas (V1 variantas)

Paveiksle (žr. 4.3 pav.) pateikiamas dviejų dalių modelių palyginimas pagal geriausius rezultatus, gautus su V1 anotavimo variantu. Kairėje diagrama vaizduoja trijų metrių profilį kiekvienam modeliui: mAP@0.5, AR@100 bei normalizuotą greitį (inferavimo laiko inversija, kuo arčiau išorinio žiedo, tuo greitesnis modelis). YOLOv11m užima didžiausią plotą visose trijose dimensijose, o tai vizualiai patvirtina jo bendrą pranašumą. Faster R–CNN pasižymi geresniu AR@100 nei SSD, tačiau gerokai praranda pozicijas greičio dimensijoje. SSD, nors ir prastesnis tikslumo atžvilgiu, išlaiko konkurencingą greičio rodiklį.

Dešinėje pateikiamas inferavimo laiko palyginimas su standartinio nuokrypio juostomis. Matosi, kad SSD pasižymi stabiliausiu inferavimo laiku (std: 6,85 ms), o Faster R–CNN labai dideliu nestabilumu (std: 84,09 ms), o tai gali būti kritinis trūkumas realiojo laiko gamybinėse sistemose, kur reikalingas nuspėjamas ir stabilus apdorojimo laikas. YOLOv11m užima tarpinę poziciją (std: 29,87 ms), išlaikant priimtina stabilumą kartu su aukščiausiu tikslumu.

4.3. PatchCore anomalijų aptikimo rezultatai

PatchCore modelio vertinimas atliktas su atskiru testavimo rinkiniu, apimančiu 22 kokybiškų ir 91 brokuotų butelių nuotraukas iš dviejų butelių tipų, rudos giros buteliuko (Nr. 3) ir skaidraus

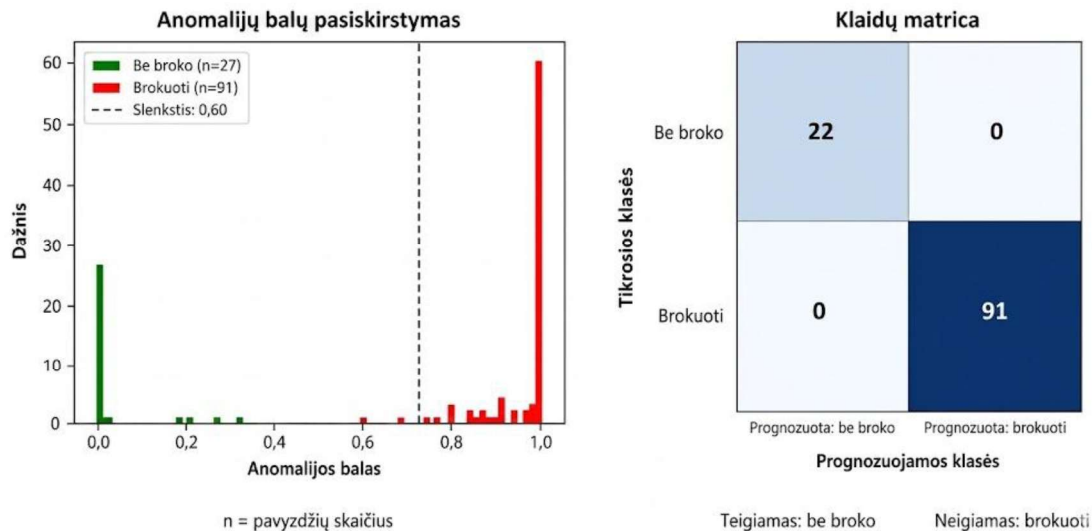
Coca-Cola buteliuko (Nr. 5). Visos nuotraukos apdorotos fono pašalinimo žingsniu prieš testavimą, atitinkamai mokymo duomenų paruošimo metodologijai.

4.3.1. Klasifikacijos rezultatai

PatchCore modelis pasiekė tobulus klasifikacijos rezultatus vaizdo lygyje, visos 113 testavimo nuotraukų teisingai klasifikuotos (žr. 4.6 lent.).

4.6 lentelė. PatchCore klasifikacijos rezultatai

Metrika	Reikšmė
Image-level AUROC	1,000
Precision	1,000
Recall	1,000
F1 Score	1,000
Teisingai klasifikuoti geri	22/22
Teisingai klasifikuoti brokuoti	91/91
Klaidingai teigiami (FP)	0
Praleisti defektai (FN)	0



4.4 pav. Anomalijų balo pasiskirstymo histograma ir konfūzijos matrica

Konfūzijos matrica (žr. 4.4 pav.) vizualiai patvirtina šiuos rezultatus, visos 22 nebrotuotų butelių nuotraukos teisingai klasifikuotos kaip geros (TN), o visos 91 brokuotų butelių nuotraukos kaip defektinės (TP), be jokių klaidų. Anomalijų balo pasiskirstymas taip pat rodo aiškų atskyrimą tarp dviejų klasių, nebrotuotų butelių anomalijų balai sutelkti ties 0,0, o brokuotų butelių balai ties

0,8–1,0. Toks ryškus pasiskirstymas rodo, kad modelis išmoko labai stabilią nebrokuotų butelių paviršiaus reprezentaciją, leidžiančią aiškiai identifikuoti bet kokius nukrypimus.

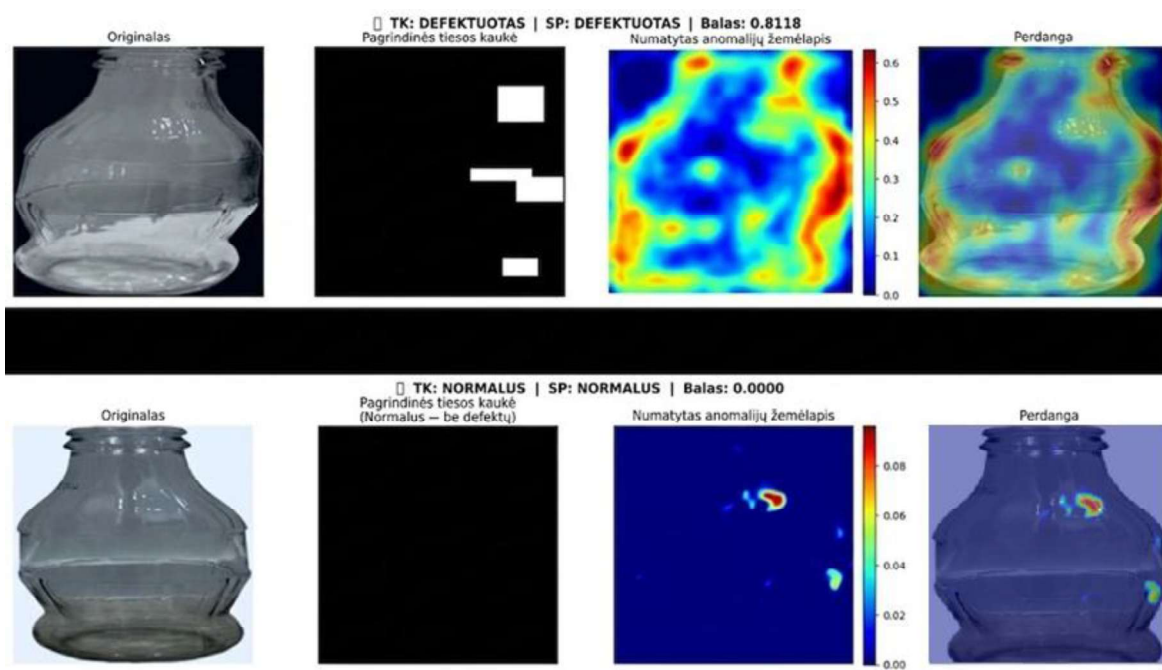
4.3.2. Lokalizacijos rezultatai

Nors klasifikacijos rezultatai tobuli, pikselių lygio lokalizacijos tikslumas gerokai mažesnis. Rezultatai pateikiami 4.7 lentelėje (žr. 4.7 lent.).

4.7 lentelė. PatchCore lokalizacijos rezultatai

Metrika	Reikšmė
Pixel-level AUROC	0,810
Pixel-level F1	0,320

Pikselių lygmens AUROC 0,810 rodo, kad modelis pakankamai tiksliai nustato anomalijų zonas pikselių lygyje, tačiau šilumos žemėlapis ne visada tiksliai atitinka tikrąsias defektų vietas. Pikselių lygmens F1 0,320 yra gerokai žemesnis rodiklis, atspindintis tai, kad modelio generuojami anomalijų žemėlapiai dažnai per platūs. Šis skirtumas tarp tobulos klasifikacijos ir vidutinio lokalizacijos tikslumo svarbus radinys. Jį lemia keli veiksniai. ResNet18 *layer2* požymiai turi ribotą erdvinę raišką, 128 matmenų požymių vektoriai užfiksuoja semantinę informaciją apie anomalijų buvimą, tačiau nepakankamą erdvinį tikslumą tiksliai pikselių lygio žymėjimui. Atrankos poaibio koeficientas 10 % sumažina atminties banką iki 10 % originalių požymių, tai padeda atminties efektyvumui, tačiau gali sumažinti lokalizacijos tikslumą. Binarinės kaukės, sugeneruotos iš bounding box anotacijų, yra stačiakampės, jos neatskleidžia tikslų defektų formų, todėl pikselių lygio vertinimas apytikslis.



4.5 pav. PatchCore anomalijų žemėlapių vizualizacijos pavyzdžiai

Paveiksle (žr. 4.5 pav.) pateikiami PatchCore anomalijų žemėlapių vizualizacijos pavyzdžiai dviem atvejais. Kiekviena eilutė rodo keturias kolonas: originalią nuotrauką, tikrąją defektų kaukę (angl. *Ground Truth Mask*), prognozuojamą anomalijų žemėlapi (angl. *Predicted Anomaly Map*) ir perdangą (angl. *Overlay*).

Viršutinėje eilutėje pavaizduotas brokuotas butelis (anomalijų balas: 0,8118). Tikrojoje kaukėje matomi trys stačiakampiai, anotuotų defektų vietos. Prognozuojamas anomalijų žemėlapis rodo aukštą anomalijų intensyvumą (raudona spalva) didžioje butelio paviršiaus dalyje, o tai paaiškina žemą pikselių lygio lokalizacijos tikslumą – modelis teisingai identifikuoja defektinį butelį, tačiau anomalijų zona platesnė nei tikroji defektų vieta. Apatinėje eilutėje pavaizduotas nebrokuotas butelis (anomalijų balas: 0,0000). Tikroji kaukė tuščia, defektų nėra. Prognozuojamas anomalijų žemėlapis rodo itin žemą anomalijų intensyvumą (mėlyna spalva) visame vaizde, su keliomis nedidelėmis zonomis, kurias modelis interpretuoja kaip silpnas anomalijas. Tačiau jų intensyvumas žymiai per žemas, kad butelis būtų priskirtas brokuotų kategorijai, o galutinis anomalijų balas 0,0000 patvirtina tobulą klasifikaciją.

4.3.3. PatchCore palyginimas su prižiūrimo mokymosi modeliais

Tiesioginis PatchCore ir prižiūrimojo mokymosi modelių palyginimas reikalauja metodologinio atidumo, šie metodai sprendžia skirtingas užduotis ir naudoja skirtingas vertinimo metrikas. mAP@0.5 metrika PatchCore atveju netaikoma, kadangi anomalijų aptikimo modelis nepateikia ribojančių langelių. Dalinį palyginimą leidžia atlikti aptikimo jautrumo rodikliai – Recall

YOLOv11m ir PatchCore atveju bei AR@100 SSD ir Faster R–CNN atveju, nors šios metrikos nėra tiesiogiai palyginamos dėl skirtingų vertinimo protokolų. Rezultatai pateikiami 4.8 lentelėje (žr. 4.8 lent.).

4.8 lentelė. PatchCore vs. prižiūrimo mokymosi modelių dalinis palyginimas

Modelis	Metodas	AR@100 / Recall	mAP@0.5	Lokalizacija
YOLOv11m (V1)	Prižiūrimas	0,802 (Recall)	0,844	Bounding box
Faster R–CNN (V1)	Prižiūrimas	0,498 (AR@100)	0,812	Bounding box
SSD (V1)	Prižiūrimas	0,537 (AR@100)	0,781	Bounding box
PatchCore	Neprižiūrimas	1,000 (Recall)	N/A	Heatmap

PatchCore pasiekė tobulą Recall klasifikacijos lygyje, pranokdamas visus prižiūrimojo mokymosi modelius šiuo aspektu. Tačiau šis palyginimas reikalauja kelių svarbių pastabų.

PatchCore pranašumai: modelis apmokytas išskirtinai su nebrokuotų butelių vaizdais, nereikalaujant rankinio defektų anotavimo. Tai ypač vertinga pramoninėje aplinkoje, kur brokuotų gaminių kiekis ribotas arba defektų tipai iš anksto nežinomi. Be to, PatchCore gali aptikti ir nežinomus defektų tipus, o tai esminis privalumas prieš prižiūrimojo mokymosi modelius, kurie gali atpažinti tik tas defektų kategorijas, su kuriomis apmokyti[44].

PatchCore apribojimai: tobuli klasifikacijos rezultatai gali būti iš dalies paaiškinami ribotu testavimo rinkinio dydžiu (113 nuotraukų) ir tuo, kad testuoti tik du butelių tipai. Prižiūrimojo mokymosi modeliai testuoti su aštuonių tipų buteliais, o tai sukuria sudėtingesnes ir realesniam scenarijui artimesnes testavimo sąlygas. Taip pat PatchCore nepateikia tikslių defektų ribų, šilumos žemėlapių lokalizacija gerokai mažiau tiksli nei bounding box metodas, o tai gali būti kritiškai svarbu gamybinėse situacijose, kur reikalinga tiksli defekto vietos ir dydžio informacija.

Praktinė rekomendacija: PatchCore tinkamas kaip pirminis filtravimo įrankis, nes jis gali greitai ir tiksliai atskirti brokuotus gaminius nuo kokybiškų, ypač tais atvejais kai defektų tipai iš anksto nežinomi. Prižiūrimojo mokymosi modeliai, ypač YOLOv11m, tinkami, kai reikalinga tiksli defektų lokalizacija ir klasifikacija pagal tipus bei kai turimas pakankamas kiekis anotuotų defektuotų pavyzdžių.

4.4. Diskusija ir praktinio pritaikymo galimybės

4.4.1. Pagrindinių rezultatų aptarimas

Ryškiausias ir labiausiai netikėtas šio tyrimo radinys yra fono pašalinimo efektas. Vidutinis $+0,214 \text{ mAP}@0.5$ pagerėjimas visuose trijuose modeliuose rodo, kad stiklo gaminių skaidrumas ir atspindžiai yra fundamentali problema, kurios negalima išspręsti vien modelio architektūros pasirinkimu ar hiperparametrų derinimu, ji turi būti sprendžiama duomenų paruošimo etape. Praktiškai tai reiškia, kad skirtumas tarp geriausio (YOLOv11m V1: 0,844) ir blogiausio (SSD V2: 0,542) rezultato didžiąją dalimi paaiškinamas ne modelių architektūrų skirtumu, o tuo, ar buvo pašalintas fonas, o tai keičia požiūrį į tokių sistemų kūrimą, kai vietoj ilgų eksperimentų su skirtingomis architektūromis verta pradėti nuo duomenų paruošimo kokybės.

YOLOv11m rezultatai išsiskyrė tuo, kad modelis dominavo visais vertinimo aspektais vienu metu, aukščiausiu tikslumu ($0,844 \text{ mAP}@0.5$), aukščiausiu Recall ($0,802$) ir greičiausiu inferavimu ($68,63 \text{ ms}$), o tai gana retas atvejis, kai nereikia rinktis tarp tikslumo ir greičio. Ypač įdomu, kad YOLOv11m pasiekė praktiškai identišką tikslumą tiek su perkėlimo mokymusi, tiek mokantis nuo nulio (skirtumas tik $0,001 \text{ mAP}@0.5$), o tai rodo, jog C3K2 blokų ir C2PSA dėmesio mechanizmų architektūra itin efektyviai išnaudoja turimus duomenis net be iš anksto apmokytų svorių. Tuo tarpu Faster R-CNN perkėlimo mokymasis davė $+0,260 \text{ mAP}$ pranašumą prieš mokymą nuo nulio, tai patvirtina, kad dideli dviejų etapų detektoriai su $\sim 41\text{M}$ parametrų tiesiog negali efektyviai išmokti iš 1710 nuotraukų be papildomo pradinio taško, o SSD atveju ši priklausomybė nuo perkėlimo mokymosi yra mažesnė, bet vis tiek pastebima ($+0,073 \text{ mAP}$).

PatchCore pasižymėjo tobula klasifikacija (Precision = 1,0, Recall = 1,0) su dviejų tipų buteliais be jokių anotacijų ir tai yra įspūdingas praktinis rezultatas, tačiau reikia nepamiršti, kad prižiūravimo mokymosi modeliai testuoti su visais aštuoniais butelių tipais, kas sukuria gerokai sudėtingesnes sąlygas. Be to, pikselių lygio lokalizacijos tikslumas ($F1 = 0,320$) rodo, kad modelis teisingai identifikuoja brokuotą butelį, bet negali tiksliai nurodyti, kur defektas yra, o tai gali būti kritiškai svarbu gamybinėse situacijose, kur reikalinga tiksli pažeidimo vietos informacija.

4.4.2. Tyrimo apribojimai

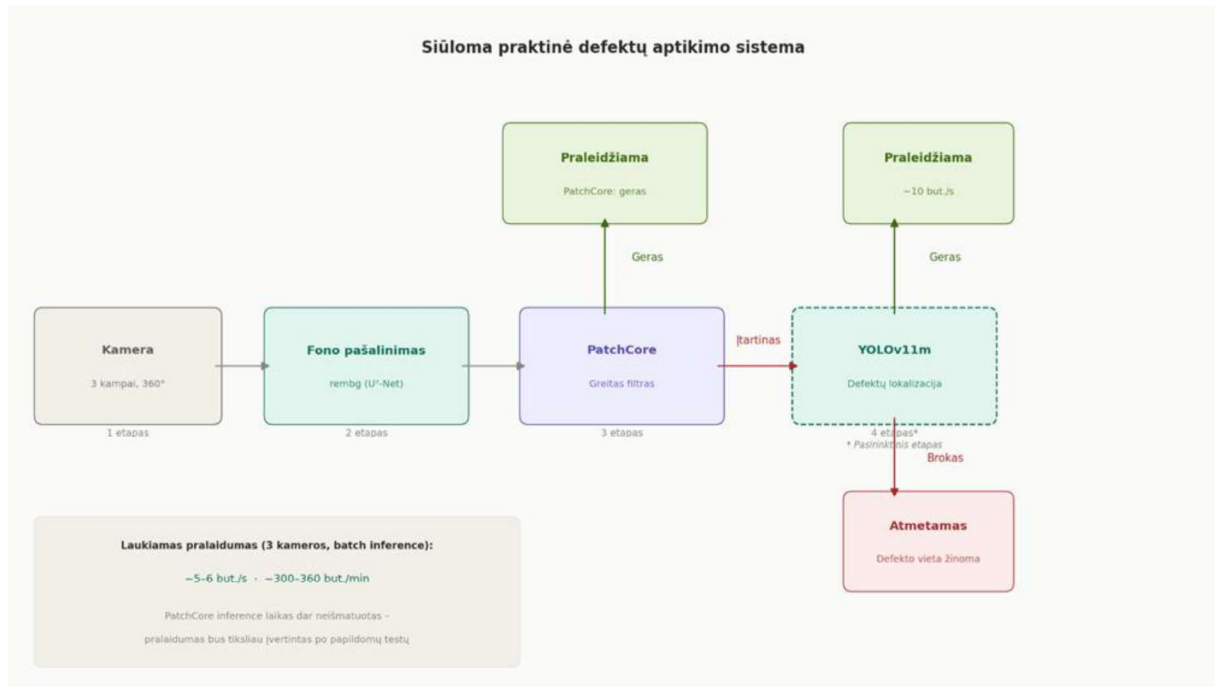
Duomenų rinkinio dydis ir įvairovė yra pagrindinis apribojimas. 708 nuotraukų rinkinys yra mažas lyginant su pramoniniais duomenų rinkiniais, o fotografavimas nekomercinėmis sąlygomis negali visiškai atkartoti pramoninės gamybos linijos aplinkos. Taip pat dirbtinai sukurti defektai, nors ir vizualiai autentiški, gali skirtis nuo realių gamybos proceso metu atsirandančių defektų subtilių charakteristikų atžvilgiu. Testavimo rinkinio dydis taip pat riboja rezultatų statistinį patikimumą. 34 nuotraukų testavimo rinkinys prižiūravimo mokymosi modeliams ir 113 nuotraukų PatchCore

testavimui yra nedideli, tad rezultatai gali kisti priklausomai nuo konkrečios testavimo imties sudėties. Anotavimo nuoseklumas yra kitas potencialus apribojimas. Kadangi anotavimas atliktas vieno žmogaus, egzistuoja subjektyvumo rizika, jog sprendžiant ar žymėti smulkius įbrėžimus ir kaip apibrėžti artimų įbrėžimų grupių ribas. Kelių anotuotojų sutarimo koeficiento (*inter-annotator agreement*) matavimas būtų sustiprinęs anotavimo kokybės vertinimą.

4.4.3. Praktinio pritaikymo galimybės

Nepaisant apribojimų, tyrimo rezultatai turi reikšmingų praktinių implikacijų, ypač smulkioms ir vidutinėms įmonėms, svarstančioms automatizuotų kokybės kontrolės sistemų diegimą. Rekomenduojama sistema ribotų išteklių aplinkoje apimtų tris komponentus. Automatinis fono pašalinimas naudojant rembg biblioteką kaip privalomą pirmą žingsnį prieš defektų aptikimą. Tuomet YOLOv11m modelis kaip pagrindinis defektų detektorius dėl optimalaus tikslumo ir greičio balanso bei gebėjimo veikti be perkėlimo mokymosi ir galiausiai PatchCore kaip papildomas kokybės kontrolės sluoksnis pirminiam butelių filtravimui, ypač tais atvejais kai gali pasirodyti naujų, anksčiau nematytų defektų tipų. Tyrimas parodė, kad veiksminga defektų aptikimo sistema gali būti sukurta naudojant vartotojų klasės įrangą (nešiojamasis kompiuteris su RTX 3060), atviro kodo bibliotekas ir santykinai nedidelį duomenų rinkinį. Tai gerokai sumažina įėjimo barjerą mažesnėms įmonėms, kurios negali investuoti į specializuotą pramoninę optinės inspekcijos įrangą[21].

Ateityje sistemą verta plėtoti keliomis kryptimis: PatchCore stuburo tinklo palyginimas su gilesniais modeliais (pvz., WideResNet50, EfficientNet) leistų įvertinti lokalizacijos tikslumo pagerėjimą. Realiojo laiko sistemos su konvejerio integracija reikalautų papildomo optimizavimo, modelių kvantavimo (angl. *quantization*) ar žinių distiliavimo (angl. *knowledge distillation*), siekiant pasiekti pramonės linijai reikalingą 30+ FPS greitį. Taip pat defektų tipų atskyrimas į atskiras klases įbrėžimai, oro burbuliukai, nudaužtos galvutės) leistų tiksliau įvertinti modelių gebėjimą klasifikuoti skirtingus pažeidimus.



4.6 pav. Siūloma praktinė stiklinių butelių defektų aptikimo sistemos architektūra

Paveiksle (žr. 4.6 pav.) pateikiama siūloma praktinė stiklinių butelių defektų aptikimo sistemos architektūra, pagrįsta šio tyrimo rezultatais. Sistema susideda iš keturių etapų. Pirmajame etape butelis fotografuojamas iš trijų kampų, užtikrinant pilną 360° paviršiaus aprėptį. Antrajame etape visų trijų nuotraukų fonas automatiškai pašalinamas naudojant rembg biblioteką, pagrįstą U²-Net modeliu. Trečiajame etape PatchCore modelis atlieka greitą binarinę klasifikaciją, buteliai identifikuoti kaip nebrokuoti iš karto praleidžiami toliau, o įtartini perduodami ketvirtajam etapui. Ketvirtajame etape YOLOv11m modelis tiksliai lokalizuoja defektus ir nustato jų vietą, taip atmetami tik patvirtinti brokuoti buteliai. Ketvirtasis etapas pažymėtas punktyrine linija kaip pasirinktinis, jo būtinumas priklauso nuo gamybinės linijos tikslumo reikalavimų ir greičio poreikių. Jei pakanka binarinio sprendimo (geras/blogas) be tikslios defekto lokalizacijos, sistema gali veikti tik su PatchCore, pasiekiant aukštesnį pralaidumą. Sistemos pralaidumas naudojant trijų kamerų konfigūraciją ir paketinį apdorojimą siekia ~5–6 butelius per sekundę (~300–360 butelių per minutę). Pažymėtina, kad PatchCore inferavimo laikas šiame tyrime nebuvo išmatuotas, dėl to tikslus pralaidumo įvertis bus galimas po papildomų eksperimentų.

4.5. Trumpas skyriaus apibendrinimas

Šiame skyriuje pateikti visų keturių modelių eksperimentinio tyrimo rezultatai ir išsami lyginamoji analizė pagal mAP@0.5, Precision, Recall ir F1 metrikas. Geriausius prižiūrimojo mokymosi rezultatus pasiekė YOLOv11m su fono pašalinimo variantu 0,844 mAP@0.5 ir 0,802

Recall yra aukščiausi rodikliai tarp visų tirtų modelių. SSD MobileNetV2 FPN išsiskyrė stabiliausiu inferavimo laiku ir mažiausiu parametų skaičiumi, todėl tinkamiausias itin ribotų išteklių aplinkose, tačiau pasiekė žemesnius tikslumo rodiklius. Faster R-CNN ResNet50 parodė vidutinį tikslumą, tačiau pasitvirtino kaip stabiliausiai reaguojantis į perkėlimo mokymąsi, šis veiksnys buvo esminis esant mažam duomenų kiekiui. PatchCore pasiekė tobulą klasifikacijos tikslumą ($\text{Precision} = 1,0$; $\text{Recall} = 1,0$) dviejų tipų buteliams testuojant, tačiau pikselių lygmens lokalizacijos tikslumas liko ribotas, o rezultatai negali būti apibendrinti visiems aštuoniems butelių tipams. Vienas svarbiausių tyrimo radinių yra, jog fono pašalinimas vidutiniškai pagerino visų modelių tikslumą $+0,214$ mAP@0.5, o tai rodo, kad duomenų paruošimo kokybė turi didesnę įtaką galutiniam rezultatui nei modelio architektūros pasirinkimas. Taip pat nustatyta, kad butelio bounding box anotavimas pagerino YOLOv11m rezultatus, tačiau neprilygo fono pašalinimo efektyvumui, o perkėlimo mokymasis buvo būtinas SSD ir Faster R-CNN modeliams, bet nebūtinai YOLOv11m dėl efektyvesnės architektūros.

APIBENDRINIMAS. IŠVADOS

Atlikta giliojo mokymosi metodų lyginamoji analizė ir eksperimentinis tyrimas parodė, kad veiksminga stiklinių butelių defektų aptikimo sistema gali būti sukurta naudojant vartotojų klasės įrangą ir atviro kodo bibliotekas, tai svarbu smulkioms ir vidutinėms įmonėms, kurios negali investuoti į specializuotą pramoninę infrastruktūrą. Literatūros analizė atskleidė, kad prižiūrimojo mokymosi metodai (SSD, Faster R–CNN, YOLO) ir neprižiūrimojo mokymosi metodai (PatchCore) papildo vienas kitą, todėl tyrimui atrinkti keturi modeliai, atstovaujantys skirtingoms paradigmoms. Šių modelių mokymui paruoštas nuosavas 708 nuotraukų duomenų rinkinys, apimantis aštuonių tipų stiklinius butelius su 1 070 anotuotų defektų atvejų, kadangi viešai prieinami rinkiniai pasirodė netinkami dėl didelio domeno neatitiktens. Eksperimentų eigoje labiausiai netikėtas buvo fono pašalinimo efektas, vidutiniškai $+0,214$ mAP@0.5 pagerėjimas visuose modeliuose parodė, kad duomenų paruošimas šiame kontekste turėjo didesnę įtaką galutiniam tikslumui nei pats architektūros pasirinkimas, o tai retai pabrėžiama panašiuose tyrimuose. Geriausius rezultatus pasiekė YOLOv11m su fono pašalinimo duomenų variantu (0,844 mAP@0.5, 0,802 Recall), taip pat būdamas greičiausias iš trijų prižiūrimojo mokymosi modelių. PatchCore, nors testuotas tik su dviejų tipų buteliais, pasiekė tobulus klasifikacijos rodiklius be jokių anotacijų, tačiau jo pikselių lygio lokalizacijos tikslumas liko ribotas. Visgi tyrimas turi keletą apribojimų, duomenų rinkinys nedidelis, defektai dirbtinai sukurti, o testavimo imtis mažoka statistiniam patikimumui užtikrinti, todėl ateities tyrimuose verta išplėsti rinkinį naudojant realios gamybos metu atsirandančius defektus, išbandyti gilesnius PatchCore stuburo tinklus lokalizacijos tikslumui gerinti ir optimizuoti sistemas realiojo laiko konvejerio integracijai.

1. Literatūros analizė atskleidė, kad šiuolaikiniai defektų aptikimo metodai skirstomi į dvi pagrindines paradigmas, prižiūrimojo mokymosi (SSD, Faster R–CNN, YOLO) ir neprižiūrimojo mokymosi (PatchCore), kurios papildo viena kitą: vieno etapo detektoriai išsiskiria greičiu, dviejų etapų lokalizacijos tikslumu, o anomalijų aptikimo metodai gebėjimu identifikuoti nežinomus defektų tipus be anotavimo. Kadangi visi tirti modeliai apdoroja atskirus kadrus nepriklausomai, jų taikymas vaizdų sekose, pavyzdžiui, gamybos linijos kamerų srautuose, metodologiškai nereikalauja papildomų architektūrinių modifikacijų.

2. Tyrimui atrinkti keturi giliojo mokymosi metodai, atstovaujantys skirtingoms paradigmoms ir tinkami įvairaus dydžio defektų aptikimui: SSD MobileNetV2 FPN ir YOLOv11m pasirinkti dėl greičio bei efektyvumo realaus laiko sistemose, Faster R–CNN ResNet50 dėl aukšto lokalizacijos tikslumo, o PatchCore dėl unikalaus gebėjimo veikti be brokuotų gaminių anotacijų ir aptikti defektų tipus, kurių modelis anksčiau nematė. Taigi tokia metodų kombinacija leidžia lanksčiai spręsti skirtingus defektų aptikimo uždavinius.

3. Paruoštas nuosavas stiklo butelių broko aptikimo metodų eksperimentinis duomenų rinkinys: 708 nuotraukos su 1 070 anotuotų defektų atvejų, apimantis aštuonių tipų stiklinius butelius, fotografuotus trijose skirtingose aplinkose ir sudarant nuoseklias vaizdų sekas iš skirtingų kampų. Papildomai anomalijų aptikimui sudarytas 2 147 gerų nuotraukų rinkinys. Viešai prieinami duomenų rinkiniai pasirodė netinkami dėl reikšmingo domenų neatitikimo.

4. Visi keturi modeliai sėkmingai įgyvendinti naudojant atitinkamas bibliotekas ir API: SSD ir Faster R-CNN naudojant TensorFlow Object Detection API, YOLOv11m naudojant Ultralytics, o PatchCore naudojant anomalib. Eksperimentų metu nustatyta, kad fono pašalinimas naudojant rembg biblioteką yra kritiškai svarbus paruošimo žingsnis, kurio įtaka galutiniam tikslumui visuose modeliuose (+0,214 mAP@0.5 vidutiniškai) viršijo skirtumą tarp pačių architektūrų.

5. Eksperimentiniai rezultatai parodė, kad geriausius rodiklius pasiekė YOLOv11m su fono šalinimo duomenų variantu, 0,844 mAP@0.5 ir 0,802 Recall, t. y. aukščiausius rezultatus tarp visų tirtų prižiūrimojo mokymosi modelių. PatchCore pasiekė tobulus klasifikacijos rodiklius (Precision = 1,0; Recall = 1,0) testuojant dviejų tipų buteliais, nors pikselių lygio lokalizacijos tikslumas liko ribotas. SSD MobileNetV2 FPN, nors ir prastesnis tikslumo atžvilgiu, išsiskyrė stabiliausiu inferavimo laiku ir mažiausiu parametų skaičiumi, todėl lieka patraukliu pasirinkimu itin ribotų išteklių aplinkose.

LITERATŪRA

1. Advantech. (2022, September 5). *Artificial intelligence-based defect inspection for glass manufacturing*. <https://buyemt.advantech.com/CMS/CmsDetail.aspx?CMSID=878b11d6-a7fd-4ec1-99be-85cdb87bd472&country=Lithuania&token=639145579240981402&f=AEU>
2. Anaconda. (2020). 2020 state of data science: Moving from hype toward maturity. <https://www.anaconda.com/blog/2020-anaconda-state-of-data-science-report-moving-from-hype-toward-maturity>
3. Baheti, P. (2021). Train, validation, test split: How to & best practices. V7 Labs. <https://www.v7darwin.com/blog/train-validation-test-set>
4. Benzine, A., El Hassouni, M., & Darmont, J. (2019). Complexity and accuracy analysis of common artificial neural networks on pedestrian detection. arXiv. <https://doi.org/10.48550/arXiv.1811.08287>
5. CloudFactory. (n.d.). *Faster R-CNN*. CloudFactory Wiki. <https://wiki.cloudfactory.com/docs/mp-wiki/model-architectures/faster-r-cnn>
6. Claypo, N., Jaiyen, S., & Hanskunatai, A. (2023). Inspection system for glass bottle defect classification based on deep neural network. *International Journal of Advanced Computer Science and Applications*, 14(7). <https://doi.org/10.14569/IJACSA.2023.0140738>
7. Diwan, T., Anirudh, G., & Tembhurne, J. V. (2022). Object detection using YOLO: Challenges, architectural successors, datasets and applications. *Multimedia Tools and Applications*, 82, 9243–9275. <https://doi.org/10.1007/s11042-022-13644-y>
8. Dwyer, B., Nelson, J., & Solawetz, J. (2022). *Roboflow* (Version 1.0) [Software]. <https://roboflow.com/>
9. He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2961–2969). <https://doi.org/10.1109/ICCV.2017.322>
10. He, Z., Wang, K., Fang, T., Su, L., Chen, R., & Fei, X. (2024). Comprehensive performance evaluation of YOLOv11, YOLOv10, YOLOv9, YOLOv8 and YOLOv5 on object detection of power equipment. arXiv. <https://doi.org/10.48550/arXiv.2411.18871>
11. Jégou, S., Drozdal, M., Vazquez, D., Romero, A., & Bengio, Y. (2017). The one hundred layers tiramisu: Fully convolutional DenseNets for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 11–19). <https://doi.org/10.1109/CVPRW.2017.156>
12. Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., & Qu, R. (2019). A survey of deep learning-based object detection. *IEEE Access*, 7, 128837–128868. <https://doi.org/10.1109/ACCESS.2019.2939201>
13. Khanam, R., & Hussain, M. (2024). YOLOv11: An overview of the key architectural enhancements. arXiv. <https://doi.org/10.48550/arXiv.2410.17725>
14. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84–90. <https://doi.org/10.1145/3065386>

15. Kumari, P., et al. (2024). A comprehensive investigation of anomaly detection methods in deep learning and machine learning: 2019–2023. *IET Information Security*. <https://doi.org/10.1049/2024/8821891>
16. kuzand. (2022). *One stage vs two stage object detection* [Stack Overflow answer]. Stack Overflow. <https://stackoverflow.com/questions/65942471/one-stage-vs-two-stage-object-detection>
17. Latina, M. A., Dela Cruz, J. V. R., & Delos Santos, F. D. (2022). Empty glass bottle defect detection based on deep learning with CNN using SSD MobileNetV2 model. In *2022 IEEE 14th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)* (pp. 1–6). <https://doi.org/10.1109/HNICEM57413.2022.10109368>
18. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
19. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot MultiBox detector. In *European Conference on Computer Vision (ECCV 2016)* (pp. 21–37). Springer. https://doi.org/10.1007/978-3-319-46448-0_2
20. Lu, J., & Lee, S. H. (2023). Real-time defect detection model in industrial environment based on lightweight deep learning network. *Applied Sciences*, 13(14), 8125. <https://doi.org/10.3390/electronics12214388>
21. Luo, H., Zhao, H., & Yin, S. (2018). Data-driven design of fog-computing-aided process monitoring system for large-scale industrial processes. *IEEE Transactions on Industrial Informatics*, 15(7), 4334–4342. <https://doi.org/10.1109/TII.2018.2843124>
22. Malamas, E. N., Petrakis, E. G., Zervakis, M., Petit, L., & Legat, J. D. (2003). A survey on industrial vision systems, applications and tools. *Image and Vision Computing*, 21(2), 171–188. [https://doi.org/10.1016/S0262-8856\(02\)00152-X](https://doi.org/10.1016/S0262-8856(02)00152-X)
23. Mao, M., et al. (2025). Workpiece surface defect detection based on YOLOv11 and edge computing. *PLoS ONE*, 20(1). <https://doi.org/10.1371/journal.pone.0327546>
24. Pang, G., Shen, C., Cao, L., & Van Den Hengel, A. (2021). Deep learning for anomaly detection: A review. *ACM Computing Surveys*, 54(2), 1–38. <https://doi.org/10.1145/3439950>
25. Qin, X., Zhang, Z., Huang, C., Dehghan, M., Zaiane, O. R., & Jagersand, M. (2020). U2-Net: Going deeper with nested U-structure for salient object detection. *Pattern Recognition*, 106, 107404. <https://doi.org/10.1016/j.patcog.2020.107404>
26. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 779–788). <https://doi.org/10.1109/CVPR.2016.91>
27. Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137–1149. <https://doi.org/10.1109/TPAMI.2016.2577031>

28. Rohan, A., Rabah, M., & Kim, S.-H. (2019). *Convolutional neural network-based real-time object detection and tracking for Parrot AR Drone 2*. *IEEE Access*, 7, 69575–69584. <https://doi.org/10.1109/ACCESS.2019.2919332>
29. Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 234–241). Springer. https://doi.org/10.1007/978-3-319-24574-4_28
30. See, J. E. (2012). Visual inspection: A review of the literature. *Sandia National Laboratories*. <https://doi.org/10.2172/1055636>
31. Shelhamer, E., Long, J., & Darrell, T. (2017). Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4), 640–651. <https://doi.org/10.1109/TPAMI.2016.2572683>
32. Shirzad, A., Ren, J., & El-Gindy, M. (2023). Defect detection methods for industrial products using deep learning techniques: A review. *Applied Sciences*, 13(4), 2244. <https://doi.org/10.3390/a16020095>
33. Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), 1–48. <https://doi.org/10.1186/s40537-019-0197-0>
34. Shperber, G. (2019, October 6). *Background removal with deep learning*. Shibumi AI. <https://www.shibumi-ai.com/post/background-removal-with-deep-learning-gidi-shperber>
35. Terven, J., Córdova-Esparza, D. M., & Romero-González, J. A. (2023). A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction*, 5(4), 1680–1716. <https://doi.org/10.3390/make5040083>
36. Wang, A., Chen, H., Liu, L., et al. (2024). YOLOv10: Real-time end-to-end object detection. *arXiv*. <https://doi.org/10.48550/arXiv.2405.14458>
37. Wang, C. Y., Yeh, I. H., & Liao, H. Y. M. (2024). YOLOv9: Learning what you want to learn using programmable gradient information. *arXiv*. <https://doi.org/10.48550/arXiv.2402.13616>
38. Wang, J., Zhao, Y., Wang, Y., Hu, H., Li, J., Du, J., & Ding, H. (2022). Application of an improved U2-Net model in ultrasound median neural image segmentation. *Diagnostics*, 12(12), 3041. <https://doi.org/10.1016/j.ultrasmedbio.2022.08.003>
39. Wang, L., et al. (2024). A defect detection method for industrial aluminum sheet surface based on improved YOLOv8 algorithm. *Frontiers in Physics*, 12, 1419998. <https://doi.org/10.3389/fphy.2024.1419998>
40. Wang, Q., Wang, M., Sun, J., Chen, D., & Shi, P. (2025). Review of surface-defect detection methods for industrial products based on machine vision. *IEEE Access*, 13, 90668–90697. <https://doi.org/10.1109/ACCESS.2025.3571297>
41. Xia, H., et al. (2022). Improved Faster R-CNN based surface defect detection algorithm for plates. *Computational Intelligence and Neuroscience*, 2022. <https://doi.org/10.1155/2022/3248722>

42. Xie, X. (2008). A review of recent advances in surface defect detection using texture analysis techniques. *ELCVIA Electronic Letters on Computer Vision and Image Analysis*, 7(3), 1–22. <https://doi.org/10.5565/rev/elevia.268>
43. Yang, Q., & Guo, R. (2024). An unsupervised method for industrial image anomaly detection with Vision Transformer–based autoencoder. *Sensors*, 24(8), 2440. <https://doi.org/10.3390/s24082440>
44. Yang W. (2022). A survey of surface defect detection based on deep learning. *Advances in Social Science, Education and Humanities Research* (Vol. 703, pp. 362–367). Atlantis Press. https://doi.org/10.2991/978-2-494069-51-0_51
45. Zhang, T., et al. (2025). A lightweight YOLOv11–based framework for small steel defect detection with a newly enhanced feature fusion module. *Scientific Reports*, 15(1). <https://doi.org/10.1038/s41598-025-16619-9>
46. Zhao, J., Zhou, S., Zheng, Q., & Mei, S. (2022). Fabric defect detection based on transfer learning and improved Faster R–CNN. *Journal of Engineered Fibers and Fabrics*, 17. <https://doi.org/10.1177/15589250221086647>
47. Zhong, R. Y., Xu, X., Klotz, E., & Newman, S. T. (2017). Intelligent manufacturing in the context of Industry 4.0: A review. *Engineering*, 3(5), 616–630. <https://doi.org/10.1016/J.ENG.2017.05.015>

PRIEDAI

A priedas. 13-oji tarptautinė elektronikos ir informacinių mokslų konferencija „eStream 2026“.

Defect Detection Strategies for Glass Bottle Inspection

Radoslav TITKO, Dalius MATUZEVIČIUS

E-mail: radoslav.titko@stud.vilniustech.lt, dalius.matuzevicius@vilniustech.lt,
Department of Electronic Systems, Vilnius Gediminas Technical University (VILNIUS TECH)

Introduction

Quality control is a critical challenge in manufacturing, where defects such as scratches, cracks, and air bubbles reduce product quality and increase costs. Traditional manual inspection is inefficient and inaccurate ($\approx 70\text{--}80\%$ detection rate).

Recent advances in deep learning, especially convolutional neural networks (CNNs), enable automated and highly accurate defect detection. However, most solutions require expensive industrial hardware, limiting their applicability for small and medium enterprises (SMEs).

This study evaluates whether modern object detection models can achieve high performance under non-industrial, resource-constrained conditions.

Objective

- Compare three object detection models:
 - SSD MobileNetV2
 - Faster R-CNN (ResNet50)
 - YOLOv11m
- Evaluate:
 - Training from scratch vs transfer learning
 - Performance on limited data and consumer hardware

Methodology

- Dataset:
 - 708 images $\rightarrow$ 1,848 after augmentation
 - Defects: scratches, air bubbles, and chipped necks



Figure 1: Types of defects: scratch, air bubble, and chipped neck

- Training Setup:
 - Hardware: RTX 3060 (6GB)
 - Metric: mAP@0.5
 - Training:
 - From scratch
 - Transfer learning (COCO)

Models

- SSD MobileNetV2 FPN
 - Parameters: $\sim 6.8\text{M}$
 - Inference time: 111.67 ms / image
 - Std: 6.85 ms
 - Speed: 8.96 FPS
 - Input resolution: 640×640 (VRAM-limited setup)
 - Strength: stable performance and lightweight architecture
- Faster R-CNN ResNet50
 - Parameters: $\sim 41\text{M}$
 - Inference time: 109.44 ms / image

- Std: 84.09 ms
- Speed: 9.14 FPS
- Input resolution: 640×640 (VRAM-limited setup)
- Strength: higher accuracy but unstable latency

YOLOv11m

- Parameters: $\sim 20\text{--}25\text{M}$
- Inference time: 68.63 ms / image
- Std: 29.87 ms
- Speed: 14.57 FPS
- Input resolution: 896×896
- Strength: best speed-accuracy-scalability trade-off

Results

Model	mAP@0.5	mAP@0.5-95
SSD MobileNetV2 (Scratch)	0.6654	0.4689
SSD MobileNetV2 (Transfer)	0.7973	0.5660
Faster R-CNN (Scratch)	0.5100	0.3384
Faster R-CNN (Transfer)	0.8140	0.5966
YOLOv11m (Scratch)	0.9071	0.7142
YOLOv11m (Transfer)	0.9132	0.7261

Table 1: Model performance comparison



Figure 2: 1 - YOLOv11, 2 - SSD, 3 - Faster R-CNN, 4 - Ground truth

Analysis

- YOLOv11m outperformed all models, even when trained from scratch
- Transfer learning significantly improved older architectures
- Minimal gains for YOLOv11m with transfer learning
- Small defect detection remains challenging
- Some false positives due to reflections

Conclusions

- Modern architectures (YOLOv11m) provide superior performance
- Transfer learning is crucial for older models
- Deep learning is feasible even with limited data and hardware
- Suitable for SMEs and edge deployment scenarios

Future Work

- Improve small defect detection
- Reduce false positives
- Explore:
 - Model pruning
 - Knowledge distillation
 - Edge optimization



13th International Conference "Electrical,
Electronic and Information Sciences"

eStream 2026

CERTIFICATE

Radoslav Titko

Delivered an Poster Presentation

**DEFECT DETECTION STRATEGIES FOR GLASS
BOTTLE INSPECTION**

Chair of Science Committee

A handwritten signature in black ink, appearing to read 'D. Navakas'.

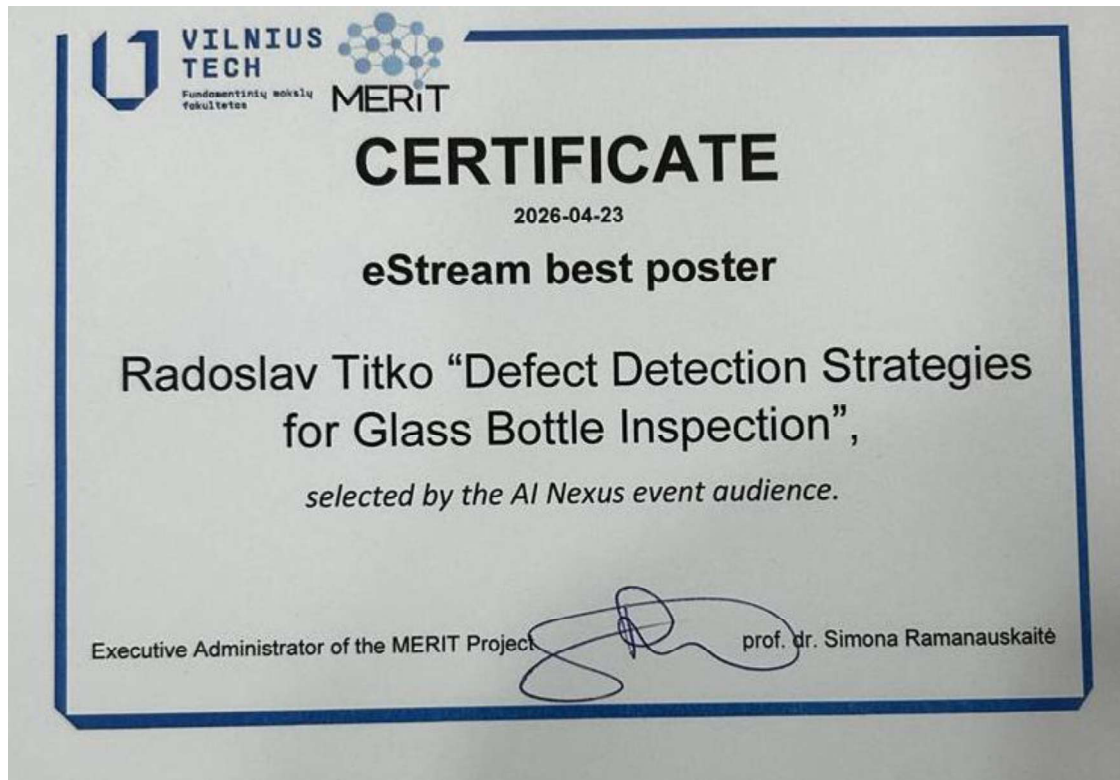
Prof. Dr. Dalius Navakas

Conference Secretary

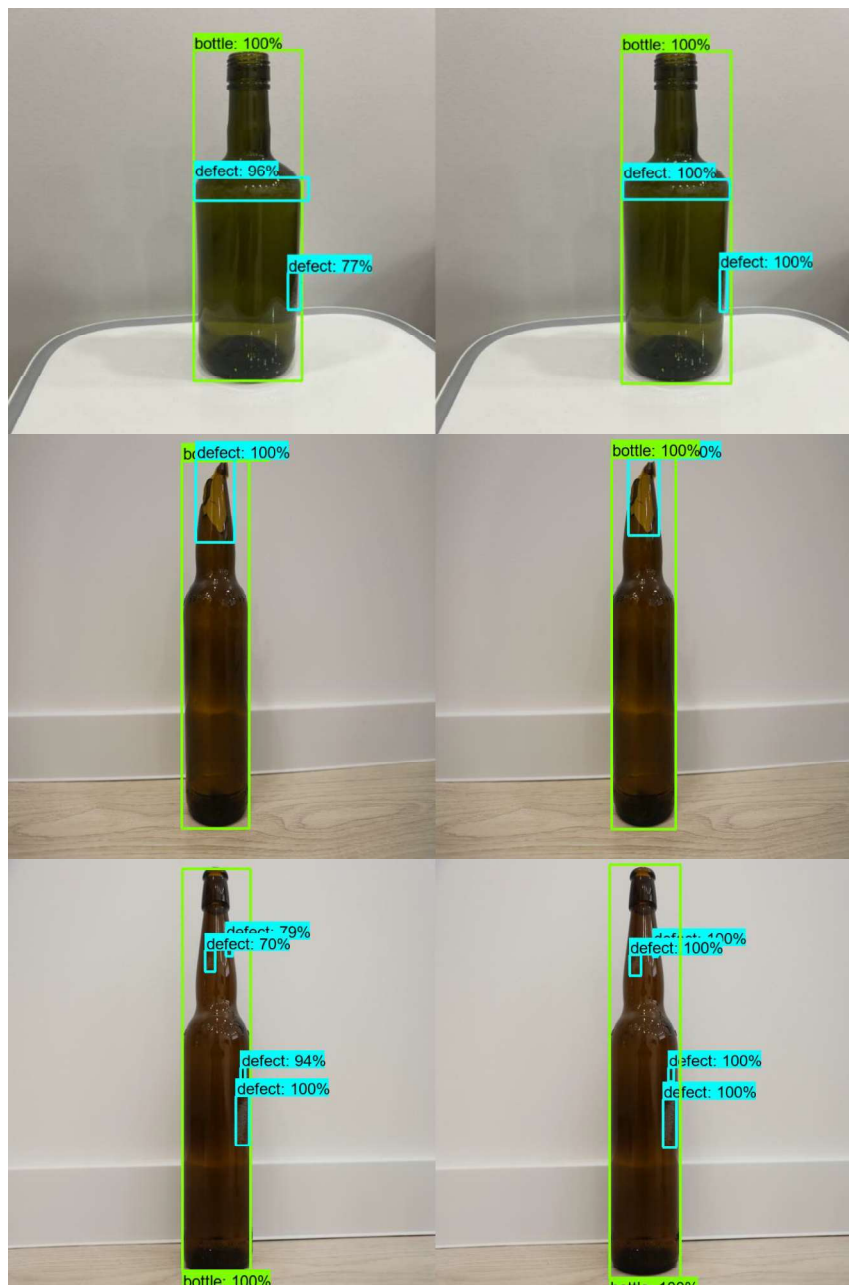
A handwritten signature in black ink, appearing to read 'J. Anulytė'.

Assoc. Prof. Dr. Justina Anulytė

April 23–24, 2026
Vilnius



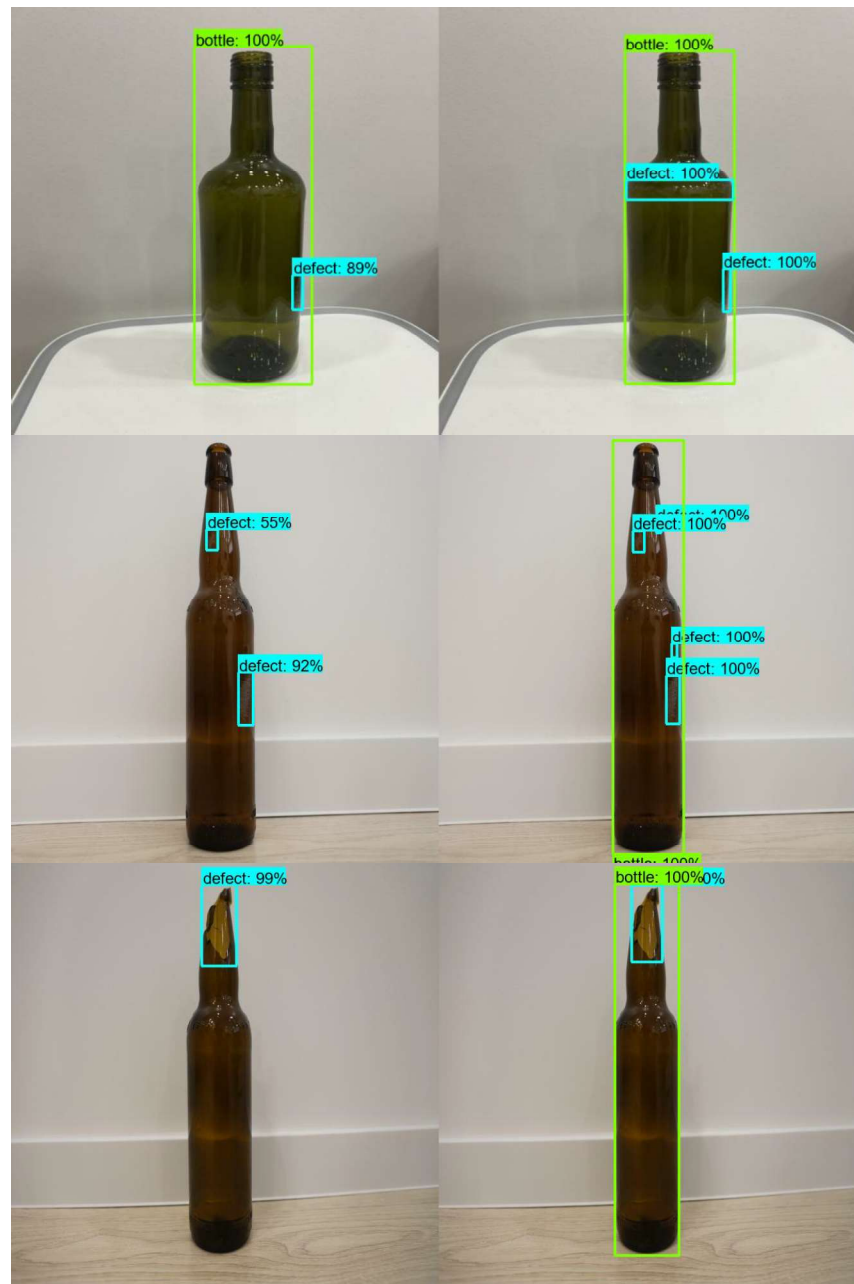
B priedas. Stiklinių butelių brokų detekcijų rezultatai naudojant skirtingus modelius.



Faster R-CNN perkėlimo mokymosi vizualiniai rezultatai



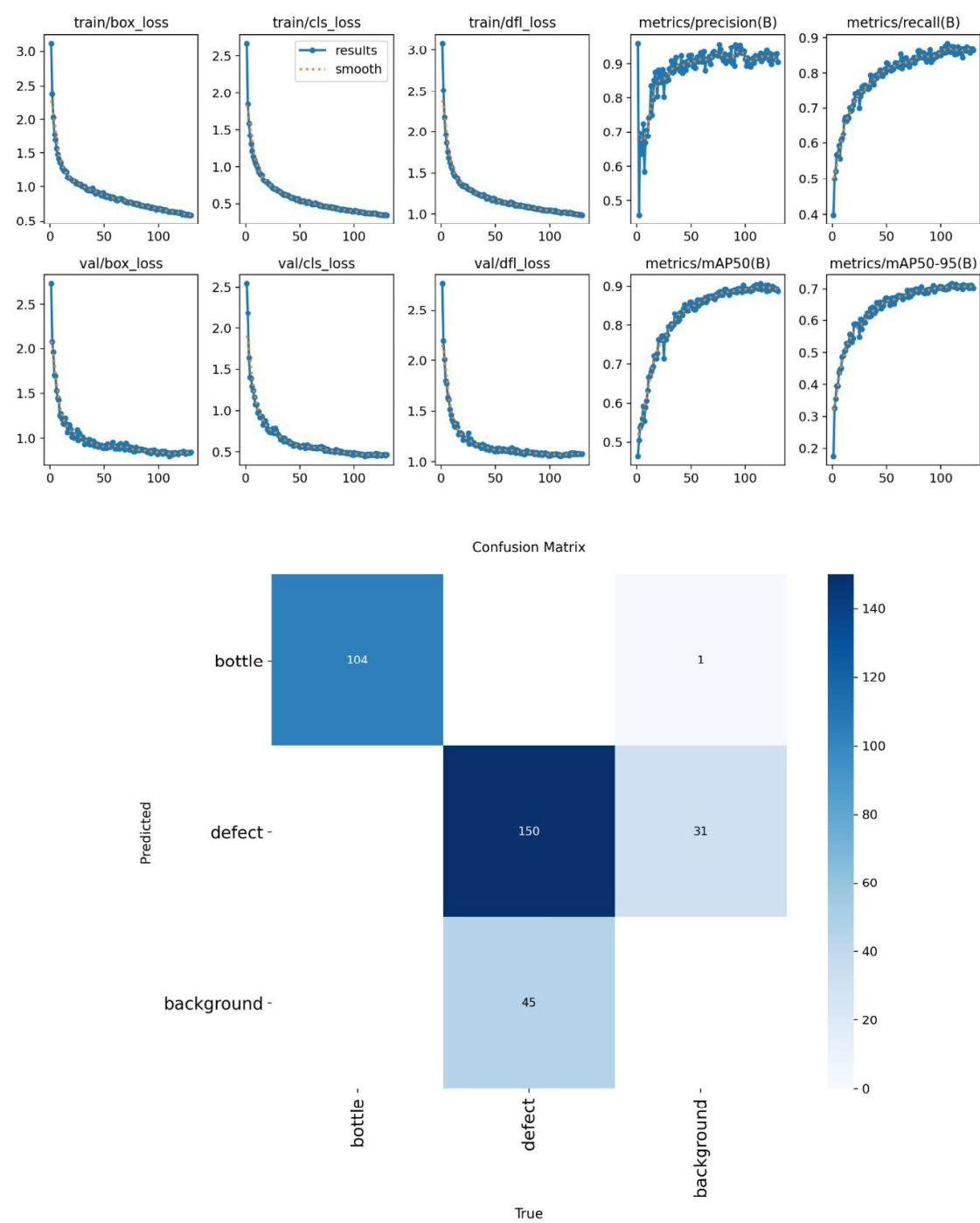
Faster R-CNN mokymosi nuo nulio vizualiniai rezultatai



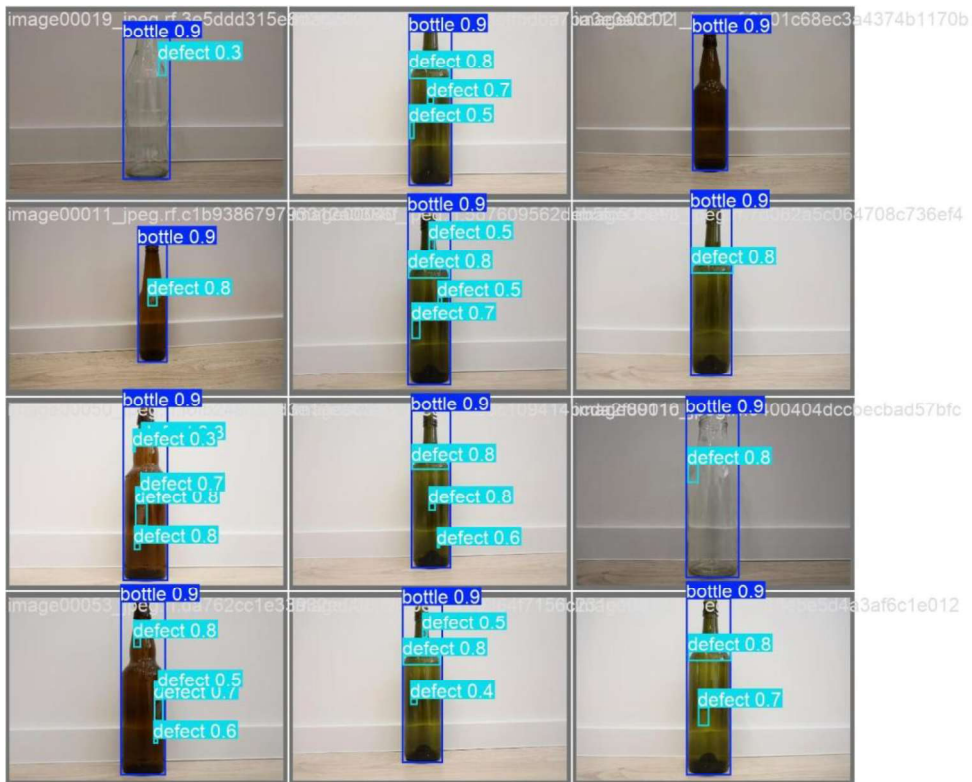
SSD perkėlimo mokymosi vizualiniai rezultatai



SSD mokymosi nuo nulio vizualiniai rezultatai



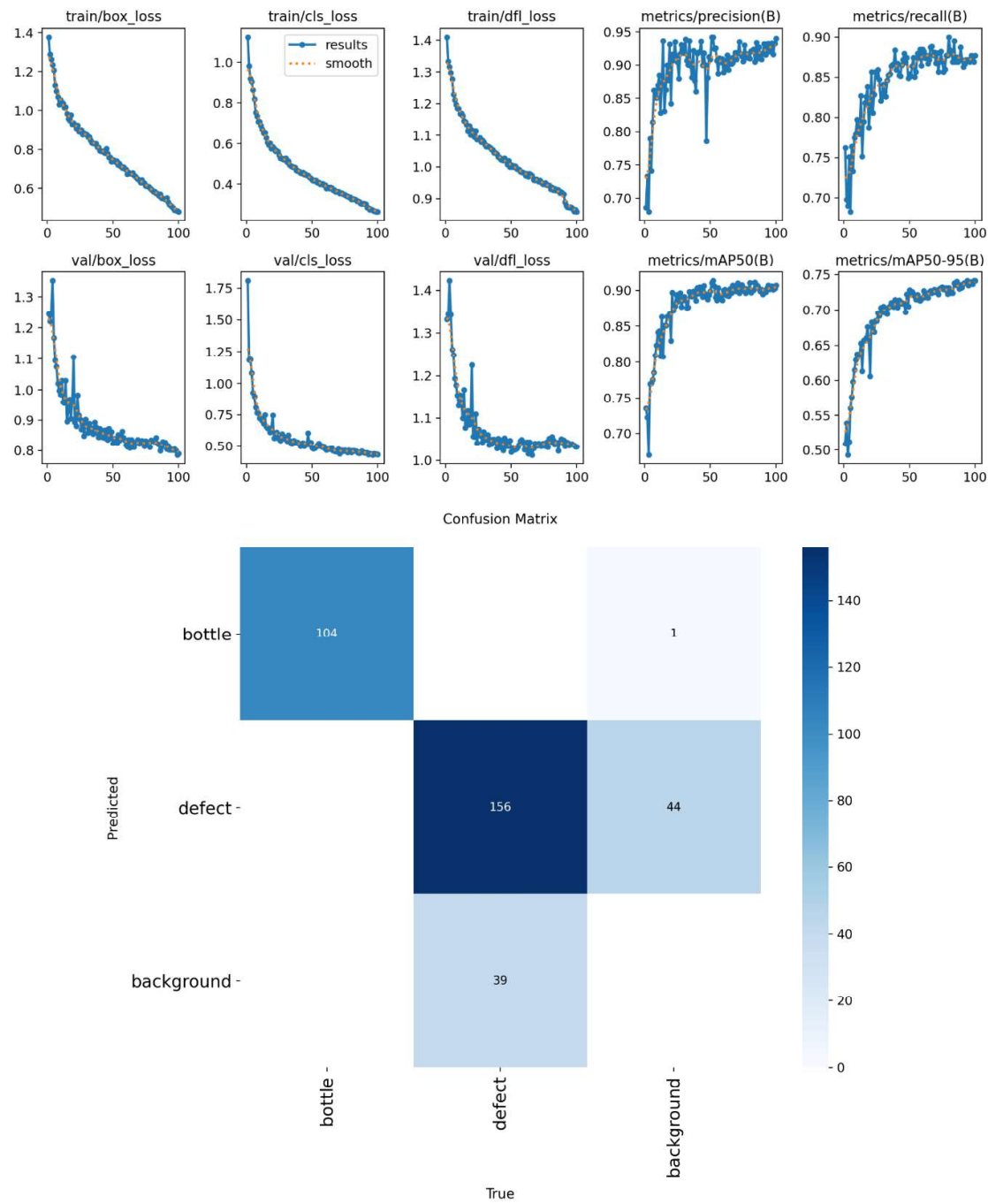
YOLO mokymosi nuo nulio rezultatai



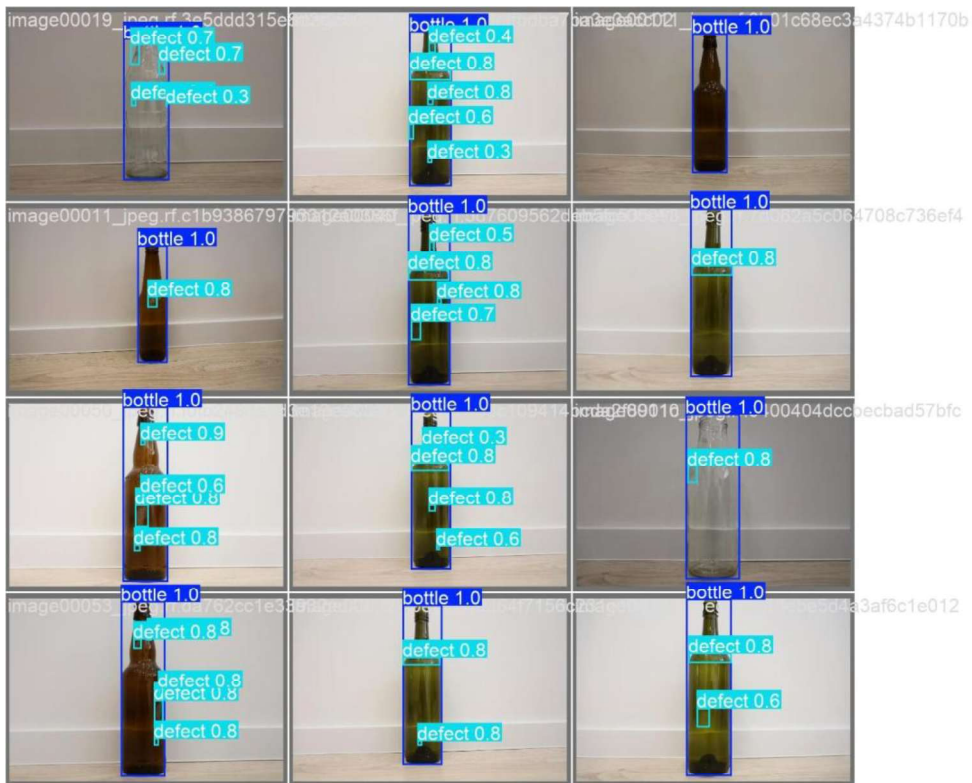
YOLO mokymosi nuo nulio spėjimai



YOLO mokymosi nuo nulio tikros kaukės



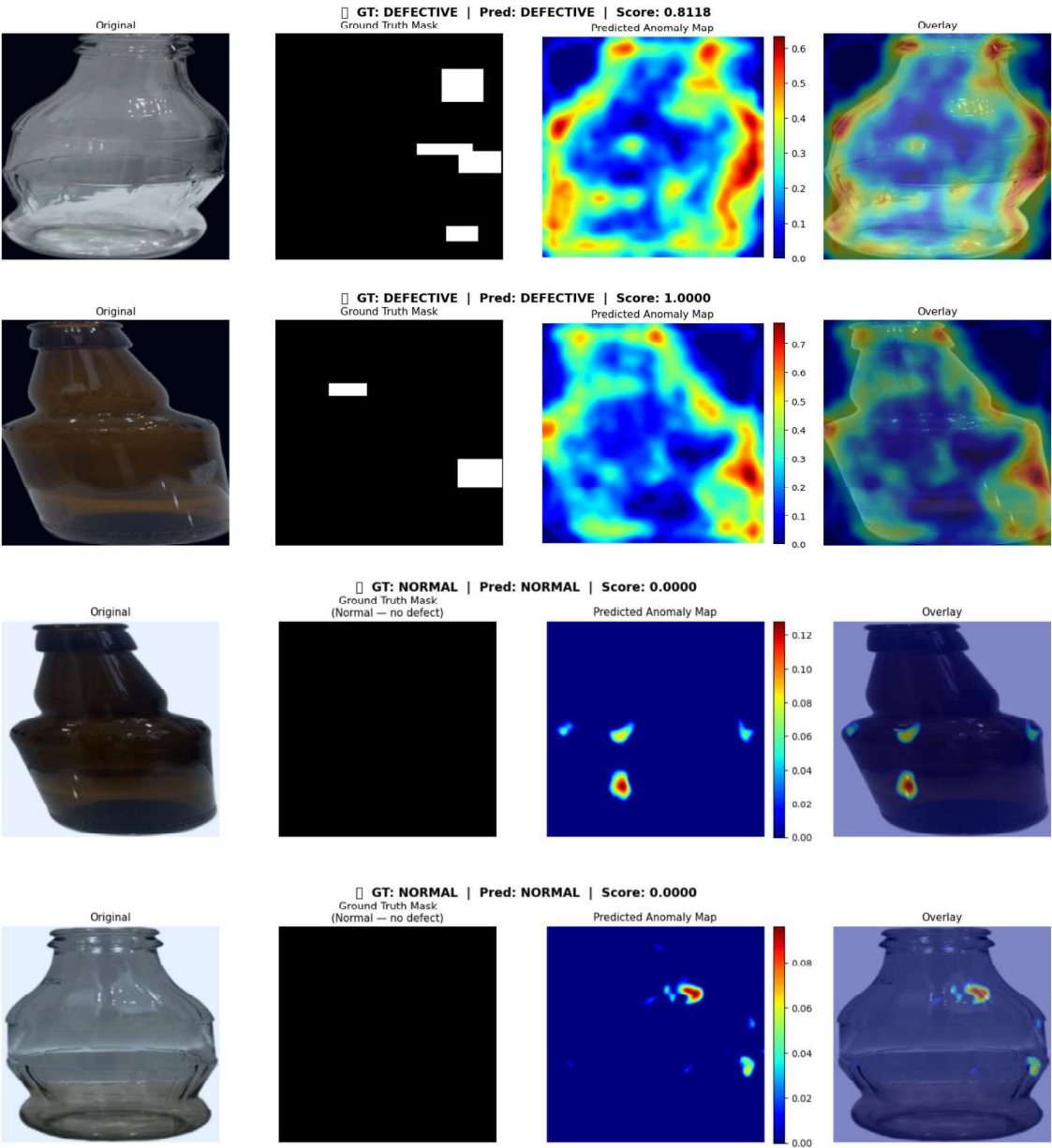
YOLO perkėlimo mokymas



YOLO perkėlimo mokymosi spėjimai



YOLO perkėlimo mokymosi tikros kaukės



Anomalijų detekcijos vizualiniai rezultatai