



VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
ANTANO GUSTAIČIO AVIACIJOS INSTITUTAS
AERONAUTIKOS INŽINERIJOS KATEDRA

Anita Kušlevič

TAIKINIŲ SEKIMO SISTEMOS TYRIMAS
RESEARCH OF THE TARGET TRACKING SYSTEM

Baigiamasis magistro darbas

Aerokosmoso inžinerijos studijų programa, valstybinis kodas 6211EX060

Aeronautikos inžinerijos studijų kryptis

Vilnius, 2023

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
ANTANO GUSTAIČIO AVIACIJOS INSTITUTAS
AERONAUTIKOS INŽINERIJOS KATEDRA

TVIRTINU
Katedros vedėjas

(Parašas)
Darius Rudinskas

(Data)

Anita Kušlevič

TAIKINIŲ SEKIMO SISTEMOS TYRIMAS
RESEARCH OF THE TARGET TRACKING SYSTEM

Baigiamasis magistro darbas

Aerokosmoso inžinerijos studijų programa, valstybinis kodas 6211EX060
Aeronautikos inžinerijos studijų kryptis

Vadovas

dr. Ivan Suzdalev
(Moksl. laipsnis/pedag. vardas, vardas, pavardė)

(Parašas)

(Data)

Lietuvių kalbos konsultantė

lekt. Angelika Petrėtienė
(Moksl. laipsnis/pedag. vardas, vardas, pavardė)

(Parašas)

(Data)

Vilnius, 2023

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
ANTANO GUSTAIČIO AVIACIJOS INSTITUTAS
AERONAUTIKOS INŽINERIJOS KATEDRA

Aeronautikos inžinerijos studijų kryptis

Aerokosmoso inžinerijos studijų programa, valstybinis kodas 6211EX060

TVIRTINU
Katedros vedėjas

(Parašas)

____Darius Rudinskas____

(Vardas, pavardė)

(Data)

BAIGIAMOJO MAGISTRO DARBO

UŽDUOTIS

Nr. AIfmu-21–3668

Vilnius

Studentui (ei)

Anitai Kušlevič

(Vardas, pavardė)

Baigiamojo darbo tema:

Taikinių sekimo sistemos tyrimas

Research of the Target Tracking System

Baigiamojo darbo užbaigimo terminas 2023 m. ____ d.

BAIGIAMOJO DARBO UŽDUOTIS:

Sukurti vizualinio bepiločių orlaivių sekimo metodą ir tuo metodu paremtą algoritmą. Atlikti literatūros apžvalgą bei analizę ir ištirti esamus taikinių sekimo algoritmus. Išsirinkti metodus, tinkamus naujo taikinių sekimo algoritmo, paremto konvoliucinių neuroninių tinklų algoritmais, sintezei. Aprobuoti sukurtą algoritmą ir pateikti gautų rezultatų apžvalgą. Pateikti išvadas apie tinkamumą arba netinkamumą naudoti sukurtą metodą.

Baigiamojo darbo rengimo konsultantai:

lietuvių kalbos konsultantė lekt. Angelika Petrėtienė

(Moksl. laipsnis/pedag. vardas, vardas, pavardė)

Vadovas

.....

(Parašas)

dr. Ivan Suzdalev

(Moksl. laipsnis/pedag. vardas, vardas, pavardė)

Užduotį gavau

.....

(Parašas)

Anita Kušlevič

(Vardas, pavardė)

2023 m. ____ d.

(Data)

<table><tr><td>Vilniaus Gedimino technikos universitetas</td></tr><tr><td>Antano Gustaičio aviacijos institutas</td></tr><tr><td>Aeronautikos inžinerijos katedra</td></tr></table>	Vilniaus Gedimino technikos universitetas	Antano Gustaičio aviacijos institutas	Aeronautikos inžinerijos katedra	<table><tr><td>ISBN</td><td>ISSN</td></tr><tr><td>Egz. sk.</td><td></td></tr><tr><td>Data-.....-.....</td><td></td></tr></table>	ISBN	ISSN	Egz. sk.		Data-.....-.....	
Vilniaus Gedimino technikos universitetas										
Antano Gustaičio aviacijos institutas										
Aeronautikos inžinerijos katedra										
ISBN	ISSN									
Egz. sk.										
Data-.....-.....										
Antrosios pakopos studijų Aerokosmoso inžinerijos programos magistro baigiamasis darbas										
Pavadinimas	Taikinių sekimo sistemos tyrimas									
Autorius	Anita Kušlevič									
Vadovas	Ivan Suzdalev									
	Kalba: lietuvių									
Anotacija										
Baigiamajame magistro darbe kuriamas vizualinio bepiločių orlaivių sekimo metodas. Atlikta literatūros apžvalga ir analizė, išnagrinėti taikinių sekimo sistemose taikomi metodai. Kuriamam sekimo metodui pasirinkti „YOLOv8“ ir „ByteTrack“ algoritmai. Pažangių kompiuterinės regos technologijų derinys suteikia patvarų ir efektyvų sprendimą bepiločių orlaivių sekimui. Siūlomas metodas sujungia abiejų algoritmų pranašumus, kad būtų pasiektas geresnis veikimas, siekiant didesnio tikslumo, greičio ir efektyvumo. Siūlomo metodo veikimo įvertinimo rezultatai parodė, kad sistema gali aptikti ir sekti dronus, esant aukštiesiems preciziškumo ir jautrumo įverčiams. Sistemos lankstumas buvo patvirtintas įvairiais scenarijais, įskaitant fono sudėtingumą ir drono uždengimą, o tai parodo algoritmo praktinio pritaikymo galimybes įvairiose srityse. Atliktas tyrimas pabrėžia giliojo mokymosi metodų derinimo potencialą sprendžiant realaus pasaulio problemas. Būsimi tyrimai gali būti pritaikyti naudojant pažangesnes architektūras ir duomenų papildymo metodus, siekiant pagerinti siūlomos sistemos našumą. Darbą sudaro 6 dalys: įvadas, literatūros apžvalga ir analizė, metodų apžvalga ir analizė, bepiločio orlaivio sekimo sistemos kūrimas, išvados ir rekomendacijos, literatūros sąrašas. Darbo apimtis – 98 p. teksto be priedų, 66 iliustracijos, 10 lentelių, 83 bibliografiniai šaltiniai. Atskirai pridedami darbo priedai.										
Prasminiai žodžiai: sekimas, aptikimas, neuroniniai tinklai, gilusis mokymasis, bepilotis orlaivis										

<table><tr><td>Vilnius Gediminas Technical University</td></tr><tr><td>Antanas Gustaitis' Aviation Institute</td></tr><tr><td>Department of Aeronautical Engineering</td></tr></table>		Vilnius Gediminas Technical University	Antanas Gustaitis' Aviation Institute	Department of Aeronautical Engineering	<table><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											
Antanas Gustaitis' Aviation Institute											
Department of Aeronautical Engineering											
ISBN	ISSN										
Copies No.											
Date-.....-.....											
Master Degree Studies Aerospace Engineering study programme Master Graduation Thesis											
Title	Research of the Target Tracking System										
Author	Anita Kušlevič										
Academic supervisor	Ivan Suzdalev										
<table><tr><td></td><td>Thesis language: Lithuanian</td></tr></table>			Thesis language: Lithuanian								
	Thesis language: Lithuanian										
Annotation The master's thesis develops a method for visual drone tracking. The current thesis reviews the literature and provides a comprehensive analysis of the scientific discourse on the topic discussed; it also examines the methods for target tracking systems. The “YOLOv8” and “ByteTrack” algorithms were chosen to develop the tracking method. By leveraging advanced computer vision technologies, a robust and efficient solution for drone tracking is achieved. Performance evaluation results demonstrate the system's ability to detect and track drones precisely. The system's flexibility was validated in diverse scenarios, including complex backgrounds and drone occlusion, showcasing the algorithm's potential for practical applications in various fields. The research underscores the potential of deep learning techniques in addressing real-world challenges. Future research can explore more advanced architectures and data augmentation techniques to improve the performance of the proposed system. The current thesis has the following structure: introduction, literature review and analysis, methods review and analysis, drone tracking system development, conclusions, and references. The thesis contains – 98 pages of plain text (appendixes excluded), 66 figures, 10 tables, and 83 bibliographical entries. Appendixes included.											
<table><tr><td>Keywords: tracking, detection, neural network, deep learning, drone</td></tr></table>		Keywords: tracking, detection, neural network, deep learning, drone									
Keywords: tracking, detection, neural network, deep learning, drone											

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

	Anita Kušlevič, 20163598	
	(Studento vardas ir pavardė, studento pažymėjimo Nr.)	
	Antano Gustaičio aviacijos institutas	
	(Fakultetas)	
	Aerokosmoso inžinerija, AIfmu-21	
	(Studijų programa, akademinė grupė)	

BAIGIAMOJO DARBO (PROJEKTO)
SAŽININGUMO DEKLARACIJA

2023 m. gegužės 20 d.

Patvirtinu, kad mano baigiamasis darbas tema „Taikinių sekimo sistemos tyrimas“ yra savarankiškai parašytas. Šiame darbe pateikta medžiaga nėra plagijuota. Tiesiogiai ar netiesiogiai panaudotos kitų šaltinių citatos pažymėtos literatūros nuorodose.

Mano darbo vadovas daktaras Ivan Suzdalev.

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

		Anita Kušlevič
	(Parašas)	(Vardas ir pavardė)

TURINYS

SANTRUMPŲ SĄRAŠAS	9
ILIUSTRACIJŲ SĄRAŠAS	11
LENTELIŲ SĄRAŠAS.....	14
ĮVADAS	15
1. LITERATŪROS APŽVALGA IR ANALIZĖ.....	17
1.1. Radiolokacijos sistemos.....	17
1.2. Vieno taikinio sekimo radaru principas	19
1.2.1. Būsenos įvertinimo algoritmas	20
1.2.2. Taikinio judėjimo modelis.....	22
1.3. Taikinių sekimo sistemos.....	25
1.3.1. Taikinio sekimo algoritmas, naudojant kampinių taškų atitikimą kartu su suspaudimo metodu.....	27
1.3.2. Aktyvi įvairiarūšių jutiklių sistema taikiniams atpažinti ir sekti.....	29
1.3.3. Infraraudonųjų mažų taikinių sekimo metodas.....	31
1.4. Taikinių atpažinimas ir sekimas, taikant giliuosius neuroninius tinklus	35
1.4.1. Taikinių sekimo algoritmas, pagrįstas konvoliuciniu neuroniniu tinklu	37
1.4.2. Taikinio sekimo algoritmas, pagrįstas tankiu paremta klasterizacija	41
1.4.3. Kelių taikinių sekimas, remiantis taikinių aptikimu ir bendrąja informacija	43
1.5. Vieno etapo algoritmai.....	47
1.6. Bepiločių orlaivių aptikimo metodai	50
1.6.1. Bepiločių orlaivių aptikimas, taikant konvoliucinio neuroninio tinklo metodus	53
1.6.2. Bepiločių orlaivių aptikimas, naudojant YOLO šeimos algoritmus.....	56
1.7. Skyriaus išvados	59
2. METODŲ APŽVALGA IR ANALIZĖ.....	61
2.1. Objektų aptikimo metodai	62
2.1.1. Rekursiniai metodai	64
2.1.2. Nerekursiniai metodai.....	66
2.2. Objektų klasifikavimo metodai.....	67
2.3. Objekto sekimo metodai	69
2.3.1. Taško sekimo metodai	70
2.3.2. Branduolio sekimo metodai.....	71
2.3.3. Silueto sekimo metodai.....	73
2.4. Krašto aptikimo operatoriai	73
2.5. Metodai, taikomi objektams sekti	77

2.5.1. Kalmano filtras	78
2.5.2. Išplėstinis Kalmano filtras	80
2.5.3. SORT algoritmas	81
2.5.4. BYTE algoritmas	83
2.6. Skyriaus išvados	86
3. BEPILOČIŲ ORLAIVIŲ SEKIMO SISTEMOS KŪRIMAS.....	88
3.1. Duomenų rinkinio paruošimas.....	89
3.2. Naudojamos metrikos	91
3.3. Modelio mokymas	94
3.4. Bepiločių orlaivių sekimas.....	104
3.5. Skyriaus išvados	109
IŠVADOS IR REKOMENDACIJOS.....	111
LITERATŪROS SĄRAŠAS.....	113
PRIEDAI	120
A priedas. Bepiločių orlaivių sekimo sistemos kūrimo algoritmas	
B priedas. „YOLOv8“ tinklo mokymo programos kodas	
C priedas. Bepiločių orlaivių sekimo, naudojant „YOLOv8“ tinklą ir „ByteTrack“ sekiklį, programos kodas	
D priedas. Pranešimo 26-ojoje Lietuvos jaunųjų mokslininkų konferencijoje medžiaga	

SANTRUMPŲ SĄRAŠAS

- ANN (angl. *Artificial Neural Network*) – dirbtinis neuroninis tinklas
- AP (angl. *Average Precision*) – vidurkio tikslumas
- CA (angl. *Constant Acceleration*) – pastovus pagreitis
- CKF (angl. *Cubature Kalman Filter*) – tūrio matavimo Kalmano filtras
- CNN (angl. *Convolutional Neural Network*) – konvoliucinis neuroninis tinklas
- CT (angl. *Compressive Tracking*) – sekimas suspaudimo metodu
- CV (angl. *Constant Velocity*) – pastovus greitis
- DCNN (angl. *Deep Convolutional Neural Network*) – gilusis konvoliucinis neuroninis tinklas
- DNN (angl. *Deep Neural Network*) – gilusis neuroninis tinklas
- DPA (angl. *Dynamic Programming Algorithm*) – dinaminio programavimo algoritmas
- EAO (angl. *Expected Average Overlap*) – laukiamoji sanklotos reikšmė
- EKF (angl. *Extended Kalman Filter*) – išplėstinis Kalmano filtras
- EMA (angl. *Expectation Model Algorithm*) – laukiamo modelio algoritmas
- FCOS (angl. *Fully Convolutional One-Stage*) – visiškai konvoliucinis vienos pakopos metodas
- FN (angl. *False Negative*) – klaidingai neigiamas
- FP (angl. *False Positive*) – klaidingai teigiamas
- FPS (angl. *Frames Per Second*) – per vieną sekundę apdorotų kadro skaičius
- GPB (angl. *Generalized Pseudo-Baysian*) – apibendrintas pseudo-Bajeso algoritmas
- GPU (angl. *Graphics Processing Unit*) – grafinis procesorius
- HCF (angl. *Hierarchical Convolutional Features*) – hierarchinių konvoliucijos požymių algoritmas
- HSV (angl. *Hue, Saturation, Value*) – spalvų modelis
- ID (angl. *identification*) – identifikavimas
- IDE (angl. *Integrated Development Environment*) – integruota programų kūrimo aplinka
- IMM (angl. *Interacting Multiple Model*) – keli sąveikaujantys modeliai
- IoU (angl. *Intersection over Union*) – aibių sankirtos ir sąjungos santykis
- KF – Kalmano filtras
- LMS (angl. *Likely Model Set*) – tikėtinas modelių rinkinys
- mAP (angl. *Mean Average Precision*) – vidutinio tikslumo vidurkis
- MGS (angl. *Model Group Switching*) – modelių grupės perjungimas
- MHT (angl. *Multiple Hypothesis Tracking*) – kelių hipotezių sekimas
- MoG (angl. *Mixture of Gaussians*) – Gauso skirstinių mišinys
- MOSSE (angl. *Minimum Output Sum of Squared Error*) – minimali kvadratinės paklaidos išvesties suma

MOT (angl. *Multi-Object Tracking*) – kelių objektų sekimas

MRS (angl. *Multistatic Radar System*) – multistatinių radarų sistema

NCC (angl. *Normalized Cross Correlation*) – normalizuota abipusė koreliacija

PF (angl. *Particle Filter*) – dalelių filtras

RCNN (angl. *Region-based Convolutional Neural Network*) – regioninis konvoliucinis neuroninis tinklas

RGB (angl. *Red, Green, Blue*) – spalvų modelis

ROI (angl. *Region of Interest*) – dominantis regionas

RPN (angl. *Region Proposal Network*) – regioninis pasiūlymų tinklas

SDAE (angl. *Stacked Denoising Autoencoder*) – triukšmą slopinantis autoenkoderis

SGD (angl. *Stochastic Gradient Descent*) – stochastinis gradiento nusileidimas

SIS (angl. *Sequential Important Sampling*) – nuoseklioji svarbioji atranka

SNR (angl. *Signal-to-Noise Ratio*) – signalo ir triukšmo santykis

SORT (angl. *Simple Online Realtime Tracking*) – paprastas internetinis sekimo realiuoju laiku algoritmas

SSD (angl. *Single Shot Multibox Detector*) – vieno kadro kelių langų detektorius

SVM (angl. *Support-Vector Machine*) – atraminių vektorių klasifikatorius

TN (angl. *True Negative*) – tikrai neigiamas

TP (angl. *True Positive*) – tikrai teigiamas

UKF (angl. *Unscented Kalman Filter*) – sigma-taškų Kalmano filtras

VSMM (angl. *Variable Structure Multi-Model*) – kintamos struktūros kelių modelių algoritmas

YOLO (angl. *You Only Look Once*) – objektų aptikimo algoritmas

ILIUSTRACIJŲ SĄRAŠAS

1.1 pav. Paprasčiausios radiolokacijos sistemos veikimo principas (sudaryta autorės, remiantis Бакулев (2004))	18
1.2 pav. Vieno taikinio sekimo radaru schema (sudaryta autorės, remiantis Hao et al. (2017))	20
1.3 pav. Pirmosios kartos kelių modelių algoritmo schema (sudaryta autorės, remiantis Hao et al. (2017))	23
1.4 pav. IMM algoritmo schema (sudaryta autorės, remiantis Hao et al. (2017))	24
1.5 pav. Sistemos veikimas (sudaryta autorės, remiantis Qu et al. (2017))	30
1.6 pav. Koreliacijos filtro sekimo algoritmo schema (sudaryta autorės, remiantis Xinlong ir Hamdulla (2020))	33
1.7 pav. Siūlomo metodo struktūra (sudaryta autorės, remiantis Liu et al. (2017))	34
1.8 pav. Konvoliucinio neuroninio tinklo „LeNet-5“ sluoksnių struktūros schema (sudaryta autorės, remiantis Zhang et al. (2018))	37
1.9 pav. Taikinių sekimo algoritmas, pagrįstas konvoliuciniu neuroniniu tinklu (sudaryta autorės, remiantis Zhang et al. (2018))	39
1.10 pav. Regiono pasirinkimo algoritmo blokinė schema (sudaryta autorės, remiantis Zhang et al. (2018))	40
1.11 pav. Patobulinto taikinio aptikimo algoritmo, pagrįsto konvoliuciniu neuroniniu tinklu ir Kalmano filtru, blokinė schema (sudaryta autorės, remiantis Zhang et al. (2018))	41
1.12 pav. Siamo tinklas (sudaryta autorės, remiantis Chen et al. (2020))	42
1.13 pav. Bendra algoritmo, pagrįsto tankiu paremta klasterizacija, struktūra (sudaryta autorės, remiantis Chen et al. (2020))	43
1.14 pav. Greitesniojo RCNN algoritmo blokinė schema (sudaryta autorės, remiantis Zhang ir Fang (2020))	45
1.15 pav. RPN tinklas ir jo veikimo procesas (sudaryta autorės, remiantis Zhang ir Fang (2020))	45
1.16 pav. Vieno etapo algoritmų architektūra (sudaryta autorės, remiantis Bochkovskiy et al. (2020))	49
1.17 pav. Vizualinio objektų aptikimo ir sekimo algoritmo schema (sudaryta autorės, remiantis Bathija ir Sharma (2019))	49
1.18 pav. Kokybinė siūlomos sistemos analizė (Bathija & Sharma, 2019)	50
1.19 pav. Eksperimento schema (sudaryta autorės, remiantis Singha ir Aydin (2021))	52
1.20 pav. Dronų ir paukščių aptikimo rezultatas su ribojančiais langais ir klasių tikimybėmis (Singha & Aydin, 2021)	52
1.21 pav. Duomenų rinkinio vaizdų pavyzdžiai (Tiwari, 2022)	53

1.22 pav. „Tiny-YOLOv4“ ir „apkarpyto“ „YOLOv4“ modelių našumo palyginimas (Liu et al., 2021)	55
1.23 pav. „Tiny-YOLOv4“ ir „apkarpyto“ „YOLOv4“ modelio našumo palyginimas po duomenų papildymo (Liu et al., 2021)	56
1.24 pav. Išmokyty modelių rezultatai, naudojant testavimui skirtus vaizdus (Sharma et al., 2022)	57
2.1 pav. Pagrindiniai objekto aptikimo ir sekimo žingsniai (sudaryta autorės, remiantis Balaji ir Karthikeyan (2017))	61
2.2 pav. Bendrojo fono atimties algoritmo schema (sudaryta autorės, remiantis Cheung ir Kamath (2004)).....	63
2.3 pav. Objekto sekimo būdai (sudaryta autorės, remiantis Balaji ir Karthikeyan (2017)).....	70
2.4 pav. Canny kraštų aptikimo algoritmo schema (sudaryta autorės, remiantis Rashmi et al. (2013))	76
2.5 pav. Kalmano filtro veikimo schema (sudaryta autorės, remiantis Youngjoo ir Hyochoong (2018))	79
2.6 pav. SORT algoritmo schema (sudaryta autorės, remiantis Hong et al. (2021))	82
2.7 pav. Metodo, susiejančio kiekvieną aptikimo langą, veikimo pavyzdys (sudaryta autorės, remiantis Zhang et al. (2022))	83
2.8 pav. BYTE duomenų susiejimo metodo algoritmas (sudaryta autorės, remiantis Zhang et al. (2022))	84
3.1 pav. Supaprastintas bepiločių orlaivių sekimo sistemos kūrimo algoritmas (sudaryta autorės).	88
3.2 pav. „Drone_dataset“ duomenų rinkinio vaizdų pavyzdžiai (sudaryta autorės)	89
3.3 pav. Failo su informacija apie žymėtą vaizdą pavyzdys (sudaryta autorės)	89
3.4 pav. „Drone_dataset“ duomenų rinkinio pakas (sudaryta autorės).....	90
3.5 pav. „Drones“ duomenų rinkinio vaizdų pavyzdžiai (sudaryta autorės).....	91
3.6 pav. Duomenų rinkinio paruošimo schema (sudaryta autorės).....	91
3.7 pav. „Drones_test“ duomenų rinkinio vaizdų pavyzdžiai (sudaryta autorės)	91
3.8 pav. YOLO tinklo mokymo schema (sudaryta autorės)	94
3.9 pav. Tinklo konfigūracijos failas (sudaryta autorės).....	94
3.10 pav. Pirmojo bandymo tinklo mokymo pabaiga (sudaryta autorės)	95
3.11 pav. Pirmojo bandymo bepiločio orlaivio aptikimo rezultatai (sudaryta autorės)	95
3.12 pav. Pirmojo bandymo tinklo mokymo rezultatai (sudaryta autorės)	96
3.13 pav. Pirmojo bandymo bepiločio orlaivio aptikimo rezultatai, naudojant testavimo duomenų rinkinį (sudaryta autorės).....	97
3.14 pav. Antrojo bandymo tinklo mokymo tarpinis rezultatas po 50 epochų (sudaryta autorės) ...	97
3.15 pav. Antrojo bandymo tinklo mokymo tarpinis rezultatas po 100 epochų (sudaryta autorės) .	98

3.16 pav. Antrojo bandymo tinklo mokymo pabaiga (sudaryta autorės).....	98
3.17 pav. Antrojo bandymo bepiločio orlaivio aptikimo rezultatai (sudaryta autorės)	98
3.18 pav. Antrojo bandymo tinklo mokymo rezultatai (sudaryta autorės)	99
3.19 pav. Antrojo bandymo bepiločio orlaivio aptikimo rezultatai, naudojant testavimo duomenų rinkinį (sudaryta autorės).....	99
3.20 pav. Antrojo bandymo metu klaidingas bepiločio orlaivio aptikimo rezultatas, naudojant testavimo duomenų rinkinį (sudaryta autorės)	100
3.21 pav. Trečiojo bandymo tinklo mokymo tarpinis rezultatas po 100 epochų (sudaryta autorės)	100
3.22 pav. Trečiojo bandymo tinklo mokymo tarpinis rezultatas po 65 epochų (sudaryta autorės)	101
3.23 pav. Trečiojo bandymo tinklo mokymo tarpinis rezultatas po 80 epochų (sudaryta autorės)	101
3.24 pav. Trečiojo bandymo tinklo mokymo pabaiga (sudaryta autorės).....	101
3.25 pav. Trečiojo bandymo bepiločio orlaivio aptikimo rezultatai (sudaryta autorės)	102
3.26 pav. Trečiojo bandymo tinklo mokymo rezultatai (sudaryta autorės)	102
3.27 pav. Trečiojo bandymo bepiločio orlaivio aptikimo rezultatai, naudojant testavimo duomenų rinkinį (sudaryta autorės).....	103
3.28 pav. Bepiločių orlaivių sekimo schema (sudaryta autorės).....	104
3.29 pav. Vaizdo įrašo, naudojamo eksperimentui, kadras (sudaryta autorės)	105
3.30 pav. Pirmojo išmokyto tinklo sekimo rezultatai (sudaryta autorės).....	106
3.31 pav. Antrojo išmokyto tinklo sekimo rezultatai (sudaryta autorės)	107
3.32 pav. Trečiojo išmokyto tinklo sekimo rezultatai (sudaryta autorės).....	108
3.33 pav. Išmokyto tinklo bepiločio orlaivio aptikimo tikslumo pasiskirstymas sekimo metu (sudaryta autorės)	108
3.34 pav. Kelių objektų aptikimo ir anotavimo rezultatas (sudaryta autorės)	109

LENTELIŲ SĄRAŠAS

1.1 lentelė. Keturių algoritmų sekimo rezultatų palyginimas (sudaryta autorės, remiantis Zhang ir Fang (2020))	47
1.2 lentelė. Kiekybinė siūlomos sistemos analizė (sudaryta autorės, remiantis Bathija ir Sharma (2019)).....	50
1.3 lentelė. Modelių veikimo rezultatų palyginimas (sudaryta autorės, remiantis Liu et al. (2021))	54
1.4 lentelė. Modelių išvedimo laiko ir vidutinio pasiklivimo palyginimas, naudojant testavimui skirtus vaizdus (sudaryta autorės, remiantis Sharma et al. (2022)).....	57
1.5 lentelė. Palyginimas dviejų siūlomų modelių (sudaryta autorės, remiantis Aydin ir Singha (2023)).....	58
2.1 lentelė. Objektų klasifikavimo metodų palyginimas (sudaryta autorės, remiantis Balaji ir Karthikeyan (2017))	67
2.2 lentelė. Skirtingų duomenų susiejimo metodų palyginimas, naudojant MOT17 duomenų rinkinį (sudaryta autorės, remiantis Zhang et al. (2022)).....	86
3.1 lentelė. Tinklo mokymo rezultatai (sudaryta autorės).....	103
3.2 lentelė. YOLO tinklų palyginimas, aptinkant bepiločius orlaivius (sudaryta autorės, remiantis Aydin ir Singha (2023)).....	104
3.3 lentelė. Sekimo rezultatai (sudaryta autorės)	105

IVADAS

Vienas iš saugos tikslų aviacijoje ar laivyboje yra susidūrimo vengimas. Šiose srityse, kad būtų išvengta susidūrimų, naudojami radarai, sukurti XX a. pirmojoje pusėje kaip labai paprastas prietaisas, skirtas orlaiviams aptikti. Toliau šis prietaisas buvo išplėtotas į sudėtingas sistemas, naudojamas civilinėse ir karinėse srityse prevencijos bei perėmimo strategijų tikslais (Chernyak, 2002). Pagrindinis šių sistemų uždavinys yra ne tik aptikti taikinius ar kliūtis, bet ir įvertinti tokius parametrus, kaip greitis, nuotolis, taikinių azimutas. Radaro tikslumas ir taikinio aptikimo tikimybė yra riboti dėl įvairių trukdžių, kurie yra aplinkoje, ir iškraipymų, atsirandančių signalo sklaidimo metu.

Vystantis mokslui ir technologijoms, vis labiau skatinama sparti dirbtinio intelekto technologijų plėtra. Svarbi dirbtinio intelekto šaka yra kompiuterinė rega, naudojama taikiniams sekti, kuris taikomas vaizdo sekoms sekti, autonominiam valdymui, žvalgybai ir kitur. Taikinio sekimas – tai procesas, kurio metu be išankstinių žinių numatoma taikinio padėtis ir matmenys (plotis bei aukštis) kituose vaizdo sekos kadruose pagal taikinio vaizdo koordinacių informaciją, gautą iš pirmojo vaizdo kadro. Taikinio sekimo procesas gali būti skirstomas į tris etapus: požymių išskyrimas, modelio sudarymas ir adaptyvusis atnaujinimo modelis (Xinlong & Hamdulla, 2020). Pagrindinė problema taikinio sekimo metu – pasiekti reikiamą taikinio padėties nustatymo tikslumą vaizduose, kuriuose taikiny s yra uždengtas, yra apšvietimo pokyčių, atsiranda neryškumų dėl judėjimo, yra sudėtingas fonas ir taikinio deformacija.

Sukurta įvairių efektyvių taikinių sekimo algoritmų, tačiau esami sekimo algoritmai vis dar turi trūkumų, kurie reikalauja tolesnių algoritmų tyrimų, siekiant pagerinti tikslumą, efektyvumą realiuoju laiku ir patvarumą. Naujų taikinių sekimo algoritmų kūrimas yra svarbus dėl kelių priežasčių:

1. Tikslumas: esami algoritmai ne visada gali būti tikslūs aptinkant ir sekant objektus, ypač sudėtingoje ir dinamiškoje aplinkoje. Taikant naujus metodus, tokius kaip gilusis mokymasis, kuris gali tiksliau aptikti ir sekti objektus, gali būti pagerintas objektų sekimo sistemų veikimas.

2. Greitis: norint sekti objektus realiuoju laiku, reikalingi algoritmai, galintys greitai apdoroti didelius duomenų kiekius. Nauji algoritmai gali būti optimizuoti, tai leidžia greičiau ir efektyviau sekti objektus.

3. Patvarumas: kuriant naujus, patikimesnius algoritmus, galima užtikrinti, kad objektą būtų galima sekti įvairiose situacijose.

4. Lankstumas: nauji algoritmai gali būti sukurti taip, kad būtų galima sekti objektus esant įvairiems scenarijams, pavyzdžiui, sudėtingoje aplinkoje arba kai juda keli objektai. Šis lankstumas leidžia plačiau naudoti objektų sekimą įvairiose programose, tokiose kaip saugumo stebėjimas, autonominis vairavimas ir robotika.

Apibendrinant galima teigti, kad norint pagerinti objektų sekimo sistemų tikslumą, greitį, patvarumą ir lankstumą, yra poreikis kurti naujus objektų sekimo algoritmus.

Siauresnė sritis yra bepiločių orlaivių sekimo algoritmai. Susidomėjimas bepiločiais orlaiviais sparčiai didėja, ir nauji sekimo algoritmai gali padėti užtikrinti dronų ir kitų orlaivių saugumą oro erdvėje. Bepiločiai orlaiviai gali kelti pavojų visuomenei ir nacionaliniam saugumui, kai jie skraido per arti žmonių, pastatų ar kitų orlaivių. Nauji bepiločių orlaivių sekimo algoritmai gali padėti atpažinti dronus, kurie skraido draudžiamosiose zonose arba skraido nesaugiai. Nauji sekimo algoritmai gali padėti atpažinti ir sekti bepiločius orlaivius, kurie užsiima įtartina veikla. Taigi, galima teigti, kad norint užtikrinti žmonių, dronų, taip pat kitų orlaivių, esančių oro erdvėje, saugą, yra poreikis kurti naujus bepiločių orlaivių sekimo algoritmus.

Darbe iškelta hipotezė, kad naujausių metodų taikymas ir derinimas gali patobulinti bepiločių orlaivių sekimo sistemos veikimą. Dauguma tyrimų yra atlikta tik su dronų aptikimu (be jų sekimo), netestuojant veikimo vaizdo įrašuose, todėl yra sudėtinga įvertinti jų veikimą realiomis sąlygomis, nors mokymo metu gaunamas gana aukštas tikslumas.

Problema: bepiločių orlaivių sekimas kelia iššūkių (pavyzdžiui, fono pasikeitimas, mažas dronų dydis vaizdo kadruose), kuriems reikia naujų sprendimų.

Darbo tikslas – naujo vizualinio taikinių sekimo algoritmo kūrimas.

Darbo uždaviniai:

1. Išanalizuoti literatūros šaltinius ir ištirti esamus taikinių sekimo algoritmus.
2. Išsirinkti metodus, tinkamus naujo taikinių sekimo algoritmo, paremto konvoliucinių neuroninių tinklų algoritmais, sintezei.
3. Sukurti vizualinio taikinių sekimo metodą ir tuo metodu paremtą algoritmą.
4. Aprobuoti sukurtą algoritmą.

Taikoma metodika: duomenų rinkinio paruošimas ir vaizdų žymėjimas, naudojant „Roboflow“ platformą; pasirinktos vertinimo metrikos: nuostolių funkcija, preciziškumas, vidutinio tikslumo vidurkis (mAP), išvedimo greitis; bepiločiams orlaiviams aptikti naudojamas mokytas „YOLOv8“ tinklas; bepiločiams orlaiviams sekti naudojamas „ByteTrack“ paketas; eksperimentui atlikti buvo naudojama „Python“ programavimo kalba.

Darbe siūlomo bepiločių orlaivių sekimo metodo teorinė reikšmė yra pagrįsta kompiuterinės regos ir giliojo mokymosi algoritmų principais. Metodui naudojami konvoliuciniai neuroniniai tinklai gali aptikti sudėtingus vizualinius modelius ir puikiai tinka objektų aptikimo ir atpažinimo užduotims. Tiksliai stebint dronus realiuoju laiku, siūlomas metodas gali padėti išvengti susidūrimų su kitais orlaiviais ar objektais. Be to, šis metodas gali padėti aptikti neteisėtą dronų veiklą. Taigi, siūlomas bepiločių orlaivių sekimo metodas turi praktinę reikšmę didinant dronų naudojimo saugą ir saugumą.

Darbą sudaro įvadas, trys skyriai, išvados ir rekomendacijos, literatūra. Taip pat yra pateikti keturi priedai. Darbo apimtis – 98 puslapiai teksto be priedų, tekste panaudota 40 formulių, 66 paveikslai ir 10 lentelių. Rašant baigiamąjį magistro darbą buvo panaudoti 83 literatūros šaltiniai.

1. LITERATŪROS APŽVALGA IR ANALIZĖ

1.1. Radiolokacijos sistemos

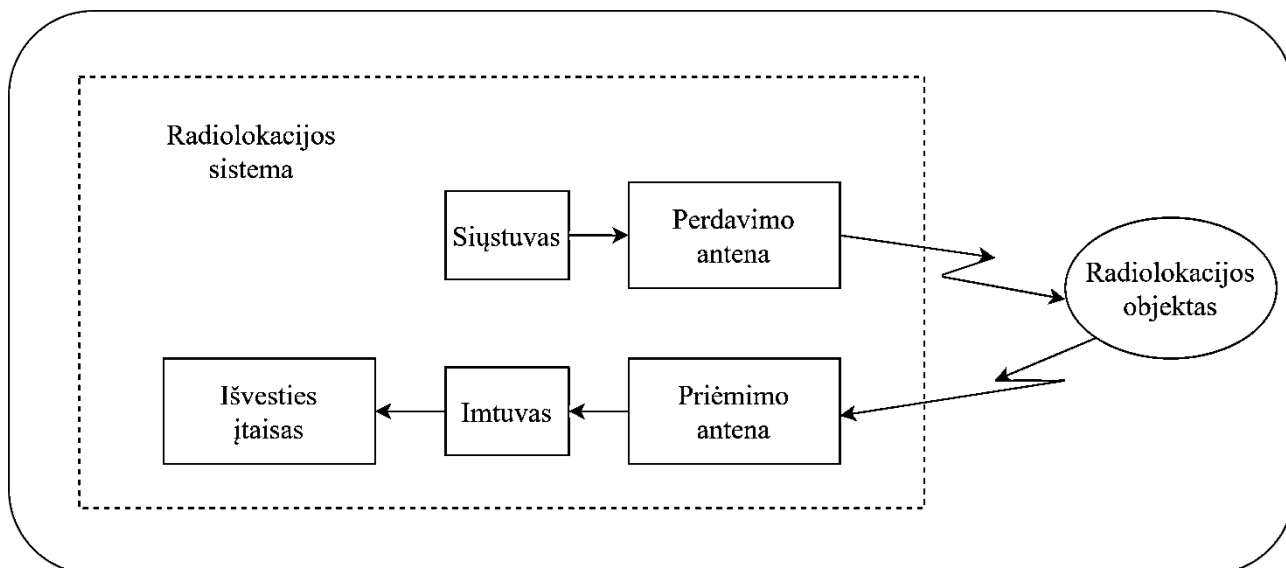
Radiolokacija yra radiotechniniai metodai ir priemonės objektams aptikti ir atpažinti, naudojant radijo bangas, taip pat objektų vietai ir judėjimo parametrus erdvėje nustatyti (Svilainis, 2019). Radiolokacijos objektai – tai radiolokacijos taikiniai. Radiolokacijoje paprastai naudojami signalai, atspindėti nuo taikinio, arba paties taikinio ir jame sumontuotų radijo prietaisų skleidžiami signalai (Бакулев, 2004).

Radiolokacijos sistemos yra skirtos informacijai apie objektus išskirti iš priimto radijo signalo. Taigi, radiolokacijos sistema ieško ir aptinka radijo signalą, o po to išmatuoja jo parametrus, kuriuose yra naudingos informacijos. Radiolokacijos objekto vietai radiolokacijos sistemoje nustatyti naudojamos taikinio koordinatės. Kai kuriais atvejais taip pat būtina žinoti taikinio greičio vektoriaus komponentus. Geometriniai arba mechaniniai dydžiai, apibūdinantys taikinio padėtį ir judėjimą, vadinami lokacijos elementais (Бакулев, 2004).

Objektų skleidžiamos radijo bangos skiriasi savo elektrinėmis charakteristikomis (elektrine skvarba, dielektrine skvarba ir elektriniu laidumu) nuo atitinkamų aplinkos charakteristikų, kai jos yra apšvitintos. Radijo bangų sklaidos ar atspindžio intensyvumas (antrinio lauko intensyvumas) priklauso nuo objekto ir terpės elektrinių charakteristikų skirtumo laipsnio, objekto formos, jo matmenų ir bangos ilgio santykio bei radijo bangų poliarizacijos (Бакулев, 2004). Gautą antrinį elektromagnetinį lauką sudaro atspindėtas laukas, sklindantis link švitinančio pirminio lauko, ir šešėlinis laukas, sklindantis už objekto (ta pačia kryptimi kaip ir pirminis laukas).

Naudojant priėmimo anteną ir imtuvą, galima priimti dalį išskleisto signalo, konvertuoti jį ir sustiprinti (kad vėliau jis būtų aptiktas). Paprasčiausią radarą sudaro (1.1 pav.):

- siųstuvas (radijo signalams formuoti ir generuoti),
- perdavimo antena (šiems radijo signalams skleisti),
- priėmimo antena (atspindėtiems signalams priimti),
- radijo imtuvas (signalams sustiprinti ir konvertuoti),
- išvesties įtaisas (atspindėtiems signalams aptikti).



1.1 pav. Paprasčiausios radiolokacijos sistemos veikimo principas (sudaryta autorės, remiantis Бакулев (2004))

Paprastai gaunamo signalo amplitudė yra maža, o pats signalas yra atsitiktinio pobūdžio. Galimos mažos signalo amplitudės priežastys: didelis atstumas iki taikinio ir signalo energijos absorbcija jo sklaidimo metu (Бакулев, 2004). Taip pat taikinių dydis turi įtakos atspindėto signalo intensyvumui. Atsitiktinis signalo pobūdis yra atspindėto signalo svyravimų pasekmė, atsirandanti dėl atsitiktinio kompleksinės formos taikinio elementų judėjimo (kai atspindimos radijo bangos), radijo bangų sklaidimo skirtingais keliais, chaotiškų signalo amplitudės pokyčių sklaidimo metu ir kitų veiksnių (Бакулев, 2004). Dėl to gaunamas signalas savo forma, intensyvumu ir pobūdžiu yra panašus į triukšmą ir trukdžius. Taigi, pirmoji ir pagrindinė radiolokacijos užduotis yra aptikti naudingąjį radijo signalą, t. y. priimti sprendimą dėl naudingojo signalo buvimo naudingojo signalo mišinyje su trikdžiais, patenkančiais į priimančio kelio įvestį. Šią statistinę problemą išsprendžia detektorius, kuriame turi būti naudojamas optimalus aptikimo algoritmas (Бакулев, 2004). Aptikimo proceso kokybei būdinga teisingo aptikimo tikimybė, kai aptinkamas įėjime esantis signalas, ir klaidingo aptikimo tikimybė, kai trukdžiai laikomi naudinguoju signalu, o paties signalo nėra. Kuo didesnė teisingo aptikimo tikimybė ir mažesnė klaidingo aptikimo tikimybė, tuo geresnis yra detektorius (Бакулев, 2004). Dauguma gaunamo signalo parametrų yra nežinomi, todėl aptikimo metu tenka ieškoti reikiamo radijo signalo parametro, kuris jį skiria nuo lydinčio triukšmo ir trukdžių.

Informacija apie geometrinius elementus, apibūdinančius taikinio padėtį ir judėjimo elementus, yra saugoma radijo signalo parametruose (informacinis signalo parametras). Paprastai elektromagnetinis laukas priėmimo taške yra laiko ir erdvės koordinačių funkcija. Todėl informatyvūs radiolokacijos signalo parametrai gali būti atvykimo laikas, dažnis, pradinė fazė, amplitudė, signalo atvykimo kryptis (du kampai erdvėje) ir lauko poliarizacijos parametrai (Бакулев, 2004).

Signalą ieškoma pagal jo informacinį parametą: arba peržiūrimos visos šio parametro reikšmės po vieną, arba jos stebimos vienu metu (Бакулев, 2004). Pirmuoju metodu pagrįsti paieškos įrenginiai (nuoseklūs paieškos įrenginiai) yra paprasčiau įgyvendinami, tačiau reikia daugiau laiko, kad būtų peržiūrėtos visos galimos parametrų vertės. Lygiagrečiosios paieškos įrenginiai sugeba signalą aptikti per žymiai trumpesnę laiką, tačiau juos yra sudėtingiau įgyvendinti.

Radaro sekimo charakteristikoms patobulinti naudojamos multistatinių radarų sistemos (angl. *Multistatic Radar System* – MRS). Siekiant maksimaliai padidinti tikslumą ir aptikimo tikimybę, MRS turi dvi ar daugiau perdavimo ar priėmimo antenų, kurios skiriamos dideliais atstumais, lyginant su pačių antenų dydžiu (IEEE, 2017).

Kuo daugiau antenų yra radarų sistemoje, tuo sudėtingesnis taikinių sekimas. Pirma, visos sistemos informacijai derinti reikalingos matomumo linijos tarp radarų stočių ir taikinių, taip pat sinchronizavimui, fazių nustatymui ir stočių išdėstymui bei atskaitos dažnių ir signalų perdavimo reikalavimams atitikti (Chernyak, 2002). Antra, sekant kelis taikinius, kyla bendrųjų iššūkių, pavyzdžiui, nežinomas ir atsitiktinai kintantis taikinių skaičius, aptikimo neapibrėžtumas, trukdžiai, duomenų susiejimo neapibrėžtumas, Doplerio matavimo netiesiškumas ir mažas stebimumas (Vo et al., 2015). Naudojant Doplerio efektą, aptikimas iš stacionarių radarų gali būti suderintas tik su judančiais taikiniais, nes atstatyti signalai iš nejudančių taikinių (arba labai lėtai judančių) yra praktiškai identiški perduodamų signalų dažniams.

Naudojant radarą bepiločiams orlaiviams sekti svarbu atsižvelgti į keletą trūkumų. Radarų sistemos paprastai suteikia mažesnę skiriamąją gebą, lyginant su kitomis sekimo technologijomis (pavyzdžiui, vaizdiniu sekimu arba LiDAR pagrįstos sistemos). Dėl to gali sumažėti tikslumas sekant mažus arba arti esančius objektus, o tai yra būdinga dronų sekimo scenarijams. Taip pat radaro signalus gali paveikti aplinka (pastatai, medžiai ar kiti objektai). Tai gali lemti klaidingus drono aptikimus arba jų neaptikimą. Be to, radarų sistemos gali būti gana brangios ir sudėtingos, lyginant su kitomis sekimo technologijomis. Joms dažnai reikia specializuotos aparatinės įrangos ir signalų apdorojimo algoritmų. Radaro integravimas su kitais jutikliais ir pažangiais sekimo algoritmais gali padėti pašalinti kai kuriuos iš šių sistemų apribojimų ir pagerinti bendrą stebėjimo našumą, tačiau radarų naudojimas taip pat gali padidinti sistemos sudėtingumą ir bendrą jos kainą.

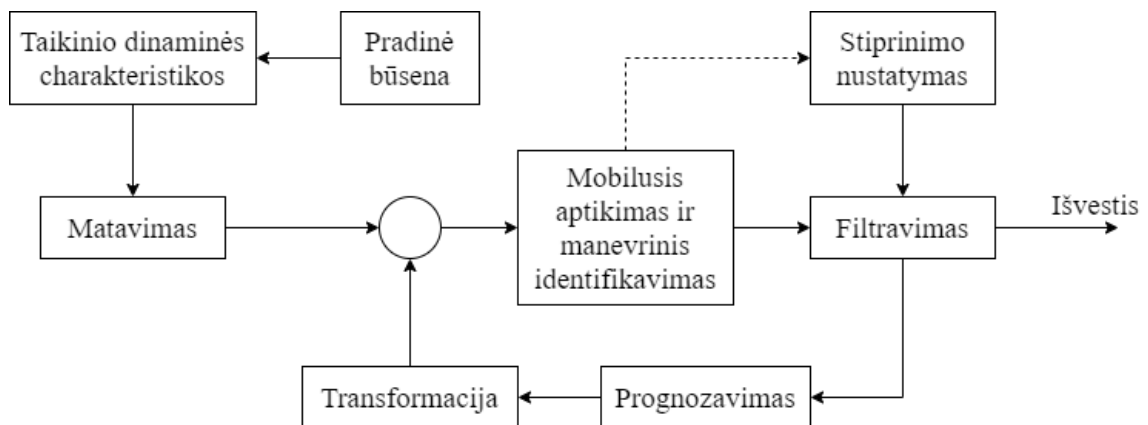
1.2. Vieno taikinio sekimo radaru principas

Radariai naudojami išankstiniam perspėjimui, taikiniais aptikti ir sekti, navigacijai valdyti ir kitoms daugiafunkcėms užduotims. Radarų informacijos apdorojimo pajėgumai tiesiogiai lemia kovinį ginklų ir įrangos efektyvumą (Zhang et al., 2013a). Pagrindinė taikinio sekimo algoritmo esmė – užtikrinti tikslų taikinio sekimą, t. y. užtikrinti algoritmo tikslumą, taip pat reikia atsižvelgti į algoritmo realizavimo laiką ir garantuoti veikimą realiuoju laiku.

Radaro vieno taikinio sekimo algoritmas susideda iš dviejų dalių (Hao et al., 2017):

- taikinio judėjimo būsenos įvertinimas įvairiais metodais realiuoju laiku,
- taikinio judėjimo būsenos pokyčių dėsnio apibūdinimas, kuriant įvairius modelius.

Taikiniui sekti naudojamas radaras seka ir analizuoja taikinį, gauna taikinio greičio, vietos ir kitą informaciją bei nustato taikinio judėjimo būsenos dinaminį modelį. Tada kompiuteris nuspėja ir įvertina šią taikinio informaciją kitą akimirką, taikant filtravimo metodų seriją, kad būtų tiksliai nustatyta taikinio trajektorija (Hao et al., 2017). Taikinio sekimo algoritmas yra svarbi radaro duomenų apdorojimo dalis (sekimo principas pateiktas 1.2 paveiksle).



1.2 pav. Vieno taikinio sekimo radaru schema (sudaryta autorės, remiantis Hao et al. (2017))

Kai taikiny maneuvruoja, matavimų vertė padidės – manevravimo aptikimas pagrįstas liekamuoju pokyčiu. Tada kompiuteris nustato filtro stiprinimą (angl. *filter gain*), kovariacinę matricą (angl. *covariance matrix*) ir kitus parametrus, gaunama filtro išvestis ir užbaigiama maneuvruojančio taikinio sekimo funkcija (Hao et al., 2017).

1.2.1. Būsenos įvertinimo algoritmas

Gebėjimas tiksliai įvertinti taikinio būseną yra būtina sąlyga tolesniam duomenų apdorojimui. Taikinio būsenos įvertinimą galima suskirstyti į tiesinio ir netiesinio būsenos įvertinimo algoritmus (Hao et al., 2017).

Ankstyviausias būsenos įvertinimo algoritmas siejamas su XIX amžiaus pradžia, kai matematikas Karlas Frydrichas Gausas pasiūlė mažiausio kvadrato metodą. Vėliau, remiantis radiolokacijos sistemų poreikiais, buvo suprojektuotas Vynerio filtras – rastas optimalaus perdavimo funkcijos analitinis sprendimas, naudojant Vynerio-Hofo lygtį. Tačiau Vynerio filtras nenaudojamas nepastovaus atsitiktinio proceso atvejams spręsti. Taikant šį filtrą reikia saugoti didelį duomenų kiekį, o tai nėra naudinga inžinerijai (Hao et al., 2017). Antroje XX a. pusėje, nuolat tobulinant ir plėtojant optimalaus įvertinimo teoriją, atsižvelgiant į Vynerio filtro trūkumus, buvo išvestas Kalmano filtras (KF). Algoritmas supažindina su būsenos erdvės samprata, naudoja būsenos lygtį santykiui tarp įvesties ir išvesties išreikšti ir įvertina reikiamą informaciją iš matavimo metu gautų duomenų

(Hao et al., 2017). Naudojant Kalmano filtrą, nereikia saugoti visų praeities duomenų ir griežtai nereikalaujamas sistemos sklandumas. KF yra paprastai suprojektuotas ir plačiai naudojamas. Tačiau realizacijai yra naudojamas kompiuteris, dėl to atsiranda apvalinimo paklaidų ir atmetimo klaidų, kurios kaupiasi, ir filtro rezultatai gali būti nestabilūs (Hao et al., 2017).

Nors Kalmano filtras turi daug privalumų ir yra taikomas daugelyje sričių, KF gali būti naudojamas tik tiesinėms sistemoms. O realiomis sąlygomis taikinio manevravimas paprastai būna netiesinis. Atlikta netiesinio filtravimo pagal Kalmano filtrą tyrimų, pavyzdžiui, taikomi tokie metodai, siekiant patobulinti KF algoritmą (Hao et al., 2017):

- Funkcijų aproksimacija. Pirmosios eilės Taylora išplėtimas 1970 m. buvo panaudotas netiesinei problemai, paremtai KF, linearizuoti ir buvo pasiūlytas išplėstinis Kalmano filtras (angl. *Extended Kalman Filter* – EKF). EKF yra paprastos struktūros, lengvai realizuojamas ir plataus taikymo. Tačiau dėl linearizavimo paklaidos, atsirandančios EKF proceso metu, metodas gali diverguoti, kai netiesiškumo laipsnis yra didelis (Hao et al., 2017). Siekiant sumažinti linearizavimo proceso paklaidas ir pagerinti EKF tikslumą, mokslininkai pateikė naujų patobulintų algoritmų, pavyzdžiui, antrosios eilės EKF (angl. *the second-order EKF*) algoritmas, iteracinis EKF (angl. *Iterative EKF* – IEKF) algoritmas. IEKF naudojama Gauso-Niutono iteracija, kad būtų pagerintas filtro tikslumas, tačiau metodo konvergavimas nebuvo pagerintas (Hao et al., 2017).

- Determinuotoji atranka. Siekiant išspręsti EKF netiesiškumo paklaidų problemą, buvo pasiūlytas sigma-taškų Kalmano filtro (angl. *Unscented Kalman Filter* – UKF) algoritmas (Julier & Uhlmann, 1997). Algoritmas naudoja tikimybės tankio funkciją, kad netiesinei funkcijai apytiksliai būtų pasirinkti $(2n + 1)$ sigmos taškai, vertinimo tikslumas gali pasiekti antrąją eilę. Siekiant išspręsti UKF nelokalinės atrankos ir aukštos eilės kryžminės matricos (angl. *cross matrix*) problemą, kai sistemos matmenys yra dideli, kitas pasiūlytas metodas leido sumažinti pavyzdžių imties dydį (Julier & Uhlmann, 2002). Atsižvelgiant į lėtą UKF konvergavimo greitį buvo siūlomi įvairūs iteraciniai UKF algoritmai, padidinantys konvergavimo greitį, tačiau atitinkamai padidėdavo reikalingas skaičiavimų kiekis (Hao et al., 2017). Vėliau buvo pateiktas hibridinis iteracinis UKF algoritmas (Liu et al., 2010), kuris garantavo konvergavimo greitį ir supaprastino skaičiavimus.

- Tūrio matavimo kriterijus. Pasiūlytas tūrio matavimo Kalmano filtro (angl. *Cubature Kalman Filter* – CKF) algoritmas (Arasaratnam et al., 2010) buvo pagrįstas trečiosios eilės sferiniu-radialiniu tūrio matavimo kriterijumi. CKF pranašumai yra paprastas realizavimas, nedidelis skaičiavimų kiekis ir didelis tikslumas (Hao et al., 2017). Vėliau pasiūlytas kvadratinės šaknies CKF (angl. *square root CKF*) algoritmas pagerino CKF stabilumą ir užtikrino kovariacinės matricos nenegatyvų determinizmą (Hao et al., 2017). Siekiant sumažinti pradinės paklaidų matavimo lygties netiesiškumo įtaką būsenai įvertinti, buvo pasiūlytas iteracinis CKF (angl. *iterative CKF*) algoritmas (Mu & Cai, 2011), derinantis Niutono-Gauso iteraciją su CKF (tai pagerino įvertinimo tikslumą).

Atlikus tyrimus, taip pat buvo įrodyta, kad aukštesnė eilė gali pagerinti algoritmo įvertinimo tikslumą, tačiau padidėja skaičiavimų sudėtingumas (Hao et al., 2017).

- Atsitiktinė atranka. Ankščiau minėti algoritmai gali būti susiję tik su Gauso netiesinės sistemos triukšmo statistinėmis charakteristikomis. Ne Gauso triukšmui būtina įdiegti dalelių filtrą (angl. *Particle Filter* – PF), kuris pagrįstas Monte Karlo metodu ir Bajeso įvertinimu (Hao et al., 2017). 1950-aisiais amerikiečių mokslininkas Hammersley pasiūlė nuosekliosios svarbiosios atrankos (angl. *Sequential Important Sampling* – SIS) algoritmą (Hu et al., 2018). Vėliau, atlikus tyrimus, buvo nustatyta SIS algoritmo pagrindinė problema – dalelių degradacija. Tai lėmė skaičiavimo išteklių švaistymą ir rezultatų nukrypimą (Hao et al., 2017). Problema nebuvo išspręsta, kol Gordon 1993 m. nepristatė pakartotinės atrankos algoritmo, kuris išsprendė dalelių degradacijos problemą ir pagerino skaičiavimus (Hao et al., 2017). 1999 m. Carpenter pirmą kartą pasiūlė dalelių filtro koncepciją, remiantis SIS algoritmo apibendrinimu (Hao et al., 2017). Pasiūlytas tūrio matavimo dalelių filtro (angl. *cubature particle filter*) algoritmas (Mu et al., 2011) pasižymi trumpu veikimo laiku ir maža įvertinimo paklaida.

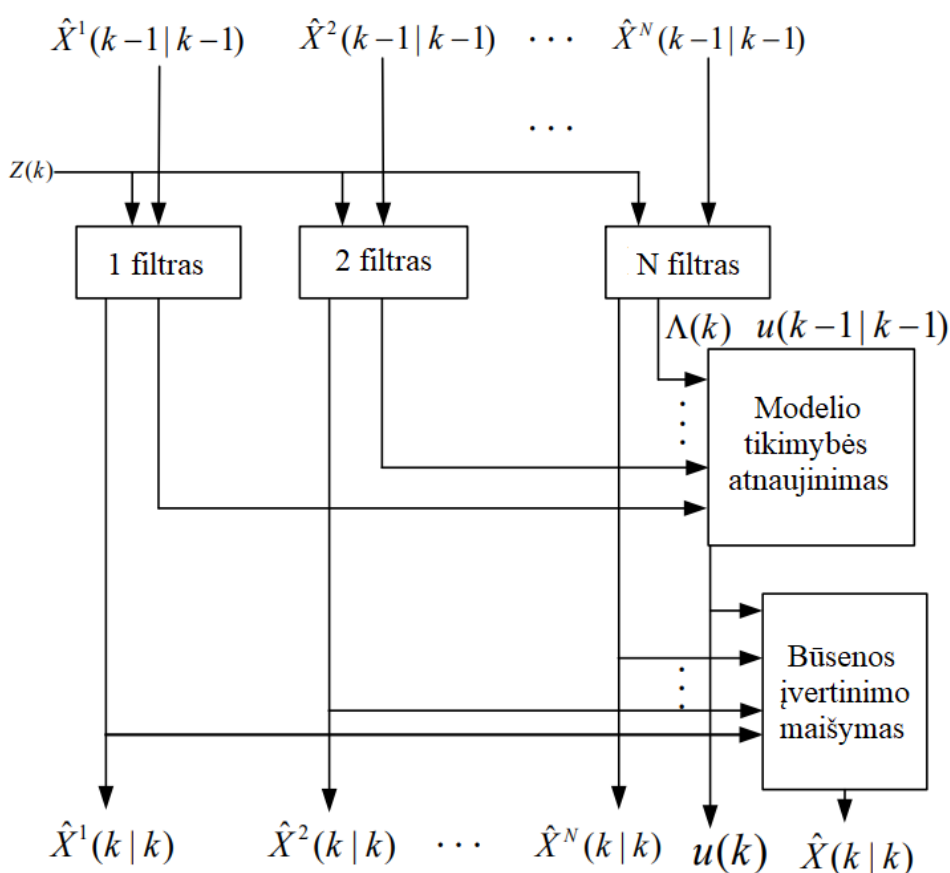
1.2.2. Taikinio judėjimo modelis

Siekiant užtikrinti taikinio sekimo sistemai naudojamo filtro stabilų veikimą ir pastovų konvergavimą, reikia, kad sistemos modelis būtų nešališkas (Hao et al., 2017). Todėl svarbu sukurti tokį matematinį modelį, kad būtų gautas geras taikinio sekimo proceso rezultatas. Manevruojančio taikinio judėjimo modelis pagal modelių skaičių skirstomas į vieno modelio ir kelių modelių algoritmus (Hao et al., 2017).

1973 m. Bridgewater ir Friedland pasiūlė pastovaus greičio (angl. *Constant Velocity* – CV) ir pastovaus pagreičio (angl. *Constant Acceleration* – CA) modelius (Hao et al., 2017). CV modelio atveju laikoma, kad taikinio judėjimas yra tiesiaeigis, greitis pastovus, o pagreitis Gauso baltojo triukšmo formos. CA modelio atveju daroma prielaida, kad taikinio judėjimas yra tiesiaeigis ir pagreitis yra pastovus. 1983 m. Zhou Hong-ren teigė, kad taikinio pagreitis kitą akimirką negali visiškai skirtis nuo ankstesnio momento vertės, vertė yra esamo pagreičio kaimynystės intervale (Hao et al., 2017). Todėl jis pasiūlė „esamą“ statistinį modelį, vadinamą vidutinio adaptyvaus pagreičio modeliu. 1992 m. Waston ir Blair sukūrė koordinuoto posūkio (angl. *coordinated turn*) modelį (Hao et al., 2017). Buvo laikoma, kad taikinyas atlieka vienodą apskritiminį judėjimą (angl. *circular motion*), tada greičio ir pagreičio santykis buvo naudojamas perdavimo matriciai (angl. *transfer matrix*) pasirinkti (matricos parametras – kampinis greitis, apibūdinantis taikinio trajektoriją, siekiant nustatyti taikinio judėjimą). Matsuzaki pateikė CAV modelį, pagrįstą apskritiminio judėjimo charakteristikomis (Hao et al., 2017). Šiame modelyje judėjimo kreivė skaidoma į vienodą tiesiaeigį judėjimą išilgai greičio krypties ir pastovų kampinio greičio judėjimą

apie dvi ašis, statmenas greičio kryptčiai (Matsuzaki et al., 2000). Visi modeliai negalėjo tiksliai apibūdinti sudėtingo taikinio judėjimo (Hao et al., 2017). Po atliktų tyrimų buvo nustatyta, kad būsenos vektoriaus išvestinės eilės ribojimas buvo svarbi to priežastis.

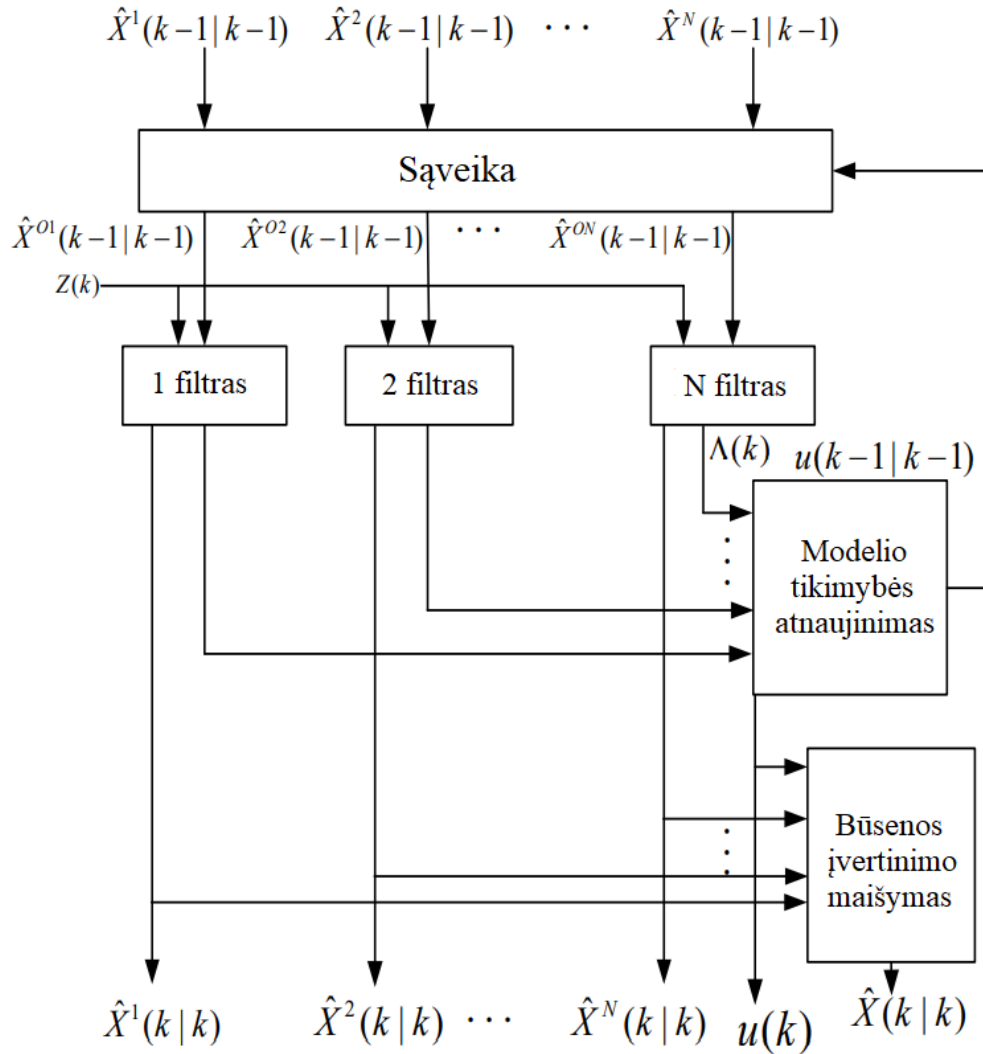
1965 m. Magill pasiūlė pirmosios kartos kelių modelių algoritmą, kuris buvo pagrįstas skirtingais filtrų modeliais, veikiančiais lygiagrečiai, sujungiant jų išvestis (Hao et al., 2017). Algoritmo schema pateikta 1.3 paveiksle. Lyginant su vieno modelio algoritmu, pasirenkant modelių rinkinį, kuris gali apimti visas judėjimo formas, algoritmas gali pasiekti geriausią filtravimo rezultatą. Akivaizdus trūkumas – modelių skaičius yra fiksuotas. Taip pat nebuvo atsižvelgta į perėjimą tarp sistemos režimų ir negalima buvo pasirinkti visiškai tinkančio modelio. Kai modelių yra daugiau, sekimo efektyvumas tampa žemas ir trūksta sąveikos tarp modelių.



1.3 pav. Pirmosios kartos kelių modelių algoritmo schema (sudaryta autorės, remiantis Hao et al. (2017))

Antrosios kartos kelių modelių algoritmo modelių skaičius vis dar yra fiksuotas, tačiau pasirinkti modeliai sąveikauja tarpusavyje. 1970 m. Ackerson pasiūlė apibendrinto pseudo-Bajeso (angl. *Generalized Pseudo-Baysian* – GPB) algoritmą (Hao et al., 2017). Laikoma, kad kiekvienas iš pasirinktų modelių atitinka taikinio judėjimo būseną ir pateikia tą pačią įvestį kiekvienam modeliui. Tačiau algoritme neatsižvelgiama į kiekvieno modelio charakteristikas ir nenaudojama informacija apie modelį, todėl algoritmo taikymas yra pakankamai ribotas (Hao et al., 2017).

1988 m. Blom ir Bar-Shalom pasiūlė naują sąveikaujančių kelių modelių (angl. *Interacting Multiple Model* – IMM) algoritmą taikinio judėjimo būsenai aproksimuoti, tikimybiškai pasveriant kelis modelius, pagrįstus GPB algoritmu (Hao et al., 2017). 1.4 paveiksle pateikta IMM algoritmo schema. IMM algoritmas perėjimą tarp modelių traktuoja kaip Markovo procesą ir transformuoja juos per tikimybinę perdavimo matricą. Algoritmas buvo plačiai naudojamas dėl jo pranašumų: nereikia manevruoti aptikimo, paprasta struktūra, nedaug skaičiavimų (Hao et al., 2017).



1.4 pav. IMM algoritmo schema (sudaryta autorės, remiantis Hao et al. (2017))

Nors IMM padarė didelę pažangą, lyginant su pirmosios kartos kelių modelių algoritmu, vis dar buvo tam tikrų trūkumų. Pavyzdžiui, kai taikinyis naudoja įvairius manevravimo metodus, mažas modelių rinkinys negali tiksliai apibūdinti manevravimo situacijos (Hao et al., 2017). Be to, padidėjus modelių skaičiui, skaičiavimų sudėtingumas padidėja, konkurencija tarp įvairių modelių taip pat didėja, tai neigiamai veikia algoritmo našumą.

1992 m. buvo pasiūlytas kintamos struktūros kelių modelių (angl. *Variable Structure Multi-Model* – VSMM) algoritmas, turintis laike kintantį modelių rinkinį ir adaptyvų perjungimą (Yang et al., 2010), kad būtų pakeista fiksuota apribojimų struktūra (Hao et al., 2017). VSMM

algoritmo veikimas yra geresnis nei IMM algoritmo, ir kuo didesnis taikinio mobilumas, tuo našesnis VSMM algoritmas. Siekiant nustatyti modelių kandidatų rinkinį ir pasirinkti optimalų modelių rinkinį, buvo naudojami keturi algoritmai: modelių grupės perjungimas (angl. *Model Group Switching* – MGS), tikėtinas modelių rinkinys (angl. *Likely Model Set* – LMS), adaptyvusis tinklulis (angl. *adaptive grid*) ir laukiamo modelio algoritmas (angl. *Expectation Model Algorithm* – EMA) (Hao et al., 2017).

Anksčiausiai pasiūlytas buvo MGS – tai dviraidžio perjungimo (angl. *digraph switching*) modelių rinkinio adaptyvusis algoritmas. Jo veikimas yra geresnis nei IMM algoritmo, tačiau algoritmas gali būti aktyvuotas netinkamu modelių rinkiniu (Hao et al., 2017). LMS yra aktyvaus dviraidžio modelių rinkinio adaptyvusis lengvai realizuojamas algoritmas. LMS trūkumas – jei topologijos dizainas nėra tinkamas, keičiant algoritmą, du skirtingi modeliai gali būti nesuvaldyti ir sukelti didesnę paklaidą (Hao et al., 2017). EMA gali sugeneruoti modelį, atitinkantį pavyzdį, realiuoju laiku. EMA įvertinimo tikslumas yra labai didelis, tačiau reikalauja daug skaičiavimų ir yra sudėtingai realizuojamas (Hao et al., 2017).

Kai išorinė aplinka yra sudėtinga ir kintanti, ankstesnio matavimo triukšmo statistinės charakteristikos dažnai negali tiksliai apibūdinti tikrųjų triukšmo charakteristikų. Standartinio sekimo algoritmo rezultatas yra paveiktas ir net gali diverguoti (Hao et al., 2017). Kai kurie mokslininkai filtro algoritme įdiegė adaptyvųjį mechanizmą, kuris padeda išspręsti šią problemą. Taip pat mokslininkai taiko netiesinį filtrą antrosios ir trečiosios kartos kelių modelių algoritmuose, tai turi gerą poveikį ilgalaikio nuolatinio manevruojančio taikinio sekimo metu (Hao et al., 2017). Minėtos įvairios algoritmo formos buvo ne tik pakartotinai panaudotos taikinių sekimo srityje, bet ir išplėstos kitose srityse (gedimų diagnostika, dirbtinis intelektas, kompiuterinė rega).

1.3. Taikinių sekimo sistemos

Objektų sekimo metodai skirstomi į generatyvinius ir diskriminacinius metodus. Generatyvinis metodas nustato taikinio modelį dabartiniame kadre (angl. *frame*) ir kitame kadre ieško srities, turinčios didžiausią panašumą į taikinio modelį (Xinlong & Hamdulla, 2020). Algoritmas daugiausia apibūdina paties taikinio charakteristikas, o modelis retai atsižvelgia į fono (angl. *background*) informaciją. Pakeitus arba užblokavus taikinį lengva sugeneruoti sekimo nuokrypį.

Diskriminacinis metodas nurodo taikinio regioną kaip teigiamą pavyzdį (angl. *sample*) dabartiniame kadre, o fono regioną – kaip neigiamą pavyzdį (klasifikatoriui mokyti), ir ieško optimalaus regiono kaip sekamo taikinio kitame kadre (Xinlong & Hamdulla, 2020). Skirtumas tarp diskriminacinių ir generatyvinių metodų yra tai, kad diskriminaciniame modelyje yra informacijos apie foną ir jis turi stiprią sekimo galimybę, kai taikiny yra iš dalies uždengtas arba paslėptas. Todėl

diskriminacinio metodo modelis yra tikslesnis ir patvaresnis ir tapo pagrindiniu metodu taikinių sekimo srityje (Xinlong & Hamdulla, 2020).

Judančio taikinio sekimas yra susijęs su objekto judėjimo parametrais, tokiais kaip vieta, greitis, pagreitis ir pozicija iš eilės einančiuose vaizdo kadruose. Tiriant judančio taikinio sekimą, procesą galima vykdyti dviem būdais (Jing et al., 2015):

- tiesiogiai aptikti ir identifikuoti taikinį vaizdų sekose, nepriklausančiose nuo ankstesnių žinių, ir tada rasti dominančio taikinio vietą;
- pirma sukurti modelį pagal turimas išankstines žinias, o tada realiuoju laiku tiksliai aptikti taikinį kituose kadruose.

Vizualinis taikinių sekimas yra plačiai tiriamas, pastaruoju metu padaryta didelė pažanga kompiuterinės regos srityje (Wu et al., 2015). Vizualinio taikinių sekimo sistemos gali būti apžvelgtos iš dviejų perspektyvų: sekimo modeliai ir taikinių požymių pateikimas (Pang et al., 2017).

Taikant vietinį retų išvaizdų (angl. *local sparse appearance*) metodą, yra pasiūlytas modelis (Liu et al., 2011), kuris padidina vizualinio taikinio sekimo stabilumą. Šis modelis derinamas su vidutinio poslinkio (angl. *mean shift*) algoritmu, siekiant aptikti taikinius. Taikinyss gali būti laikomas skirtingų vietinių vaizdų dalių, kurių erdvinis išdėstymas pagrįstas retais kodais (angl. *sparse codes*), sudėtimi (Jia et al., 2012). Norėdami susidoroti su radikaliais apšvietimo pokyčiais ir greitu judėjimu, Kwon ir Lee (2010) naudojo kelis sekimo ir judėjimo modelius, kuriant taikinio sekimo skaidymo būdą. Tai sukėlė pakankamai didelį išvaizdos pokytį. Dėl nenutrūkstamų sekamo taikinio išvaizdos pokyčių Ross et al. (2008) sukūrė sistemą, kuri palaipsniui atnaujiną savąją bazę ir prisitaiko prie sekimo proceso pokyčių. Pateikti metodai yra generatyvinių metodų pavyzdžiai.

Diskriminacinio metodo algoritmo atveju taikinio sekimo problema laikoma tam tikra klasifikavimo problema, kuriai naudojamas integruotas modelis, siekiant atskirti taikinį nuo fono. Egzistuoja įvairūs internetiniai (angl. *online*) taikinių sekimo algoritmai, vienas jų – daugialypių pavyzdžių mokymosi (angl. *multiple instance learning*) sistema (Babenko et al., 2009). Kitas pavyzdys – mokomas atraminių vektorių klasifikatorius (angl. *Support-Vector Machine* – SVM), integruotas į optinio srauto sistemą, kad būtų galima susidoroti su taikinio išvaizdos pokyčiais (Avidan, 2004). „Struck“ algoritme (Pang et al., 2017) naudojamas struktūrizuotas SVM su branduoliu (angl. *kernelized*), kad būtų sukurtas taikinių sekimo modelis, kuris galėtų panaudoti prognozuojamų rezultatų apribojimus. Henriques et al. (2015) gavo naują branduolių koreliacijos filtrą (angl. *kernelized correlation filter*) su diskrečiąja Furjė transformacija. Filtro naudojimo taikinių sekimo užduočių sprendimui rezultatas – sumažėjo reikalingos atminties apimtis ir skaičiavimų kiekis. Sekimo metodai, taikant aptikimą, ir giliojo mokymosi (angl. *deep learning*) metodai taikiniams sekti, gali būti laikomi atskira diskriminacinių modelių kategorija.

Sprendžiant vizualinio taikinių sekimo užduotį, naudojami keli pateikimo požymiai, kurie gali būti skirstomi į dvi kategorijas: rankų darbo (angl. *hand-crafted*) ir gilieji požymiai (angl. *deep features*). Rankų darbo požymiai paprastai laikomi dirbtiniais, žemo lygio požymiais. Šios kategorijos pavyzdžiai yra spalvų histogramos, orientuotų gradientų histogramos (angl. *histograms of oriented gradients*), kovariantiškumo sričių aprašai (angl. *covariance region descriptors*) ir Haar-tipo požymiai (Pang et al., 2017). Rankų darbo požymiai sukurti vaizdo elemento (pikselio) lygyje (angl. *pixel-level*). Sekimo proceso metu vaizdas gali būti nestabilus dėl trikdžių. Be to, pikselių lygio požymiai priklauso paprastiesiems požymiams (angl. *shallow features*), kurie negali užfiksuoti giliosios ir semantinės taikinio informacijos, tai gali sukelti taikinio sekimo poslinkį, esant sudėtingoms vaizdo sekoms (Pang et al., 2017).

Kompiuterinės regos užduotims naudojami gilieji požymiai. Šie glausti ir semantiniai požymiai aptinkami nuolat kartojant mokymąsi (angl. *ceaseless iterative training*) dideliuose vaizdų rinkiniuose (Pang et al., 2017). Gilieji požymiai yra pakankamai efektyvūs semantiniams požymiams išskirti ir įvairių kategorijų objektams klasifikuoti. Vizualinio taikinių sekimo proceso metu, atvaizduojant taikinio sekimą, gilieji požymiai naudojami kaip juodosios dėžės. Kita vertus, naujesniuose vizualinio sekimo algoritmuose gilusis modelis yra tiesiogiai mokomas esamomis taikinių sekimo sekomis, kad įgytų realiųjų ir semantinių taikinio požymių (Pang et al., 2017).

Didelio tikslumo taikinių atpažinimo ir sekimo sistemos, naudojančios vieną jutiklį arba pasyvų daugiasensorių rinkinį (angl. *multisensor set*), yra jautrios išoriniams trukdžiams ir priklauso nuo aplinkos veiksnių. Šie sunkumai daugiausia kyla dėl turimų atvaizdavimo dažnių juostų apribojimų ir duomenų, susijusių su taikiniais, trūkumo (Qu et al., 2017).

Skirtingai nuo pasyvosios įvairiarūšių jutiklių sistemos, kurios pagrindinė užduotis yra duomenų rinkimas ir duomenų perdavimas duomenų apdorojimo programinei įrangai, aktyvioji sistema geba koreguoti jutiklių poziciją, kad duomenų apdorojimo metu būtų gauta papildomos informacijos (Qu et al., 2017). Skirtingi modaliniai jutikliai renka papildomus duomenis apie taikinį, kad būtų išvengta dviprasmių atpažinimo ir sekimo sprendimų dėl neteisingų duomenų. Tai padeda pagerinti rezultatus, bendrą tikslumą ir padidinti sistemos patikimumą.

1.3.1. Taikinio sekimo algoritmas, naudojant kampinių taškų atitikimą kartu su suspaudimo metodu

Siekiant išvengti dingusio taikinio problemos (atsirandančios dėl pozicijos pasikeitimo, apšvietimo pokyčių, objekto uždengimo), sekant taikinius suspaudimo metodu (angl. *Compressive Tracking* – CT) algoritmais, buvo pasiūlytas naujas sekimo algoritmas (Jing et al., 2015), kuriam naudojamas kampinių taškų atitikimas kartu su suspaudimo metodu (angl. *Angular Point Matching Combined with Compressive Tracking* – APMCCT). Haar-tipo požymiams išskirti naudojama reta

matavimo matrica (angl. *sparse measurement matrix*). Numatomos taikinio padėties poslinkis integruotas į kampinio taško atitikimą bei apskaičiuojama nauja taikinio padėtis (Jing et al., 2015).

Suspaudimo koncepciją 2006 m. pasiūlė Candes ir Donoho. Pagrindinė mintis buvo atsitiktiniu būdu atrinkti signalą, kurį galima suspausti, kai atrankos intervalai yra daug mažesni nei Naikvisto atrankos, ir tada netiesinio atkūrimo algoritmu atstatyti pradinį signalą tam tikrame paklaidų diapazone (Candes & Wakin, 2008). Zhang et al. (2012) pateikė paprastą ir efektyvų sekimo algoritmą, taikant suspaudimo metodą. Buvo sukurta reta atsitiktinio matavimo matrica. Suspaudus pradinį požymį, pagal atsitiktinio matavimo matricą buvo gautas modelis, pagrįstas retu atvaizdavimu, ir, naudojant paprastąjį Bajeso klasifikatorių, taikinyis buvo atskirtas nuo fono (Zhang et al., 2012). Tačiau taikinio deformacijos, apšvietimo pasikeitimo atvejais ar taikiniui dingus, suspaudimo sekimo algoritmas neveiks reikiamu tikslumu (Jing et al., 2015).

Kampinis taškas yra svarbi lokali vaizdo charakteristika. Šiuose taškuose yra vertingos dviejų matmenų struktūros informacijos, kurios pilkos skalės transformacijos vertė yra pakankamai didelė visomis kryptimis (Jing et al., 2015). Harrisio kampo aptikimo algoritmas yra efektyvus sprendimas požymiams išgauti, naudojant tik pilkos spalvos pirmosios eilės skirtumą ir filtravimą. Taigi, tai yra pakankamai stabilus ir patvarus algoritmas, veikiantis sukimosi, triukšmo ir vizualinės transformacijos atvejais (Jing et al., 2015). Pagrindinė Harrisio kampų aptikimo idėja yra suprojektuoti lokalų langą vaizde ir tada apskaičiuoti šio lango energiją visomis kryptimis.

Lokalūs atitikimo metodai (pagrįsti regionu) apima absoliučių skirtumų reikšmių sumą (angl. *sum of absolute differences*), kvadratinų skirtumų reikšmių sumą (angl. *sum of squared differences*) ir normalizuotą abipusę koreliaciją (angl. *Normalized Cross Correlation* – NCC) (Jing et al., 2015). NCC algoritmas pasižymi geriausiu triukšmo slopinimo gebėjimu ir atitikimo tikslumu, todėl Jing et al. (2015) savo APMCCT algoritme NCC metodą pritaikė kampinio taško atitikimui.

Neatitikimų būna nepriklausomai nuo to, koks požymio deskriptorius yra naudojamas ir koks atitikmens mato metodas yra taikomas. Jing et al. (2015) APMCCT algoritme, siekdami pašalinti kandidatų atitikimo taškų neatitikimą, naudoja Ransako algoritmą. Šis algoritmas apskaičiuoja duomenų matematinį modelį, remdamasis duomenų rinkinių pavyzdžiais, kuriuose yra anormaliųjų duomenų, ir tada gauna naudingų pavyzdžių (Jing et al., 2015).

Šablonui atnaujinti gali būti naudojamas Gauso skirstinys (Zhang et al., 2012). Pavyzdžių vidurkis ir dispersija atnaujinami po sekimo. Tačiau tai sukelia sekimo paklaidas dėl taikinio uždengimo, apšvietimo pokyčių ir pozicijos kitimo (Jing et al., 2015). Jei šablonas bus atnaujintas po sekimo, tai sukels didesnę nuokrypį – šablono ir taikinio panašumas bus mažas. Jing et al. (2015) taikinio padėtį palygina su ankstesniu šablonu per NCC, o didesnio panašumo šablonas laikomas dabartinio kadro šablonu.

Kampinio taško atitikimo kartu su suspaudimo metodu (APMCCT) algoritmo žingsniai (Jing et al., 2015):

1) Inicijavimas: rankiniu būdu pasirenkamas taikinyis ir nustatomi pradiniai parametrai; sukuriami projekcijos matricą; nustatomi klasifikatoriaus pradinių charakteristikų skaičius ir pradiniai parametrai.

2) Skaičiavimas: nuskaitomas dabartinio kadro vaizdas, tada viename regione surenkami keli teigiami pavyzdžiai, kitame regione – keli neigiami pavyzdžiai. Kandidato srityje apskaičiuojamas kiekvieno pavyzdžio suspausto domeno požymio vektorius. Naudojant Harrisio operatorių, ištraukiamas dabartinio vaizdo kampinis taškas. Kadangi pirmojo kadro vaizdo šablonas apskaičiuojamas pagal taikinio srities apimtį, likusi vaizdo dalis, atsižvelgiant į taikinio judėjimo charakteristikas, apskaičiuojama išplėstoje 10 % srityje.

3) Sekimas: apskaičiuojama kiekvieno kandidato pavyzdžio atsako vertė, o pavyzdys, turintis didžiausią vertę, pasirenkamas taikinio padėties poslinkiui apskaičiuoti; apskaičiuojamas kampinio taško poslinkis, vėliau taikinio padėčiai apskaičiuoti naudojami du poslinkiai.

4) Klasifikatoriaus atnaujinimas: apskaičiuojama normalizuota abipusė koreliacija tarp esamos vietos ir pradinio šablono; jei normalizuota abipusė koreliacija yra didesnė už slenkstį, šablonas atnaujinamas.

5) Grįžtama prie 2 žingsnio kito kadro apdorojimui.

Skirtingų vaizdo sekų eksperimentai rodo, kad APMCCT algoritmas gali tiksliai prisitaikyti prie taikinio ir fono pokyčių bei tiksliai sekti taikinį. Tikslumas ir patvarumas akivaizdžiai pagerėjo, lyginant su CT algoritmu (Jing et al., 2015). APMCCT pagrindinis trūkumas – algoritmas negali adaptyviai koreguoti sekimo lango mastelio.

1.3.2. Aktyvi įvairiarūšių jutiklių sistema taikiniams atpažinti ir sekti

Taikinių atpažinimo ir sekimo sistemose duomenų rinkimas ir apdorojimas dažniausiai yra visiškai atskirti – jutikliai naudojami tik duomenims rinkti, o duomenų apdorojimo sistema naudojama tik jutiklių surinktiems duomenims apdoroti. Dėl šio atskyrimo gaunamas tikslumas ir patikimumas gali būti nepakankami, jei apdorojimo metu nustatoma, kad duomenys yra neišsamūs arba dviprasmiai ir nėra jokių papildomų duomenų šiems trūkumams pašalinti (Qu et al., 2017).

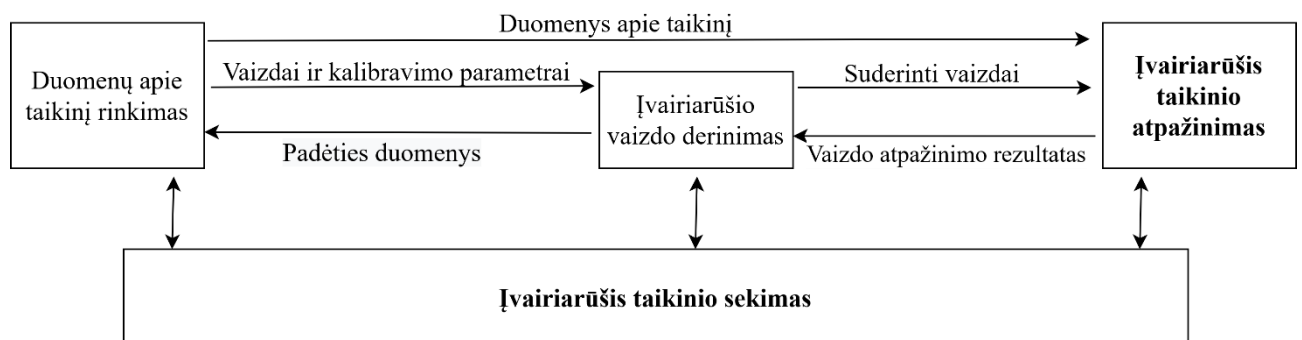
Qu et al. (2017) siūlo aktyvią įvairiarūšių jutiklių sistemą (angl. *active multimodal sensor system*), skirtą taikiniui atpažinti ir sekti. Sistema susideda iš regimojo spektro, infraraudonojo spektro ir hiperspektrinio jutiklių. Ji visapusiškai išnaudoja jutiklių informacijos rinkimo galimybes – gali aktyviai valdyti skirtingus jutiklius, kad surinktų papildomų duomenų, atsižvelgiant į taikinio atpažinimo ir sekimo realiuoju laiku procesų poreikius. Yra pateikti du pagrindiniai siūlymai.

Pirma, siūloma aktyvi įvairiarūšių jutiklių atpažinimo ir sekimo sistema, kurioje aparatinės įrangos duomenų rinkimas ir programinės įrangos duomenų apdorojimas gali sąveikauti. Duomenų rinkimo metu duomenų apdorojimo algoritmas nurodo ir valdo jutiklius, kad būtų optimizuotas duomenų apie taikinį rinkimas (Qu et al., 2017). Duomenų apdorojimo metu skirtingi modaliniai jutikliai renka papildomą informaciją pagal duomenų apdorojimo sistemos poreikius. Tai padeda patvirtinti atpažinimo ir sekimo rezultatus, pagerinti bendrą tikslumą ir padidinti sistemos patikimumą (Qu et al., 2017).

Antra, taikiniui atpažinti ir sekti siūloma naudoti hiperspektrinį jutiklį ir hiperspektriniais duomenimis pagrįstą atpažinimo algoritmą. Taikinio spektras priklauso tik nuo jo sudėties ir nėra jautrus aplinkai. Vadinasi, spektru pagrįsto atpažinimo ir sekimo metodo tikslumas yra didesnis nei metodų, pagrįstų vaizdais (Qu et al., 2017). Visų pirma, kai taikiny yra toli ir informacijos apie jo formą nėra (šiuo atveju taikiny yra suprantamas kaip taškas, kelių ar kelių dešimčių pikselių dydžio), hiperspektriniai duomenys vis tiek gali būti naudojami taikiniams atpažinti ir sekti. Taip pat pasiūlyta sistema gali atpažinti ir sekti taikinius sudėtingomis sąlygomis, pavyzdžiui, kai yra du tos pačios formos taikiniai, kai sekamas taikiny buvo pamestas ar kai sekamas ne tas taikiny (Qu et al., 2017). Sistema gali realiuoju laiku aptikti šias situacijas ir grįžti prie teisingo taikinio sekimo.

Qu et al. (2017) siūlomą sistemą sudaro keturi pagrindiniai funkciniai blokai (1.5 pav.):

- duomenų apie taikinį rinkimas,
- įvairiarūšio vaizdo derinimas,
- įvairiarūšis taikinio atpažinimas,
- įvairiarūšis taikinio sekimas.



1.5 pav. Sistemos veikimas (sudaryta autorės, remiantis Qu et al. (2017))

Pirmasis blokas naudojamas regimojo spektro, infraraudonojo spektro ir hiperspektriniais duomenimis apie taikinį rinkti. Įvairiarūšiai optiniai vaizdai (regimieji ir infraraudonieji vaizdai) yra suderinami įvairiarūšio vaizdo derinimo bloke, o atstumą tarp taikinio ir sistemos galima apskaičiuoti naudojant trianguliaciją tarp regimojo spektro jutiklio, infraraudonojo spektro jutiklio ir taikinio (Qu et al., 2017).

Įvairiarūšio taikinio atpažinimo blokas naudojamas esamam taikiniui identifikuoti. Taikinio atpažinimas atliekamas pagal optinius vaizdus (regimuosius, infraraudonuosius ir suderintus regimųjų-infraraudonųjų) ir hiperspektrinius duomenis. Dėl hiperspektrinio jutiklio atvaizdavimo veikos (vienu metu galima gauti tik vieną plyšio vaizdą), reikia pasukti rotatorių, esantį žemiau jutiklio, kad atvaizdavimo spindulys būtų orientuotas į taikinį (Qu et al., 2017). Čia panaudojami erdviniai parametrai, gauti įvairiarūšio vaizdo derinimo metu. Tokiu būdu sistema gali surinkti tikslius hiperspektrinius duomenis apie taikinį ir atlikti papildomą patvirtinimą hiperspektriniu atpažinimu.

Aptiktam taikiniui sekti naudojamas įvairiarūšio taikinio sekimo blokas, kuris bendradarbiauja su duomenų rinkimo, derinimo ir atpažinimo blokais. Duomenų rinkimo blokas aktyviai renka regimuosius ar infraraudonuosius vaizdus ir nukreipia hiperspektrinius duomenis, kai taikinio sekimo blokui reikia papildomų duomenų (Qu et al., 2017). Esant šioms papildomiems duomenims, sistema gali gauti papildomą patvirtinimą apie taikinio pobūdį (teisingas ar klaidingas) ir užtikrinti sekimo tikslumą bei patvarumą. Pametus taikinį, taip pat bus iškvieistas įvairiarūšio atpažinimo blokas, kuris atpažins tikrąjį sekimo taikinį visame vaizde, taip padėdamas sekimo blokui atkurti tikrąjį taikinio kelią (Qu et al., 2017).

Ekspimentiniu būdu buvo įrodyta, kad šis aparatinės įrangos surinkimo kontrolės ir duomenų apdorojimo integravimo lygis efektyviai pagerina taikinių atpažinimo ir sekimo sistemų tikslumą ir patvarumą (Qu et al., 2017).

1.3.3. Infraraudonųjų mažų taikinių sekimo metodas

Dėl sudėtingo fono ypatumų, taikinio signalo silpnumo, neraiškių kraštų, vaizdo blokavimo ir trukdžių infraraudonojo blankaus mažo taikinio (angl. *infrared dim small target*) sekimas laikomas sudėtingu. Geresnis veikimas realiuoju laiku, didesnė aptikimo tikimybė ir mažesnis menamojo vaizdo greitis tapo priežastimi mokslininkų dėmesiui atkreipti ir tyrimams atlikti šioje srityje (Xinlong & Hamdulla, 2020).

Infraraudonųjų mažų taikinių sekimas plačiai naudojamas žvalgybai, išankstiniam įspėjimui ir valdymui galutiniame trajektorijos ruože, taip pat tai yra viena iš kompiuterinės regos tyrimų srities kryptų. Infraraudonasis vaizdas – tai vaizdas, fiksuojantis objekto infraraudonųjų spindulių intensyvumą (Xinlong & Hamdulla, 2020). Taigi, infraraudonųjų blankių mažų taikinių vaizdas pasižymi mažu plotu ir intensyvumu.

Mažas taikiny – tai toks taikiny, kurio bendras erdvinis mastas yra mažesnis nei 80 pikselių (9×9), o tai yra mažiau nei 0,12 % 256×256 pikselių vaizdo (Xinlong & Hamdulla, 2020). Jūros ir dangaus fono infraraudonojoje fotoelektrinėje žvalgybos įrangoje tolimi orlaivių ar raketų taikiniai paprastai neviršija 6×6 pikselių vaizdo.

Ankstyvojo taikinio sekimo algoritmo kūrimą galima suskirstyti į du etapus (Xinlong & Hamdulla, 2020). Pradiniam tyrimui pagrindinis dėmesys buvo skiriamas pagrindinio vieno žingsnio taikinio sekimo realizavimui, kuriam naudojami globaliojo artimiausio kaimyno (angl. *global nearest neighbor*), tikimybinio duomenų ryšio (angl. *probabilistic data association*) ir jungtinio tikimybinio duomenų ryšio (angl. *joint probabilistic data association*) algoritmai.

Toliau aprašyti kai kurie mokslininkų darbai, sprendžiant taikinių sekimo problemą, kai signalo ir triukšmo santykis (angl. *Signal-to-Noise Ratio* – SNR) yra mažas. Vieni iš tyrimų rezultatų yra dalelių filtro (PF), dinaminio programavimo (angl. *Dynamic Programming Algorithm* – DPA) ir kelių hipotezių sekimo (angl. *Multiple Hypothesis Tracking* – MHT) algoritmai.

PF yra žinomas kaip nuosekliojo diskretizavimo (angl. *sampling*) metodas, skirtas Bajeso filtravimo rekursijai realizuoti per Monte Karlo integracijos modeliavimą (Xinlong & Hamdulla, 2020). PF algoritmas naudojamas netiesinėms ir ne Gauso problemoms spręsti bei tradicinio išplėstinio Kalmano filtro netiesinio paklaidų kaupimo problemai išspręsti. Dėl tikimybės pasiskirstymui imituoti naudojamo didelio dalelių skaičiaus ir didelio skaičiavimų kiekio dalelių praskiedimas lengvai įvyksta atrankos proceso metu ir veda prie sekimo nuokrypio.

MHT daroma prielaida, kad lokalaus tolygiojo judėjimo taikinyje yra tiesioje linijoje. Atsižvelgiant į greitį ir kryptį, galima daugiau nei viena įmanoma taikinio trajektorija pagal „medžio“ struktūrą. Kiekviename „medžio“ struktūros sluoksnyje hipotezių testavimo metodas naudojamas struktūrai modifikuoti, pašalinant nepaėjusį testavimo „medį“, kad sumažėtų skaičiavimų kiekis ir naudojama atmintis. MHT yra plačiai taikomas metodas objektui aptikti, naudojant daugiažingsnį sekimą ir vėlinimo logiką (angl. *delay logic*). MHT yra efektyvus algoritmas, turintis mažą skaičiavimų kiekį ir mažą atminties talpą bei galimybę vienu metu aptikti kelis tiesiai įvairiomis kryptimis judančius taikinius (Xinlong & Hamdulla, 2020).

DPA yra sekimo prieš aptikimą (angl. *track-before-detect*) algoritmas. DPA poskyrio optimizavimo idėja buvo panaudota taikinio trajektorijos paieškos problemai išskaidyti į hierarchinę optimizavimo problemą (Xinlong & Hamdulla, 2020). DPA pagrindu atliekamas sekimas yra pagrįstas pikselių veikimo lygiu, tinkančiu aparatinei įrangai įgyvendinti, ir gali nustatyti taikinio tiesiaiečio judėjimo trajektoriją, esant mažam SNR, atlikus nedaug skaičiavimų.

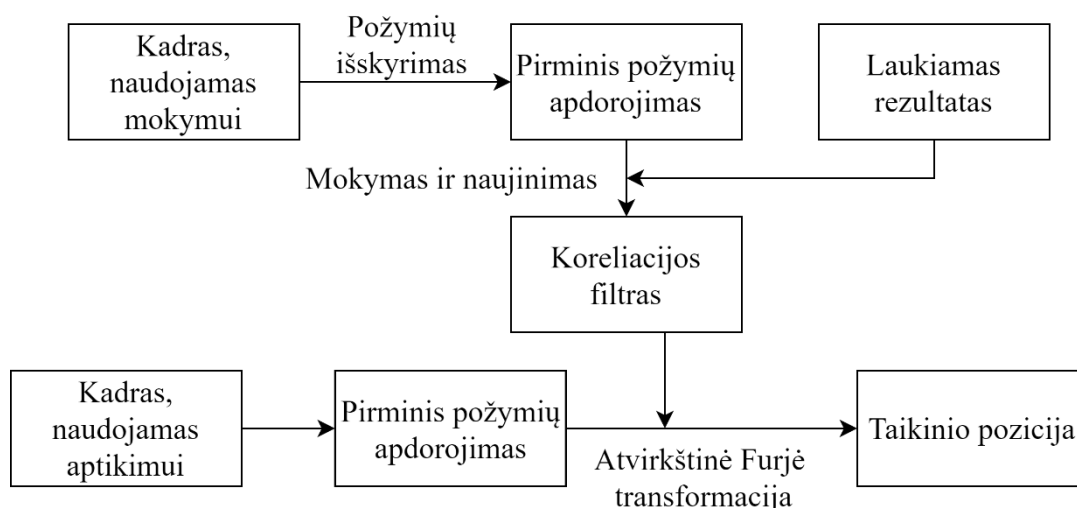
Vienu metodu galima išspręsti tik atitinkamą problemą su dideliais apribojimais. Todėl keli metodai yra taikomi kartu, siekiant kompensuoti vieno būdo trūkumus. Vienas iš pasiūlytų sprendimų yra adaptyvusis netiesinis taikinio sekimo algoritmas, derinantis gradiento histogramos požymius su vidutinio poslinkio metodu (Xinlong & Hamdulla, 2020). Patobulintas gradiento histogramos požymis naudojamas mažų ir blankių taikinių bei mažo kontrasto taikinių vaizdų išsamiems požymiams atvaizduoti, esant pereinamojo vaizdo spinduliavimo sąlygoms. Atitinkama sekimo sistema sukurta taikant vidutinio poslinkio metodą, kuris efektyviai padidina kontrastą tarp taikinio

ir aplinkos triukšmo. Palyginti su sekimo algoritmu, pagrįstu klasikiniu vidutinio poslinkio metodu, akivaizdžiai sumažėja sekimo lango plotis taikinio atžvilgiu, o tai pagerina sekimo algoritmo patvarumą (Xinlong & Hamdulla, 2020).

Kitas pavyzdys yra patobulintas PF algoritmas, derinantis PF algoritmą su neneigiamos matricos skaidymo metodu (Xinlong & Hamdulla, 2020). Sekimo algoritmas įvertina gerų neneigiamos matricos skaidymo požymių išskyrimo rezultatų vidurkį, optimizuoja paskirstymo svorį dalelių atrankos proceso metu ir sumažina paklaidą, kurią sukelia klasikinio PF charakteristikų blogėjimas ir divergencija. Šis metodas pasižymi geromis sekimo galimybėmis ir patvarumu, taikiniui laikinai dingus iš vaizdo.

Dar vienas pasiūlytas patobulintas metodas derina glodinimo filtro „Guided Filter“ pranašumus ir erdvines bei laiko konteksto charakteristikas (Xinlong & Hamdulla, 2020). Šis metodas naudoja filtro pranašumus, kad būtų galima atskirti foninį kraštų triukšmą, ir sukuria sekimo modelį kartu su erdvinėmis bei laiko konteksto charakteristikomis. Metodas pasižymi geresniu sekimo efektyvumu nei klasikinis erdvės bei laiko filtravimo sekimo būdas, esant mažam SNR. Apdorojimo greitis siekia 60 kadrų per sekundę, tai labai pagerina našumą realiuoju laiku (Xinlong & Hamdulla, 2020).

Koreliacijos filtras taikinių sekimo srityje buvo pritaikytas 2010 m., kai mokslininkai pasiūlė koreliacijos filtro taikinių sekimo metodą, pagrįstą mažiausia vidutine kvadratine paklaida. Metodas pakeičia vaizdo sąsūką į dažninę charakteristiką pagal Furjė transformaciją, tokiu būdu sumažinamas skaičiavimų kiekis (Xinlong & Hamdulla, 2020). Bendra koreliacijos filtro sekimo sistemos schema pateikta 1.6 paveiksle. Koreliacijos filtras mokomas pagal taikinio regioną ir laukiamą rezultatą, o maksimali paieškos atsakymo padėtis kitame kadre yra taikinio padėtis.



1.6 pav. Koreliacijos filtro sekimo algoritmo schema (sudaryta autorės, remiantis Xinlong ir Hamdulla (2020))

Pateiktas naujo tipo koreliacijos filtras – minimalios kvadratinės paklaidos išvesties sumos (angl. *Minimum Output Sum of Squared Error* – MOSSE) filtras, kuris sukuria stabilius koreliacijos

filtrus, kai yra inicijuojamas naudojant vieną kadrą. MOSSE filtrų pagrindu veikiantis sekiklis yra patvarus apšvietimo, masto, pozicijos ir nestandžių deformacijų pokyčiams, veikdamas 669 kadrų per sekundę greičiu (Bolme et al., 2010).

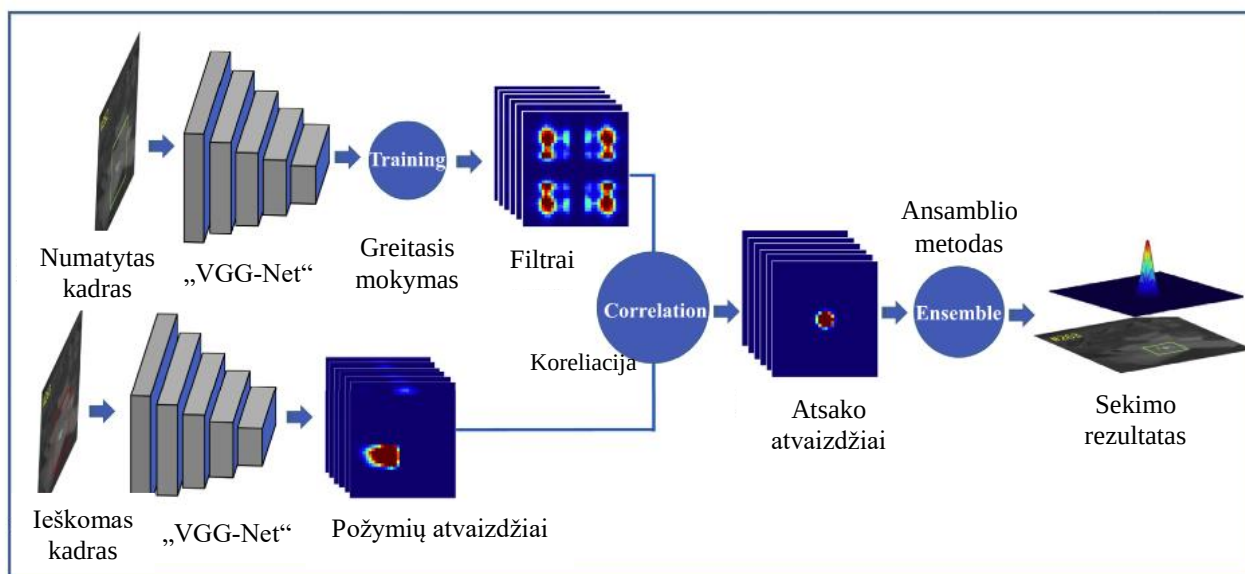
Yra daug būdų, kaip galima pagerinti šį sekiklį. Pavyzdžiui, jei taikinio išvaizda yra pakankamai stabili, dreifą galima sumažinti atsitiktinai iš naujo centruojant pradinio kadru pagrįstą filtrą. Sekiklį taip pat galima išplėsti, kad būtų galima įvertinti masto ir sukimosi pokyčius, filtruojant sekimo lango loginį ir poliarinį transformavimą po atnaujinimo (Bolme et al., 2010).

Šiuo metu giliojo mokymosi taikymą taikinių sekimo srityje galima suskirstyti į dvi kategorijas (Xinlong & Hamdulla, 2020):

- gilusis mokymasis naudojamas kaip požymių išskyrimo metodas kartu su atitinkamu filtravimo metodu, ir atitinkamas filtravimo metodas naudojamas kaip sekimo sistema;
- taikinio sekimo metodas, pagrįstas neuroninio mokymosi sistema.

Liu et al. (2017) pasiūlytas infraraudonųjų taikinių sekimo algoritmas, derinantis giliąsias charakteristikas ir koreliacijos filtro algoritmą. Taikant šį metodą, iš anksto išmokytas konvoliucinių neuroninių tinklų (angl. *Convolutional Neural Networks* – CNN) modelis yra tiksliai sureguliuotas infraraudonųjų spindulių duomenų rinkinyje, kad būtų galima išskirti infraraudonųjų taikinių charakteristikas. Naudojant hierarchinius požymius, sukonstruojamas silpnai koreliuojantis filtras, siekiant gauti mažos skyros (angl. *image resolution*) infraraudonojo vaizdo ir nepakankamos taikinio informacijos apie vienasluoksnį neuroninį tinklą charakteristikas. Tada gaunamas taikinio atsakas ir vykdomas sekimas pagal hierarchinį atsako adaptyvųjį svorių pasiskirstymą.

Visą algoritmą (1.7 pav.) galima suskirstyti į dvi dalis: mokymą ir paiešką.



1.7 pav. Siūlomo metodo struktūra (sudaryta autorės, remiantis Liu et al. (2017))

Mokymui naudojamas iš anksto išmokytas „VGG-Net“ tinklas, kad būtų išskirtos šiluminio infraraudonojo taikinio konvoliucijos požymiai numatyame kadre. Tada, siekiant gauti kiekvieno

sąsūkos sluoksnio filtrus, naudojami šie konvoliucijos požymiai ir Gauso formos žymė. Iš anksto išmokytas „VGG-Net“ tinklas taip pat naudojamas ieškomo kadro konvoliucijos požymiams išskirti. Tada filtrai iš mokymo dalies ir požymių atvaizdžiai iš paieškos dalies surenkami, kad būtų apskaičiuota jų abipusė koreliacija. Kiekvienas filtras ir požymių atvaizdas sukuria atsako atvaizdį. Galiausiai atsako atvaizdžiai sujungiami ansamblio metodu (angl. *ensemble method*), kad būtų gautas geresnis sekiklis (Liu et al., 2017).

Wang ir Yeung (2013) taip pat taiko gilųjį mokymąsi taikiniams sekti. Siekiant gauti universalią taikinių požymių išskyrimo sistemą, buvo sukonstruotas ir išmokytas sudurtinis, triukšmą slopinantis autoenkoderis (angl. *Stacked Denoising Autoencoder* – SDAE). Skirtingai nuo daugelio esamų sekiklių, kurie sužino sekamo objekto išvaizdą tik virtualiai (angl. *online*), autoriai labiau akcentavo požymių mokymosi (be priežiūros) problemą (Wang & Yeung, 2013). Naudojant pagalbinį tikrą vaizdą, SDAE mokomas iš anksčiau (angl. *offline*), kad būtų išmokyti bendro vaizdo požymiai, kurie yra patvaresni prieš variacijas. Po to gautos žinios pridamos prie virtualiojo sekimo proceso. Virtualusis sekimas apima klasifikacijos neuroninį tinklą, kuris yra sudarytas iš mokyto autoenkoderio dalies kaip požymių išskyrimo ir papildomo klasifikacijos sluoksnio. Tiek požymių išskyrimą, tiek klasifikatorių galima toliau derinti, kad prisitaikytų prie judančio objekto išvaizdos pokyčių. Palyginus su pažangiausiais sudėtingų vaizdo sekų sekikliais, pasiūlytas giliojo mokymosi sekiklis yra tikslesnis (Wang & Yeung, 2013), išlaikant mažas skaičiavimo sąnaudas realiuoju laiku, kai sekiklio realizavimas atliktas naudojant programinės įrangos MATLAB grafinį procesorių (angl. *Graphics Processing Unit* – GPU).

Dėl fono sudėtingumo ir taikinio judėjimo infraraudonųjų blankių mažų taikinių sekimas išlieka viena iš sudėtingiausių tyrimų kryptų kompiuterinės regos srityje. Dėl atitinkamų filtravimo ir giliojo mokymosi metodų trūkumo infraraudonųjų mažų taikinių sekimo srityje, dabartiniai tokiais metodais pagrįsti sekimo algoritmai nėra tobuli (Xinlong & Hamdulla, 2020). Tačiau, lyginant su klasikiniiais metodais, gilieji neuroniniai tinklai turi stiprias požymių išgavimo (angl. *feature extraction*) galimybes ir pasiekia didelį tikslumą, todėl sparčiai vystantis giliajam mokymuisi, CNN pagrįstų objektų aptikimo ir sekimo modelių veikimas nuolat gerėja.

1.4. Taikinių atpažinimas ir sekimas, taikant giliuosius neuroninius tinklus

Pastaraisiais metais gilieji neuroniniai tinklai kuriami ir taikomi atliekant daugybę kompiuterinės regos (angl. *computer vision*) užduočių, pavyzdžiui, vaizdų klasifikavimas ir atpažinimas, objektų aptikimas, vaizdo segmentavimas, veido atpažinimas, žmogaus pozos įvertinimas (Pang et al., 2017). Pagrindinės giliųjų neuroninių tinklų populiarumo priežastys yra jų tikslumas, efektyvumas ir lankstumas. Techninės įrangos pažanga leidžia giliesiems neuroniniams tinklams spręsti problemas, kurių sudėtingumas neleistų taikyti tokių metodų dar prieš keletą metų. Giliųjų neuroninių tinklų

hierarchiniai paslėptųjų požymių sluoksniai pranoksta rankų darbo požymius, todėl jų populiarumas kompiuterinės regos srityje auga. Tačiau, sprendžiant vizualinio taikinių sekimo užduotį, gilieji neuroniniai tinklai susiduria su problema – daugybės žymėtų vaizdų (angl. *labeled images*) surinkimas (Pang et al., 2017).

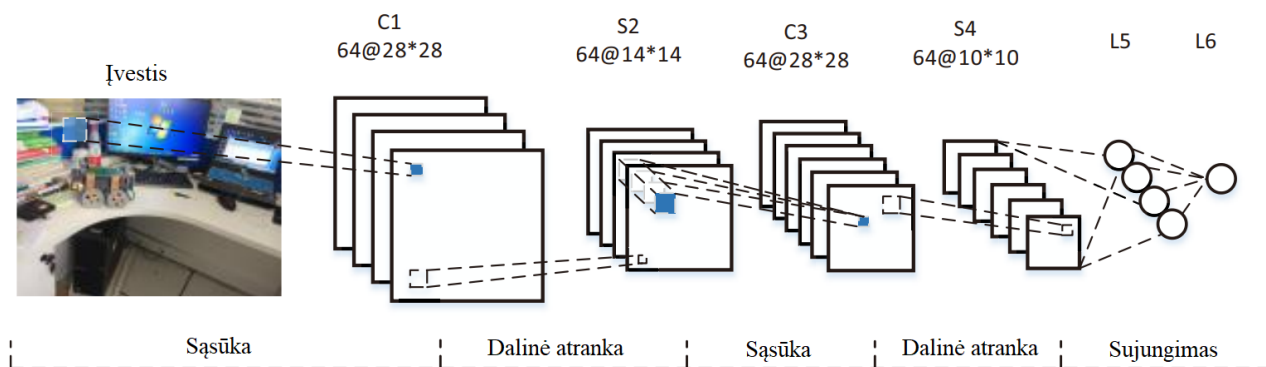
Siekiant sušvelninti šią duomenų stygiaus problemą, siūloma perkelti giliuosius požymius iš anksčiau išmokyto (angl. *offline training stage*) giliojo neuroninio tinklo į didelį pagalbinį duomenų rinkinį (Pang et al., 2017). Pavyzdžiui, anksčiau minėtame Wang ir Yeung (2013) darbe, giliuoju modeliu buvo pasirinktas sudurtinis, triukšmą slopinantis autoenkoderis (SDAE), kuris buvo išmokytas iš anksčiau be priežiūros (angl. *unsupervised learning*), naudojant didelį „Tiny“ duomenų rinkinį (Pang et al., 2017). Siekiant išvengti poslinkio triktį, toliau buvo atliktas tikslus modelio parametrų derinimas pagal skirtingus vaizdo įrašus taikinio sekimo atžvilgiu. Šio tipo gilieji neuroniniai tinklai buvo plačiai naudojami, nes mokymui be priežiūros nereikia žymėtų vaizdų. Tačiau visiškai sujungti neuroniniai tinklai turi milijonus parametrų, kuriuos reikia suderinti, taip pat jie nepaiso erdvinės vaizdų struktūros. Šie klausimai paskatino mokslininkus pasiūlyti konvoliucinius neuroninius tinklus (CNN), kurių naudojimas plinta visose srityse, susijusiose su vaizdų apdorojimu (Pang et al., 2017).

CNN yra specialiai sukurti vaizdams kaip įvestims tvarkyti, todėl tam tikrus požymius galima užkoduoti architektūroje (Pang et al., 2017). Tai lemia persiūtos funkcijos (angl. *forward function*) efektyvumą, taip pat sumažina parametrų kiekį tinkle. CNN turi šias savybes (Pang et al., 2017):

- Sluoksniai yra išdėstyti 3 dimensijose (plotis, aukštis ir gylis), o kiekvienas sluoksnio apdorojimo vienetas (neuronas) yra prijungtas tik prie nedidelio ankstesnio sluoksnio srities. CNN susideda iš skirtingų tipų sluoksnių: lokaliai sujungtų ir visiškai sujungtų.
- Vietinis ryšys išnaudoja vietos koreliaciją vaizduose. Lokaliai sujungtų neuronų pogrupiai gerai parodo mažas įvesties vaizdo dalis. Išmokius tinklą, jie veikia kaip vietiniai filtrai. Sudėtingesnis vaizdavimas gaunamas suduriant kelis lokaliai sujungtus sluoksnius.
- Lokaliai sujungtų sričių svoris yra padalijamas visiems to paties sluoksnio neuronams – šis filtras yra vienodas kiekvienam tam tikro sluoksnio neuronui. Filtras gauna įvestį iš skirtingos įvesties vaizdo srities, tai leidžia aptikti požymius, neatsižvelgiant į srities padėtį vaizde.

Lyginant su tradiciniu neuroniniu tinklu, konvoliucinis neuroninis tinklas prideda konvoliucijos sluoksnį, kuris gali atlikti požymių išskyrimą, ir mažesnės atrankos (angl. *downsampling*) sluoksnį, kuris užtikrina poslinkį (Zhang et al., 2018). Konvoliucijos sluoksnis paprastai yra sujungtas su įvesties sluoksniu konvoliuciniame neuroniniame tinkle, įvesties vaizdas gaunamas iš įvesties sluoksnio. Vidurinis tinklo sluoksnis yra suformuotas pakaitomis sujungiant konvoliucijos sluoksnį ir mažesnės atrankos sluoksnį. Išvesties sluoksnis naudoja taikinio žymės informaciją ir mokymo metodą su priežiūra tinklo konvergavimo skaičiavimams atlikti (Zhang et al., 2018). Taip pat

neuroninis tinklas leidžia sumažinti skaičiavimo sudėtingumą, pasidalinti neuronų svorį neuronų plokštumoje ir išskirti skirtingus požymius per skirtingus sąsūkos branduolius. Viena iš klasikinių konvoliucinių neuroninių tinklų struktūrų yra „LeNet-5“, pavaizduota 1.8 paveiksle.



1.8 pav. Konvoliucinio neuroninio tinklo „LeNet-5“ sluoksnių struktūros schema (sudaryta autorės, remiantis Zhang et al. (2018))

„LeNet-5“ iš viso turi 7 sluoksnius, kiekviename sluoksnyje yra sujungimo svoris ir daugybė požymių. Kiekvienas požymis (turi daugiau nei vieną neuroną) išskiria po vieną įvesties požymį per sąsūkos filtrą (Zhang et al., 2018). Taikant vietinį vaizdo koreliacijos principą, vaizdas lokaliai atrenkamas, o duomenų apdorojimo suma sumažinama, išsaugant svarbią informaciją (Zhang et al., 2018).

1.4.1. Taikinių sekimo algoritmas, pagrįstas konvoliuciniu neuroniniu tinklu

Daugiasluoksnis dirbtinis neuroninis tinklas (angl. *Artificial Neural Network* – ANN) pasižymi puikiais mokymosi gebėjimais. Taikinio charakteristikų mokymasis per ANN yra natūralesnis, palankesnis atskiriant skirtingus taikinius. Konvoliucinis neuroninis tinklas (CNN) yra naujo tipo tinklas, sujungiantis ANN su sąsūkos operacijomis (Zhang et al., 2018). Naudojant CNN, galima nustatyti įvairias taikinio būsenas, tinklas turi patvarumą tam tikro laipsnio iškraipymams ir atsirandančiai taikinio deformacijai. Naudojamas retas sujungimas ir svorio pasidalijimas sumažina tradicinio neuroninio tinklo parametrų skaičių. Remiantis šiais pranašumais, daugelis pasiūlytų CNN aptikimo metodų pasižymi gerais rezultatais, aptinkant įvairius taikinius (Zhang et al., 2018).

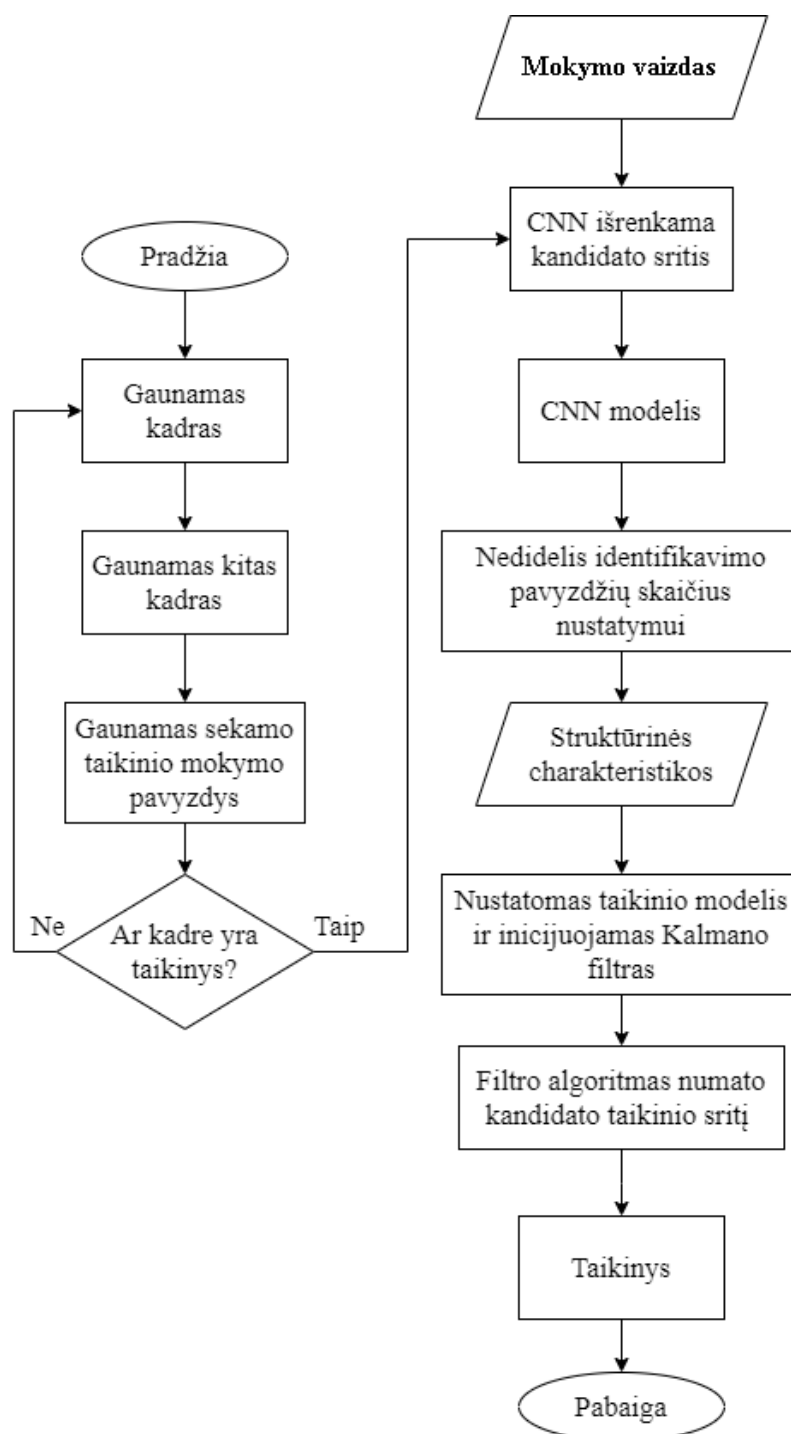
Tačiau vis dar yra daug aktualių problemų, atsirandančių CNN modeliu pagrįsto objektų aptikimo ir atpažinimo metu. Dažniausiai modelis naudojamas paprastesnės struktūros objektams aptikti, (skaičiams, raidėms (Chen et al., 2006)) arba kai kuriems sudėtingesniems taikiniams, pavyzdžiui, pėstiesiems, dideliems gyvūnams ir pan. Modeliui paprastai reikia daug paslėptų mazgų, todėl mokymo laikotarpis yra labai ilgas, o galutinis rezultatas dažnai nepatenkina lūkesčių. Taip pat CNN aptikimo tinklas nepaiso taikinio padėties, greičio, krypties ir kitos judėjimo informacijos. Dėl nenuspėjamumo sekimo algoritmas būna labai paveikiamas, pavyzdžiui, kai objektas yra uždengtas per vidurį arba uždengtas visiškai, taikinytis tokiu atveju dažnai yra prarandamas ir sekimas

nepavyksta. Kai taikiny s vėl pasirodys, realiojo laiko sekimas bus paveiktas dėl neteisingo vertinimo. Todėl yra ieškoma būdų, kaip padaryti CNN modelį tinkamesnį sudėtingiems taikiniams aptikti, esant įvairiems scenarijams.

Zhang et al. (2018) savo darbe giliojo mokymosi konvoliucinį neuroninį tinklą įveda į sekimo algoritmą. Kandidatų regionai vaizdų sekoje parenkami kaip įvestis į CNN. Neuroninio tinklo sąsūkos operacija apskaičiuojami taikinio išvaizdos modelio požymiai. CNN išvestis (taikinio požymis) naudojama kaip tiesinio atraminių vektorių klasifikatoriaus (SVM) įvestis, siekiant nustatyti, ar dabartinis kandidato regionas yra sekamas taikiny s (Zhang et al., 2018). Tuo pačiu metu, naudojant įvertinimo rezultatą kartu su atrankos procedūra, surenkami teigiami ir neigiami dabartinio kadro sekamo taikinio pavyzdžiai (SVM atnaujinimui).

Taikinio struktūrinio požymio modeliui mokyti Zhang et al. (2018) naudojama didelė duomenų rinkinio CIFAR-10 atpažinimo biblioteka, skirta išankstiniam mokymui, o parametrų derinimo mokymui naudojama mažesnė bandomoji pavyzdžių biblioteka. Sekikliui naudojamas Kalmano filtras, kuris prideda nuspėjamumo, kad būtų sumažinta klaidingai teigiamo rezultato tikimybė. Kalmano filtras tam tikru mastu gali tiksliai sekti taikinį, išspręsti sekimo problemą, kurią sukelia sekimo paklaida, ir padidinti algoritmo patvarumą (Zhang et al., 2018).

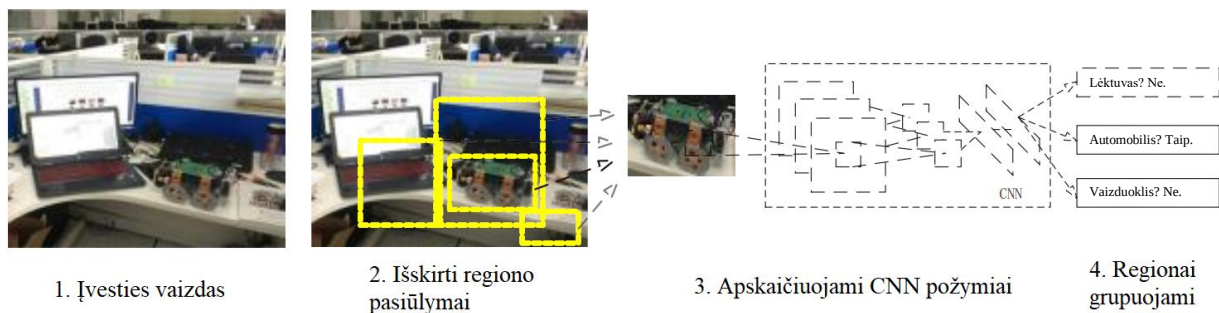
Zhang et al. (2018) siūlomas algoritmas pateiktas 1.9 paveiksle. CNN naudojamas taikinio struktūrinių požymių talpyklai mokyti kartu su regiono pasirinkimo algoritmu. Pirmiausia generuojama pradinė kandidato sritis, tada apskaičiuojamas gretimos srities poros spalvų ir tekstūros panašumas, toliau filtruojama pradinė kandidato sritis. Galutinis kandidato regionas yra konvoliucinio neuroninio tinklo, veikiančio regiono pagrindu, įvestimi (Zhang et al., 2018). Giliojo mokymosi metodu išskiriami taikinio regiono požymiai, kurie vertinami pagal SVM panašumą, kad galiausiai būtų apskaičiuota sekamo taikinio padėties informacija (Hu et al., 2012).



1.9 pav. Taikinių sekimo algoritmas, pagrįstas konvoliuciniu neuroniniu tinklu (sudaryta autorės, remiantis Zhang et al. (2018))

Eksperimentiniai rezultatai rodo, kad konvoliucinio neuroninio tinklo algoritmas kartu su kandidato regiono algoritmu veikia gerai tiek taikinio sekimo tikslumo, tiek algoritmo patvarumo atžvilgiu (Zhang et al., 2018).

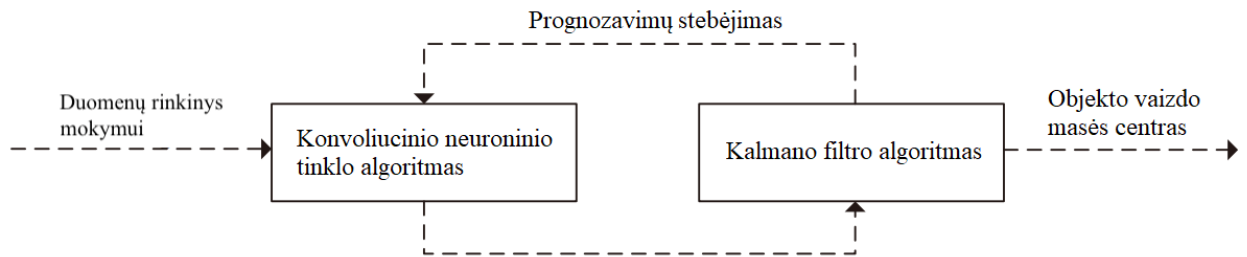
Kai regiono pasirinkimo algoritmas atlieka reikalingus skaičiavimus, vaizdo kadre pasirenkamas kandidato regionas, kuris gali būti sekamas taikiny. Tokiu būdu pateikiama konvoliucinio neuroninio tinklo įvestis. 1.10 paveiksle pavaizduota algoritmo blokinė schema.



1.10 pav. Regiono pasirinkimo algoritmo blokinė schema (sudaryta autorės, remiantis Zhang et al. (2018))

Išankstiniam požymių talpyklos mokymui naudojamas CIFAR-10 duomenų rinkinys, kurį sudaro 60 000 spalvotų vaizdų (32×32 dydžio), suskirstytų į 10 kategorijų: 50 000 pavyzdžių skirti mokymui ir 10 000 – testavimui (Zhang et al., 2018). Šis duomenų rinkinys yra pritaikomas daugybei klasifikacijų. Konvoliucinis neuroninis tinklas iš anksto išmokomas naudojant šį duomenų rinkinį, atsižvelgiant į vaizdų struktūrines charakteristikas talpykloje.

Konvoliucinio neuroninio tinklo sekimo algoritme dabartinio vaizdo kadro paieškos langas nustatomas pagal ankstesnio kadro rezultata. Sekimo procese neišvengiamai pasitaiko taikinio uždengimo atvejų. Naujas paieškos langas, apskaičiuotas pagal ankstesnį kadrą, gali būti toli nuo taikinio arba net gali būti nesusijęs su juo, dėl to netenkama taikinio ir padidėja klaidingai teigiamo taikinio aptikimo tikimybė (Zhang et al., 2018). Siekiant išspręsti šią problemą, gali būti naudojamas Kalmano filtras, skirtas išmokyto taikinio aptikimo modelio judėjimo būsenai stebėti (Zhang et al., 2018). Filtro optimalus taikinio padėties vaizdo plokštumoje įvertinimas laikomas CNN taikinio atpažinimo tašku kiekviename kadre. Tai atitinka taikinio vietą, naudojant taikinio padėties globalinį išankstinį įvertinimą. Galutinę padėtį nustato ir identifikuoja CNN algoritmo mokymo modelis, o filtro vėlesnė vertės reikšmė pakeičiama galutine kito kadro filtravimo įvertinimo padėtimi (Zhang et al., 2018). Dabartinė taikinio padėtis prognozuojama pagal judėjimo būsenos informaciją prieš taikinį. Kalmano filtras yra automatinio duomenų apdorojimo metodų optimizavimas – reikia gauti tik paskutinės akimirkos būsenos vertę. Taip pat, stebint būklę dabartiniu momentu, galima apskaičiuoti esamos būklės įvertinimą. Todėl nereikia įrašyti ar įvertinti praeities informacijos (Zhang et al., 2018). Konvoliucinio neuroninio tinklo kartu su Kalmano filtru taikinio aptikimo algoritmo blokinė schema pateikta 1.11 paveiksle.



1.11 pav. Patobulinto taikinio aptikimo algoritmo, pagrįsto konvoliuciniu neuroniniu tinklu ir Kalmano filtru, blokinė schema (sudaryta autorės, remiantis Zhang et al. (2018))

Taigi, Zhang et al. (2018) atkreipė dėmesį į judančių objektų sekimo sudėtingumą (šviesos pokyčiai, objekto uždengimas, greitas judėjimas, sudėtingas fonas) ir pasiūlė patobulintą Kalmano filtro integravimą į konvoliucinio neuroninio tinklo modelį algoritmą. Filtro optimalus taikinio padėties vaizdo plokštumoje įvertinimas laikomas CNN taikinio atpažinimo tašku kiekviename kadre. Tai prilygsta globaliniam išankstiniam taikinio padėties įvertinimui (Zhang et al., 2018). Galutinę padėtį nustato ir identifikuoja konvoliucinio neuroninio tinklo algoritmo mokymo modelis, o vertė po įvertinimo filtre pakeičiama galutine kito kadro filtravimo įvertinimo padėtimi. Šie du algoritmai naudojami pakaitomis ir papildo vienas kitą (Zhang et al., 2018).

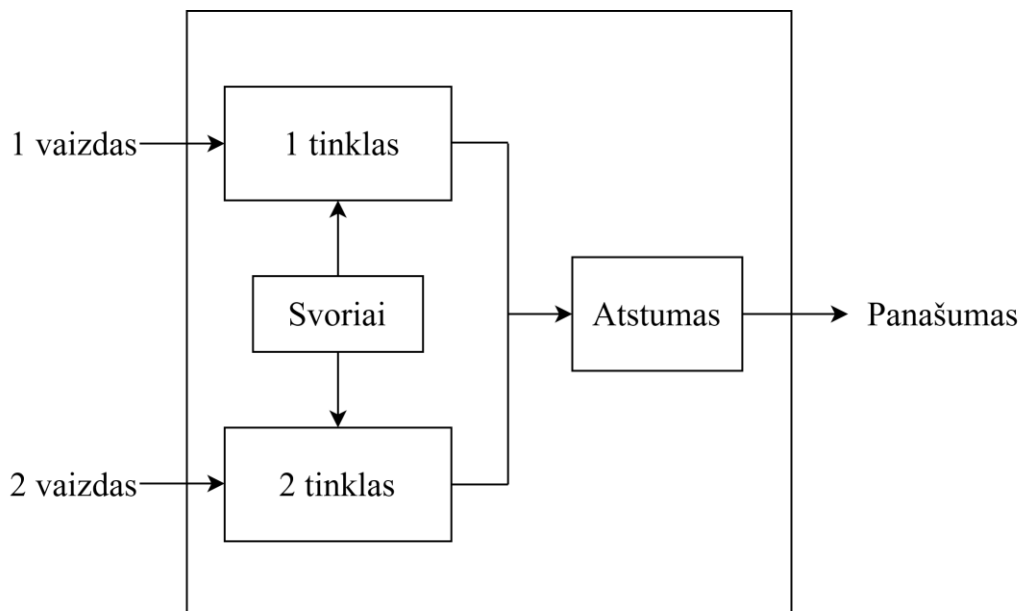
1.4.2. Taikinio sekimo algoritmas, pagrįstas tankiu paremta klasterizacija

Tradiciiniame Siamos tinkle (angl. *Siamese network*) taikinio sekimo algoritmui taikinio vietai nuskaityti reikia naudoti taikinio konvoliucijos požymius, numatant taikinio vietą kitame vaizdo kadre, ir atlikti panašumo skaičiavimus, kad būtų gauta panašumo vertinimo matrica su aukščiausiu įverčiu (tai kito kadro taikinio padėtis) (Chen et al., 2020). Aukščiausias panašumo įvertis dažnai neatspindi tikslios taikinio padėties, nes įtakos turi slinkimo žingsnio (angl. *sliding step*) dydis nuskaitymo metu. Siekiant išspręsti šią problemą, Chen et al. (2020) siūlo taikinio sekimo metodą, pagrįstą tankiu paremta klasterizacija.

Klasterizacija yra procesas, kurio metu duomenų objektų rinkinys padalijamas į vieną ar daugiau panašių grupių (klasterių), kad kiekvienas tos pačios grupės objektas būtų labai panašus į kitus to klasterio objektus, o skirtingų grupių objektai – labiau nepanašūs vienas į kitą (Chen et al., 2020). Pagal klasterio skalę klasterizacijos metodus galima suskirstyti į paremtus atstumu, sujungimu ir tankiu (Chen et al., 2020). Atstumu paremtas klasterizacijos metodas paprastai vertina taikinių panašumą pagal atstumą tarp taikinių. Tankiu paremtas klasterizacijos metodas dažniausiai naudoja tankio funkciją, taikinių panašumui nustatyti. Sujungimais paremti klasterizacijos metodai paprastai sujungia objektus į vieną klasę, remdamiesi grafu ar hipergrafu.

Siamos tinklas yra gyliu paremtas atitikmens matas (angl. *similarity measure*), kuris išmoka duomenų atitikmens matą, o paskui panaudoja gautas žinias naujų nežinomų kategorijų pavyzdžiams palyginti ir suderinti (Chen et al., 2020). Siamos tinklą sudaro du identiški tinklai, kurie dalijasi

svoriais (1.12 pav.). Sekimo proceso metu taikinio vaizdo konvoliucijos požymiai naudojami kito kadro vaizdo konvoliucijos požymiams nuskaityti pagal tam tikrą žingsnio dydį, tada apskaičiuojamas dabartinio taikinio vaizdo panašumas kiekviename kito kadro vaizde. Galiausiai taikinio padėtis nustatoma pagal panašumo vertinimo matricą.



1.12 pav. Siamo tinklas (sudaryta autorės, remiantis Chen et al. (2020))

Kaip parodyta 1.12 paveiksle, abu tinklai yra tos pačios struktūros neuroninių tinklų grupė, turinti bendrus svorius. Chen et al. (2020) naudoja konvoliucinį neuroninį tinklą. Tinklo įvestis yra du skirtingi vaizdai, o atitinkamas sąsūkos požymių atvaizdavimas gaunamas per neuroninio tinklo išvestį. Apskaičiuojant atstumą tarp dviejų vektorių, gaunamas vaizdų panašumo vektorius. Kuo panašesni abu vaizdai, tuo mažesnis bus atstumas ir tuo didesnis skirtumas (Chen et al., 2020). Atstumas apskaičiuojamas pagal (1.1) formulę:

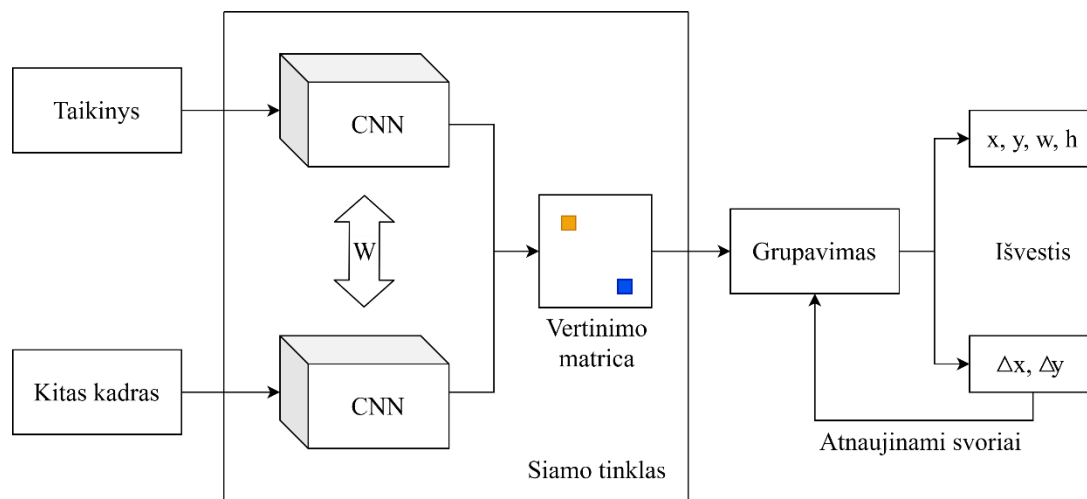
$$D_w(x_1, x_2) = ||G_w(x_1) - G_w(x_2)||, \quad (1.1)$$

čia: x_1 ir x_2 – dvi vienodo dydžio nuotraukos (tinklo įvestis); w – tinklo mokymo svoris; G_w – vektorius, gautas iš vaizdo per tinklą; D_w – dviejų tinklo išvesčių vektorius.

Taikant metodą, siūlomą Chen et al. (2020), judėjimo trajektorijos informacija prieš taikinį pridedama klasterizacijos proceso metu, todėl klasterizacija atliekama du kartus. Pirma yra sugrupuojama taikinio vertinimo matrica su taikinio trajektorijos informacija, kad būtų gauta tiksli taikinio vieta. Antra grupuojami taikinio plotis ir aukštis – gaunama tiksli informacija apie taikinio plotį ir aukštį.

Taigi, sujungiant Siamo tinklą, kad būtų galima numatyti kito kadro taikinio padėtį, ir pridedant informaciją apie taikinio judėjimo trajektoriją, taikinio judėjimo kryptiai suteikiama daugiau svorio, kitoms kryptims suteikiamas mažesnis svoris (Chen et al., 2020). Galiausiai taikinio padėtis numatoma pagal tankio grupavimo metodą.

Bendra Chen et al. (2020) metodo struktūra pateikta 1.13 paveiksle. Esamas taikinio vaizdas ir kito kadro taikinio vaizdas įvedami į tinklą. Siamo tinklas naudojamas taikinio panašumo vertinimo matricai kitame kadre apskaičiuoti. Tada, atsižvelgiant į judėjimo trajektoriją prieš taikinį, vertinimo matricai suteikiami atitinkami svoriai. Galiausiai, grupuojant taikinio padėtį ir plotį bei aukštį, naudojamas tankio klasterizacijos algoritmas, gaunama tiksli taikinio padėtis ir atnaujinama taikinio judėjimo krypties informacija.



1.13 pav. Bendra algoritmo, pagrįsto tankiu paremta klasterizacija, struktūra (sudaryta autorės, remiantis Chen et al. (2020))

Eksperimentiniai rezultatai rodo, kad siūlomas algoritmas pagerina Siamo tinklo taikinio vietos prognozavimo tikslumą, sekant taikinius (Chen et al., 2020).

1.4.3. Kelių taikinių sekimas, remiantis taikinių aptikimu ir bendrąja informacija

Norint sekti taikinį, reikia visiškai segmentuoti jį, išskirti požymius ir tiksliai jį identifikuoti. Taip pat būtina atsižvelgti į algoritmo įgyvendinimo laiką, kad būtų užtikrintas tikslumas ir sekimas realiuoju laiku.

Vykdant judėjimo aptikimo užduotį, rezultatui pagerinti dažnai naudojamas regioninio konvoliucinio neuroninio tinklo (angl. *Region-based Convolutional Neural Network* – RCNN) metodas. RCNN metodo idėja yra pirmiausia gauti kelis dominančius regionus, tada vienas jų įvedamas į konvoliucinį neuroninį tinklą. Tinklas išskiria jo giliuosius požymius ir naudoja regioną klasifikacijai (dominantis regionas – tai aptikta lokacijos informacija) (Zhang & Fang, 2020). Vėliau atsirado greitis ir greitesnysis RCNN (angl. *Fast RCNN*, *Faster RCNN*) taikinio aptikimo metodai, turintys didesnę aptikimo tikslumą.

Sąvoką „bendroji informacija“ (angl. *mutual information*) 1948 m. pasiūlė Shannon straipsnyje apie informacijos teoriją (Zhang & Fang, 2020). Kadangi tai remiasi tikimybių teorija ir matematine statistika, galima taikyti statistikos metodus. Dviejų vaizdų atveju tai yra statistinė koreliacija tarp šių

dviejų atsitiktinių kintamųjų. Kai du vaizdai pasiekia geriausią atitikimą, jų atitinkamų pikselių pilka bendroji informacija pasiekia maksimalią vertę (Zhang & Fang, 2020).

Zhang ir Fang (2020) naudoja greitesniojo RCNN tinklo struktūrą, siekdami suvienodinti kandidatų aptikimo regionų išskyrimą ir klasifikavimą į vieną konvoliucinio tinklo architektūrą. Kelių simbolių (angl. *multi-character*) mokymosi mechanizmas naudojamas tinklo parametrų mokymui užbaigti, kad būtų pagerinti fizinio aptikimo efektyvumas ir tikslumas. Registruojant vaizdus pagal bendrosios informacijos suderinimą, kai erdvėje užregistruojami du vaizdai, bendroji informacija apie pikselių porų pilkuosius lygius, atitinkančius jų sanklotos vietas (angl. *overlapping parts*), pasiekia didžiausią vertę. Pagal tai galima nustatyti geriausią vaizdo atitikties tašką, norint tiksliai sekti kelis taikinius.

RCNN aptikimo metodo procesas skirstomas į penkis etapus (Zhang & Fang, 2020):

1. Regionų prielaidų eilutės kūrimas, naudojant tam tikrus įrankius (pavyzdžiui, „EdgeBox“, „Selective Search“ ir kt.).

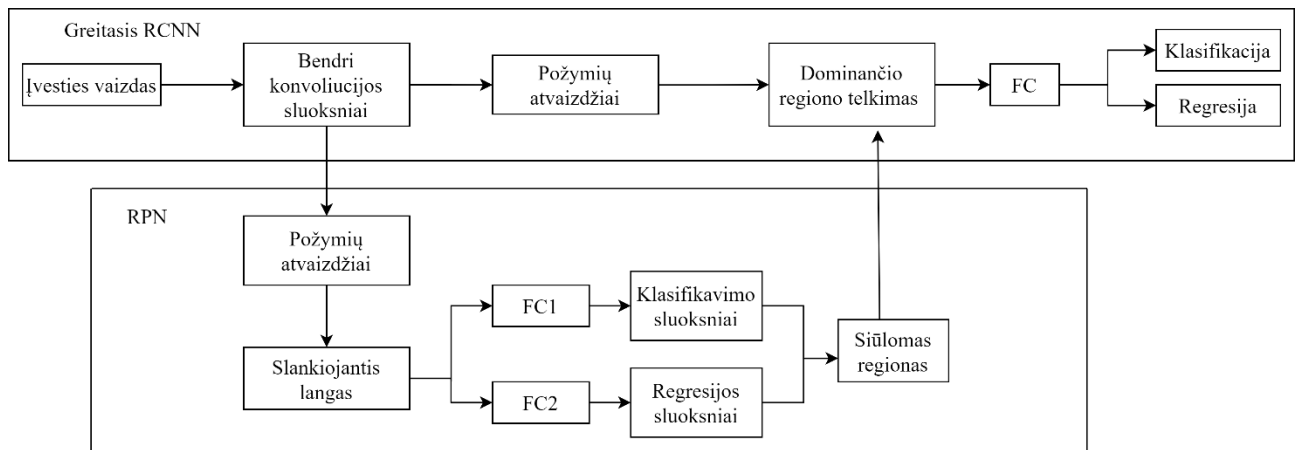
2. Naudojant CNN, paeiliui išskiriami sukurtų regionų prielaidų požymiai (tam gali būti naudojami „AlexNet“, „CaffeNet“ ir kt.).

3. Klasifikatorius (pavyzdžiui, SVM) naudojamas išskirtiems požymiams klasifikuoti į teigiamus ir neigiamus pavyzdžius.

4. Regresijos įtaisas naudojamas padėties regresijai atlikti srityje, klasifikuojamoje kaip priekinis planas.

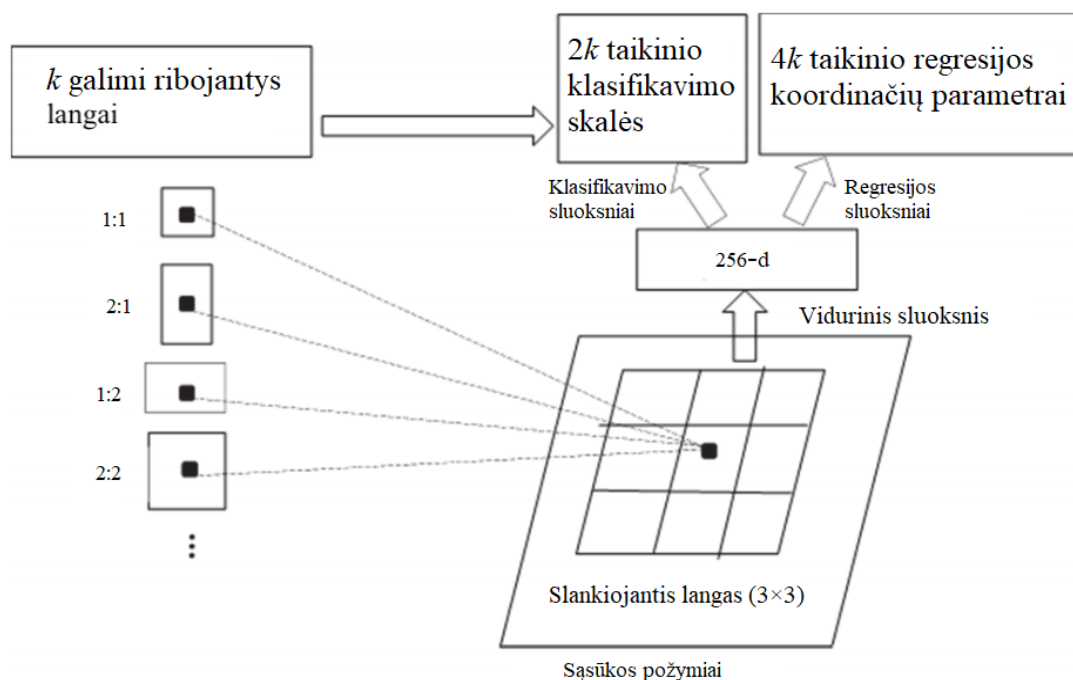
5. Galiausiai naudojamas ne-maksimalusis malšinimas (angl. *non-maximum suppression*), kad būtų pašalintas labai susiliejęs kandidato langas.

Greitesniojo RCNN algoritmo blokinė schema pateikta 1.14 paveiksle. Greitesniojo RCNN konvoliucinis neuroninis tinklas pirmiausia naudojamas įvesties vaizdo charakteristikoms gauti. Tada dalijamas būdingasis vaizdas paduodamas į greitojo RCNN konvoliucijos sluoksnį ir regioninį pasiūlymų tinklą (angl. *Region Proposal Network* – RPN). RPN tinklo įvesties charakteristikos vektoriai generuojami iš slankiojančio lango (angl. *sliding window*), prijungiant jį prie viso sluoksnio (FC1, FC2). Vektoriaus išvestis yra du sluoksniai, sujungiami tame pačiame lygyje (Zhang & Fang, 2020). Aplinkos lango regresijos sluoksnis ir aplinkos lango klasifikavimo sluoksnis pateikia informaciją apie dominančio regiono lokaciją ir pasiklovimo informaciją apie atitinkamą regioną, kuriame taikiny yra arba jo nėra, ir tada sukuria klasifikavimo ir regresijos pasiūlymų regioną. Taip pat per konvoliucinį neuroninį tinklą gaunamas požymių atvaizdis (angl. *feature map*). Pasiūlymo regionas ir požymių atvaizdis siunčiami į dominančio regiono telkimą (angl. *the pool of the region of interest*), kad būtų nustatytas konkretus dydis (Zhang & Fang, 2020). Tada vyksta klasifikacija ir regresija per dvi momentines funkcijas, kad būtų sukurta numatoma koordinačių vertė ir gautas taikinio aptikimo langas.



1.14 pav. Greitesniojo RCNN algoritmo blokinė schema (sudaryta autorės, remiantis Zhang ir Fang (2020))

Regioninis pasiūlymų tinklas yra visiškai konvoliucinis tinklas (1.15 pav.). RPN struktūroje vaizdas yra greitesniojo RCNN tinklo įvestis, kad paskleistų išvesties sluoksnį bendrojo konvoliucijos sluoksnio požymių atvaizdžio pabaigoje. Sluoksnis naudoja slankiojantį langą (3×3 dydžio) požymių atvaizdžio sąsūkos operacijai atlikti ir kartotiniams 256 vektoriams išvestyje gauti, kurie visiškai sujungiami su klasifikavimo sluoksniais ir regresijos sluoksniais (Zhang & Fang, 2020). Slankiojantis langas numato k srities pasiūlymus (trys skalės yra 128^2 , 256^2 ir 512^2 ir trys proporcijos yra 1:1, 1:2, 2:1). Regresijos sluoksnis turi numatyti aplinkos sritį – lango plotį ir centro koordinatės. Klasifikavimo sluoksnyje nuo priekinio plano ar fono atskiriama siūlomo lango tikimybė. Todėl regresijos sluoksnyje yra $4k$ išvestys, t. y. k taikinį ribojančių langų koordinatinis kodavimas, o klasifikavimo sluoksnio išvestys – $2k$ skalės (Zhang & Fang, 2020). Langas nurodo apskaičiuotą taikinio buvimo/nebuvimo tikimybę.



1.15 pav. RPN tinklas ir jo veikimo procesas (sudaryta autorės, remiantis Zhang ir Fang (2020))

Greitesnysis RCNN – tai kartu sujungtų RPN ir greitojo RCNN struktūra. Greitesnysis RCNN taip pat dalijasi RPN ir greitojo RCNN konvoliucijos sluoksnių parametrais ir palaiko ištisinį mokymą (angl. *end-to-end training*). Klasifikavimo tikslumas yra panašus į greituoju RCNN metodu gaunamą tikslumą. RPN naudojimas galimų regionų gavimui žymiai pagerina metodo greitį (Zhang & Fang, 2020).

Taikinio sekimo žingsniai, remiantis greitesniu RCNN tinklu ir bendrąja informacija (Zhang & Fang, 2020):

1. RPN tinklo mokymas. Tinklas pirmiausia inicijuojamas naudojant iš anksto išmokytą „ImageNet“ modelį, o tada taikomas ištisinis mokymas ir tiksliai suderinami tinklo parametrai.

2. Atskiras greitesniojo RCNN tinklo modulio mokymas. Inicijavimo modelyje taip pat naudojamas iš anksto išmokytas „ImageNet“ modelis, o dominančią sritį, įvestą į tinklo mokymą, gauna ir pateikia RPN tinklas, išmokytas 1 žingsnyje.

3. Mokymo pradžioje atsižvelgiama į bendrus RPN ir greitesniojo RCNN du konvoliucijos sluoksnius. Siekiant inicijuoti RPN, naudojamas 2 žingsnyje išmokytas tinklas. RPN tinklui mokyti bendrasis konvoliucijos sluoksnis nekeičiamas, tik tiksliai suderinama tinklo sluoksnio dalis, būdinga RPN.

4. Greitojo RCNN tinklo mokymas. Bendra konvoliucijos sluoksnio dalis lieka nepakitusi. Reikia tiksliai suderinti tinklo sluoksnio dalį, būdingą greitajam RCNN tinklui. Šiame žingsnyje baigiamas visas greitesniojo RCNN mokymo procesas.

5. Išmokytas greitesniojo RCNN tinklas naudojamas informacijai apie taikinį vaizdų sekoje numatyti, kiekvieno taikinio padėčiai pirmajame kadre aptikti ir atsitiktiniu būdu taikinio eilės numeriui pažymėti.

6. Pradedant nuo antrojo kadro, pirmiausia aptinkama taikinio pozicija vaizde, naudojant greitesnį RCNN tinklą, tada apskaičiuojama bendroji informacija su taikiniu, aptiktu ankstesniame kadre. Taikiny, turintis maksimalią bendrąją informaciją, yra tas pats objektas. Siekiant tikslaus sekimo, taikinio, atitinkančio ankstesnį kadrą, eilės numeris pažymimas kaip esamas eilės numeris.

Klasikiniai sekimo algoritmai yra hierarchinių konvoliucijos požymių (angl. *Hierarchical Convolutional Features* – HCF) algoritmas, visiškai konvolucinių Siamo tinklų (angl. *Fully-Convolutional Siamese Networks* – SiamFC) algoritmas, ištisinio srauto koreliacijos sekimo struktūra (angl. *end-to-end flow correlation tracking framework* – FlowTrack):

- HCF algoritme naudojama iš trijų sluoksnių gauta išvestis, užuot naudojant paskutinio viso konvoliucijos sluoksnio išvestį kaip požymių išskyrimo sluoksnį (Zhang & Fang, 2020). Požymiai mokomi naudojant koreliacijos filtrus, siekiant gauti skirtingus šablonus. Tada gauti trys patikimumo atvaizdžiai yra įvertinami ir gaunama galutinė taikinio padėtis.

- *SiamFC* algoritmas moko Siamo tinklą, kuris ieško vaizdų šablonų didelėje paieškos srityje (Zhang & Fang, 2020). Šis Siamo tinklas yra visiškai konvoliucinis tinklas, esantis aplink paieškos sritį, o galutinės taikinio padėties įvertinimas apskaičiuojamas abipuse dviejų įvesčių koreliacija (angl. *cross-correlation*). Tada atliekama tanki ir efektyvi interpoliacija.

- *FlowTrack* algoritmas naudoja optinį srauto įvertinimą, požymių išskyrimą, konvergavimą ir koreliacijos filtrų sekimą kaip specialius tinklo sluoksnius, kad būtų pasiektas ištisinis mokymasis (Zhang & Fang, 2020). Tada srauto informacija nukreipiama, kad ankstesnis kadras būtų perkreiptas į nurodytą kadrą ir jie būtų sujungti, toliau atliekami koreliacijos filtravimas ir sekimas.

Zhang ir Fang (2020) palygino HCF, *SiamFC*, *FlowTrack* ir jų siūlomo algoritmo (remiantis greitesniu RCNN tinklu ir bendrąja informacija) veikimą (1.1 lent.).

1.1 lentelė. Keturių algoritmų sekimo rezultatų palyginimas (sudaryta autorės, remiantis Zhang ir Fang (2020))

Sekiklis	Laukiamoji sanklotos reikšmė	Tikslumas	Patvarumas
HCF	0,257	0,541	0,337
<i>SiamFC</i>	0,277	0,549	0,382
<i>FlowTrack</i>	0,334	0,578	0,241
Zhang ir Fang (2020) siūlomas algoritmas	0,331	0,581	0,238

Lyginant Zhang ir Fang (2020) siūlomą sekimo algoritmą su trimis klasikiniiais sekimo algoritmais (1.1 lent.), siūlomas algoritmas yra pranašesnis už HCF ir *SiamFC* pagal laukiamąją sanklotos reikšmę (angl. *Expected Average Overlap* – EAO), o *FlowTrack* algoritmo EAO vertė yra nežymiai didesnė. Pagal gautus rezultatus siūlomas algoritmas yra geriausias sekimo tikslumo požiūriu (Zhang & Fang, 2020).

1.5. Vieno etapo algoritmai

Esamus metodus, kuriuose naudojamas gilusis mokymasis objektų klasifikacijai ir regresijai, galima suskirstyti į dvi kategorijas (Tan et al., 2021):

1) Dviejų etapų algoritmai (pavyzdžiui, anksčiau aptarti RCNN, greitis ir greitesnysis RCNN). Šiuose algoritmuose pirmiausia naudojama atrankioji paieška arba regioninis pasiūlymų tinklas (RPN), kad būtų gautas pasiūlymo regionas, o tada atliekama to pasiūlymo regiono klasifikacija ir regresija. Šie metodai pasižymi dideliu tikslumu, tačiau apriboja aptikimo greitį (Tan et al., 2021).

2) Vieno etapo algoritmai (pavyzdžiui, toliau šiame poskyryje pristatyti YOLO ir SSD). Tai pagrįsti regresija ištisiniai objektų aptikimo ir atpažinimo algoritmai, kurie naudoja vieną tinklą objektą ribojančiam langui ir kategorijos tikimybės reikšmei numatyti tiesiai iš vaizdo. Kadangi šis

algoritmas nenaudoja RPN, yra pagerinamas aptikimo greitis (Tan et al., 2021). Tačiau dviejų etapų algoritmas greičiau aptinka mažus taikinius.

Objektų atpažinimo galimybė tiesiogiai priklauso nuo aptikimo tikslumo ir aptikimo greičio (Tan et al., 2021). Nors taikant dviejų etapų metodus buvo pasiektas didelis tikslumas, šių metodų skaičiavimo greitis paprastai yra mažesnis nei vieno etapo metodų. Todėl YOLO (angl. *You Only Look Once*) metodas yra plačiai taikomas įvairiose praktinėse užduotyse, ypač lengvose platformose.

YOLO algoritmas naudoja konvoliucinius neuroninius tinklus (CNN), kad greitai identifikuotų objektus. Norint aptikti objektus, šiam metodui reikia tik vieno perdavimo per neuroninį tinklą. Per pastaruosius kelerius metus atlikta daug mokslinių tyrimų, susijusių su YOLO algoritmais (Gillani et al., 2022), ir buvo įrodyta, kad šis algoritmas tinka objektams aptikti realiuoju laiku.

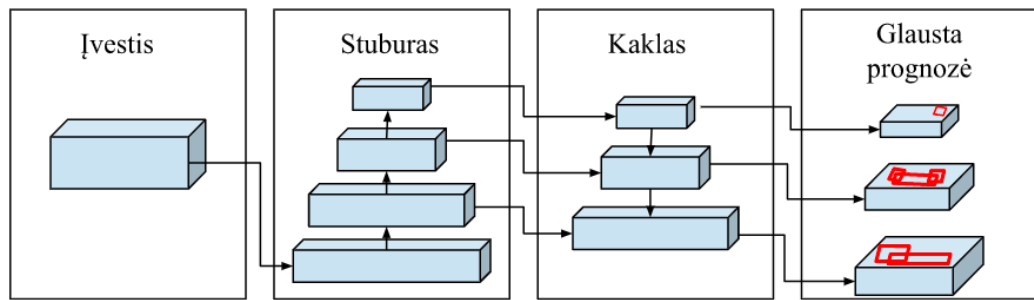
Pirmosios kartos YOLO modelis buvo sukurtas 2015–2016 m.. Šis modelis aptikimo užduotį paverčia matematinės regresijos problema (Kong et al., 2022). Tais pačiais metais buvo pateiktas ir SSD (angl. *Single Shot Multibox Detector*) modelis, derinantis skirtingų skalių požymius su numatytais ribojančiais langais (Kong et al., 2022). YOLO šeimos algoritmai vystosi labai sparčiai, palaipsniui pridedant daugiau metodų, dėl to jų aptikimo tikslumas gerėja.

Pavyzdžiui, „YOLOv3“ yra plačiai taikomas YOLO šeimos algoritmas, nes buvo atrastas kompromisas tarp aptikimo greičio ir aptikimo tikslumo. „YOLOv3“ naudoja požymių piramidės tinklą (angl. *feature pyramid network*), kuris gali numatyti taikinį iš trijų skirtingų skalių (Kong et al., 2022). Tokiu būdu pagerinamos mažų taikinių aptikimo galimybės.

Testavimo procesų metu buvo nustatyta, kad „YOLOv3“ nėra pakankamai efektyvus kai kurių scenarijų atvejais, ir vis dar yra nemažai klaidingų aptikimų (Kong et al., 2022). Todėl taikinių aptikimo technologijų tyrimuose daug dėmesio yra skiriama ir dideliems konvoliuciniams neuroniniams tinklams. Tačiau didelio masto tinklus sunku įdiegti įrenginiuose, kuriuose yra nedaug aparatinės įrangos išteklių (Kong et al., 2022). Taip pat mažesnis per vieną sekundę apdorotų kadro skaičius (angl. *Frames Per Second* – FPS) ir dideli vėlavimai tampa svarbiais veiksniais, ribojančiais tokių tinklų praktinį taikymą. Taigi, mokslininkų tyrimai rodo, kad „YOLOv3“ tinklas (ir kiti šios šeimos algoritmai) gali būti patobulintas, siekiant pagerinti aptikimo našumą pagal tam tikro scenarijaus reikalavimus (Kong et al., 2022).

Vieno etapo algoritmų architektūra susideda iš trijų komponentų (1.16 pav.):

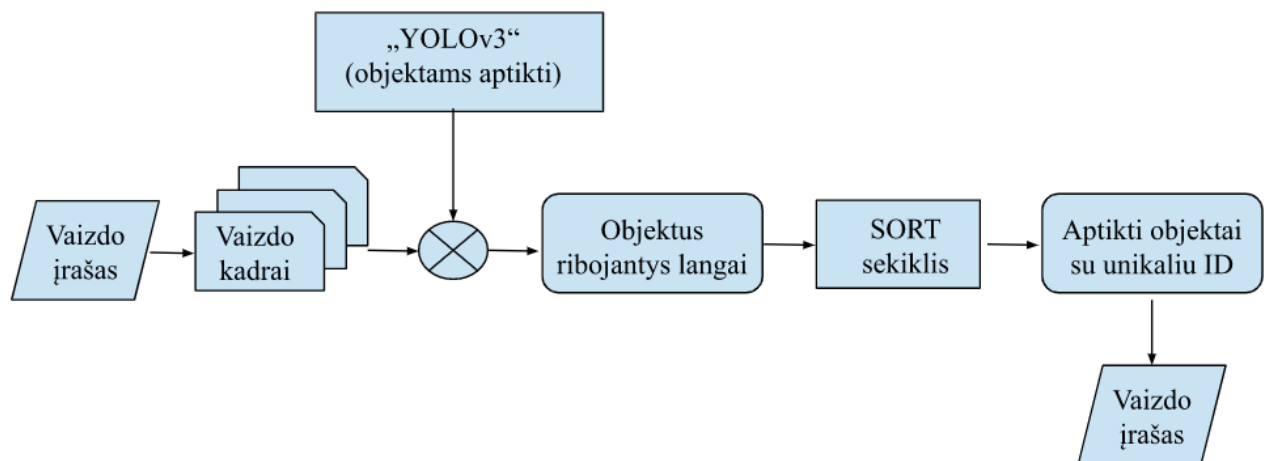
- stuburo (angl. *backbone*),
- kaklo (angl. *neck*),
- galvos (angl. *head*).



1.16 pav. Vieno etapo algoritmų architektūra (sudaryta autorės, remiantis Bochkovskiy et al. (2020))

Modelio „stuburas“ yra iš anksto išmokytas tinklas, naudojamas vaizdų išsamiems požymiams išskirti. Tai padeda sumažinti vaizdo erdvinę skiriamąją gebą ir padidinti jo požymių (kanalo) skyrą. Modelio „kaklas“ naudojamas požymių piramidėms išskirti. Tai padeda modeliui gerai apibendrinti skirtingų dydžių ir mastelių objektus. Modelio „galva“ naudojama paskutinės stadijos operacijoms atlikti. Ji pritaiko inkaro (angl. *anchor*) langus požymių atvaizdžiuose ir pateikia galutinį rezultatą: klases, pasiklivimo įverčius ir ribojančius langus.

Bathija ir Sharma (2019) siūlo vizualinio objektų sekimo metodą (1.17 pav.), kuriame naudojami YOLO tinklas (objektams aptikti) ir SORT (angl. *Simple Online Realtime Tracking*) algoritmas (objektams sekti). „YOLOv3“ tinklui mokyti buvo naudojamas paruoštas duomenų rinkinys, kurį sudaro 800 vaizdų (išskirtos 6 klasės).



1.17 pav. Vizualinio objektų aptikimo ir sekimo algoritmo schema (sudaryta autorės, remiantis Bathija ir Sharma (2019))

Objektams aptikti naudojamas YOLO tinklas generuoja ribojančius langus su unikaliu klasės identifikavimu (angl. *identification* – ID) ir kiekvieno ribojančio lango pasiklivimo (angl. *confidence*) įverčiu. Kadangi šiame algoritme laikomasi sekimo pagal aptikimą metodo, YOLO aptikimai persiunčiami į SORT sekiklį (Bathija & Sharma, 2019).

SORT metodas naudoja Vengrų algoritmą ir Kalmano filtrą aptiktiems objektams sekti (Bathija & Sharma, 2019). SORT seka kiekvieną aptikimą, priskirdamas unikalų ID kiekvienam

ribojančiam langui. Kai objektas prarandamas, pavyzdžiui, dėl jo už dengimo ar neteisingo identifikavimo, sekiklis priskiria naują ID ir pradeda sekti naujai rastą objektą.

Po YOLO tinklo mokymo gauti svoriai naudojami objektams aptikti vaizdo įrašuose. Įvesties vaizdo įrašas yra skirstomas į kadrus, ir kiekvienas vaizdo kadras perduodamas į išmokytą tinklą. Atlikus aptikimą, ribojančio lango informacija perduodama SORT sekimo algoritmui ir atliekamas objekto sekimas. Gauti rezultatai pateikti 1.2 lentelėje ir 1.18 paveiksle.

1.2 lentelė. Kiekybinė siūlomos sistemos analizė (sudaryta autorės, remiantis Bathija ir Sharma (2019))

Vaizdo įrašas	Kadrų skaičius	Tikslumas	Preciziškumas	Jautrumas
1	812	0,794	0,843	0,934
2	930	0,851	0,958	0,93
3	1 160	0,75	0,9	0,9
4	835	0,781	0,833	0,892
5	590	0,444	0,5	0,889



1.18 pav. Kokybinė siūlomos sistemos analizė (Bathija & Sharma, 2019)

Kaip galima matyti 1.18 paveiksle, siūlomas algoritmas aptinka pėsčiuosius ir transporto priemones, o ribojantys langai skirtinguose vaizdo sekos kadruose lieka tos pačios spalvos (su tuo pačiu ID), tai reiškia, kad sekamas objektas nebuvo aptiktas kaip kitas objektas kitame kadre. Tikslumą ir preciziškumą galima pasiekti išmokant modelį daugiau epochų ir tiksliai derinant YOLO tinklo parametrus (Bathija & Sharma, 2019). SORT sekimo priemonės našumas visiškai priklauso nuo tinklo veikimo, nes sekiklis gauna duomenis apie aptiktą objektą iš YOLO tinklo.

1.6. Bepiločių orlaivių aptikimo metodai

Bepiločių orlaivių aptikimas yra plačiai tiriamos objektų aptikimo problemos dalis. Dronų aptikimas gali būti atliekamas naudojant radarą, garso, vaizdo ir radijo dažnių technologijas. Radaras

yra tradicinis ir efektyvus antidroninių sistemų įrankis. Tačiau jam įdiegti reikia brangių įrenginių, ir šis būdas gali būti netinkamas perpildytose miesto vietose arba ten, kur yra sudėtingas fonas, nes nedideliame aukštyje sunku atskirti dronus nuo sudėtingų fonų (Farlik et al., 2017). Daugelyje tyrimų dronams aptikti naudojamas akustinis signalas (drono atvykimo krypties absoliuti įvertinimo paklaida, gauta tyrimo metu, buvo ne didesnė kaip 6° (Yang et al., 2018)). Be akustinio metodo, objektams aptikti ir lokalizuoti buvo naudojama sistema, pagrįsta priimto radijo dažnio signalo stiprumu. Tačiau aplinkos triukšmas lengvai paveikia akustinį aptikimą (Azari et al., 2018). Be to, sprendimai, kuriuose naudojamas mašininis mokymasis arba gilusis mokymasis, sulaukė vis didesnio dėmesio dėl dirbtinio intelekto plėtros. Palaipsniui giliojo mokymusi pagrįsti sprendimai tobulėjo, bet vis reikalavo didelės skaičiavimo galios (Sharma et al., 2022). Giliojo mokymosi metodai dominuoja dabartiniuose pažangiausiuose objektų aptikimo metoduose, tačiau tik kai kurie metodai yra tinkami naudoti realiuoju laiku.

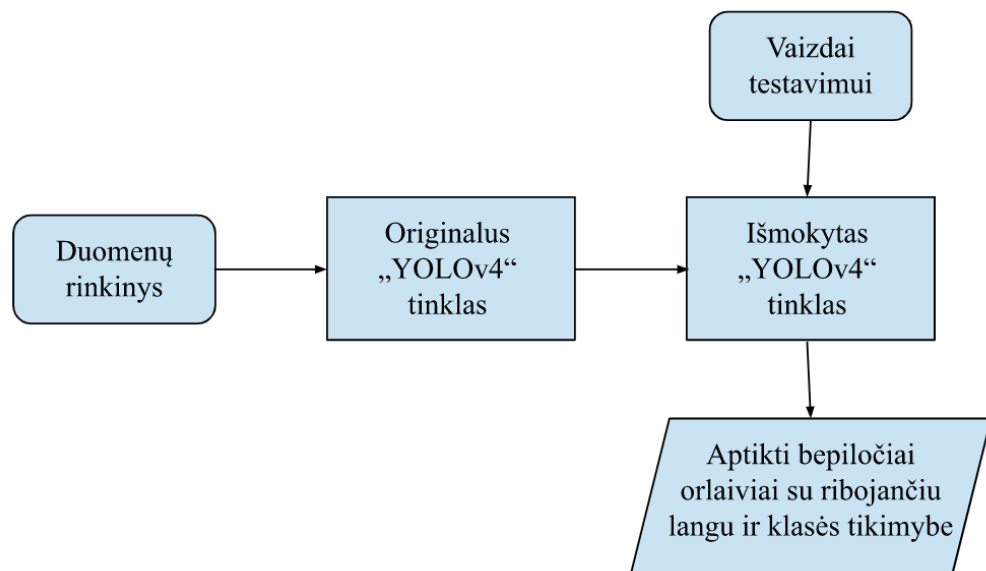
Bepiločių orlaivių atpažinimo sudėtingumas, pavyzdžiui, keičiantis fonas, dronų panašumas į paukščius, mažesnio dydžio dronai, didesni atstumai iki objektų, vaizdo apšvietimo problemos, reikalauja naujų sprendimų, todėl mokslininkai vis kuria naujus ir tobulina esamus algoritmus.

Dėl mažo drono dydžio aptikimo metodai paprastai yra nepatikimi. Didelis klaidingų aptikimų skaičius, mažas tikslumas ir mažas aptikimo greitis yra dažniausiai pasitaikantys šios užduoties iššūkiai. Vienas iš siūlomų naujų dronų aptikimo algoritmų, veikiančių realiuoju laiku, yra pagrįstas patobulintu „YOLOv3“ algoritmu (Alsanad et al., 2022). Šis algoritmas susideda iš kelių fazių ir parodė, kad gali pranokti tradicinius aptikimo būdus. Siūlomas algoritmas tobulina tinklo našumą, pagerinant ryšį tarp CNN sluoksnių ir naudojant kelių skalių (angl. *multiple-scale*) aptikimą, siekiant padidinti mažų objektų, tokių kaip dronai, aptikimo tikslumą (Alsanad et al., 2022).

Tinklui mokytį buvo naudojamas duomenų rinkinys, sudarytas iš 5 000 vaizdų (Alsanad et al., 2022). Po „YOLOv3“ algoritmo ir siūlomo algoritmo mokymų, naudojant paruoštą duomenų rinkinį, buvo palyginti rezultatai: originalaus tinklo gautas aptikimo tikslumas yra 92,35 %, o siūlomo tinklo, pagrįsto „YOLOv3“ algoritmu, gautas tikslumas lygus 95,6 % (Alsanad et al., 2022). Taip pat originalaus tinklo mokymas užtruko ilgiau nei Alsanad et al. (2022) siūlomo modelio mokymo laikas. Taigi, Alsanad et al. (2022) atliktas tyrimas patvirtino „YOLOv3“ galimybes ir efektyvumą, aptinkant bepiločius orlaivius vaizduose.

Singha ir Aydin (2021) siūlomoje dronų aptikimo sistemoje naudojamas „YOLOv4“ algoritmas. Dėl paukščių panašumų į dronus modeliui mokytį buvo naudojamas dronų (1 916 vaizdų) ir paukščių (479 vaizdų) duomenų rinkinys.

YOLO algoritmas pranoko kitus objektų aptikimo algoritmus, tokius kaip RCNN ir SSD, tiksliai aptinkant objektus realiuoju laiku (Singha & Aydin, 2021). Eksperimento schema yra pateikta 1.19 paveiksle.



1.19 pav. Eksperimento schema (sudaryta autorės, remiantis Singha ir Aydin (2021))

Mokymo metu paruoštas duomenų rinkinys buvo pateiktas kaip „YOLOv4“ tinklo įvestis. Kai kurie originalaus tinklo parametrai buvo pakeisti (pavyzdžiui, tinklas turi aptikti dronus ir paukščius, todėl klasių skaičius buvo pakeistas į 2). Gauti geriausi svoriai buvo išsaugoti ir testavimui skirti vaizdai buvo pateikti kaip išmokyto tinklo įvestis. Išmokyto tinklo išvestis yra aptikti objektai su ribojančiu langu ir klasės tikimybe (1.20 pav.).



1.20 pav. Dronų ir paukščių aptikimo rezultatas su ribojančiais langais ir klasių tikimybėmis (Singha & Aydin, 2021)

Didžiausias per 4 000 iteracijų pasiektas mAP metrikos įvertis buvo 74,36 %, sistemos testavimo metu vieno kadro apdorojimas truko 20,5 ms pirmam vaizdo įrašui ir 19 ms antram vaizdo įrašui (Singha & Aydin, 2021).

Tiwari (2022) siūlomas giliojo mokymosi metodas, naudojant „YOLOv5“ tinklą, gali aptikti ir atpažinti dviejų tipų dronus (daugiarotorius ir vieno rotoriaus) ir atskirti juos nuo paukščių ir

sraigtasparnių. Šiame darbe tinklui mokytui naudojamas duomenų rinkinys, kurį sudaro 4 000 vaizdų (1.21 pav.).



1.21 pav. Duomenų rinkinio vaizdų pavyzdžiai (Tiwari, 2022)

Šio modelio pasiektas preciziškumas po 300 epochų yra 0,976, jautrumas – 0,98, mAP50 įvertis – 0,98 ir mAP50–95 – 0,62 (Tiwari, 2022).

Taigi, dėl panašaus dronų ir paukščių elgesio bei išvaizdos danguje, taip pat dėl didelio jų greičio, fono sudėtingumų, uždengimo, bepiločių orlaivių dydžių, esant dideliems atstumams, mokslininkai siūlo naujus, giliuoju mokymusi pagrįstus, bepiločių orlaivių aptikimo ir atpažinimo metodus. YOLO šeimos algoritmai pasižymi tikslu aptikimu realiuoju laiku, todėl pastaruoju metu yra dažnai naudojami dronams aptikti.

1.6.1. Bepiločių orlaivių aptikimas, taikant konvoliucinio neuroninio tinklo metodus

Bepiločių orlaivių aptikimas, pagrįstas vaizdo apdorojimu, pastaraisiais metais tapo vis populiaresnis. Atsiradus ir labai išaugus giliojo neuroninio tinklo (angl. *Deep Neural Network* – DNN) taikymui įvairiose srityse, šiuo metu sprendžiama daug problemų, kurių nebuvo galima išspręsti anksčiau. Ypač gilusis CNN (DCNN) yra plačiai naudojamas įvairioms užduotims, susijusioms su vaizdais, atlikti (Liu et al., 2021).

Liu et al. (2021) naudoja CNN metodus aukštam dronų aptikimo tikslumui pasiekti. Šiame tyrime buvo vertinami keturi DCNN: „RetinaNet“, visiškai konvoliucinis vienos pakopos (angl. *Fully Convolutional One-Stage* – FCOS) metodas, „YOLOv3“ ir „YOLOv4“.

Dronų aptikimą realiuoju laiku daugiausia riboja dvi problemos. Viena iš jų – dronai juda greitai, todėl reikalingi greitesni detektoriai. DCNN turi itin gilųjį sluoksnį, dėl to tinklui reikia daug laiko apskaičiuoti milijonus parametrų, kad pateiktų rezultatą. Taigi, siekiant aptikti dronus realiuoju laiku, Liu et al. (2021) sumažino parametrų ir skaičiavimo operacijų skaičių.

Kita dronų aptikimo problema kyla dėl to, kad dronų pikseliai vaizde užima itin mažą dalį. Dėl šios priežasties detektorius sudėtinga atpažinti dronus. Be to, tarp mažų ir didelių objektų aptikimo yra didelis CNN veikimo skirtumas. Todėl, siekdami pagerinti mažų bepiločių orlaivių aptikimo tikslumą ir sumažinti prarandamą modelio tikslumą (dėl panaikintų sluoksnių), Liu et al. (2021) įdiegė specialų mažų objektų duomenų papildymą (kopijuojant ir įklijuojant mažus dronus vaizduose, kuriuose jų yra). Šis papildymas padidina mažų dronų skaičių kiekviename vaizde ir pateikia mažų dronų padėties įvairovę.

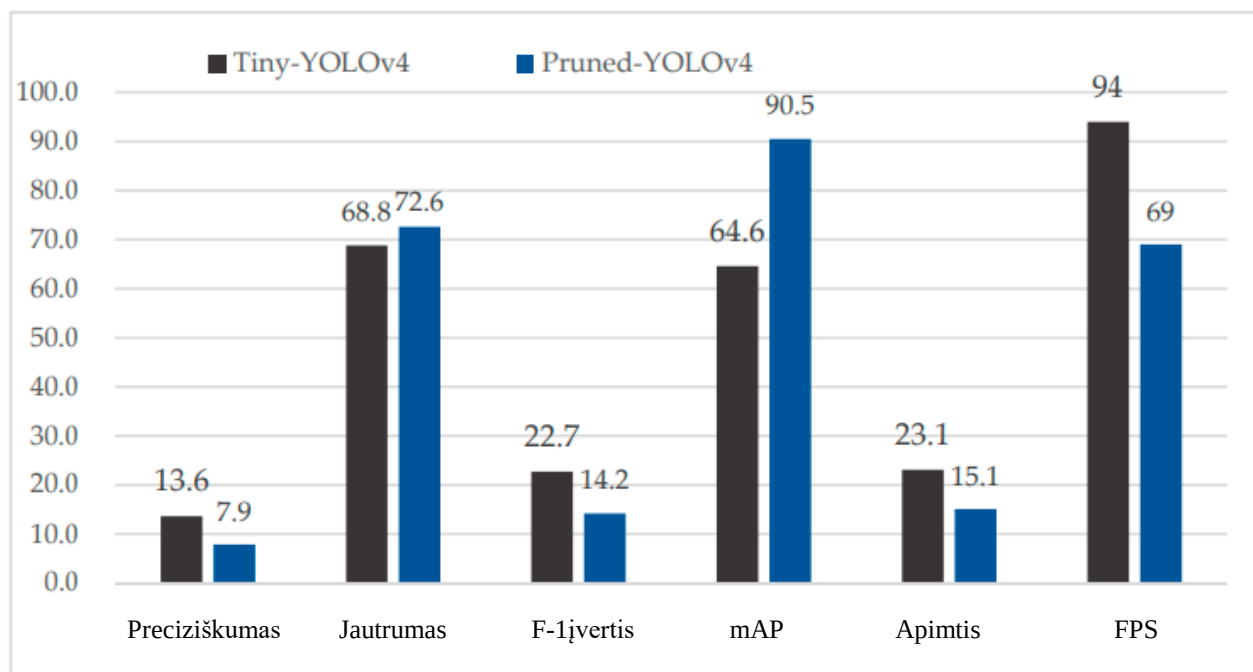
Šiame darbe autoriai naudoja preciziškumo, jautrumo ir mAP metrikas kiekvieno metodo veikimui įvertinti (Liu et al., 2021). Gauti rezultatai pateikti 1.3 lentelėje.

1.3 lentelė. Modelių veikimo rezultatų palyginimas (sudaryta autorės, remiantis Liu et al. (2021))

Modelis	Preciziškumas	Jautrumas	mAP
„ResNet-50“	12,9	94,9	85,5
„ResNet-101“	26,7	78,6	90,3
„RetinaNet“	68,5	91,7	90,5
„YOLOv3“	61,7	91,5	89,1
„YOLOv4“	74,2	93,1	93,6

Kaip galima matyti iš pateiktų rezultatų (1.3 lent.), palyginus FCOS (su „ResNet-50“ ir „ResNet-101“ modelio „stuburu“), „RetinaNet“ ir YOLO modelius, „YOLOv4“ našumas visais aspektais yra geresnis nei kitų algoritmų. Po atlikto skirtingų modelių mokymo ir rezultatų palyginimo originalus „YOLOv4“ tinklas buvo pakeistas, sumažinant sluoksnių ir parametrų skaičių (Liu et al., 2021).

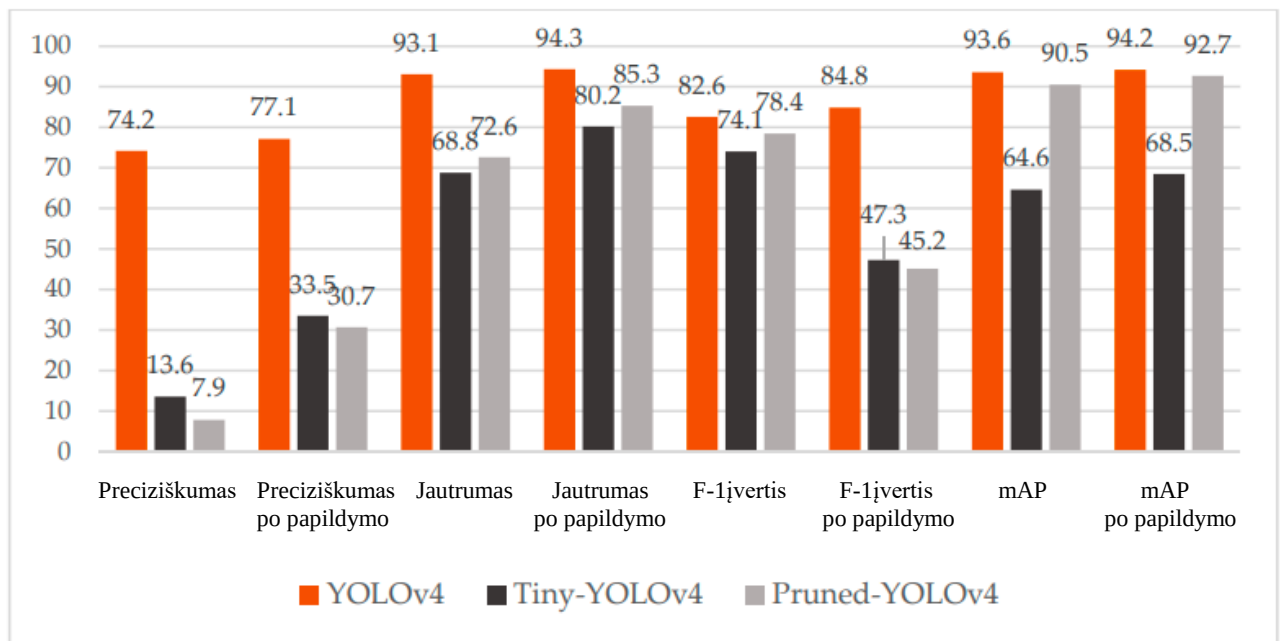
Sukurto tinklo veikimas buvo palygintas su „Tiny-YOLOv4“ tinklu (labai supaprastinta „YOLOv4“ versija). Gauti rezultatai pateikti 1.22 paveiksle.



1.22 pav. „Tiny-YOLOv4“ ir „apkarpyto“ „YOLOv4“ modelių našumo palyginimas
(Liu et al., 2021)

„Apkarpytų“ modelių trūkumai yra akivaizdūs: tikslumo, atšaukimo ir F1-įverčio vertės ženkliai sumažėjo (Liu et al., 2021). Pavyzdžiui, preciziškumo vertė sumažėjo nuo 74,2 % iki 7,9 %, lyginant su originalaus „YOLOv4“ tinklo gautais rezultatais po mokymo. Kuo mažesnis tikslumas, tuo daugiau klaidingų aptikimo langų pateiks modelis. Taip pat jautrumas sumažėjo nuo 93,1 % iki 72,6 %. Mažesnis jautrumas rodo, kad didėja neaptiktų bepiločių orlaivių tikimybė. „Tiny-YOLOv4“ turi nedidelį pranašumą preciziškumo ir F1-įverčio atžvilgiu (1.22 pav.). Taip pat dėl mažesnio sluoksnių skaičiaus „Tiny-YOLOv4“ yra pranašesnis FPS atžvilgiu. Tačiau Liu et al. (2021) siūlomas „YOLOv4“ modelis turi didelį pranašumą prieš „Tiny-YOLOv4“ mAP atžvilgiu.

Straipsnio autoriai (Liu et al., 2021) daro prielaidą, kad pagrindinė atsiradusių problemų priežastis yra tai, kad „apkarpytas“ modelis sunkiai aptinka mažus objektus. Todėl buvo įdiegtas mažų objektų duomenų papildymas mažų dronų aptikimui pagerinti. Papildyti vaizdai pakeičia originalius ir yra saugomi mokymui skirtame duomenų rinkinyje. Po duomenų papildymo rezultatai žymiai pagerėjo (1.23 pav.).



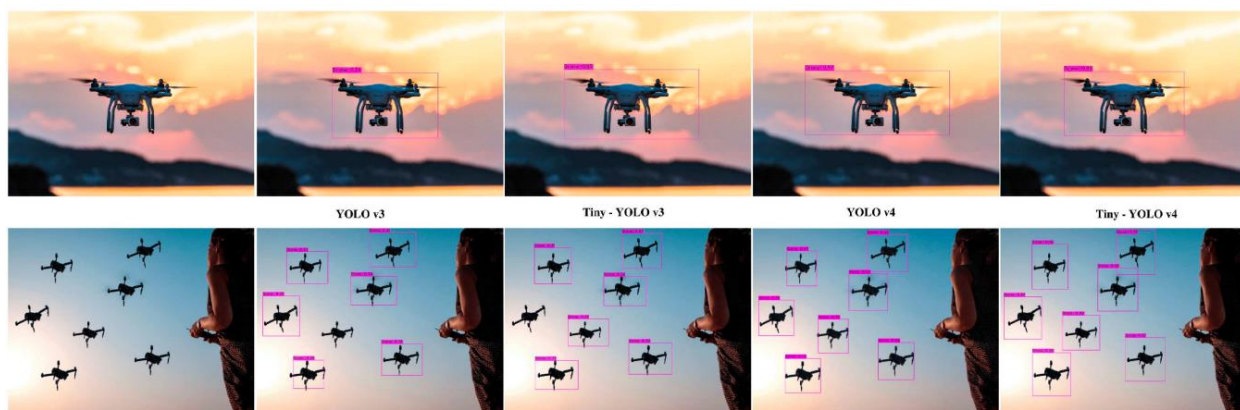
1.23 pav. „Tiny-YOLOv4“ ir „apkarpyto“ „YOLOv4“ modelio našumo palyginimas po duomenų papildymo (Liu et al., 2021)

Iš gautų rezultatų matoma (1.23 pav.), kad po duomenų papildymo siūlomo „apkarpyto“ „YOLOv4“ modelio preciziškumas padidėja daugiau nei 3 kartus, bet vis tiek lieka žemiausias, lyginant su kitais pateiktais modeliais (Liu et al., 2021). Taip pat nuo 30,7 % iki 72,6 % padidėja jautrumas. Panašiai buvo patobulintas ir „Tiny-YOLOv4“ našumas, o „YOLOv4“ modeliui taikytas duomenų papildymo metodas nebuvo reikšmingas našumo atžvilgiu.

„YOLOv4“ originalaus modelio apimtis yra 245,8 MB. Panaikinus konvoliucinį kanalą ir 24 sluoksnius (lengvesniam modeliui gauti), pavyko pasiekti 90,5 % mAP, 69 FPS ir 15,1 MB modelio apimtį (Liu et al., 2021). Tai reiškia, kad „apkarpyto“ „YOLOv4“ modelio apdorojimo greitis buvo padidintas 60,4 %, bet sumažėjo modelio tikslumas (Liu et al., 2021).

1.6.2. Bepiločių orlaivių aptikimas, naudojant YOLO šeimos algoritmus

Sharma et al. (2022) bepiločiams orlaiviams aptikti pasirinko konvoliucinio neuroninio tinklo YOLO modelį. Šis darbas apima keturių YOLO modelio versijų („YOLOv3“, „Tiny-YOLOv3“, „YOLOv4“, „Tiny-YOLOv4“) mokymą, siekiant pasirinkti tinkamiausią tinklą šiai užduočiai. Sharma et al. (2022) gauti rezultatai (aptinkant vieną ir kelis dronus) pateikti 1.24 paveiksle ir 1.4 lentelėje.



1.24 pav. Išmokyty modelių rezultatai, naudojant testavimui skirtus vaizdus (Sharma et al., 2022)

1.4 lentelė. Modelių išvedimo laiko ir vidutinio pasiklivimo palyginimas, naudojant testavimui skirtus vaizdus (sudaryta autorės, remiantis Sharma et al. (2022))

	Vienas bepilotis orlaivis		Keli bepiločiai orlaiviai	
YOLO versija	Vidutinis pasiklivimas	Išvedimo laikas, ms	Vidutinis pasiklivimas	Išvedimo laikas, ms
„YOLOv3“	94 %	215,25	42 %	202,09
„Tiny-YOLOv3“	90 %	196,53	59 %	190,2
„YOLOv4“	94 %	212,19	75 %	195,25
„Tiny-YOLOv4“	98 %	197,27	94 %	165,07

Iš gautų rezultatų galima pastebėti, kad „YOLOv4“ ir „Tiny-YOLOv4“ pranoksta „YOLOv3“, nes „YOLOv3“ nesugeba aptikti visų dronų dėl mažesnių pasiklivimo įverčių (1.24 pav.). Be to, buvo įvertintas išvedimo greitis, kuris yra svarbus matas, siekiant veikimo realiuoju laiku. Iš 1.4 lentelėje pateiktų duomenų galima daryti išvadą, kad „Tiny-YOLOv4“ veikia geriausiai, lyginant su kitais modeliais, atsižvelgiant į vidutinį pasiklivimą ir išvedimo greitį. Kaip teigiama literatūroje, „YOLOv4“ turi pranokti visus kitus 1.4 lentelėje išvardytus modelius (pagal vidutinį pasiklivimą), tačiau ši hipotezė Sharma et al. (2022) nepasiteisino. Kadangi „YOLOv4“ yra labai tankus tinklas, jam reikalingas didelis duomenų rinkinys tiek kokybės, tiek kiekybės požiūriu.

Pagrindiniai parametrai, pagal kuriuos buvo palyginti keturi YOLO modeliai, yra numatymo pasiklivimas, išvedimo laikas ir tikslumas aptinkant kelis dronus. Remiantis šiais rezultatais buvo pastebėta, kad (Sharma et al., 2022):

- Naujesnės YOLO versijos veikia geriau tiek tikslumo, tiek preciziškumo požiūriu.

- Nors „YOLOv4“ yra daug tankesnis tinklas, „Tiny-YOLOv4“ vis tiek jį pranoksta, ypač aptinkant kelis objektus.

- „Tiny-YOLOv4“ numatymo laikas yra daug geresnis nei kitų YOLO versijų, išlaikant geriausią aptikimo pasiklovimą.

Aydin ir Singha (2023) palygino „YOLOv4“ ir „YOLOv5“ tinklus (1.5 lent.). Modeliui mokytis buvo naudojamas duomenų rinkinys, kurį sudaro vaizdai su skirtingu fonu (saulėtu, debesuotu, tamsiu ir kt.), įvairia drono padėtimi (vaizdas iš šono, vaizdas iš viršaus ir kt.), dronų vaizdais iš ilgesnių atstumų ir keliais objektais viename vaizde (Aydin & Singha, 2023). Duomenų rinkinys buvo suformuotas iš 479 paukščių ir 1 916 dronų vaizdų (iš viso 2 395 vaizdai).

1.5 lentelė. Palyginimas dviejų siūlomų modelių (sudaryta autorės, remiantis Aydin ir Singha (2023))

Modelis	Preciziškumas	Jautrumas	mAP50	Vaizdo įrašas	Vieno kadro apdorojimo laikas, ms
„YOLOv4“	0,95	0,68	0,7436	Su „DJI Mavic Pro“	19
				Su „DJI Phantom III“	20,5
„YOLOv5“	0,918	0,875	0,904	Su „DJI Mavic Pro“	23,9
				Su „DJI Phantom III“	31,2

Kaip galima matyti iš 1.5 lentelės, naudojant „YOLOv4“ buvo pasiekti 0,74 mAP, 0,95 preciziškumo ir 0,68 jautrumo įverčiai. Taip pat vieno kadro apdorojimo laikas buvo 20,5 ms vaizdo įrašui su „DJI Phantom III“ ir 19 ms vaizdo įrašui su „DJI Mavic Pro“. Naudojant „YOLOv5“ tinklą, testavimo metu gauti įverčiai: preciziškumo – 0,92, jautrumo – 0,88, mAP – 0,904, vieno kadro apdorojimo laikas atitinkamai lygus 31,2 ms ir 23,9 ms.

„YOLOv5“ yra sukurtas naudojant „PyTorch“ karkasą (angl. *framework*), o ne „Darknet“ karkasą. Tai yra viena iš priežasčių, kodėl buvo pasiektas didesnis tikslumas ir greitis nei taikant ankstesnes metodikas (Aydin & Singha, 2023). Taip pat paskutiniai originalios „YOLOv5“ architektūros sluoksniai buvo tiksliai pritaikyti, kad mokymas, naudojant Aydin ir Singha (2023) duomenų rinkinį, būtų efektyvesnis. Išskyrus sluoksnių derinimą, buvo pakeistos numatytosios mokymosi greičio, paketo dydžio ir kt. reikšmės (Aydin & Singha, 2023).

Taigi, Aydin ir Singha (2023) atlikto tyrimo metu buvo naudojami „YOLOv4“ ir „YOLOv5“ tinklai, ir buvo pasiektas geresnis modelio našumas, nes dėl lengvesnės architektūros „YOLOv5“ objektus gali atpažinti greičiau nei „YOLOv4“. Kaip teigia Aydin ir Singha (2023), dėl duomenų

rinkinio kintamumo, naujų klasių pridėjimo ir su duomenų papildymu siūloma metodika yra lanksti ir gali būti taikoma realaus pasaulio scenarijams.

1.7. Skyriaus išvados

1. Pagrindinė taikinio sekimo algoritmo užduotis yra užtikrinti tikslų taikinio sekimą, taip pat svarbu atsižvelgti į algoritmo realizavimo laiką ir garantuoti veikimą realiuoju laiku.

2. Naudojant vieną jutiklį, taikinių atpažinimo ir sekimo sistemos yra jautrios išoriniams trukdžiams ir priklauso nuo aplinkos veiksnių. Geresni rezultatai ir didesnis tikslumas gaunami naudojant skirtingus modalinius jutiklius, renkančius papildomus duomenis apie taikinį. Tokiu būdu gali būti išvengiama dviprasmių atpažinimo ir sekimo sprendimų dėl neteisingų duomenų bei padidinamas sistemos patikimumas.

3. Sekant taikinius, diskriminacinio metodo modelis taikomas dažniau, nes yra tikslesnis ir patvaresnis. Modelyje yra informacijos apie foną, tai suteikia sekimo galimybę, pavyzdžiui, kai taikiny yra uždengtas.

4. Esamų standartinių sekimo algoritmų rezultatas dažniausiai yra paveiktas ir net gali diverguoti. Dėl to kai kurie mokslininkai filtro algoritme įdiegė adaptyvųjį mechanizmą, padedantį išspręsti šią problemą. Pasirenkant modelių rinkinį, galintį apimti skirtingas judėjimo formas, algoritmas pasiekia geriausią filtravimo rezultatą.

5. Naudojant Kalmano filtrą sekikliui pridedamas nuspėjamumas ir tokiu būdu sumažinama klaidingai teigiamo rezultato tikimybė. Atsižvelgiant į judančių objektų sekimo sudėtingumą (šviesos pokyčiai, objekto uždengimas, greitas judėjimas, sudėtingas fonas), mokslininkai tobulina Kalmano filtro integravimo į CNN modelį algoritmus.

6. Objektams aptikti buvo naudojami įvairūs jutikliai (pavyzdžiui, radarai, LiDAR ir kt.) kartu su įvairiais naujais sprendimais, o tikrasis proveržis įvyko, kai objektams aptikti ir sekti atsirado metodai, pagrįsti giliuoju mokymusi. Taikant šiuos metodus, buvo aptiktos dvi pagrindinė kliūtys: skaičiavimo aparatinės įrangos poreikis ir veikimo realiuoju laiku užtikrinimas.

7. CNN aptikimo tinklas nepaiso taikinio padėties, greičio, krypties ir kitos judėjimo informacijos. Dėl nenuspėjamumo sekimo algoritmas gali būti paveikiamas. Pavyzdžiui, kai objektas yra uždengtas, taikiny gali būti prarastas ir sekimas nepavyks. Vyksta tyrimai ir ieškoma būdų, kaip padaryti CNN modelį tinkamesnį sudėtingiems taikiniams aptikti, esant įvairiems scenarijams.

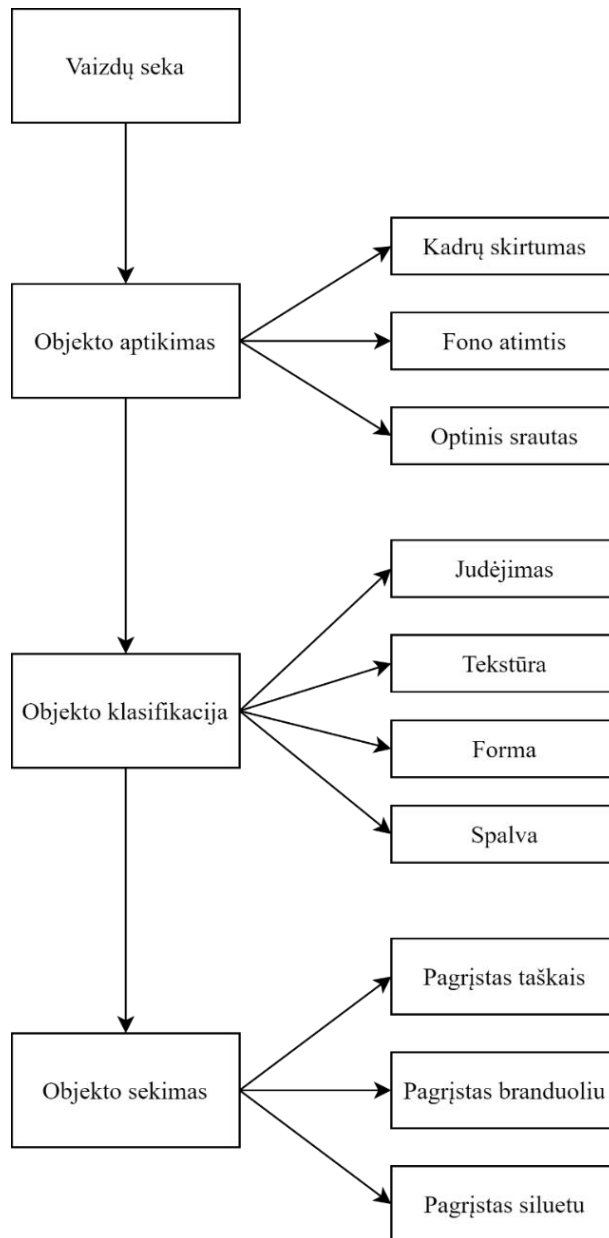
8. Sprendžiant vizualinio taikinių sekimo užduotį, gilieji neuroniniai tinklai turi surinkti daugybę žymėtų vaizdų. Siekiant išspręsti duomenų stygiaus problemą, gilieji požymiai gali būti perkelti iš anksčiau išmokyto giliojo neuroninio tinklo į didelį pagalbinį duomenų rinkinį.

9. YOLO šeimos algoritmų populiarumą lemia objektų aptikimo greičio ir aptikimo tikslumo kompromisas. Šie algoritmai leidžia aptikti objektus realiuoju laiku, todėl gali būti naudojami įvairioms užduotims.

10. Bepiločių orlaivių sekimas kelia iššūkių (pavyzdžiui, fono pasikeitimas, mažas dronų dydis vaizdo kadruose, panašumas į paukščius), kuriems reikia naujų sprendimų. Dronai gali skristi dideliu greičiu, todėl taikomų metodų aptikimo greitis taip pat turi būti didelis. Pastaruoju metu mokslininkai renkasi YOLO algoritmus, pasižyminčius dideliu našumu ir greitu objektų aptikimu bei veikiančius realiuoju laiku.

2. METODŲ APŽVALGA IR ANALIZĖ

Vaizdo apdorojimas – tai naudingos informacijos išskyrimo būdas, konvertuojant vaizdą į skaitmeninę informaciją ir atliekant tam tikras operacijas. Objektų aptikimas ir sekimas yra aktuali ir sudėtinga užduotis, reikalaujanti optimalaus sprendimo, kuris priklauso nuo iškeltos užduoties. 2.1 paveiksle pateikti pagrindiniai objektų aptikimo ir sekimo žingsniai.



2.1 pav. Pagrindiniai objekto aptikimo ir sekimo žingsniai (sudaryta autorės, remiantis Balaji ir Karthikeyan (2017))

Vaizdo apdorojimas yra terminas, nurodantis vaizdų sekos arba vaizdo kadro apdorojimą, kurio rezultatas – susijusių vaizdo parametrų rinkinys (Panchal et al., 2015). Sekant objektą, vaizdo analizė apima judančio objekto aptikimą, objekto klasifikavimą ir objekto sekimą paeiliui einančiuose kadruose (Panchal et al., 2015):

- 1) Objektų aptikimas skirtas objektų buvimui vaizdo kadre patikrinti ir tam objektui aptikti.

2) Tada aptiktą objektą galima priskirti įvairioms kategorijoms, pavyzdžiui, žmonėms, transporto priemonėms ir kitiems judantiems objektams.

3) Objektų sekimo proceso metu yra segmentuojama dominanti sritis vaizdo kadruose, taip sekamas objekto judėjimas bei jo padėtis.

Toliau apžvelgiami metodai, taikomi šiems trimis etapams realizuoti.

2.1. Objektų aptikimo metodai

Yra trys pagrindiniai vaizdo tyrimo etapai: dominančių objektų aptikimas vaizde, šių objektų sekimas kiekviename kadre (paeiliui einančiuose kadruose) ir objektų judėjimo analizė, siekiant atpažinti jų veiklą (Panchal et al., 2015). Objektų aptikimas yra susijęs su objektų atpažinimu vaizdo kadre ir šių objektų pikselių grupavimu (Parekh et al., 2014). Judančių objektų aptikimas yra pradinis vaizdo sekos analizės etapas. Tai daroma kiekviename kadre arba kai vaizdo sekoje pasirodo dominantis objektas (Joshi & Thakore, 2012).

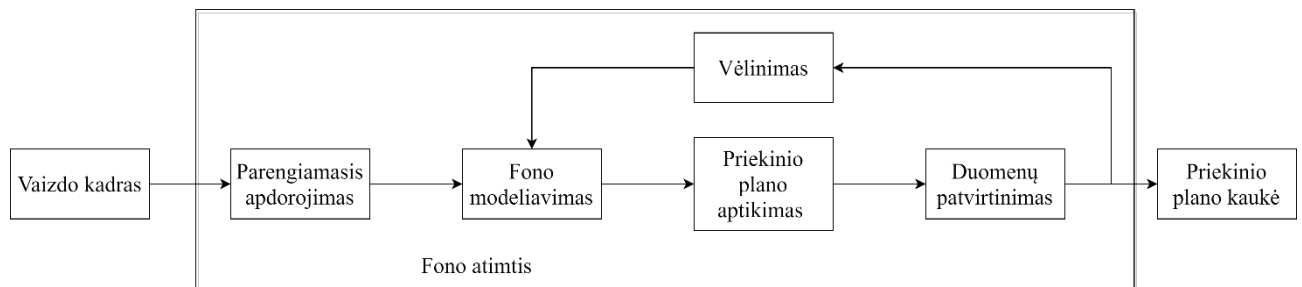
Objektui aptikti gali būti taikomi labai paprasti algoritmai, bet geresniam rezultatui pasiekti reikalingi sudėtingesni metodai. Metodo pasirinkimas priklauso nuo reikalaujamo apdorojimo greičio, naudojamos atminties, siekiamo tikslumo. Kai kurie siūlomi metodai gali susidoroti su realiuoju laiku kylančiais iššūkiais, tokiais kaip sniegas, lietus, judančios šakos, kiti objektai, kurie sutampa su dominančiu objektu, keičiantis šviesos intensyvumas ar lėtai judantys objektai (Panchal et al., 2015).

Paprastas objektų aptikimo būdas yra informacijos naudojimas viename kadre. Tačiau kai kurie objektų aptikimo metodai naudoja laikinąją informaciją, apskaičiuotą iš kadrų sekos, kad būtų sumažintas klaidingų aptikimų skaičius (Joshi & Thakore, 2012). Ši informacija paprastai yra pateikiama kadrų skirtumo forma, kuri išryškina iš eilės einančiuose kadruose dinamiškai kintančius regionus.

Populiarūs judančių objektų aptikimo metodai yra kadrų skirtumo metodas, fono atimties metodas ir optinio srauto metodas (Rakibe & Patil, 2013). Taikant kadrų skirtumo metodą, skaičiuojamas skirtumas tarp dviejų vienas po kito einančių vaizdų, siekiant identifikuoti judantį objektą. Šiame etape atliekami nesudėtingi skaičiavimai. Šis metodas gali būti puikiai pritaikytas įvairioms dinaminėms aplinkoms. Tačiau paprastai šiuo metodu yra sunku gauti visą judančio objekto kontūrą, todėl aptikimo rezultatas gali būti netikslus (Rakibe & Patil, 2013).

Optinio srauto metodas yra skirtas vaizdo optinio srauto laukui apskaičiuoti ir grupavimui atlikti, remiantis vaizdo optinio srauto pasiskirstymo charakteristikomis (Rakibe & Patil, 2013). Taikant šį metodą, iš fono galima gauti išsamią informaciją apie objekto judėjimą ir tiksliau jį aptikti. Tačiau dėl didelio skaičiavimų kiekio, jautrumo triukšmui ir prasto triukšmo slopinimo šis metodas yra nepatikimas realiuoju laiku veikiančioms programoms (Rakibe & Patil, 2013).

Bet kuri judėjimo aptikimo sistema, pagrįsta fono atimties metodu, turi spręsti daugybę kritinių situacijų (Rakibe & Patil, 2013), pavyzdžiui, dėl prastos vaizdo šaltinio kokybės atsirandantys triukšmai, šviesos sąlygų pokyčiai ir kt. Egzistuoja įvairių fono atimties algoritmų, tačiau daugumą jų sudaro keturi pagrindiniai žingsniai (Cheung & Kamath, 2004): parengiamasis apdorojimas (angl. *preprocessing*), fono modeliavimas, priekinio plano aptikimas ir duomenų patvirtinimas (2.2 pav.).



2.2 pav. Bendrojo fono atimties algoritmo schema (sudaryta autorės, remiantis Cheung ir Kamath (2004))

Parengiamąjį apdorojimą sudaro paprastų vaizdo apdorojimo užduočių rinkinys, kuriuo neapdorotas įvesties vaizdas pakeičiamas formatu, kuris tinka tolesniems apdorojimams atlikti (Cheung & Kamath, 2004). Daugumai kompiuterinės regos sistemų parengiamojo apdorojimo žingsnyje naudojamas paprastas glodinimas (angl. *smoothing*), siekiant sumažinti vaizdo kameros triukšmą, taip pat glodinimas gali būti naudojamas pašalinant trumpalaikį aplinkos triukšmą (pavyzdžiui, lietų ir sniegą) (Cheung & Kamath, 2004).

Fono modeliavimas yra naudojamas atraminiam (angl. *reference*) modeliui gauti. Šis atraminis modelis yra lyginamas su kiekvienu vaizdo sekos kadru, siekiant nustatyti galimą pokytį (Balaji & Karthikeyan, 2017). Judančio objekto buvimą lemia dabartinio vaizdo kadro ir ankstesnio kadro skirtumas pikselių atžvilgiu (Athanesious & Suresh, 2012). Šis fono modelis pateikia visos fono scenos statistinį aprašymą (Cheung & Kamath, 2004). Priekinio plano aptikimo žingsnyje nustatomi vaizdo rėmelyje esantys pikseliai, kurių fono modelis negali tinkamai paaiškinti, ir pateikia juos kaip dvejetainę kandidato priekinio plano kaukę (angl. *foreground mask*) (Cheung & Kamath, 2004).

Duomenų patvirtinimo žingsnyje nagrinėjama kandidato kaukė, pašalinami taškai, kurie neatitinka tikrųjų judančių objektų, ir pateikiama galutinė priekinio plano kaukė. Tikrinant duomenis dažnai naudojamos domeno žinios ir skaičiavimų reikalaujantys kompiuterinės regos algoritmai (Cheung & Kamath, 2004). Naudojant šį algoritmą, apdorojimas realiuoju laiku yra įmanomas, nes šie sudėtingi algoritmai taikomi tik nedaugeliui kandidatų priekinio plano pikselių.

Toliau apžvelgiamos adaptyviosios technikos, kurioms atlikti reikia mažiau išteklių. Paprastai algoritmai yra skirstomi į rekursinius ir nerekursinius.

2.1.1. Rekursiniai metodai

Taikant rekursinius metodus yra atnaujinamas vienas fono modelis, remiantis kiekvienu įvesties kadru. Todėl tolimes praeities įvesties kadrai gali turėti įtakos dabartiniam fono modeliui (Cheung & Kamath, 2004). Šių algoritmų pagrindinis privalumas – reikia mažiau atminties duomenų saugojimui (Balaji & Karthikeyan, 2017). Tačiau, atsiradus klaidai fono modelyje, ji egzistuos ilgą laiką. Kai kuriuose algoritmuose yra naudojamas eksponentinis svoris, kad būtų galima atmesti praeities duomenis, ir įtraukiamas teigiamas sprendimų grįžtamasis ryšys, kad atnaujinimui būtų naudojami tik fono pikseliai (Cheung & Kamath, 2004).

Kai kurie dažnai taikomi rekursiniai metodai:

- Apytikslis vidutinis filtras (angl. *approximated median filter*). Metodo esmė: einamasis vidutinis įvertis padidinamas vienetu, jei įvesties pikselis yra didesnis už įvertį, ir sumažinamas vienetu, jei jis yra mažesnis. Šis įvertis galiausiai sutampa su verte, kuriai pusė įvesties taškų yra didesni ir pusė yra mažesni už šią vertę – tai yra mediana (Cheung & Kamath, 2004). Vykdamas vaizdo apdorojimą realiuoju laiku ir sprendžiant kompiuterinės regos užduotis, yra nuolat siūlomi patobulinimai, leidžiantys pasiekti geresnius rezultatus, lyginant su anksčiau sukurtomis apytikslio vidutinio filtro variacijomis (Appiah et al., 2020).

- Kalmano filtras. Šis filtras yra plačiai naudojamas tiesinėms dinaminėms sistemoms sekti, esant Gauso triukšmui (Cheung & Kamath, 2004). Paprasčiausia šio rekursinio metodo versija naudoja tik skaisčio intensyvumą, taip pat gali būti naudojamas intensyvumas ir jo išvestinė laiko atžvilgiu arba intensyvumas ir jo išvestinė vietos atžvilgiu. Pavyzdžiui, kai vidinę sistemos būseną apibūdina fono intensyvumas B_t ir jo išvestinė laiko atžvilgiu B'_t , parametrai rekursyviai atnaujinami taip (Cheung & Kamath, 2004):

$$\begin{bmatrix} B_t \\ B'_t \end{bmatrix} = A \cdot \begin{bmatrix} B_{t-1} \\ B'_{t-1} \end{bmatrix} + K_t \cdot \left(I_t - H \cdot A \cdot \begin{bmatrix} B_{t-1} \\ B'_{t-1} \end{bmatrix} \right), \quad (2.1)$$

čia: B_t – fono intensyvumas t laiko momentu; B'_t – fono intensyvumo išvestinė laiko atžvilgiu t laiko momentu; A – matrica, apibūdinanti fono dinamiką; B_{t-1} – fono intensyvumas $(t - 1)$ laiko momentu; B'_{t-1} – fono intensyvumo išvestinė laiko atžvilgiu $(t - 1)$ laiko momentu; K_t – Kalmano stiprinimo matrica; I_t – einamasis pikselis; H – matavimo matrica.

Konkrečios A ir H matricų vertės parenkamos atsižvelgiant į sprendžiamą užduotį. Kalmano stiprinimo matrica K_t persijungia tarp lėto adaptacijos greičio α_1 ir greito adaptacijos greičio $\alpha_2 > \alpha_1$, atsižvelgiant į tai, ar I_{t-1} yra priekinio plano pikselis (Cheung & Kamath, 2004): $K_t = \begin{bmatrix} \alpha_1 \\ \alpha_1 \end{bmatrix}$, kai I_{t-1} yra priekinio plano pikselis, priešingu atveju – $K_t = \begin{bmatrix} \alpha_2 \\ \alpha_2 \end{bmatrix}$.

- Gauso skirstinių mišinys (angl. *Mixture of Gaussians* – MoG). Skirtingai nuo Kalmano filtro, kuris stebi vieno Gauso skirstinio išskyrimą, MoG metodas vienu metu stebi kelis Gauso skirstinius

(Cheung & Kamath, 2004). Panašiai kaip neparametrinis modelis (aprašytas 2.1.2. skyrelyje), MoG palaiko kiekvieno pikselio tankio funkciją. Taigi, šiuo metodu galima valdyti kelių modelių fono skirstinius (Cheung & Kamath, 2004). Kita vertus, kadangi MoG yra parametrinis, modelio parametrus galima adaptyviai atnaujinti, nenaudojant didelio vaizdo kadrų buferio. Pikselių pasiskirstymas $f(I_t = u)$ modeliuojamas kaip K Gauso skirstinių mišinys (Cheung & Kamath, 2004):

$$f(I_t = u) = \sum_{i=1}^K \omega_{i,t} \cdot \eta(u; \mu_{i,t}, \sigma_{i,t}), \quad (2.2)$$

čia: $\eta(u; \mu_{i,t}, \sigma_{i,t})$ – i -tasis Gauso komponentas, kurio intensyvumo vidurkis $\mu_{i,t}$ ir standartinis nuokrypis $\sigma_{i,t}$; $\omega_{i,t}$ – duomenų dalis, patenkanti į i -tajį komponentą.

Paprastai K svyruoja nuo trijų iki penkių, atsižvelgiant į turimą atmintinę (angl. *storage*). Kiekvienam įvesties pikseliui I_t pirmiausia reikia nustatyti komponentą i , kurio vidurkis yra arčiausiai šio pikselio. Komponentas i yra deklaruojamas kaip atitinkantis (angl. *matched*) komponentas, jei $|I_t - \mu_{i,t-1}| \geq D \cdot \sigma_{i,t-1}$, kur D apibrėžia mažą teigiamo nuokrypio ribą (Cheung & Kamath, 2004). Tada atitinkančio komponento parametrai atnaujinami taip (Cheung & Kamath, 2004):

$$\omega_{i,t} = (1 - \alpha)\omega_{i,t-1} + \alpha, \quad (2.3)$$

$$\mu_{i,t} = (1 - \rho)\mu_{i,t-1} + \rho I_t, \quad (2.4)$$

$$\sigma_{i,t}^2 = (1 - \rho)\sigma_{i,t-1}^2 + \rho(I_t - \mu_{i,t})^2, \quad (2.5)$$

čia: α – nustatytas mokymosi greitis (angl. *learning rate*), kai $0 \leq \alpha \leq 1$; ρ – parametrų mokymosi greitis, apytiksliai apskaičiuojamas pagal formulę $\rho \approx \frac{\alpha}{\omega_{i,t}}$.

Jei nepavyksta rasti atitinkančio komponento, mažiausią svorį turintis komponentas pakeičiamas nauju komponentu, kurio vidurkis I_t , pradinė dispersija σ_o ir svoris ω_o . Likusių komponentų vidurkių ir dispersijų reikšmės nekeičiamos, o jų svoris mažinamas, kad būtų pasiektas eksponentinis skilimas (angl. *exponential decay*): $\omega_{i,t} = (1 - \alpha)\omega_{i,t-1}$ (Cheung & Kamath, 2004).

Galiausiai visi svoriai yra normalizuojami, kad jų suma būtų lygi vienetui. Norint nustatyti, ar tai yra pirmojo plano pikselis, pirmiausia visi komponentai rikiuojami pagal jų $\omega_{i,t}/\sigma_{i,t}$ vertes (Cheung & Kamath, 2004). Aukštesnę vietą užimantys komponentai turi mažą dispersiją ir didelę tikimybę – tai yra tipiškos fono savybės. Kai $i_1, i_2, \dots, i_K$ yra komponentų eilė po rikiavimo, pirmieji M komponentai, kurie atitinka šį kriterijų, yra skelbiami kaip fono komponentai (Cheung & Kamath, 2004):

$$\sum_{k=i_1}^{i_M} \omega_{k,t} \geq \Gamma, \quad (2.6)$$

čia: Γ – svorio riba (angl. *weight hold*).

MoG metodo vystymasis apima parametrų jautrumo analizę, sudėtingumo ir pritaikymo patobulinimus, išplėtimą panoraminiam fonui sukurti (Cheung & Kamath, 2004), o tai leidžia taikyti šį metodą skirtingoms užduotims.

2.1.2. Nerekursiniai metodai

Taikant nerekursinius algoritmus, fonui įvertinti taikomas slankiojančio lango metodas. Buferyje saugomi ankstesni N vaizdo kadrai ir įvertinamas fonas, remiantis kiekvieno pikselio, esančio buferyje, kitimu laike. Nerekursiniai metodai yra labai adaptyvūs, nes jie nepriklauso nuo ankstesnių kadrų, išskyrus saugomus buferyje.

Nerekursinių metodų pavyzdžiai:

- Kadru skirtumas (angl. *frame differencing*). Paprasčiausia fono modeliavimo technika kadrams diferencijuoti naudoja vaizdo sekos kadrą $(t - 1)$ laiko momentu kaip fono modelį kadrui t laiko momentu. Šiam metodui naudojamas tik vienas ankstesnis kadras (Cheung & Kamath, 2004), todėl, taikant šį metodą, gali kilti sunkumų aptinkant didelį tolygiai spalvotą judantį objektą.

- Vidutinis filtras (angl. *median filter*). Tai yra vienas iš dažniausiai naudojamų fono modeliavimo būdų. Fono įvertis (angl. *background estimate*) apibrėžiamas kaip visų esančių buferyje kadrų kiekvieno atitinkamo pikselio vietos vidutinė vertė (Cheung & Kamath, 2004). Daroma prielaida, kad pikselis lieka fone daugiau nei pusei buferio kadrų.

- Tiesinis prognozuojantysis filtras (angl. *linear predictive filter*). Dabartinį fono įvertį galima apskaičiuoti buferyje esantiems pikseliams taikant tiesinį prognozuojantįjį filtrą (Toyama et al., 1999). Filtro koeficientai apskaičiuojami kiekvienam kadrui, remiantis imties kovariantiškumu, todėl šį metodą sudėtinga taikyti realiuoju laiku.

- Neparametrinis modelis. Skirtingai nuo ankstesnių metodų, kai kiekvienoje pikselių vietoje yra naudojamas vienas fono įvertis, buvo pasiūlyti metodai (Elgammal et al., 2000), kuriuose naudojami ankstesni kadrai, kad būtų sudarytas neparametrinis pikselių tankio funkcijos $f(I_t = u)$ įvertis (Cheung & Kamath, 2004):

$$f(I_t = u) = \frac{1}{N} \sum_{i=t-N}^{t-1} K(u - I_i), \quad (2.7)$$

čia: $K(\cdot)$ – branduolio įvertis (pasirinktas Gauso skirstinys); N – ankstesnių vaizdo kadrų skaičius.

Dabartinis pikselis I_t yra skelbiamas kaip priekinis planas, jei pagal šį pasiskirstymą yra maža tikimybė, t. y. $f(I_t)$ yra mažesnis už tam tikrą iš anksto nustatytą slenkstį. Priekinio plano aptikimas šiuo būdu gali būti sudėtingesnis, lyginant su kitais metodais, nes reikia atlikti skaičiavimus pagal (2.7) lygtį kiekvienam pikseliui (Cheung & Kamath, 2004). Šio modelio naudojimo pranašumas prieš vieną įvertį – galimybė valdyti kelių modelių fono pasiskirstymus, kaip jau buvo minėta aptariant MoG metodą.

2.2. Objektų klasifikavimo metodai

Objektų klasifikavimo metu objektai priskiriami tam tikrai grupei. Objektų klasifikavimas atliekamas remiantis objektų formos požymiais judėjimo srityje (Balaji & Karthikeyan, 2017). Populiarius objektų klasifikavimo metodus ir jų tikslumą pateikti 2.1 lentelėje.

2.1 lentelė. Objektų klasifikavimo metodų palyginimas (sudaryta autorės, remiantis Balaji ir Karthikeyan (2017))

Metodas	Tikslumas	Komentarai
Pagrįstas judėjimu	Vidutinis	Nereikalauja iš anksto nustatytų šablonų, gali būti aptiktas ir nejudantis objektas
Pagrįstas tekstūra	Aukštas	Užtikrina geresnę kokybę, bet reikia laiko papildomiems skaičiavimams atlikti
Pagrįstas forma	Vidutinis	Paprastą modelių atitikmens metodą galima pritaikyti naudojant šablonus
Pagrįstas spalva	Aukštas	Sukuriamas Gauso mišinio modelis, apibūdinantis spalvų pasiskirstymą vaizdų sekoje ir segmentuojantis vaizdą į foną ir objektus

Judančio objekto klasifikacija turi periodinės savybės ženklą, kurį rodo nestandaus objekto judėjimas. Optinis srautas taip pat gali būti naudingas objekto klasifikacijai. Judančių objektų standumo analizė ir periodiškumas atliekami naudojant liekamąjį srautą (Balaji & Karthikeyan, 2017). Nestandūs dinaminiai objektai, tokie kaip žmogus, gali turėti didesnę vidutinę liekamąjį srautą ir parodyti periodinę komponentą, tuo tarpu tikimasi, kad standžių objektų liekamasis srautas bus mažas (Balaji & Karthikeyan, 2017).

Gradiiento orientacija, atsirandanti lokalizuotose vaizdo dalyse, skaičiuojama pagal tekstūrą pagrįstą techniką. Tada skaičiavimai atliekami naudojant sutampantį vietinį kontrasto normalizavimą per vienodai išdėstytą ląstelių tankio tinklą, siekiant pagerinti tikslumą (Balaji & Karthikeyan, 2017).

Klasifikuoti judančius objektus pagal turimą informaciją apie objekto formą galima naudojant judėjimo regionus, atvaizduotus per taškus, langą ar dėmę (Balaji & Karthikeyan, 2017). Pavyzdžiui, kiekviena dėmė kiekviename kadre yra klasifikuojama, o gauti rezultatai atvaizduojami histograma.

Spalva yra lengvai išskiriamas ir stabilus požymis (Balaji & Karthikeyan, 2017). Dėl siūlomų algoritmų nedidelių skaičiavimo apimčių spalva yra pageidaujamas požymis, kurį rekomenduojama naudoti. Pavyzdžiui, transporto priemonėms aptikti ir sekti realiuoju laiku naudojamas spalvų histograma pagrįstas algoritmas (Balaji & Karthikeyan, 2017). Norint segmentuoti vaizdą į foną ir

objektą bei apibūdinti spalvų pasiskirstymą vaizdų sekoje, gali būti sukuriamas modelis pagal Gauso mišinio metodą.

Kiekvienam objektui yra matuojami keli požymiai (Zhang et al., 2013b), pavyzdžiui, judėjimo greitis, objekto dydis, ribojantis langas (angl. *boundary box*). Tada klasifikatorius naudoja šiuos požymius ir nustato, kokiai klasei priklauso kiekvienas aptiktas objektas. Klasifikacija gali būti atliekama su priežiūra ar be priežiūros (N. Kaur & Y. Kaur, 2014). Pirmuoju atveju išankstiniame mokymui yra naudojamas rinkinys, kuriame pateikiami pavyzdiniai objektai, priskirti žinomoms klasėms. Taikant klasifikavimo metodus be priežiūros, klasė nustatoma tiesiogiai iš turimų duomenų apie objektą – nenaudojami mokymui skirti rinkiniai ir nėra iš anksto nustatytų klasių.

Taip pat yra naudojami algoritmai, kuriems taikomos abiejų metodų, su priežiūra ir be jos, savybės. Pagrindinė idėja yra mokyti du klasifikatorius pagal nepriklausomas tų pačių duomenų ypatybes, naudojant palyginti nedaug pavyzdžių (N. Kaur & Y. Kaur, 2014). Tada šie klasifikatoriai peržiūri nežymėtus pavyzdžius, juos pažymi ir prideda prognozes prie kito klasifikatoriaus žymėto rinkinio. Kitaip tariant, klasifikatoriai moko vienas kitą, naudodami nežymėtus duomenis (N. Kaur & Y. Kaur, 2014).

Taigi, klasifikatorius yra algoritmas, kuris paima objektams būdingų požymių rinkinį ir naudoja šias charakteristikas kiekvieno objekto klasei nustatyti. Klasifikatorių kūrimo procesą sudaro keturi pagrindiniai žingsniai: mokymo duomenų rinkinio kūrimas, efektyvių požymių pasirinkimas, klasifikatoriaus mokymas, reikiamo klasifikatoriaus tikslumo pasiekimas (Singh & Sharma, 2007).

Pasirinkto metodo efektyvumui didelę įtaką turi tinkamai sudarytas mokymui skirtas duomenų rinkinys ir parinktas objektų parametrų rinkinys. Mokymui skirtame rinkinyje turi būti objektų, kurių klasifikacija yra žinoma, sąrašas (White, 1996). Geram rezultatui pasiekti, šiame rinkinyje turi būti surinkta daug pavyzdžių (paprastai tūkstančiai objektų), svarbu, kad į jį būtų įtraukti ir dažnai pasirodantys, ir reti objektų tipai.

Klasifikatorių mokymas yra optimizavimo problema – pridedant daugiau parametrų, optimizavimas yra sudėtingesnis. Todėl geriau yra pateikti algoritmui tik būtinus parametrus (White, 1996). Klasifikatoriaus tikslumui nustatyti naudojamas nepriklausomas objektų rinkinys.

Teisingai parinkti požymiai turi įtakos objekto sekimo rezultatui ir yra tiesiogiai susiję su objekto atvaizdavimu (Athanesious & Suresh, 2012). Objektams sekti naudojami spalvų, kraštų ir tekstūros požymiai.

Visi vaizdo kadru formatai yra pagrįsti skirtingu spalvų erdvės modeliu. Skirtingų kadru duomenys gali būti saugomi skirtingose spalvų erdvėse, pavyzdžiui, pilkos skalės, RGB (angl. *red, green and blue*) ir HSV (angl. *hue, saturation, value*) spalvų modelių (Athanesious & Suresh, 2012). Kiekviename kadre saugomi duomenys – tai kiekvienos spektrinės juostos ryškumas. Spalvoti vaizdai

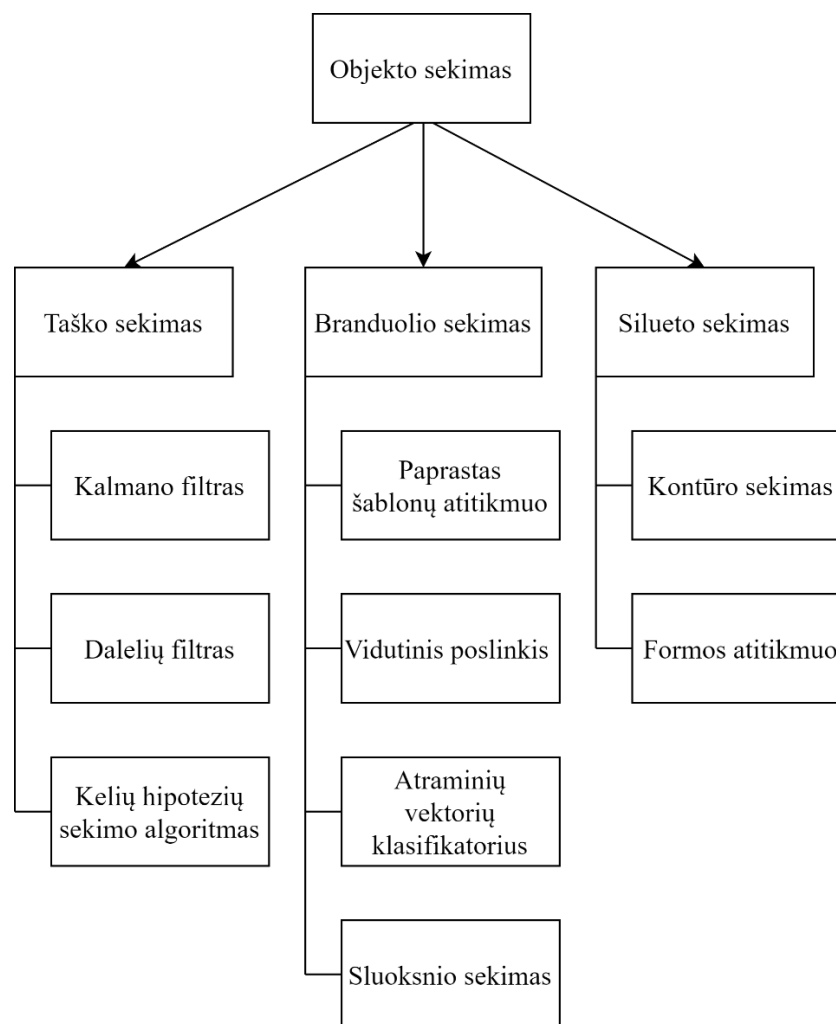
žymimi raudonais (R), žaliais (G) ir mėlynais (B) sluoksniais (arba RGB). HSV spalvų modelyje spalva apibūdinama trimis parametrais: atspalviu, gryniu ir skaisčiu (Dagienė et al., 2014).

Objektų ribos dažnai sukelia ryškius vaizdo intensyvumo pokyčius (Athanesious & Suresh, 2012). Šie pokyčiai nustatomi pagal krašto aptikimą. Kraštai pasižymi mažesniu jautrumu apšvietimo pokyčiams, lyginant su spalvų požymiais. Todėl dauguma algoritmų naudoja kraštus kaip pagrindinį požymį objektui sekti (Athanesious & Suresh, 2012).

Tekstūra parodo paviršiaus intensyvumo skirtingumą, tai leidžia išgauti glotnumo ir dėsningumo savybes (Athanesious & Suresh, 2012). Tekstūrai reikalingas apdorojimo etapas. Lyginant su spalvomis, tekstūros požymiai yra mažiau jautrūs apšvietimo pokyčiams, kaip ir kraštų požymiai (Athanesious & Suresh, 2012).

2.3. Objekto sekimo metodai

Objektų sekimas apima dominančio regiono (angl. *Region of Interest* – ROI) pasirinkimą ir objekto judėjimo bei padėties sekimą vaizdo sekos kadruose (Balaji & Karthikeyan, 2017). Sekimą galima apibrėžti kaip objekto kelio vaizdo plokštumoje apytikslio nustatymo problemą (Athanesious & Suresh, 2012). Taigi, objektų sekimas naudojamas objektų judėjimo kryptčiai sekti. Objekto sekimo metodai gali būti skirstomi pagal tai, koku būdu atvaizduojamas dominantis objektas: kaip taškas, branduolys (angl. *kernel*) ar siluetas (2.3 pav.).



2.3 pav. Objekto sekimo būdai (sudaryta autorės, remiantis Balaji ir Karthikeyan (2017))

Toliau apžvelgiami šių metodų dažnai taikomi algoritmai.

2.3.1. Taško sekimo metodai

Taško sekimo metodas yra viena iš kompiuterinės regos užduočių, naudojamų įvairios paskirties programoms. Vaizdo struktūroje judantys objektai sekimo metu atvaizduojami pagal jų požymius (Athanesious & Suresh, 2012). Nustatant šiuos taškus atliekamas atpažinimas, šiam tikslui gali būti naudojamas slenkstis (angl. *thresholding*). Taškų sekimas yra sudėtinga problema, ypač dėl dominančio objekto uždengimo ir klaidingo objekto aptikimo atvejų (Athanesious & Suresh, 2012). Vienas iš šio metodo privalumų: tinka labai mažiems objektams sekti.

Šio metodo algoritmus dažniausiai sudaro dvi fazės: prognozavimas ir korekcija (Balaji & Karthikeyan, 2017). Prognozavimas apima dabartinių stebėjimų rinkinio naudojimą ir atnaujinimą kitam etapui, kuris suteikia geresnę kitos būsenos apytikslę vertę.

Algoritmuose naudojant Kalmano filtrą (KF), taškai sekami atsižvelgiant į kriterijus, pasirinktus vertinant našumą. Optimalus taškas pasirenkamas remiantis prasmingais kriterijais (Athanesious & Suresh, 2012). Taip pat, naudojant KF, pasiekiamas balansas tarp prognozuojamų

verčių ir triukšmingų matavimų. Šio filtro esmė: prognozuojama būsenos vertė yra nuolat atnaujinama. Sviurių vertei nustatyti naudojama modeliavimo būsenos lygtis (Athanesious & Suresh, 2012).

Naudojant Kalmano filtrą sekimo algoritmuose, pateikiamas optimalus sprendimas, sumažinamas triukšmo poveikis, gali būti sekami vienas ir keli objektai (Athanesious & Suresh, 2012). Tačiau KF prastai aproksimuoja tuos būsenos kintamuosius, kurių skirstinys yra ne Gauso (Athanesious & Suresh, 2012). Šis apribojimas gali būti apeitas naudojant dalelių filtrą (PF).

Dalelių filtras sukuria visus vieno kintamojo modelius prieš pereidamas prie kito kintamojo. Algoritmo pranašumas – kintamieji generuojami dinamiškai ir jų gali būti neribotai daug (Athanesious & Suresh, 2012). Šiame algoritme paprastai naudojami kontūrai, tekstūros atvaizdavimas ir spalvų požymiai (Balaji & Karthikeyan, 2017).

Naudojant kelių hipotezių sekimo (MHT) algoritmą, kiekvienam objektui kiekvienu laiko tarpu teikiami keli pasiūlymai (Balaji & Karthikeyan, 2017). Tai leidžia sumažinti neteisingos atitikties tikimybę. Atliktus kelių kadrų stebėjimą, galima gauti geresnius sekimo rezultatus. Galutinis objekto sekimo rezultatas – tai geriausių atitikmenų rinkinys per tą stebėjimo laikotarpį (Balaji & Karthikeyan, 2017).

MHT – tai iteracinis algoritmas, kartojimas prasideda nuo esamo sekimo hipotezės rinkinio. Kiekvienai hipotezei numatoma kiekvieno objekto padėtis sekančiame kadre. Išmatuotas atstumas apskaičiuojamas ir palyginamas su darytomis prognozėmis (Athanesious & Suresh, 2012). MHT algoritmas tinka keliems objektams sekti, gali būti naudojamas objekto uždengimo atveju, apskaičiuoja optimalius sprendimus (Athanesious & Suresh, 2012).

Apibendrinant galima pasakyti, kad KF yra efektyvus, tačiau turi tiesinę dinamiką ir Gauso triukšmą. PF gali tvarkyti netiesinius ir ne Gauso scenarijus, tačiau gali prireikti daug sudėtingų skaičiavimų. MHT yra patvarus algoritmas, tačiau, esant sudėtingiems scenarijams, taip pat gali prireikti daug skaičiavimų. Algoritmo pasirinkimas priklauso nuo konkrečios sekimo problemos ir kompromisų tarp skaičiavimo sudėtingumo, tikslumo ir patvarumo, reikalingo programai.

2.3.2. Branduolio sekimo metodai

Branduolio sekimas paprastai atliekamas apskaičiuojant nestacionarų objektą, kurį parodo objekto sritis (paeiliui einančiuose kadruose). Objekto judėjimas yra parametrinės formos (Athanesious & Suresh, 2012). Šie algoritmai skiriasi priklausomai nuo sekamų objektų skaičiaus, naudojamo atvaizdavimo ir pasirinkto objekto judėjimo aproksimavimo metodo (Balaji & Karthikeyan, 2017).

Objekto atvaizdavimas, naudojant geometrinę formą, yra įprastas būdas, taikomas realiuoju laiku veikiančių programų realizavimui. Tačiau objektų dalys gali likti už apibrėžtos formos ribų, o fono

dalys gali būti formos viduje, tai yra vienas iš šio metodo taikymo apribojimų (Athanesious & Suresh, 2012).

Dominančio regiono (ROI) nagrinėjimas vaizdų sekoje gali būti atliekamas taikant paprasto šablonų atitikmens metodą. Kadras yra atskiriamas nuo vaizdų sekos ir patikrinama, ar jis atitinka šabloną. Taikant šį metodą galima sekti tik vieną judantį objektą, tačiau sekimas galimas ir esant daliniam objekto uždengimui (Athanesious & Suresh, 2012).

Taikant vidutinio poslinkio metodą, iš pradžių segmentavimo būdu apibrėžiama dominanti sritis, tada objektas sekamas kiekviename kadre (Balaji & Karthikeyan, 2017). Pradiniame kadre ROI apibrėžiamas stačiakampiu langu. Naudojant šį algoritmą, sekamas objektas atskiriamas nuo fono. Taikinio atvaizdavimo ir lokalizavimo tikslumas pagerinamas naudojant *Chamfer* atstumo transformaciją (Balaji & Karthikeyan, 2017). Atstumas tarp dviejų spalvų pasiskirstymų sumažinamas naudojant *Bhattacharya* koeficientą, taip pat šiam tikslui yra taikoma *Chamfer* atstumo transformacija (Balaji & Karthikeyan, 2017). Sekant objektą, jį galima apibūdinti atskirų pavyzdžių paskirstymu ir branduolio lokalizacija (Athanesious & Suresh, 2012).

Branduolio sekimo metodo taikymo žingsniai (Athanesious & Suresh, 2012):

1. Naudojant spalvų požymius, gaunamas tikimybinis taikinio pasiskirstymas pirmajame kadre.
2. Pirmojo kadro pasiskirstymas palyginamas su kitu iš eilės einančiu kadru.
3. Siekiant rasti kadrų panašumo laipsnį, naudojamas *Bhattacharya* koeficientas.
4. Ciklas kartojamas iki paskutinio kadro.

Atraminių vektorių klasifikatoriaus (SVM) metodas pateikia teigiamų ir neigiamų mokymo verčių rinkinį. Teigiamuose pavyzdžiuose yra sekamas vaizdas, o neigiamuose pavyzdžiuose yra visa kita, kas paliekama neseinama (Balaji & Karthikeyan, 2017). SVM metodas gali būti taikomas vienam objektui sekti, tinka dalinai uždengtam objektui sekti, tačiau šiam metodui yra būtinas mokymas (Athanesious & Suresh, 2012).

Taikant sluoksnio sekimo metodą gali būti sekami keli objektai. Formos vaizdavimas, judėjimas (perkėlimas ir pasukimas) bei sluoksnio išvaizda yra įtraukiami į kiekvieną sluoksnį, atsižvelgiant į intensyvumą (Balaji & Karthikeyan, 2017). Iš pradžių fono judėjimo kompensacija atliekama sluoksniuojant taip, kad objektų judėjimą būtų galima įvertinti, naudojant 2D parametrinį judėjimą (Balaji & Karthikeyan, 2017). Remiantis judėjimo ir formos bruožais, apskaičiuojama kiekvieno taško tikimybė. Taikant šį metodą, galima spręsti kelių vaizdų sekimo ar laikinai visiškai uždengto objekto sekimo užduotis (Athanesious & Suresh, 2012).

Tai tik keli aptarti branduolio sekimo algoritmų pavyzdžiai, o praktikoje yra daugybė šių metodų variantų ir derinių. Kiekvieno algoritmo našumas ir tinkamumas priklauso nuo sekamų objektų sudėtingumo, išvaizdos svyravimų, skaičiavimo efektyvumo reikalavimų, objekto uždengimų buvimo, sudėtingų aplinkos sąlygų bei kitų veiksnių.

2.3.3. Silueto sekimo metodai

Kai kurie objektai negali būti gerai apibrėžti geometrinėmis figūromis, todėl sudėtingesnių formų objektams sekti yra naudojamas siluėtų sekimas. Šis sekimo metodas taikomas objekto sričiai kiekviename kadre numatyti, naudojant ankstesnio kadro gautą objekto modelį. Modeliui atvaizduoti gali būti naudojami spalvų holograma, objekto kraštai ar kontūras (Athanesious & Suresh, 2012). Silueto sekimo metodai tinka įvairių formų objektams sekti, objektų uždengimo, sujungimų atvejams (Balaji & Karthikeyan, 2017).

Taikant kontūro sekimo metodus, atliekamas ankstesnio kadro pirminio kontūro iteracinis perkėlimas į naują poziciją dabartiniame kadre (Balaji & Karthikeyan, 2017). Tam tikras objektų skaičius dabartiniame kadre persidengia su objekto sritimi ankstesniame kadre. Objekto būsenos nustatymui naudojami formos parametrai ir kontūro judėjimas. Kiekvieną kartą, atsižvelgiant į didžiausią gautą tikimybę, būsena yra atnaujinama (Balaji & Karthikeyan, 2017).

Formos atitikmens metodai nagrinėja objekto modelį dabartiniame kadre. Šio metodo tikslas – surasti sutampančius siluetus, aptiktus dviejuose vienas po kito einančiuose kadruose (Athanesious & Suresh, 2012).

Aptikimas pagal siluetą atliekamas atimant foną. Objekto modeliui sudaryti gali būti naudojamos tankio funkcijos, silueto ribos, objekto kraštai (Athanesious & Suresh, 2012). Šie metodai tinka vienam objektui sekti. Objekto uždengimo atvejams spręsti taikomi Hofo transformacijos metodai (Athanesious & Suresh, 2012).

2.4. Krašto aptikimo operatoriai

Kraštų aptikimas yra esminis skaitmeninio vaizdo apdorojimo ir segmentavimo žingsnis. Kraštas – tai staigus vaizdo intensyvumo verčių pokytis. Šis elementarus vaizdo požymis turi vertingos informacijos, kuri yra naudinga atpažįstant objektą (Reddy et al., 2016).

Kraštai apibūdina tikrojo vaizdo ribas, o tai yra pagrindinis vaizdo segmentavimo uždavinys (Reddy et al., 2016). Aptinkant vaizde kraštą, sumažinamas duomenų kiekis tolesniam vaizdo apdorojimui, išfiltruojama nepageidaujama informacija, išlaikant svarbius vaizdo požymius. Kraštai paprastai aptinkami ties riba tarp dviejų skirtingų vaizdo kadro regionų. Kraštų aptikimas leidžia vartotojui aptikti tuos vaizdo požymius, kuriuose daugiau ar mažiau staigiai pasikeičia pilkumo lygis ar tekstūra, nurodant vaizde vieno regiono pabaigą ir kito pradžią.

Taigi, kraštų aptikimas yra pagrindinis vaizdo apdorojimo įrankis, iš esmės skirtas požymiams aptikti ir išgauti. Šio įrankio tikslas – nustatyti skaitmeninio vaizdo taškus, kuriuose ryškiai keičiasi vaizdo ryškumas, ir rasti netolydumus (Rashmi et al., 2013).

Kraštams aptikti yra naudojami skirtingi operatoriai, kuriuos galima suskirstyti į pirmosios eilės (arba gradientu pagrįstus operatorius) ir antrosios eilės krašto detektorius. Gradientu pagrįstų

operatorių atveju naudojama pirmos eilės išvestinė. Kai $I(i, j)$ yra įvesties vaizdas, vaizdo gradientas apskaičiuojamas pagal (2.8) formulę (Rashmi et al., 2013):

$$\nabla I(i, j) = \hat{i} \frac{\delta I(i, j)}{\delta i} + \hat{j} \frac{\delta I(i, j)}{\delta j}, \quad (2.8)$$

čia: $\frac{\delta I(i, j)}{\delta i}$ – gradientas i kryptimi; $\frac{\delta I(i, j)}{\delta j}$ – gradientas j kryptimi.

Operatoriai *Robert*, *Sobel*, *Prewitt* yra klasifikuojami kaip klasikiniai operatoriai, kuriuos lengva valdyti, tačiau jie labai jautrūs triukšmui (Rashmi et al., 2013). Naudojant *Robert* operatorių, pirmiausia per atskirą diferencijavimą apskaičiuojama įstrižai gretimų pikselių skirtumo kvadratų suma, tada apskaičiuojamas apytikslis vaizdo gradientas (Rashmi et al., 2013). Įvesties vaizdas yra sujungiamas su numatytais operatoriaus branduoliais ir apskaičiuojamas gradiento dydis bei kryptis.

Naudojamos dvi 2×2 dydžio kaukės (Blackledge, 2005): $D_x = \frac{1}{2} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ ir $D_y = \frac{1}{2} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$.

Šio operatoriaus privalumas yra jo paprastumas, tačiau, turėdamas mažą branduolį, jis yra labai jautrus triukšmui (Rashmi et al., 2013). Tokių operatorių praktiniam pritaikymui paprastai reikia papildomo parengiamojo apdorojimo, kad būtų sumažintas vaizdui būdingo triukšmo lygis (Blackledge, 2005). Todėl šiuolaikiniams algoritmams šis operatorius taikomas rečiau.

Sobel operatorius yra diskretusis diferencijavimo operatorius, naudojamas vaizdo intensyvumo funkcijos gradiento aproksimacijai apskaičiuoti, siekiant aptikti kraštą. Kiekviename vaizdo pikselyje *Sobel* operatorius pateikia vektoriui atitinkamą gradiento vektorių ar normalę, atliekama įvesties vaizdo ir branduolio sąsūkos operacija ir apskaičiuojamas gradiento dydis bei kryptis (Rashmi et al., 2013). Šiam operatoriui naudojami du 3×3 dydžio branduoliai (Blackledge, 2005):

$$D_x = \frac{1}{8} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ ir } D_y = \frac{1}{8} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}.$$

Lyginant su *Robert* operatoriumi, *Sobel* operatorius skaičiavimus atlieka lėčiau, nes turi didesnį branduolį, tačiau jis yra mažiau jautrus triukšmui (Rashmi et al., 2013). Kuo didesnis filtro masyvas, tuo labiau sumažės triukšmas, aptinkant mažiau kraštų. Tačiau per didelis filtras naudingų kraštų gali ir neaptikti (Blackledge, 2005).

Prewitt krašto detektoriaus funkcija yra beveik tokia pati kaip *Sobel* detektoriaus, tačiau skiriasi

naudojami branduoliai (Blackledge, 2005): $D_x = \frac{1}{6} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$ ir $D_y = \frac{1}{6} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$.

Prewitt krašto operatorius užtikrina geresnius rezultatus nei *Sobel* operatorius (Rashmi et al., 2013).

Antrosios eilės krašto detektorius yra pagrįstas antros eilės išvestine. Šiame operatoriuje pikselis pažymėtas kraštu toje vietoje, kur antroji vaizdo išvestinė yra lygi nuliui (Rashmi et al., 2013).

Laplaso operatorius ∇^2 , skirtas 2D $I(i, j)$ vaizdui, apibrėžiamas pagal (2.9) formulę (Blackledge, 2005):

$$\nabla^2 = I(i, j) = \frac{\delta^2}{\delta x^2} I(i, j) + \frac{\delta^2}{\delta y^2} I(i, j). \quad (2.9)$$

Vienas iš antrosios eilės krašto detektorių pavyzdžių yra Marr-Hildreth metodas, kuris buvo labai populiarus, kol Canny (1986) nepasiūlė savo algoritmo. *Marr-Hildreth* operatorius taip pat yra pagrįstas gradientu. Jam naudojamas tiek *Gauso*, tiek *Laplaso* operatorius: *Gauso* operatorius sumažina triukšmą, o *Laplaso* operatorius aptinka aštirus kraštus (Rashmi et al., 2013). Tačiau šiam operatoriui būdingi du pagrindiniai trūkumai. Pirmasis – generuojami atsakai, kurie neatitinka kraštų, vadinamieji „klaidingi kraštai“ (angl. *false edges*) (Rashmi et al., 2013). Antrasis trūkumas – atsiranda didelė lokalizacijos klaida lenktų kraštų atveju.

Canny krašto detektorius turi pažangų algoritmą, išvestą iš Marro ir Hildreto darbo (Rashmi et al., 2013). Tai yra optimali kraštų aptikimo technika, užtikrinanti gerą aptikimą, aiškų atsaką ir gerą lokalizaciją. *Canny* krašto aptikimo algoritmas yra vienas optimaliausių kraštų detektorių (Rashmi et al., 2013).

Canny kraštų aptikimo algoritmas pateiktas 2.4 paveiksle. Pirmiausia yra pašalinamas nuskaitytame vaizde esantis triukšmas: atliekama įvesties $I(i, j)$ ir Gauso filtro G sąsūkos operacija. Po glodinimo gautą vaizdą matematiškai galima pateikti taip (Rashmi et al., 2013):

$$F(i, j) = G * I(i, j). \quad (2.10)$$

Kitame etape nustatomi kraštai, atsižvelgiant į tai, kur pilkosios spalvos intensyvumo pokytis yra didžiausias. Reikiamos zonos nustatomos, naudojant vaizdų gradientą. Atlikus *Sobel* operatoriaus kaukių (D_i ir D_j) ir vaizdo, gauto po glodinimo, sąsūkos operaciją, gaunami gradientai i ir j kryptimis (Rashmi et al., 2013):

$$G_i = D_i * F(i, j), \quad (2.11)$$

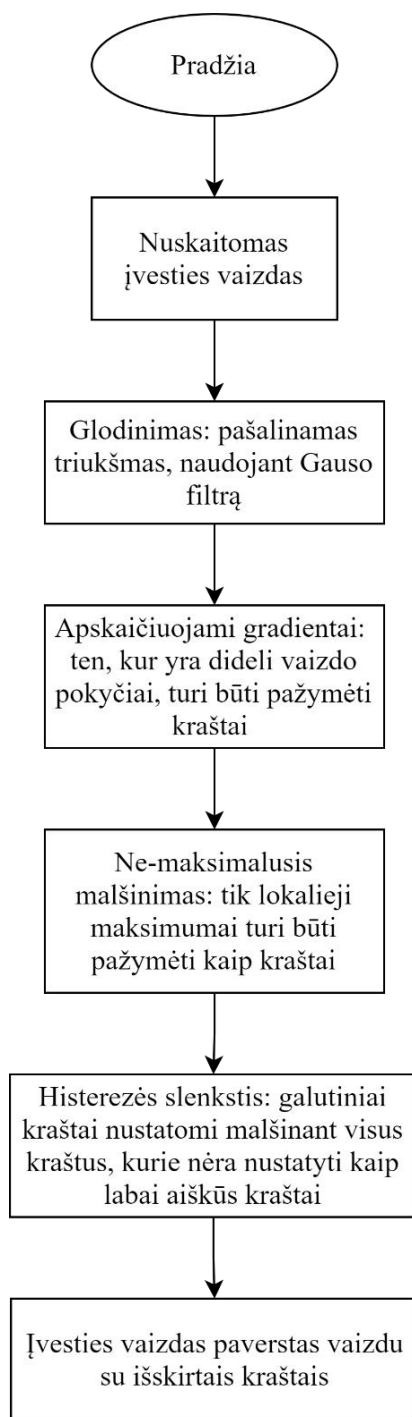
$$G_j = D_j * F(i, j). \quad (2.12)$$

Tada krašto intensyvumas arba pikselio gradiento dydis apskaičiuojamas pagal (2.13) formulę (Rashmi et al., 2013):

$$G = \sqrt{G_i^2 + G_j^2}. \quad (2.13)$$

Gradianto kryptis apskaičiuojama pagal (2.14) formulę (Rashmi et al., 2013):

$$\theta = \arctan\left(\frac{G_j}{G_i}\right). \quad (2.14)$$



2.4 pav. Canny kraštų aptikimo algoritmo schema (sudaryta autorės, remiantis Rashmi et al. (2013))

Siekiant išlaikyti visus lokaliuosius maksimumus, gradiento vaizde atliekamas ne-maksimalusis malšinimas. Taip pašalinama visa kita ir sudaromi ploni kraštai. Kiekvienam pikseliui $M(i, j)$ atliekami tokie žingsniai (Rashmi et al., 2013):

- Pirmiausia gradiento kryptis apvalinama iki artimiausių 45° , tada pikselių gradiento dydis palyginamas teigiama ir neigiama gradiento kryptimis, t. y. jei gradiento kryptis yra rytai, palyginimas atliekamas su pikselių gradientu rytų ir vakarų kryptimis, atitinkamai $E(i, j)$ ir $W(i, j)$.
- Jei pikselio $M(i, j)$ krašto intensyvumas yra didesnis nei $E(i, j)$ ir $W(i, j)$, tada gradiento vertė išsaugoma ir $M(i, j)$ pažymimas kaip krašto pikselis, jei ne – tada malšinamas ar pašalinamas.

Po ne-maksimaliojo malšinimo gautoje išvestyje vis dar yra triukšmo sukelti lokalieji maksimumai. Kitam žingsniui pasirenkami du slenksčiai (t_{high} ir t_{low}), kad būtų išvengta dviejų ribų nustatymo problemos. Pikselis $M(i, j)$, kurio gradiento dydis G , nustatomas kaip kraštas arba atmetamas pagal tokias sąlygas (Rashmi et al., 2013):

- Jei $G < t_{low}$, kraštas atmetamas.
- Jei $G > t_{high}$, kraštas išsaugomas.
- Jeigu $t_{low} < G < t_{high}$ ir bet kurio šio pikselio kaimyno 3×3 regione aplink jį gradiento dydis yra didesnis nei t_{high} , kraštas išsaugomas.
- Jeigu nė vienas iš pikselio (x, y) kaimynų neturi didelio gradiento dydžio, tačiau bent vienas patenka tarp t_{low} ir t_{high} , 5×5 regione, tikrinama, ar kurio nors iš šių pikselių gradiento dydis yra didesnis nei t_{high} . Jei taip, kraštas išsaugomas.
- Kitais atvejais, kraštas atmetamas.

Lyginant su kitais aptartais krašto detektoriais, *Canny* krašto aptikimo algoritmas turi šiuos privalumus (Rashmi et al., 2013): mažiau jautrus triukšmui (tai pasiekama naudojant Gauso filtrą), pašalinama ruožų (angl. *streaking*) problema (naudoja „histerezės“ metodika – užuot vieno, naudojami du slenksčiai), tai adaptyvus metodas (vartotojas gali keisti parametrus, pavyzdžiui, standartinio nuokrypio reikšmę, ir tokiu būdu pagerinti algoritmo rezultatą), gradiento kryptis lemia gerą lokalizaciją.

Apibendrinant, kraštų aptikimas gali būti naudojamas programoms, veikiančioms realiuoju laiku, tačiau gali kilti tam tikrų iššūkių. Pavyzdžiui, jautrumas triukšmui, dėl kurio gali atsirasti klaidingų kraštų arba gali būti praleisti svarbūs kraštai. Taip pat sunku tiksliai aptikti kraštus vaizduose su pasikartojančiais raštais arba vienalyte tekstūra, be to, kraštai yra jautrūs dydžio ir orientacijos pokyčiams. Dėl skaičiavimų sudėtingumo, ypač pažangiems algoritmams, gali prireikti optimizuoti realiojo laiko programas. Todėl, taikant kraštų aptikimo metodus, svarbu įvertinti šiuos trūkumus ir, atsižvelgiant į konkrečias vaizdo charakteristikas ir programos reikalavimus, apsvarstyti alternatyvius metodus.

2.5. Metodai, taikomi objektams sekti

Vienas iš svarbių aspektų, lemiančių sekimo algoritmų našumą, yra objektų atvaizdavimas (angl. *representation*). Objekto atvaizdavimas – tai kaip yra modeliuojamas sekamas objektas, o objekto lokalizavimas susijęs su tuo, kaip atliekama atitinkamo objekto paieška kitame kadre.

Kaip jau buvo minėta anksčiau, objekto ribos gali būti apibrėžiamos kaip kontūras. Pasirenkant objektų atvaizdavimui kontūrą, žymiai sumažinamas reikalingos atminties kiekis, nes kontūrai apibūdina tik objektą gaubiančius kraštus (Jalal & Singh, 2012). Taip pat šiuo atvaizdavimo būdu galima geriau reprezentuoti standžius objektus (Yilmaz et al., 2004).

Sekimo tikslumui pagerinti ir paieškos vietai sumažinti yra naudojami objekto vietos kiekviename kadre numatymo algoritmai. Ši prognozė taip pat turi įtakos sistemos patvarumui staigių judėjimo pokyčių ar objekto uždengimo atvejais (Jalal & Singh, 2012). Taigi, tikslus algoritmas gali papildomai pagreitinti visą sekimo procesą. Populiarus ir efektyvus būdas sekimo algoritmui patobulinti yra prognozuojančiųjų filtravimo metodų naudojimas, pavyzdžiui, tokių kaip Kalmano filtras (KF).

2.5.1. Kalmano filtras

Kalmano filtras – tai rekursinis filtravimo algoritmas, pateikiantis kai kurių nežinomų kintamųjų įvertinimus, atsižvelgiant į laikui bėgant gautus matavimus (Agarwal et al., 2016). KF įrodė savo naudingumą, taikant jį įvairioms programoms, ir lieka aktualus, kuriant naujus algoritmus.

Kalmano filtras naudojamas tiesinių dinaminių sistemų būsenoms įvertinti. Proceso modelis apibūdina būsenos pasikeitimą nuo laiko momento $(k-1)$ iki laiko momento k kaip (Youngjoo & Hyochoong, 2018):

$$x_k = Fx_{k-1} + Bu_{k-1} + w_{k-1}, \quad (2.15)$$

čia: F – būsenos perdavimo matrica, pritaikyta ankstesniam būsenos vektoriui x_{k-1} ; B – įvesties valdymo matrica, pritaikyta valdymo vektoriui u_{k-1} ; w_{k-1} – proceso triukšmo vektorius, baltasis Gauso triukšmas su nulio vidurkiu ir kovariantiškumu Q ($w_{k-1} \sim N(0, Q)$).

Sistemos matrica F aprašo, kaip sistemos būsena keičiasi laike, atsitiktinis triukšmo vektorius w apibūdina neapibrėžtumą ar patikimumą modelyje (Rudinskas, 2011). Proceso modelis yra susietas su matavimo modeliu, kuris apibūdina santykį tarp būsenos ir matavimų (Youngjoo & Hyochoong, 2018):

$$z_k = Hx_k + v_k, \quad (2.16)$$

čia: z_k – matavimų vektorius; H – matavimų matrica; v_k – matavimų triukšmo vektorius, baltasis Gauso triukšmas su nulio vidurkiu ir kovariantiškumu R ($v_k \sim N(0, R)$).

Matavimų matricioje H aprašomi sąryšiai tarp matavimų reikšmių ir sistemos būsenos (Rudinskas, 2011).

Kalmano filtro tikslas yra pateikti x_k įvertinimą k laiko momentu, atsižvelgiant į pradinį x_0 įvertį, matavimų eilę $(z_1, z_2, \dots, z_k)$ ir sistemos informaciją, aprašytą F , B , H , Q ir R matricomis. Nors kovariacijos matricos turėtų atspindėti triukšmo statistiką, tikroji triukšmo statistika daugeliui praktinių programų nėra žinoma arba nėra Gauso (Youngjoo & Hyochoong, 2018). Todėl Q ir R paprastai naudojami kaip derinimo parametrai, kuriuos vartotojas gali koreguoti, siekiant norimo našumo.

Kalmano filtro algoritmas susideda iš dviejų etapų: prognozavimo ir korekcijos (atnaujinimo). Apibendrintas KF algoritmas gali būti pateiktas taip (Youngjoo & Hyochoong, 2018):

1. Prognozavimas:

- Prognozuojamos būsenos įvertinimas:

$$\hat{x}_k^- = F\hat{x}_{k-1}^+ + Bu_{k-1}. \quad (2.17)$$

- Prognozavimo klaidos kovariacija:

$$P_k^- = FP_{k-1}^+F^T + Q. \quad (2.18)$$

2. Korekcija:

- Matavimo likinys (angl. *residual*):

$$\tilde{y}_k = z_k - H\hat{x}_k^-. \quad (2.19)$$

- Kalmano stiprinimo matrica:

$$K_k = P_k^-H^T(R + HP_k^-H^T)^{-1}. \quad (2.20)$$

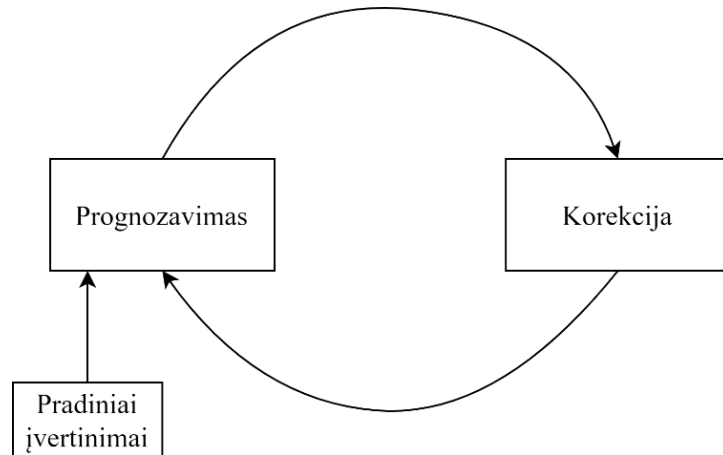
- Atnaujintas būsenos įvertinimas:

$$\hat{x}_k^+ = \hat{x}_k^- + K_k\tilde{y}_k. \quad (2.21)$$

- Atnaujinta klaidos kovariacija:

$$P_k^+ = (I - K_kH)P_k^-. \quad (2.22)$$

Anksčiau pateiktose lygtyse operatorius „ $\hat{\cdot}$ “ reiškia kintamojo įvertinimą, t. y. $\hat{x}$ yra x įvertinimas. „ $-$ “ ir „ $+$ “ indeksai atitinkamai reiškia prognozuojamus ir apskaičiuotus (pagal matavimus z_k) įvertinimus. Kalmano filtro veikimas pavaizduotas 2.5 paveiksle pateiktoje schemeje.



2.5 pav. Kalmano filtro veikimo schema (sudaryta autorės, remiantis Youngjoo ir Hyochoong (2018))

Norint taikyti Kalmano filtrą, reikia atlikti inicijavimo etapą. Turi būti apibrėžtos pradinės reikšmės: pradinės būsenos įvertinimas $\hat{x}_0^+$ ir klaidos kovariacijos matrica P_0^+ (Youngjoo & Hyochoong, 2018). Šios dvi pradinės vertės kartu su Q ir R vaidina svarbų vaidmenį, norint pasiekti gerą algoritmo našumą. Pagal „pradinio nežinomojo“ taisyklę (angl. *initial ignorance*) vartotojas turėtų pasirinkti didelę P_0^+ vertę, kad konvergavimas būtų pasiektas greičiau (Youngjoo & Hyochoong, 2018).

Įvertinta būsena $\hat{x}_{k-1}^+$ ir susijusi klaidos kovariacijos matrica P_k^+ su F , B ir Q naudojami sistemos būsenoms $\hat{x}_k^-$ spėti (kitame laiko momento žingsnyje, kai laukiama matavimų z_k) ir susijusiai klaidos kovariacijos matricai P_k^- apskaičiuoti (Rudinskas, 2011).

Korekcijos etape pirmiausia apskaičiuojamas matavimo likinys $\tilde{y}_k$. Tai tikrojo matavimo z_k ir apskaičiuoto matavimo $H\hat{x}_k^-$ skirtumas (dabartinis matavimas įvertinamas padauginant prognozuojamą būseną iš matavimo matricos) (Youngjoo & Hyochoong, 2018). Tada likinys $\tilde{y}_k$ padauginamas iš Kalmano stiprinimo matricos K_k , kad prognozuojamam įvertinimui $\hat{x}_k^-$ būtų pateikta korekcija $K_k\tilde{y}_k$. Gavus atnaujintą būsenos įvertinimą, apskaičiuojama atnaujinta klaidos kovariacijų matrica P_k^+ , kuri bus naudojama kitu laiko momentu. Po korekcijos etapo gauta klaidos kovariacijų matrica yra mažesnė nei prognozuojama klaidos kovariacijų matrica, tai reiškia, kad po korekcijos filtras tiksliau nustato būsenos įvertinimą (Youngjoo & Hyochoong, 2018).

Kalmano filtrai gaunami remiantis prielaida, kad proceso ir matavimo modeliai yra tiesiniai, t. y. juos galima išreikšti matricomis F , B ir H , o proceso ir matavimo triukšmas yra adityvusis Gauso skirstinys (Youngjoo & Hyochoong, 2018). Taigi, Kalmano filtras pateikia optimalų įvertinimą tik tuo atveju, jei šios prielaidos yra tenkinamos.

2.5.2. Išplėstinis Kalmano filtras

Išplėstinis Kalmano filtras yra efektyviai naudojamas netiesiniams modeliams. Skaičiavimo požiūriu šiam filtrui reikia mažiau resursų nei kitiems netiesiniams filtravimo metodams, pavyzdžiui, dalelių filtrui. Todėl šis filtras naudojamas įvairioms realiojo laiko programoms, pavyzdžiui, tokioms kaip navigacijos sistemos (Youngjoo & Hyochoong, 2018).

Išplėstinį Kalmano filtrą galima vertinti kaip netiesinę Kalmano filtro versiją, kuri linearizuoja modelius, atsižvelgiant į dabartinį įvertinimą. Netiesinės sistemos būsenos perdavimo ir matavimo modelius galima apibūdinti taip (Youngjoo & Hyochoong, 2018):

$$x_k = f(x_{k-1}, u_{k-1}) + w_{k-1}, \quad (2.23)$$

$$z_k = h(x_k) + v_k, \quad (2.24)$$

čia: f – ankstesnės būsenos x_{k-1} ir įvesties valdymo u_{k-1} funkcija, kuri suteikia dabartinę būseną x_k ; h – matavimo funkcija, susiejanti dabartinę būseną x_k ir matavimą z_k ; w_{k-1} – proceso modelio triukšmo vektorius, Gauso triukšmas su Q kovariantiškumu; v_k – matavimų modelio triukšmo vektorius, Gauso triukšmas su R kovariantiškumu.

Išplėstinio Kalmano filtro algoritmo įgyvendinimui kiekvienu laiko momentu reikia gauti kiekvieno modelio Jakobiano matricą, vektoriaus funkcijos pirmosios eilės dalinę išvestinę, vektoriaus atžvilgiu (Youngjoo & Hyochoong, 2018):

$$F_{k-1} = \left. \frac{\delta f}{\delta x} \right|_{\hat{x}_{k-1}^+, u_{k-1}}, \quad (2.25)$$

$$H_k = \left. \frac{\delta h}{\delta x} \right|_{\hat{x}_k^-}. \quad (2.26)$$

F ir H indeksai yra išlaikomi, nes kiekvienu laiko momentu matricos dažnai keičiasi, priklausomai nuo būsenos vektoriaus reikšmių. Tokiu būdu modeliai linearizuojami, atsižvelgiant į dabartinį įvertinimą (Youngjoo & Hyochoong, 2018). Apibendrintas šio filtravimo algoritmas (Youngjoo & Hyochoong, 2018):

1. Prognozavimas:

- Prognozuojamos būsenos įvertinimas:

$$\hat{x}_k^- = f(\hat{x}_{k-1}^+, u_{k-1}). \quad (2.27)$$

- Prognozavimo klaidos kovariacija:

$$P_k^- = F_{k-1} P_{k-1}^+ F_{k-1}^T + Q. \quad (2.28)$$

2. Korekcija:

- Matavimo likinys:

$$\tilde{y}_k = z_k - h(\hat{x}_k^-). \quad (2.29)$$

- Kalmano stiprinimo matrica:

$$K_k = P_k^- H_k^T (R + H_k P_k^- H_k^T)^{-1}. \quad (2.30)$$

- Atnaujintas būsenos įvertinimas:

$$\hat{x}_k^+ = \hat{x}_k^- + K_k \tilde{y}_k. \quad (2.31)$$

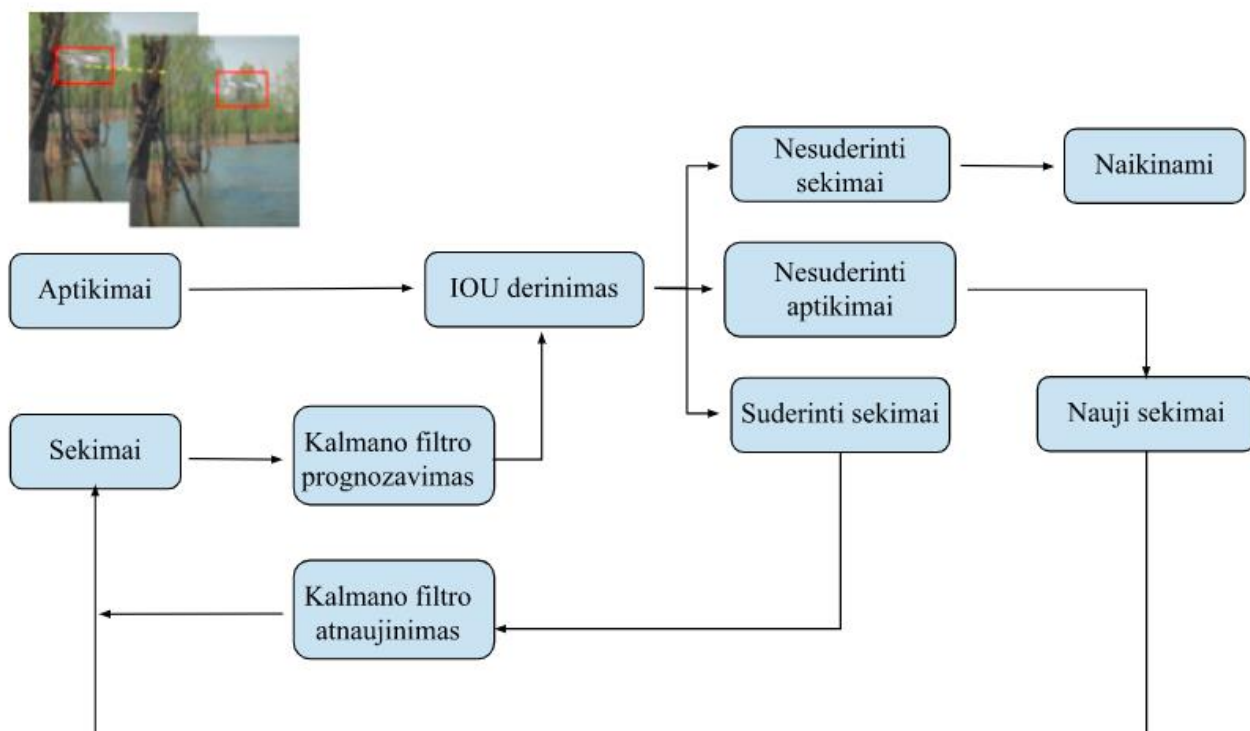
- Atnaujinta klaidos kovariacija:

$$P_k^+ = (I - K_k H_k) P_k^-. \quad (2.32)$$

Algoritmas yra labai panašus į Kalmano filtro algoritmą, pagrindinis skirtumas yra tai, kad išplėstinis Kalmano filtras gauna prognozuojamą būsenos įvertinimą ir prognozuojamą matavimą netiesinėmis funkcijomis: $f(x_{k-1}, u_{k-1})$ ir $h(x_k)$.

2.5.3. SORT algoritmas

SORT algoritmo branduolį sudaro Kalmano filtras ir Vengrų algoritmas (2.6 pav.). Kalmano filtro algoritmas yra padalintas į du procesus: prognozavimą ir atnaujinimą. Algoritmas apibrėžia taikinio judėjimo būseną kaip aštuonis normaliai paskirstytus vektorius (Hong et al., 2021). Kai taikiny pradeda judėti, dabartinio kadro taikinio aptikimo padėtis ir greitis yra nuspėjami pagal taikinio aptikimo kadrą ir taikinio greitį ankstesniame kadre. Šis procesas apibūdina Kalmano filtro prognozavimo metodą. Kalmano filtro atnaujinimo procesas yra pagrįstas numatoma ankstesnio kadro verte ir stebima dabartinio kadro verte (numatoma vertė ir stebima vertė atitinka normalųjį pasiskirstymą), kurios yra tiesiškai svertinės, kad būtų gauta esama numatoma sistemos būsena (Hong et al., 2021). Tada Vengrų algoritmas išsprendžia atitikimo problemą.



2.6 pav. SORT algoritmo schema (sudaryta autorės, remiantis Hong et al. (2021))

Siekiant apdorojimo realiuoju laiku, SORT buvo kuriamas kaip paprastas algoritmas, vengiant sudėtingų ir daug laiko reikalaujančių skaičiavimų. Tačiau vienas iš didžiausių SORT algoritmo trūkumų yra tai, kad šis algoritmas neturi atminties, ir sekamas objektas yra prarandamas, jei SORT nesugeba jo aptikti vaizdo kadre (Nyström, 2019). Taigi, SORT yra greitas ir paprastas algoritmas, pasižymintis dideliu tikslumu ir preciziškumu, tačiau šio algoritmo našumas blogėja, pavyzdžiui, kai objektas yra uždengiamas.

„Deep SORT“ algoritmas yra patobulintas remiantis SORT algoritmu. Svarbiausias šio algoritmo patobulinimas, lyginant su SORT algoritmu, yra giliųjų tinklų įvedimas, kad būtų galima išgauti išvaizdos požymius (Hong et al., 2021). Ši operacija taip pat žymiai sumažina ID pasikeitimo skaičių sekimo algoritme (kai tas pats objektas yra aptinkamas kaip kitas objektas skirtinguose kadruose). „Deep SORT“ algoritmą galima suskirstyti į tris etapus (Hong et al., 2021): trajektorijos takelių numatymas, naudojant Kalmano filtrą; numatytų trajektorijų takelių suderinimas su aptikimu dabartiniame kadre, įskaitant pakopinį atitikimą ir aibių sankirtos ir sąjungos santykio vertę (angl. *Intersection over Union* – IoU), naudojant Vengrų algoritmą; KF atnaujinimas.

Taigi, pagrindiniai SORT ir „Deep SORT“ trūkumai yra skaičiavimų sudėtingumas, objekto ID pasikeitimai ir trajektorijų suskaidymai uždengimo atveju, priklausomybė nuo tikslaus objekto aptikimo, taip pat dėl reikšmingų išvaizdos pokyčių sekimas gali nepavykti.

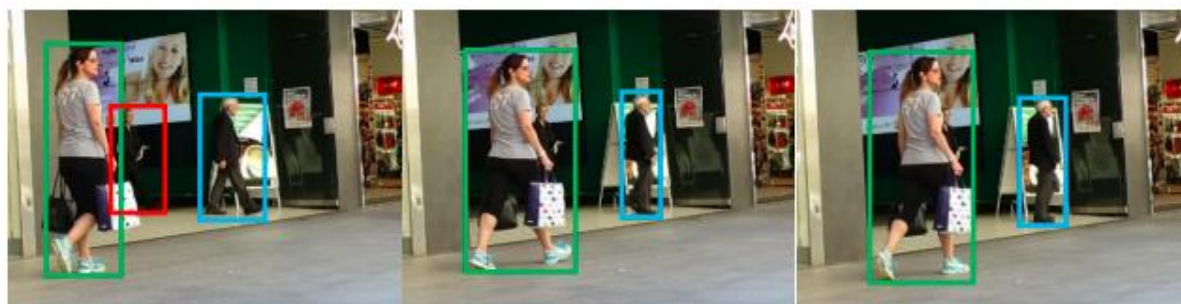
2.5.4. BYTE algoritmas

Sekimas pagal aptikimą yra efektyviausia šiuo metu prieinama kelių objektų sekimo (angl. *Multi-Object Tracking* – MOT) paradigma. Dėl sudėtingų vaizdo įrašų scenarijų detektoriai yra linkę daryti netobulas prognozes. Gavus aptikimo langus įvairiais detektoriais, dauguma MOT metodų išlaiko tik aukšto įverčio aptikimo langus, algoritme nurodant tam tikrą slenkstį, ir naudoja tuos langus kaip duomenų susiejimo įvestį (Zhang et al., 2022). Šis sprendimas yra susijęs su tuo, kad žemo įverčio aptikimo languose yra daug fono, kuris kenkia sekimo našumui. Tačiau, tuo pačiu metu, uždengti objektai gali būti teisingai aptikti tuose languose. Todėl, siekiant sumažinti trūkstamų aptikimų skaičių ir išlaikyti trajektorijų vientisumą, Zhang et al. (2022) siūlo metodą, kuriame išlaikomi visi aptikimo langai ir susiejimas taikomas kiekvienam iš jų.

Zhang et al. (2022) darbe nustatyta, kad panašumas su takelio pėdsakais (angl. *tracklets*) padeda atskirti objektus ir foną žemą įvertį turinčiuose aptikimo languose (2.7 pav.).



(a) Aptikimo langai



(b) Takelių pėdsakai, susiejant aukšto įverčio langus

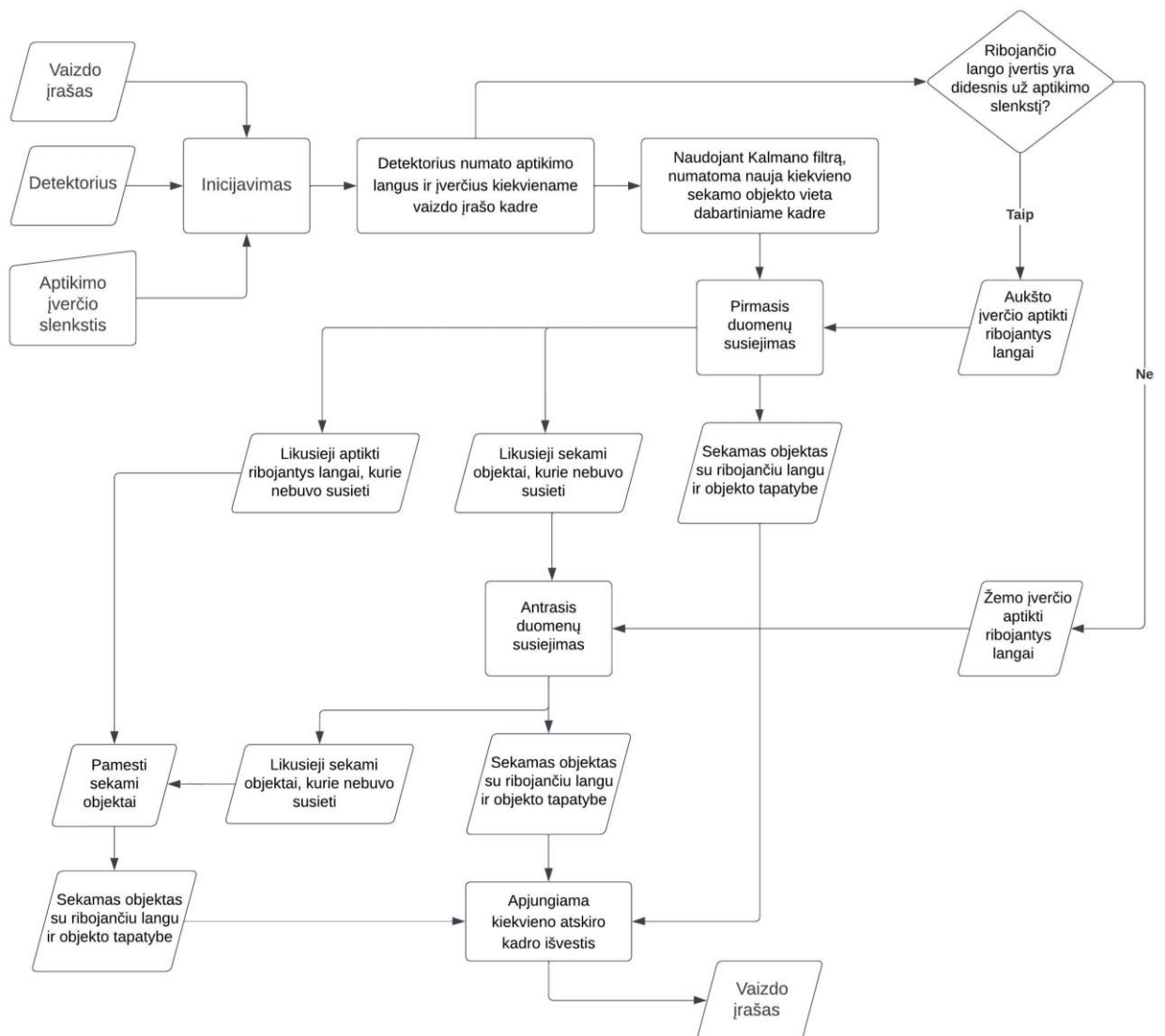


(c) Takelių pėdsakai, susiejant kiekvieną aptikimo langą

2.7 pav. Metodo, susiejančio kiekvieną aptikimo langą, veikimo pavyzdys (sudaryta autorės, remiantis Zhang et al. (2022))

2.7a paveiksle pateikti visi aptikimo langai su jų įverčiais. 2.7b paveiksle rodomi ankstesniais metodais gauti takelių pėdsakai, susiejantys aptikimo langus, kurių įverčiai yra didesni už nustatytą slenkstį (pavyzdžiui, 0,5). Ta pati lango spalva reiškia tą patį aptiktą objektą. 2.7c paveiksle pateikti Zhang et al. (2022) siūlomu metodu gauti takelių pėdsakai. Brūkšniniai langai rodo ankstesnių takelių pėdsakų numatomą langą, gautą naudojant Kalmano filtrą. Du žemo įverčio aptikimo langai yra tinkamai suderinti su ankstesniais takelių pėdsakais, remiantis aukštu IoU įverčiu. Taigi, aptiktas uždengtas objektas (raudonas aptikimo langas 2.7c paveiksle), kurio aptikimo įvertis yra žemas, tinkamai pritaikomas prie ankstesnio takelio pėdsakų, o fonas (dešinėje t_3 kadro dalyje (2.7a pav.) aptiktas langas) yra pašalinamas.

Duomenų susiejimas (angl. *data association*) yra kelių objektų sekimo (MOT) pagrindas. Šio proceso metu pirmiausia apskaičiuojamas takelių pėdsakų ir aptikimo langų panašumas ir, naudojant skirtingas strategijas, jie yra derinami pagal panašumą. Zhang et al. (2022) siūlomo BYTE duomenų susiejimo metodo algoritmas pateiktas 2.8 paveiksle.



2.8 pav. BYTE duomenų susiejimo metodo algoritmas (sudaryta autorės, remiantis Zhang et al. (2022))

BYTE algoritmo įvestis yra vaizdų seka ir objekto detektorius. Taip pat yra nustatomas aptikimo įverčio slenkstis. BYTE išvestis yra vaizdo įrašas su aptiktais sekamais objektais (kiekvienas aptiktas objektas yra su ribojančiu langu ir objekto tapatybe kiekviename kadre).

Naudojant detektorių, kiekviename vaizdo įrašo kadre numatomi aptikimo langai ir jų įverčiai. Visi aptikimo langai padalijami į dvi dalis pagal aptikimo įverčio slenkstį (Zhang et al., 2022). Aptikimo langai, kurių įverčiai yra didesni už pasirinktą slenkstį, išsaugomi kaip aukšto įverčio aptikimo langai. O langai, kurių įverčiai yra žemesni už slenkstį, išsaugomi kaip žemo įverčio aptikimo langai. Atskyrus žemo ir aukšto įverčio aptikimo langus, naudojamas Kalmano filtras, kad būtų numatyta nauja kiekvieno sekamo objekto vieta dabartiniame kadre (Zhang et al., 2022).

Pirmasis duomenų susiejimas atliekamas tarp aukšto įverčio aptikimo langų ir visų sekamų objektų (įskaitant pamestus sekamus objektus). Panašumą galima apskaičiuoti pagal IoU arba išvaizdos požymių atstumus tarp aptikimo langų (turinčių aukštą įvertį) ir numatomo sekamo objekto lango (Zhang et al., 2022). Tada pritaikomas Vengrų algoritmas, kad būtų užbaigtas atitikimas (angl. *matching*) pagal panašumą. Likusieji aptikimai ir sekami objektai, kurie nebuvo susieti, išsaugomi atskirai.

Antrasis duomenų susiejimas atliekamas tarp žemo įverčio aptikimo langų ir po pirmojo susiejimo likusiųjų sekamų objektų. Šįkart likusieji sekami objektai ir vėl išsaugomi atskirai, o likusieji žemo įverčio aptikimo langai yra tiesiog ištrinami, nes juos algoritmas mato kaip foną (Zhang et al., 2022). Antrajam susiejimui panašumas apskaičiuojamas naudojant tik IoU, nes žemo įverčio aptikimo languose paprastai yra objekto uždengimų arba judesio neryškumų (išvaizdos požymiai nėra patikimi šiuo atveju).

Po duomenų susiejimo likusieji sekami objektai bus ištrinti iš takelių pėdsakų. Tolimojo ryšio susiejimui būtina išsaugoti jų tapatumą, todėl po antrojo susiejimo jie išsaugomi kaip pamesti sekami objektai (jie gali būti ištrinti po nustatyto kadrų skaičiaus). Galiausiai, po pirmojo susiejimo likusieji aukšto įverčio aptikimo langai inicijuojami kaip nauji sekami objektai (Zhang et al., 2022). Kiekvieno atskiro kadro išvestis yra dabartiniame kadre esančių sekamų objektų ribojantys langai ir jų tapatybės (pamestų sekimų langai nėra išvedami).

BYTE buvo palygintas su kitais populiariais duomenų susiejimo metodais (Zhang et al., 2022), naudojant MOT17 duomenų rinkinį (2.2 lent.).

2.2 lentelė. Skirtingų duomenų susiejimo metodų palyginimas, naudojant MOT17 duomenų rinkinį (sudaryta autorės, remiantis Zhang et al. (2022))

Metodas	MOTA	IDF1	IDs
SORT	74,6	76,9	291
„DeepSORT“	75,4	77,2	293
MOTDT	75,8	77,6	273
BYTE	76,6	79,3	159

Iš rezultatų, pateiktų 2.2 lentelėje, galima pastebėti, kad BYTE metodo gautos MOTA ir IDF1 metrikos yra pagerinamos, lyginant su kitais metodais. Taip pat gaunamas mažesnis ID skaičius. Tai pabrėžia žemo įverčio aptikimo langų svarbą ir įrodo BYTE metodo gebėjimą atkurti objektą iš žemą įvertį turinčio lango (Zhang et al., 2022).

Taigi, siekdami panaudoti visus aptikimo langus (turinčius aukštus iki žemus įverčius), Zhang et al. (2022) pateikia paprastą ir veiksmingą duomenų susiejimo metodą BYTE, skirtą keliems objektams sekti. Skatindami pažangiausias MOT galimybes, Zhang et al. (2022) siūlo BYTE metodą naudoti sekimo algoritme „ByteTrack“.

2.6. Skyriaus išvados

1. Judančių objektų aptikimas yra pradinis vaizdo sekos analizės etapas. Kadru skirtumas yra paprasčiausias objektų aptikimo būdas, kuris puikiai veikia esant statiniam fonui. Taikant fono atimties metodą, sukuriamas tinkamas fono modelis, tačiau skaičiavimams atlikti reikia buferio su naujausiomis pikselių vertėmis. Optinis srautas pateikia visą informaciją apie judėjimą, bet tam reikia atlikti daug skaičiavimų.

2. Klasifikavimo algoritmai nustato objektų charakteristikas ir naudoja jas kiekvieno objekto klasei numatyti. Klasifikatorių mokymui svarbu teisingai parinkti požymius, kad būtų gautas aukštas tikslumas.

3. Objekto sekimo metodai gali būti klasifikuojami kaip tašku, branduoliu ir siluetu pagrįstas sekimas. Tašku pagrįsti sekimo algoritmai aptinka objektą kiekviename kadre. Branduoliu arba kontūrais pagrįsto sekimo metodų atveju, aptikimas atliekamas tik tada, kai objektas pasirodo pirmą kartą.

4. Kontūrai apibūdina tik objektą gaubiančius kraštus, todėl, pasirenkant objektų atvaizdavimui kontūrą, yra žymiai sumažinamas reikalingos atminties kiekis.

5. Sekimas vaizdo įrašė yra skirtas taikiniams nuoseklyose vaizdo kadruose susieti. Kai objektai greitai juda, lyginant su kadru dažniu, susiejimas gali būti ypač sudėtingas. Taip pat

problemos sudėtingumą padidina scenarijai, kai sekamas objektas laikui bėgant keičia judėjimo kryptį ar yra uždengiamas.

6. Sekimas pagal aptikimą yra efektyviausia šiuo metu prieinama kelių objektų sekimo paradigma. Dėl sudėtingų vaizdo įrašų scenarijų detektoriai yra linkę daryti netobulas prognozes. Todėl, sekant objektus, yra svarbu sumažinti trūkstamų aptikimų skaičių ir išlaikyti trajektorijų vientisumą.

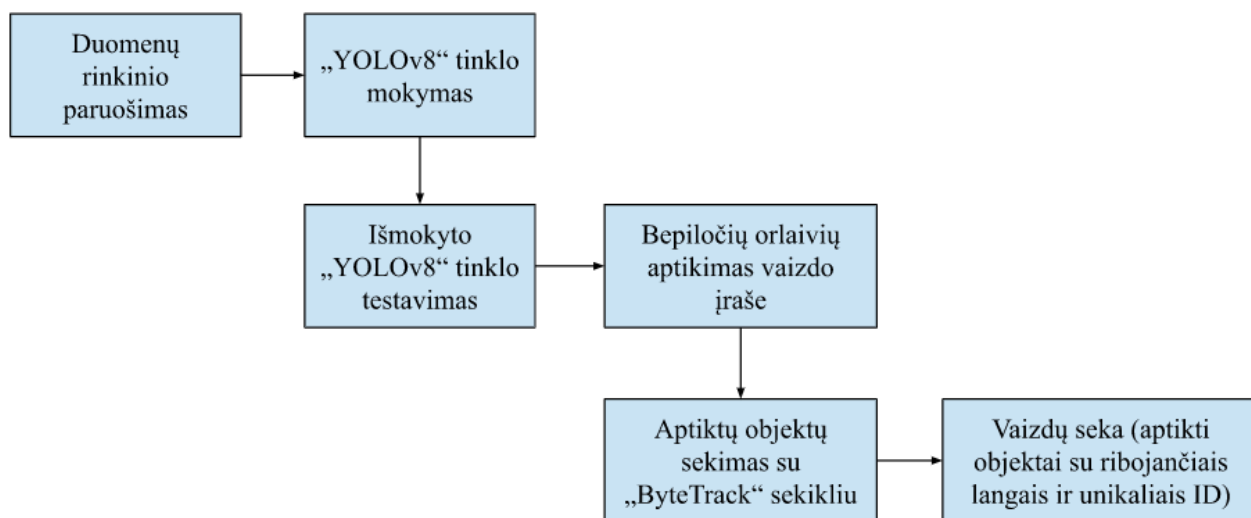
7. Atsižvelgiant į atliktą literatūros bei metodų apžvalgą ir analizę, baigiamajame darbe kuriamam vizualinio taikinių sekimo algoritmui nuspręsta pasirinkti YOLO šeimos tinklą, kuris bus naudojamas bepiločiams orlaiviams aptikti, ir keliems objektams sekti skirtą BYTE duomenų susiejimo metodą.

3. BEPILOČIŲ ORLAIVIŲ SEKIMO SISTEMOS KŪRIMAS

Objekto sekimas susideda iš trijų pagrindinių žingsnių: aptikimas, sekimas ir anotavimas. Baigiamajame darbe kiekvienam iš šių žingsnių naudojami naujausi įrankiai – „YOLOv8“, „ByteTrack“ ir „Roboflow Supervision“. Bepiločio orlaivio sekimo sistemos kūrimui buvo naudojama nemokama debesijos paslauga – integruota programų kūrimo aplinka (angl. *Integrated Development Environment* – IDE) „Google Colaboratory“.

Objektų aptikimas yra kompiuterinės regos užduotis. Aptikimo metu neuroninis tinklas numato objektus vaizdo kadre ir aplink juos nubrėžia ribojančius langus. YOLO yra objektų aptikimo vaizduose metodas, kuris gali būti taikomas realiuoju laiku. „ByteTrack“ yra objektų sekimo algoritmas, naudojamas kompiuterinės regos ir mašininio mokymosi algoritmuose objektams sekti vaizdo įrašuose arba srautinio vaizdo siuntimo metu (angl. *streaming video*). „Roboflow Supervision“ yra paketas, skirtas vaizdams anotuoti ir ženklinti, kuris gali būti integruotas su kitais kompiuterinės regos įrankiais.

Bepiločių orlaivių sekimo sistemos kūrimo algoritmas pateiktas A priede, 3.1 paveiksle pateiktas supaprastintas algoritmas.

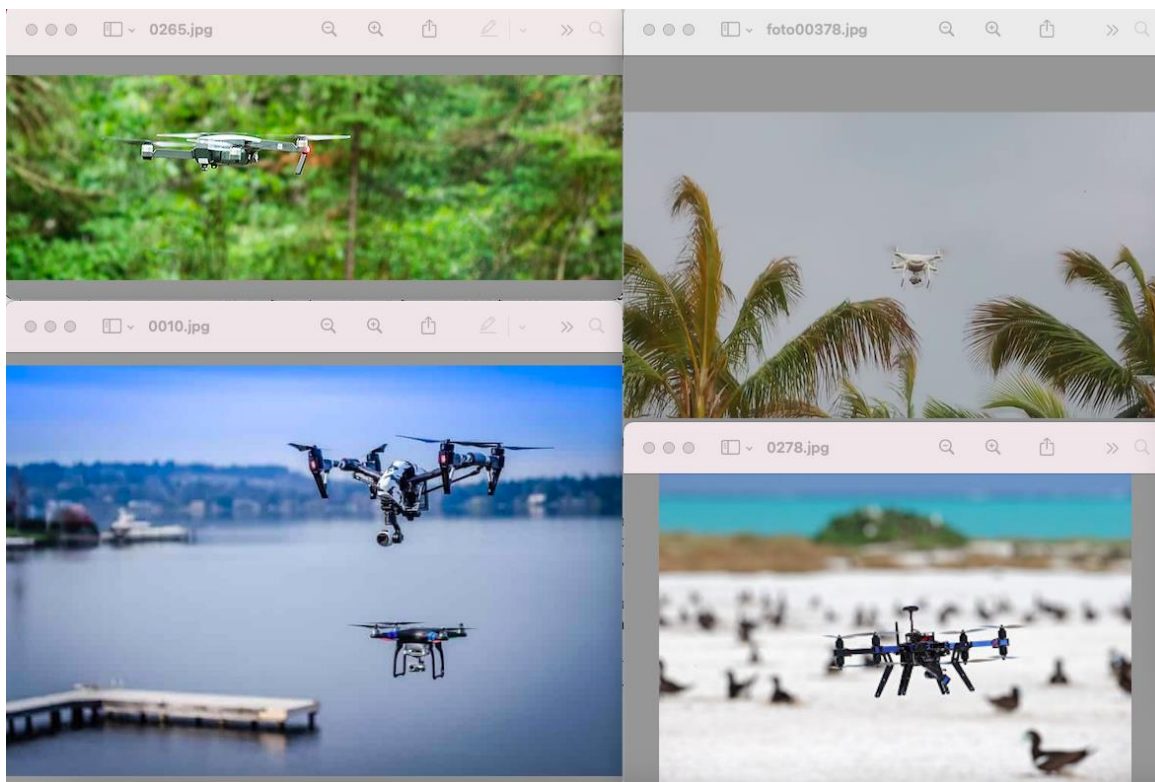


3.1 pav. Supaprastintas bepiločių orlaivių sekimo sistemos kūrimo algoritmas (sudaryta autorės)

Pirmiausia buvo paruošti duomenų rinkiniai (po žymėjimo išsaugoti vaizdai ir „.txt“ failai). Toliau paruošti vaizdai buvo naudojami „YOLOv8“ tinklui mokyti (tinklo įvestis). Pagal išmokyto tinklo testavimo metu gautus rezultatus buvo sprendžiama ar tęsti mokymą, ar išsaugoti geriausius gautus svorius, kurie toliau naudojami sekikliui. Geriausi gauti išmokyto „YOLOv8“ tinklo svoriai buvo naudojami bepiločiams orlaiviams aptikti vaizdo sekoje (aptikus dronus, tinklo išvestis yra ribojantis langas su klasės tikimybe). Gauti ribojantys langai perduodami į „ByteTrack“ sekiklį, kuris palygina aptiktus objektus kiekviename kadre ir priskiria unikalų ID. Galiausiai, apdoroti vaizdo kadrai sujungiami ir išvestyje gaunama vaizdų seka.

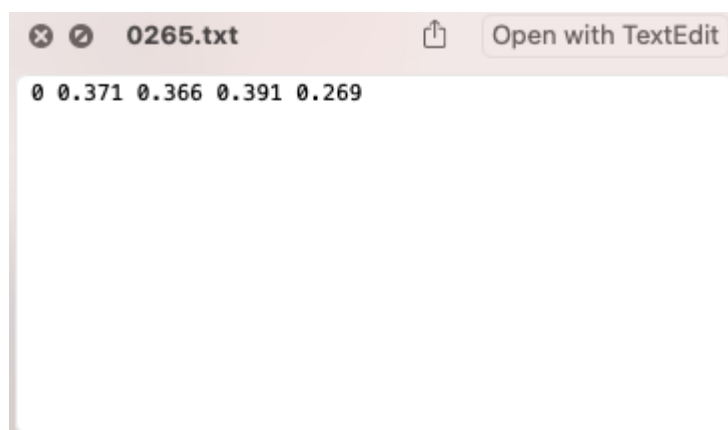
3.1. Duomenų rinkinio paruošimas

Modeliui mokytis ir jo veikimui patikrinti buvo paruošti keli duomenų rinkiniai. Pirmajam duomenų rinkiniui („Drone_dataset“) buvo panaudoti vaizdai, esantys paruoštame duomenų rinkinyje, kuris yra patalpintas „Kaggle“ platformoje (Özel, n. d.). Duomenų rinkinį sudaro 1 359 bepiločių orlaivių nuotraukos, darytos iš skirtingų aukščių, kampų ir esant skirtingiems fonams, todėl duomenų rinkinys yra įvairus (3.2 pav.).



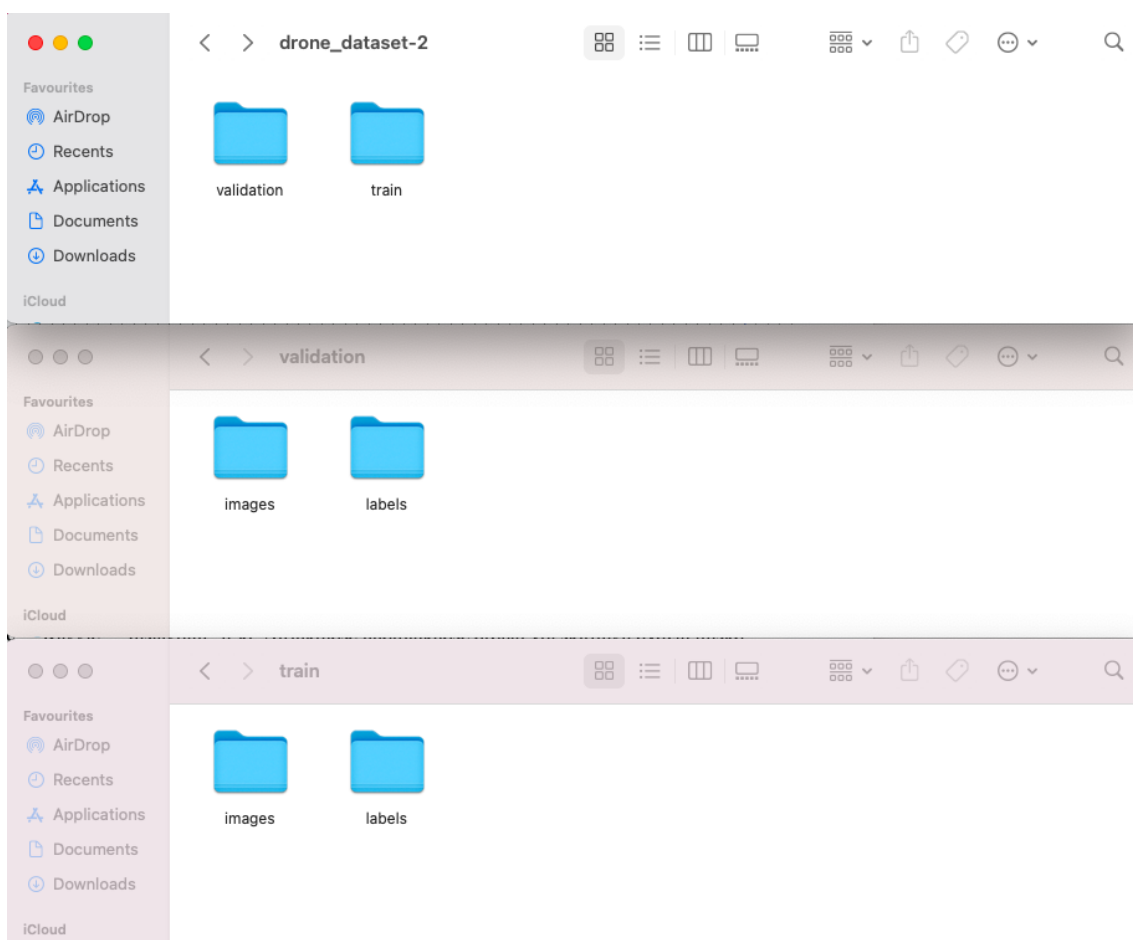
3.2 pav. „Drone_dataset“ duomenų rinkinio vaizdų pavyzdžiai (sudaryta autorės)

YOLO tinklui mokytis reikalaujama, kad informacija apie žymėtus vaizdus būtų išsaugota „.txt“ formatu (3.3 pav.). Faile turi būti saugomos keturios koordinatės (duomenys apie objektą ribojantį langą) ir klasės numeris (šiam darbe kuriamas modelis turi aptikti tik bepilotį orlaivį, todėl yra tik viena klasė – „0“).



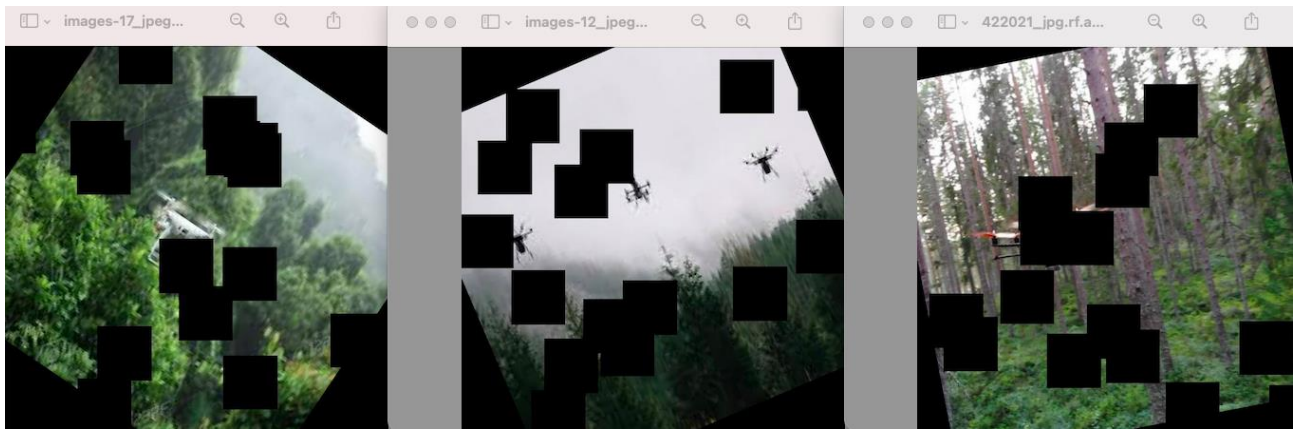
3.3 pav. Failo su informacija apie žymėtą vaizdą pavyzdys (sudaryta autorės)

Duomenų rinkinys buvo padalintas į dvi dalis (3.4 pav.), skirtas mokymui (1 132 nuotraukų) ir patikrinimui (227 nuotraukų).



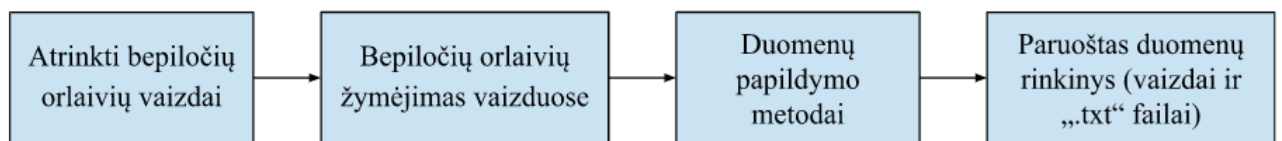
3.4 pav. „Drone_dataset“ duomenų rinkinio pakas (sudaryta autorės)

Antrajam duomenų rinkiniui („Drones“) bepiločių orlaivių vaizdai buvo renkami iš viešųjų išteklių, tokių kaip „Google“, „Kaggle“, „Instagram“ ir kt. Atrinktose nuotraukose bepiločiai orlaiviai yra miško fone, kuriame juos yra sunkiau atpažinti. Siekiant duomenų įvairovės ir dalinio objekto (drono) uždengimo vaizduose, buvo naudojami duomenų papildymo (angl. *data augmentation*) metodai. Bepiločiams orlaiviams sužymėti pasirinktuose vaizduose ir naujiems vaizdams generuoti buvo naudojama „Roboflow“ platforma. Šį duomenų rinkinį sudaro 45 mokymui skirti vaizdai ir 5 vaizdai patikrinimui (3.5 pav.).



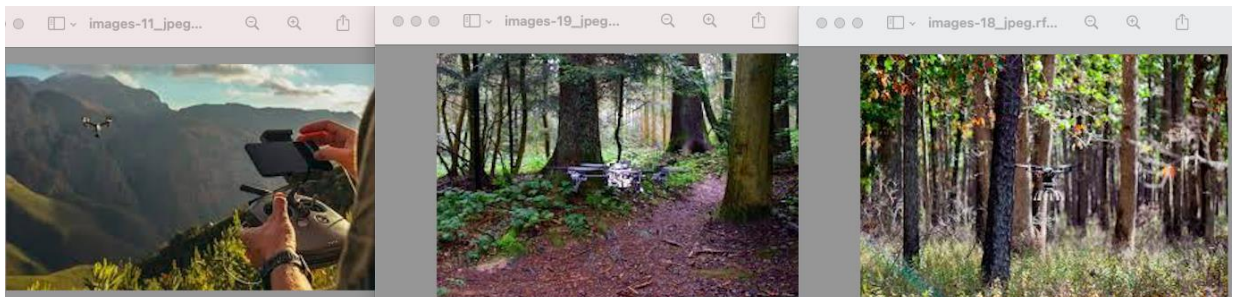
3.5 pav. „Drones“ duomenų rinkinio vaizdų pavyzdžiai (sudaryta autorės)

Duomenų rinkinio paruošimo žingsniai pateikti 3.6 paveiksle.



3.6 pav. Duomenų rinkinio paruošimo schema (sudaryta autorės)

Išmokyto modelio veikimui patikrinti viešuosiuose ištekliuose buvo atrinkta dar 20 bepiločių orlaivių nuotraukų (3.7 pav.). Šis duomenų rinkinys („Drones_test“) naudojamas išmokytiems modeliams testuoti ir palyginti tarpusavyje.



3.7 pav. „Drones_test“ duomenų rinkinio vaizdų pavyzdžiai (sudaryta autorės)

Visi trys duomenų rinkiniai buvo suglaudinti, kad vėliau būtų naudojami tinklui mokytis.

3.2. Naudojamos metrikos

Kiekybinė analizė gali būti atliekama pagal šiuos parametrus (Bathija & Sharma, 2019): tikrai teigiamas (angl. *True Positive* – TP), tikrai neigiamas (angl. *True Negative* – TN), klaidingai teigiamas (angl. *False Positive* – FP) ir klaidingai neigiamas (angl. *False Negative* – FN):

- TP – kai modelis teisingai numato teigiamą objekto klasę (modelis nusprendžia, kad objektas yra, ir jis iš tikrųjų egzistuoja – teisingai atpažino);
- FP – kai modelis neteisingai numato teigiamą objekto klasę (modelis nusprendžia, kad objektas yra, nors jis iš tikrųjų neegzistuoja – klaidingai atpažino);

○ FN – kai modelis neteisingai numato neigiamą objekto klasę (modelis nusprendžia, kad objekto nėra, nors jis egzistuoja – nesugebėjo atpažinti);

○ TN – kai modelis teisingai numato neigiamą objekto klasę (modelis nusprendžia, kad objekto nėra, kai jis iš tikrųjų neegzistuoja).

Pavyzdžiui, tikslumui (angl. *accuracy*) nustatyti, apskaičiuojamas tikrai teigiamų ir tikrai neigiamų prognozių sumos bei visų prognozių santykis (Bathija & Sharma, 2019):

$$\text{Tikslumas} = \frac{TP+TN}{\text{iš viso}}. \quad (3.1)$$

Preciziškumas (angl. *precision*) apibrėžiamas kaip tikrai teigiamų ir visų teigiamų prognozių santykis (Bathija & Sharma, 2019):

$$\text{Preciziškumas} = \frac{TP}{TP+FP}. \quad (3.2)$$

Jautrumas (angl. *recall*) apibrėžiamas kaip tikrai teigiamų prognozių bei tikrai teigiamų ir klaidingai teigiamų prognozių santykis (Bathija & Sharma, 2019):

$$\text{Jautrumas} = \frac{TP}{TP+FN}. \quad (3.3)$$

Tikslumas parodo, kiek kartų modelis buvo teisingas bendrai. Preciziškumas parodo, kaip gerai modelis numato tam tikrą kategoriją. Jautrumas parodo, kiek kartų modelis aptiko tam tikrą kategoriją (aukštesnė šios metrikos vertė parodo, kad modelis teisingai identifikuoja objektus ir yra mažesnė tikimybė klasifikuoti neteisingai).

YOLO našumas dažniausiai matuojamas naudojant tris metrikas (Zaman et al., 2021): vidutinio tikslumo vidurkis (angl. *Mean Average Precision* – mAP), preciziškumas, jautrumas.

mAP yra matas, kuris sujungia jautrumą ir preciziškumą aptikimo tikslumui nustatyti, kai yra apskaičiuojama vidutinė preciziškumo vertė jautrumo vertei (Zaman et al., 2021). mAP apskaičiuojamas nustatant vidurkio tikslumą (angl. *Average Precision* – AP) kiekvienai klasei, o tada – kelių klasių vidurkį (Zaman et al., 2021):

$$AP = \frac{\sum_{r=1}^R P_r}{R}, \quad (3.4)$$

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i, \quad (3.5)$$

čia: AP – vidurkio tikslumas; r – prognozės eilės numeris; R – prognozių skaičius; P_r – teigiamų prognozių preciziškumas; N – klasių skaičius; i – klasės eilės numeris.

mAP yra populiari objektų aptikimo modelių vertinimo metrika, nes ji užtikrina gerą preciziškumo ir jautrumo pusiausvyrą. Taip pat atsižvelgiama į tai, kad objektai gali turėti skirtingus dydžius ir proporcijas. mAP įvertis svyruoja nuo 0 (modelis negali tinkamai aptikti jokių objektų) iki 1 (modelis pasiekė puikų preciziškumą ir jautrumą).

Aibių sankirtos ir sąjungos santykio (IoU) vertė rodo numatomų ribojančių langų koordinatų sutapimą su pagrindiniu tiesos langų (angl. *ground truth box*):

$$IoU = \frac{\text{Sankirtos plotas}}{\text{Sąjungos plotas}}. \quad (3.6)$$

Didesnė IoU reikšmė rodo, kad numatomos ribojančio lango koordinatės yra labai panašios į pagrindinio tiesos lango koordinates. IoU slenkstis yra nustatomas iš anksto (baigiamajame darbe naudojamo tinklo atveju IoU slenkstis yra lygus 0,8), kad būtų nustatyta, ar aptikimas yra tikrai teigiamas, ar klaidingai teigiamas (Abhinu et al., 2021).

Baigiamajame darbe mokytam modeliui įvertinti naudojamos nuostolių kreivės, rodančios, kaip keičiasi tinklo mokymo ir vertinimo (angl. *validation*) nuostoliai. Po tinklo mokymo „Google Colaboratory“ IDE gaunamos šios diagramos:

- „Box_loss“ yra nuostolių funkcijos diagrama. Nuostolių funkcija apima du komponentus: lokalizacijos ir pasiklivimo nuostolius. Lokalizacijos nuostoliai matuoja skirtumą tarp numatyto ribojančio lango ir pagrindinio tiesos lango koordinatės, o pasiklivimo nuostoliai – skirtumą tarp numatomos ir pagrindinės tiesos klasės tikimybių ir objektyvumo įverčių. Lokalizacijos nuostoliai paprastai yra didesni ankstyvosiose mokymo stadijose, kai modelis vis dar mokosi tiksliai numatyti ribojančio lango koordinates, o pasiklivimo nuostoliai yra didesni, kai modelis mokosi tiksliai numatyti klasės tikimybes ir objektyvumo įverčius. Vykstant tinklo mokymui, „box_loss“ diagrama turėtų rodyti mažėjimo tendenciją, kas parodo, kad modelis mokosi tiksliai aptikti objektus vaizduose. Lygi arba didėjanti nuostolių funkcijos tendencija gali reikšti, kad modelis mokosi ne taip efektyviai, todėl gali prireikti pakoreguoti modelį arba mokymo parametrus.

- „Cls_loss“ diagrama parodo, kaip kinta pasiklivimo nuostoliai modelio mokymo metu. Vykstant tinklo mokymui, „cls_loss“ diagrama turėtų rodyti mažėjimo tendenciją, kas parodo, kad modelis mokosi tiksliai klasifikuoti objektus vaizduose.

- „Metrics/precision(B)“ ir „metrics/recall(B)“ diagramos rodo ryšį tarp modelio našumo metrikų (preciziškumo ir jautrumo) bei duomenų taškų klasifikavimo kaip teigiamų slenksčio. Šios diagramos naudojamos bendram modelio veikimui įvertinti. Gero modelio kreivė yra arčiau viršutinio dešiniojo grafiko kampo, kas parodo aukštus jautrumo ir preciziškumo įverčius.

- „Metrics/mAP50(B)“ yra vertinimo metrikos mAP (kai sutampa 50 % ribojančių langų) diagrama. Ši metrika apskaičiuojama lyginant pagrindinius tiesos langus su numatytais ribojančiais langais. Numatytas ribojantis langas laikomas tikrai teigiamu, jei jo IoU sutampa su pagrindiniu tiesos langu bent 50 %. Vykstant tinklo mokymui, diagrama turėtų rodyti didėjimo tendenciją, kas parodo, kad modelis tiksliau aptinka objektus. Tačiau gali būti tam tikrų našumo svyravimų, ypač ankstyvosiose mokymo stadijose.

- „Metrics/mAP50–95(B)“ diagramoje pateikiamas apskaičiuotas mAP vidurkis, esant skirtingiems IoU slenksčiams (nuo 0,5 iki 0,95 su 0,05 žingsniu). Aukštas mAP įvertis rodo, kad tinklas gali tiksliai aptikti objektus plačiame pasiklivimo slenksčio diapazone.

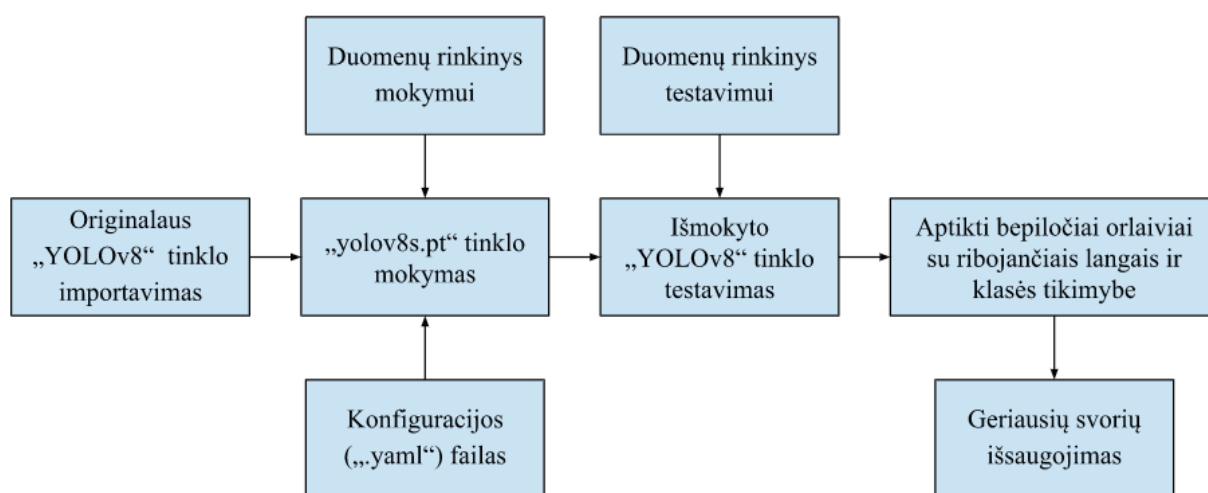
Be to, naudojamas išvedimo laikas, parodantis per kiek laiko algoritmas aptinka visus objektus viename kadre (Zaman et al., 2021). Išvedimo laikas (sekundėmis) gali būti apskaičiuotas pagal (3.7) formulę (Zaman et al., 2021):

$$\text{Išvedimo laikas} = \frac{\text{Apdorojimo laikas}}{\text{Vaizdų skaičius}}. \quad (3.7)$$

Išvedimo laiką taip pat galima išreikšti per sekundę apdorotų kadrų skaičiumi (FPS).

3.3. Modelio mokymas

Modeliui mokyti buvo naudojama nemokama debesijos paslauga „Google Colaboratory“. „YOLOv8“ tinklo mokymo programos kodas pateiktas B priede. Mokymo žingsniai pateikti 3.8 paveiksle.



3.8 pav. YOLO tinklo mokymo schema (sudaryta autorės)

YOLO tinklas buvo importuotas iš mašininio mokymosi ir diegimo platformos „Ultralytics HUB“. Siekiant gauti greitesnį ir tikslesnį rezultatą, pradiniam modelio mokymui buvo naudojamas originalus „yolov8s.pt“ tinklas, kurio svoriai gauti mokymui naudojant COCO duomenų rinkinį.

„YOLOv8“ tinklas buvo pritaikytas baigiamojo darbo užduočiai. Pakeitimai buvo išsaugoti konfigūracijos („yaml“ formato) faile (3.9 pav.).

```

data.yaml ×
1 train: /content/drone_dataset/train/images
2 val: /content/drone_dataset/validation/images
3
4 nc: 1
5 names: ['drone']
  
```

3.9 pav. Tinklo konfigūracijos failas (sudaryta autorės)

Buvo pakeistas klasių skaičius nuo 80 iki 1, nes kuriamas modelis turi aptikti tik bepiločius orlaivius. Taip pat „data.yaml“ faile nurodomas kelias iki naudojamo duomenų rinkinio. Pradiniam

mokymui buvo nuspręsta naudoti hiperparametrų reikšmes, pasiūlytas originaliame „yolov8s.pt“ tinkle.

Tuo metu, kai vyko modelio mokymas, „Google Colaboratory“ skyrė „Tesla T4“ su 15 835 MB atminties NVIDIA GPU. Modelis buvo optimizuotas naudojant stochastinį gradiento nusileidimą (angl. *Stochastic Gradient Descent* – SGD).

Modelio mokymo parametrai:

- nuotraukų skaičius: 1 132 (mokymui) ir 227 (patikrinimui);
- epochų kiekis: 50;
- mokymosi greitis: 0,01;
- paketo dydis: 16;
- aktyvacijos funkcija: SGD.

Po 50 epochų buvo gautas 94,5 % mAP50 ir 66,3 % mAP50–95 (3.10 pav.). Mokymo rezultatai pateikti 3.11 ir 3.12 paveiksluose.

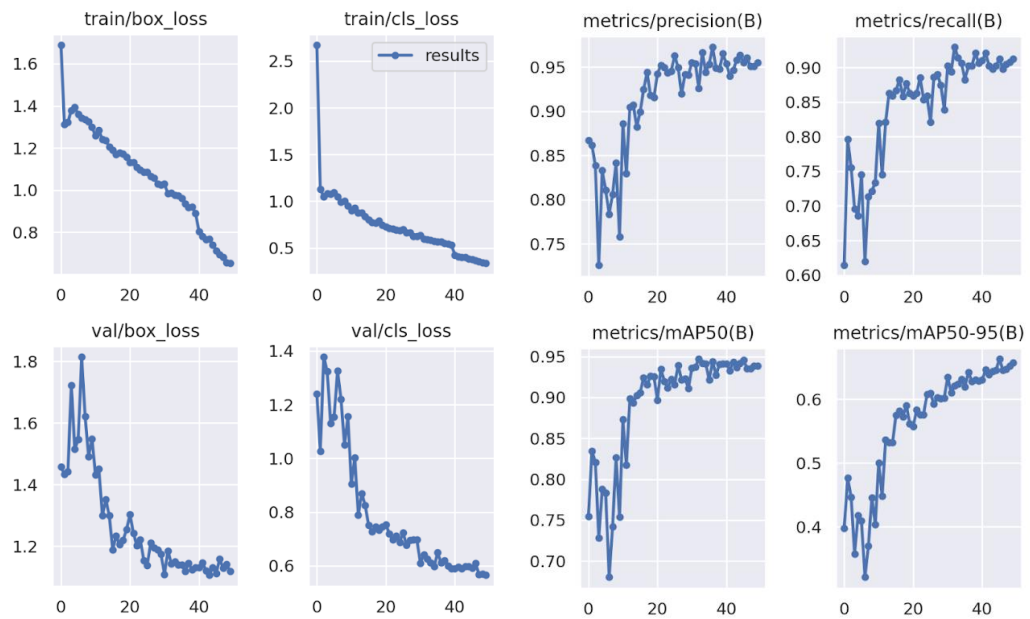
```
50 epochs completed in 1.335 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 22.5MB
Optimizer stripped from runs/detect/train/weights/best.pt, 22.5MB

Validating runs/detect/train/weights/best.pt...
Ultralytics YOLOv8.0.20 Python-3.9.16 torch-1.13.1+cu116 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11125971 parameters, 0 gradients, 28.4 GFLOPs
      Class      Images  Instances  Box(P      R      mAP50  mAP50-95): 12% 1/8 [00:03<00:21,  3.02s/it]
      Class      Images  Instances  Box(P      R      mAP50  mAP50-95): 100% 8/8 [00:09<00:00,  1.17s/it]
           all         227         255    0.955    0.912    0.945    0.663
Speed: 1.2ms pre-process, 5.7ms inference, 0.0ms loss, 1.7ms post-process per image
Results saved to runs/detect/train
```

3.10 pav. Pirmojo bandymo tinklo mokymo pabaiga (sudaryta autorės)



3.11 pav. Pirmojo bandymo bepiločio orlaivio aptikimo rezultatai (sudaryta autorės)



3.12 pav. Pirmojo bandymo tinklo mokymo rezultatai (sudaryta autorės)

Iš nuostalių kreivių (3.12 pav.) matoma, kad modelio mokymas buvo efektyvus:

- „Box_loss“ diagramoje matyti, kad ankstyvosiose mokymo stadijose nuostoliai yra didesni, bet yra mažėjimo tendencija – modelis gerai mokosi tiksliai aptikti objektus vaizduose.
- „Cls_loss“ diagramoje matyti, kad ankstyvosiose mokymo stadijose pasiklivimo nuostoliai taip pat yra didesni, o mažėjimo tendencija parodo, kad modelis gerai mokosi tiksliai klasifikuoti objektus vaizduose.
- „Metrics/precision(B)“ ir „metrics/recall(B)“ diagramose yra svyravimų, tačiau kreivė yra arčiau viršutinio dešiniojo grafiko kampo, tai parodo, kad modelio mokymas buvo sėkmingas.
- „Metrics/mAP50(B)“ diagrama rodo didėjimo tendenciją, svyravimai nenusileidžia žemiau 90 %.
- „Metrics/mAP50-95(B)“ diagramoje taip pat matoma didėjimo tendencija, tačiau, esant skirtingiems IoU slenksčiams, mAP vidurkis yra žemiau 70 %.

Išmokyto modelio veikimui patikrinti buvo panaudotas testavimui skirtas duomenų rinkinys (3.13 pav.).

```
[ ] # Drono aptikimas vaizduose
!yolo task=detect mode=predict model=/content/runs/detect/train/weights/best.pt conf=0.5 source=/content/Drones_test/images/ save=True

Ultralytics YOLOv8.0.20 Python-3.9.16 torch-1.13.1+cu116 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11125971 parameters, 0 gradients, 28.4 GFLOPs
image 1/20 /content/Drones_test/images/422021.jpg.rf.8cd1761e223f8b341c56a25f78cd870e.jpg: 608x800 1 drone, 19.7ms
image 2/20 /content/Drones_test/images/images-10.jpeg.rf.46261397b8f5f64e81764bf36dd4cb41.jpg: 416x800 17.2ms
image 3/20 /content/Drones_test/images/images-11.jpeg.rf.65062b93fc2c4c8b022bcc34e0927ba5.jpg: 480x800 1 drone, 17.1ms
image 4/20 /content/Drones_test/images/images-12.jpeg.rf.e8ec775c73a81eeef88626455a1a3c43f.jpg: 544x800 3 drones, 18.7ms
image 5/20 /content/Drones_test/images/images-13.jpeg.rf.233c7610006aa704f5de9cb69e99f678.jpg: 544x800 18.1ms
image 6/20 /content/Drones_test/images/images-14.jpeg.rf.dc1b02a01606fe6588333dcfcd3258d4.jpg: 544x800 18.1ms
image 7/20 /content/Drones_test/images/images-15.jpeg.rf.4e270804df50f4d1631b3cb797337973.jpg: 544x800 1 drone, 18.1ms
image 8/20 /content/Drones_test/images/images-16.jpeg.rf.7c52aae1e02a24671ece8f9c19eadcb.jpg: 448x800 1 drone, 17.0ms
image 9/20 /content/Drones_test/images/images-17.jpeg.rf.bde2378c6d0c3fb0126bfd95a2f18ba.jpg: 544x800 1 drone, 16.1ms
image 10/20 /content/Drones_test/images/images-18.jpeg.rf.aa254a358ec4c13b65ba4453153971b5.jpg: 544x800 12.0ms
image 11/20 /content/Drones_test/images/images-19.jpeg.rf.a3c0ad6659ea1978d0ed339b456665cc.jpg: 544x800 12.0ms
image 12/20 /content/Drones_test/images/images-2.jpeg.rf.821cfff732f83008fe1a6ae0b1c54e59.jpg: 544x800 12.1ms
image 13/20 /content/Drones_test/images/images-3.jpeg.rf.e186c5c8b9beb21cea8183aef16bf503.jpg: 480x800 11.3ms
image 14/20 /content/Drones_test/images/images-4.jpeg.rf.4fae04243cbb1494e4a352aff2efb5bb.jpg: 512x800 13.8ms
image 15/20 /content/Drones_test/images/images-5.jpeg.rf.535c0069cc53207ff3a7050dd99c70a3.jpg: 448x800 1 drone, 11.1ms
image 16/20 /content/Drones_test/images/images-6.jpeg.rf.0083c55d18db3f6abfa1fa0b1b930a2d.jpg: 576x800 13.8ms
image 17/20 /content/Drones_test/images/images-7.jpeg.rf.ff3640779cdf2d84e767d20daa1375ed.jpg: 544x800 1 drone, 12.6ms
image 18/20 /content/Drones_test/images/images-8.jpeg.rf.83b49a44b8af9283c75fc5ddb8e5e1e8.jpg: 384x800 14.1ms
image 19/20 /content/Drones_test/images/images-9.jpeg.rf.8445a96472da6f083c758d4fe8bb0f09.jpg: 544x800 1 drone, 12.5ms
image 20/20 /content/Drones_test/images/images.jpeg.rf.03b8fc245bfe2755f5599f9f2942e9c2.jpg: 448x800 1 drone, 11.0ms
Speed: 0.5ms pre-process, 14.8ms inference, 1.0ms postprocess per image at shape (1, 3, 800, 800)
Results saved to runs/detect/predict
```

3.13 pav. Pirmojo bandymo bepiločio orlaivio aptikimo rezultatai, naudojant testavimo duomenų rinkinį (sudaryta autorės)

Kaip matoma iš gautų rezultatų, pateiktų 3.13 paveiksle, bepiločiai orlaiviai buvo atpažinti 10 vaizdų iš 20. Toks modelio veikimas neatitinka darbe iškeltų tikslų, todėl buvo nuspręsta tęsti tinklo mokymą, naudojant pirmojo mokymo metu gautus geriausius svorius ir kitą paruoštą duomenų rinkinį.

Modelio mokymo parametrai:

- nuotraukų skaičius: 45 (mokymui) ir 5 (patikrinimui);
- epochų kiekis: 50;
- mokymosi greitis: 0,05;
- paketo dydis: 16;
- aktyvacijos funkcija: SGD.

Po 50 epochų buvo gautas 92,8 % mAP50 ir 48,8 % mAP50–95 (3.14 pav.). Tinklo mokymas buvo pratęstas (parametrai nebuvo keisti), siekiant geresnio rezultato.

```
50 epochs completed in 0.069 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 22.5MB
Optimizer stripped from runs/detect/train/weights/best.pt, 22.5MB

Validating runs/detect/train/weights/best.pt...
Ultralytics YOLOv8.0.20 Python-3.9.16 torch-1.13.1+cu116 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11125971 parameters, 0 gradients, 28.4 GFLOPs
Class Images Instances Box(P R mAP50 mAP50-95): 100% 1/1 [00:00<00:00, 2.30it/s]
all 5 5 0.832 0.991 0.928 0.488
Speed: 0.4ms pre-process, 9.8ms inference, 0.0ms loss, 1.1ms post-process per image
Results saved to runs/detect/train
```

3.14 pav. Antrojo bandymo tinklo mokymo tarpinis rezultatas po 50 epochų (sudaryta autorės)

Po dar 50 epochų buvo gautas 96,2 % mAP50 ir 51,9 % mAP50–95 (3.15 pav.). Tinklo mokymas dar kartą pratęstas, pakeičiant mokymosi greitį į 0,01.

```

50 epochs completed in 0.070 hours.
Optimizer stripped from runs/detect/train2/weights/last.pt, 22.5MB
Optimizer stripped from runs/detect/train2/weights/best.pt, 22.5MB

Validating runs/detect/train2/weights/best.pt...
Ultralytics YOLOv8.0.20 Python-3.9.16 torch-1.13.1+cu116 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11125971 parameters, 0 gradients, 28.4 GFLOPs

```

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95)
all	5	5	0.816	1	0.962	0.519

```

Speed: 0.4ms pre-process, 9.9ms inference, 0.0ms loss, 1.6ms post-process per image
Results saved to runs/detect/train2

```

3.15 pav. Antrojo bandymo tinklo mokymo tarpinis rezultatas po 100 epochų (sudaryta autorės)

Galiausiai, po 150 epochų buvo pasiektas 99,5 % mAP50 ir 53,3 % mAP50–95 (3.16 pav.).

Mokymo rezultatai pateikti 3.17 ir 3.18 paveiksluose.

```

50 epochs completed in 0.071 hours.
Optimizer stripped from runs/detect/train3/weights/last.pt, 22.5MB
Optimizer stripped from runs/detect/train3/weights/best.pt, 22.5MB

Validating runs/detect/train3/weights/best.pt...
Ultralytics YOLOv8.0.20 Python-3.9.16 torch-1.13.1+cu116 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11125971 parameters, 0 gradients, 28.4 GFLOPs

```

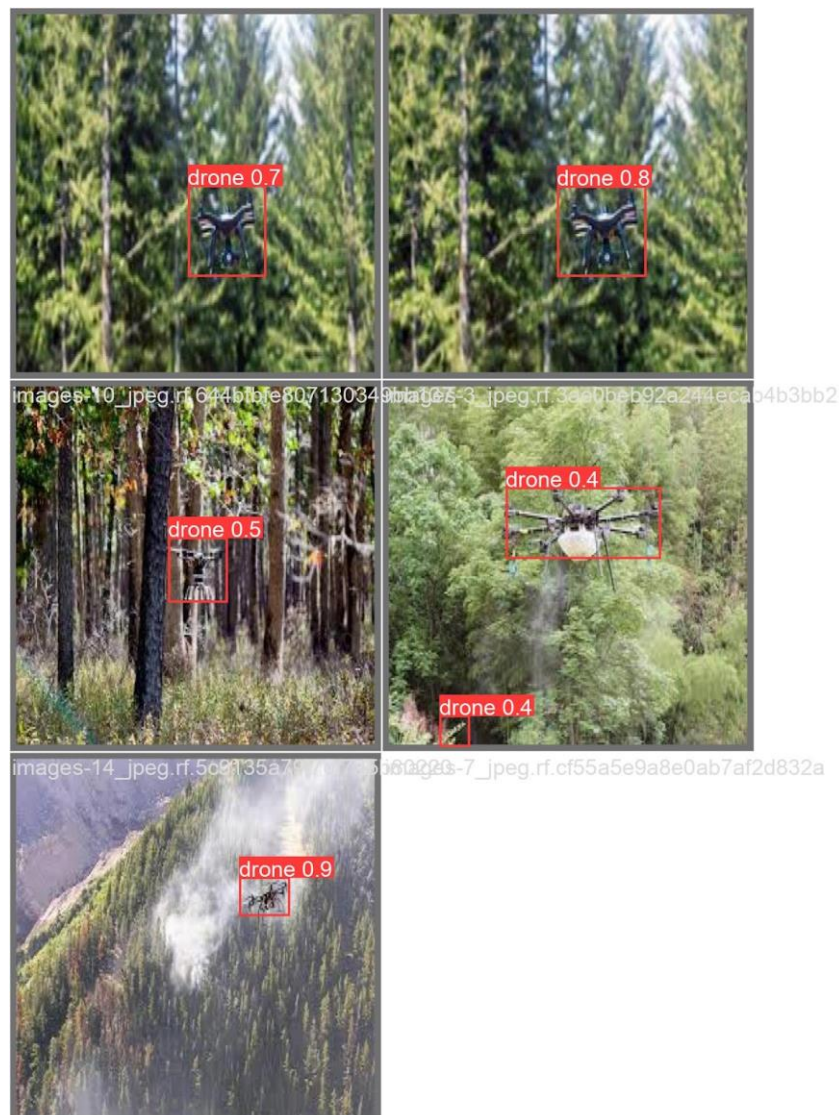
Class	Images	Instances	Box(P)	R	mAP50	mAP50-95)
all	5	5	1	0.944	0.995	0.533

```

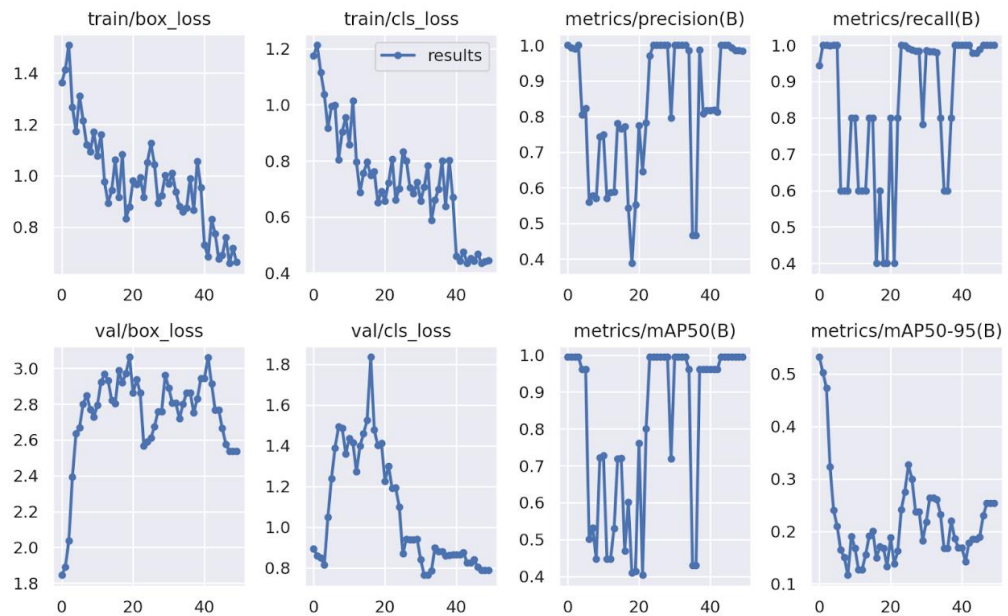
Speed: 0.4ms pre-process, 9.8ms inference, 0.0ms loss, 1.0ms post-process per image
Results saved to runs/detect/train3

```

3.16 pav. Antrojo bandymo tinklo mokymo pabaiga (sudaryta autorės)



3.17 pav. Antrojo bandymo bepiločio orlaivio aptikimo rezultatai (sudaryta autorės)



3.18 pav. Antrojo bandymo tinklo mokymo rezultatai (sudaryta autorės)

Iš nuostolių kreivių (3.18 pav.) matoma, kad modelio mokymas buvo efektyvus. Diagramos atspindi tik paskutinių 50 epochų rezultatus, todėl yra didesnių svyravimų. „Box_loss“ ir „Cls_loss“ diagramose matyti, kad pirmose epochose nuostoliai yra didesni, bet toliau yra mažėjimo tendencija. „Metrics/precision(B)“ ir „metrics/recall(B)“ diagramose yra svyravimų, tačiau kreivė yra arčiau viršutinio dešiniojo grafiko kampo, tai parodo, kad modelio mokymas buvo sėkmingas. „Metrics/mAP50(B)“ diagramoje matoma, kad reikšmės artėja prie 1 – modelis pasiekė puikų preciziškumą ir jautrumą, kai sutampa 50 % ribojančių langų. „Metrics/mAP50–95(B)“ diagramoje matoma, kad pirmomis epochomis įverčiai buvo didesni, bet pradėjo mažėti, todėl galima teigti, kad per paskutines 50 epochų nebuvo pasiektas geresnis rezultatas.

Išmokyto modelio veikimui patikrinti buvo panaudotas testavimui skirtas duomenų rinkinys (3.19 pav.).

```
[ ] # Drono aptikimas vaizduose
!yolo task=detect mode=predict model=/content/runs/detect/train3/weights/best.pt conf=0.5 source=/content/Drones_test/images/ save=True

Ultralytics YOLOv8.0.20 Python-3.9.16 torch-1.13.1+cu116 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11125971 parameters, 0 gradients, 28.4 GFLOPs
image 1/20 /content/Drones_test/images/images-422021.jpg.rf.8cd1761e223f8b341c56a25f78cd870e.jpg: 608x800 1 drone, 19.2ms
image 2/20 /content/Drones_test/images/images-10.jpg.rf.46261397b8f5f64e81764bf36dd4cb41.jpg: 416x800 1 drone, 17.0ms
image 3/20 /content/Drones_test/images/images-11.jpg.rf.65062b93fc2c4c8b022bcc34e0927ba5.jpg: 480x800 1 drone, 17.1ms
image 4/20 /content/Drones_test/images/images-12.jpg.rf.e8ec775c73a81eef88626455a1a3c43f.jpg: 544x800 3 drones, 18.8ms
image 5/20 /content/Drones_test/images/images-13.jpg.rf.233c7610006aa704f5de9cb69e99f678.jpg: 544x800 1 drone, 15.2ms
image 6/20 /content/Drones_test/images/images-14.jpg.rf.dc1b02a01606fe6588333dcfc3258d4.jpg: 544x800 1 drone, 15.2ms
image 7/20 /content/Drones_test/images/images-15.jpg.rf.4e270804df50f4d1631b3cb797337973.jpg: 544x800 1 drone, 15.2ms
image 8/20 /content/Drones_test/images/images-16.jpg.rf.7c52aae1e02a24671ece8f9c19eadcb.jpg: 448x800 1 drone, 14.7ms
image 9/20 /content/Drones_test/images/images-17.jpg.rf.bde2378c6d0c3fb0126bfed95a2f18ba.jpg: 544x800 1 drone, 15.8ms
image 10/20 /content/Drones_test/images/images-18.jpg.rf.aa254a358ec4c13b65ba4453153971b5.jpg: 544x800 15.2ms
image 11/20 /content/Drones_test/images/images-19.jpg.rf.a3c0ad6659ea1978d0ed339b456665cc.jpg: 544x800 2 drones, 15.2ms
image 12/20 /content/Drones_test/images/images-2.jpg.rf.821cfcf732f83008f1a6ae0b1c54e59.jpg: 544x800 1 drone, 15.2ms
image 13/20 /content/Drones_test/images/images-3.jpg.rf.186c5c8b9beb21cea8183aef16bf503.jpg: 480x800 1 drone, 14.2ms
image 14/20 /content/Drones_test/images/images-4.jpg.rf.4fae04243cbb1494e4a352aff2efb5bb.jpg: 512x800 1 drone, 15.0ms
image 15/20 /content/Drones_test/images/images-5.jpg.rf.535c0069cc53207ff3a7050dd99c70a3.jpg: 448x800 1 drone, 9.2ms
image 16/20 /content/Drones_test/images/images-6.jpg.rf.0083c55d18db3f6abfa1fa0b1b930a2d.jpg: 576x800 1 drone, 14.8ms
image 17/20 /content/Drones_test/images/images-7.jpg.rf.ffa3640779cdf2d84e767d20daa1375ed.jpg: 544x800 1 drone, 10.5ms
image 18/20 /content/Drones_test/images/images-8.jpg.rf.83b49a44b8af9283c75fc5ddb8e5e1e8.jpg: 384x800 1 drone, 15.7ms
image 19/20 /content/Drones_test/images/images-9.jpg.rf.8445a96472da6f083c758d4fe8bb0f09.jpg: 544x800 1 drone, 12.3ms
image 20/20 /content/Drones_test/images/images.jpg.rf.03b8fc245bfe2755f5599f92942e9c2.jpg: 448x800 9.9ms
Speed: 0.5ms pre-process, 14.8ms inference, 1.2ms postprocess per image at shape (1, 3, 800, 800)
Results saved to runs/detect/predict
```

3.19 pav. Antrojo bandymo bepiločio orlaivio aptikimo rezultatai, naudojant testavimo duomenų rinkinį (sudaryta autorės)

Kaip matoma iš gautų rezultatų, pateiktų 3.19 paveiksle, bepiločiai orlaiviai buvo atpažinti 18 vaizdų iš 20, viename paveiksle klaidingai buvo aptikti du dronai (3.20 pav.). Šis modelis žymiai geriau aptinka bepiločius orlaivius, lyginant su pirmuoju bandymu.



3.20 pav. Antrojo bandymo metu klaidingas bepiločio orlaivio aptikimo rezultatas, naudojant testavimo duomenų rinkinį (sudaryta autorės)

Trečiojo bandymo metu buvo pakeistas duomenų rinkinių eiliškumas: pradinis mokymas buvo atliktas naudojant mažesnę duomenų rinkinį ir originalų „yolov8s.pt“ tinklą.

Modelio mokymo parametrai:

- nuotraukų skaičius: 45 (mokymui) ir 5 (patikrinimui);
- epochų kiekis: 100;
- mokymosi greitis: 0,01;
- paketo dydis: 16;
- aktyvacijos funkcija: SGD.

Po 100 epochų buvo gautas 99,5 % mAP50 ir 54,9 % mAP50–95 (3.21 pav.).

```
100 epochs completed in 0.145 hours.
Optimizer stripped from runs/detect/train4/weights/last.pt, 22.5MB
Optimizer stripped from runs/detect/train4/weights/best.pt, 22.5MB

Validating runs/detect/train4/weights/best.pt...
Ultralytics YOLOv8.0.20 Python-3.9.16 torch-1.13.1+cu116 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11125971 parameters, 0 gradients, 28.4 GFLOPs
Class      Images  Instances  Box(P      R      mAP50  mAP50-95): 100% 1/1 [00:00<00:00, 1.52it/s]
   all         5           5    0.977      1    0.995    0.549
Speed: 0.4ms pre-process, 9.8ms inference, 0.0ms loss, 2.9ms post-process per image
Results saved to runs/detect/train4
```

3.21 pav. Trečiojo bandymo tinklo mokymo tarpinis rezultatas po 100 epochų (sudaryta autorės)

Tinklo mokymas buvo pratęstas (pakeistas duomenų rinkinys), po 15 epochų buvo pasiektas 92,3 % mAP50 ir 62,5 % mAP50–95 (3.22 pav.).

```

15 epochs completed in 0.305 hours.
Optimizer stripped from runs/detect/train5/weights/last.pt, 22.5MB
Optimizer stripped from runs/detect/train5/weights/best.pt, 22.5MB

Validating runs/detect/train5/weights/best.pt...
Ultralytics YOLOv8.0.20 Python-3.9.16 torch-1.13.1+cu116 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11125971 parameters, 0 gradients, 28.4 GFLOPs
      Class  Images  Instances  Box(P          R              mAP50  mAP50-95): 12% 1/8 [00:01<00:12,  1.80s/it]
      Class  Images  Instances  Box(P          R              mAP50  mAP50-95): 100% 8/8 [00:08<00:00,  1.01s/it]
      all      227      255      0.941      0.871      0.923      0.625
Speed: 1.4ms pre-process, 5.8ms inference, 0.0ms loss, 2.4ms post-process per image
Results saved to runs/detect/train5

```

3.22 pav. Trečiojo bandymo tinklo mokymo tarpinis rezultatas po 65 epochų (sudaryta autorės)

Po dar 15 epochų buvo gautas 93 % mAP50 ir 64,5 % mAP50–95 (3.23 pav.). Tinklo mokymas buvo pratęstas, naudojant geriausius gautus svorius ir nekeičiant kitų parametrų.

```

15 epochs completed in 0.304 hours.
Optimizer stripped from runs/detect/train6/weights/last.pt, 22.5MB
Optimizer stripped from runs/detect/train6/weights/best.pt, 22.5MB

Validating runs/detect/train6/weights/best.pt...
Ultralytics YOLOv8.0.20 Python-3.9.16 torch-1.13.1+cu116 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11125971 parameters, 0 gradients, 28.4 GFLOPs
      Class  Images  Instances  Box(P          R              mAP50  mAP50-95): 12% 1/8 [00:03<00:21,  3.09s/it]
      Class  Images  Instances  Box(P          R              mAP50  mAP50-95): 100% 8/8 [00:10<00:00,  1.26s/it]
      all      227      255      0.942      0.886      0.93      0.645
Speed: 1.9ms pre-process, 5.7ms inference, 0.0ms loss, 2.2ms post-process per image
Results saved to runs/detect/train6

```

3.23 pav. Trečiojo bandymo tinklo mokymo tarpinis rezultatas po 80 epochų (sudaryta autorės)

Po 25 epochų (iš viso po 105 epochų) buvo pasiektas 94,3 % mAP50 ir 66 % mAP50–95 (3.24 pav.). Mokymo rezultatai pateikti 3.25 ir 3.26 paveiksluose.

```

25 epochs completed in 0.610 hours.
Optimizer stripped from runs/detect/train7/weights/last.pt, 22.5MB
Optimizer stripped from runs/detect/train7/weights/best.pt, 22.5MB

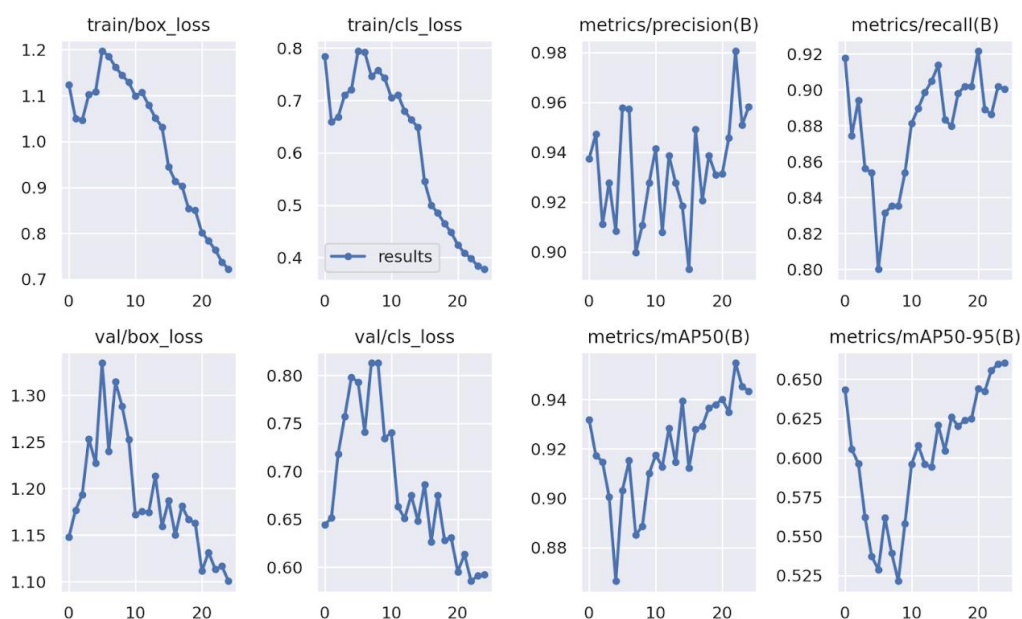
Validating runs/detect/train7/weights/best.pt...
Ultralytics YOLOv8.0.20 Python-3.9.16 torch-1.13.1+cu116 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11125971 parameters, 0 gradients, 28.4 GFLOPs
      Class  Images  Instances  Box(P          R              mAP50  mAP50-95): 12% 1/8 [00:01<00:13,  1.87s/it]
      Class  Images  Instances  Box(P          R              mAP50  mAP50-95): 100% 8/8 [00:09<00:00,  1.17s/it]
      all      227      255      0.958      0.901      0.943      0.66
Speed: 1.3ms pre-process, 6.0ms inference, 0.0ms loss, 2.5ms post-process per image
Results saved to runs/detect/train7

```

3.24 pav. Trečiojo bandymo tinklo mokymo pabaiga (sudaryta autorės)



3.25 pav. Trečiojo bandymo bepiločio orlaivio aptikimo rezultatai (sudaryta autorės)



3.26 pav. Trečiojo bandymo tinklo mokymo rezultatai (sudaryta autorės)

Iš nuostolių kreivių (3.26 pav.) matoma, kad modelio mokymas buvo efektyvus. Diagramos atspindi tik paskutines 25 epochas. „Box_loss“ ir „Cls_loss“ diagramose matyti, kad pirmose epochose nuostoliai yra didesni, bet toliau yra mažėjimo tendencija. „Metrics/precision(B)“ ir „metrics/recall(B)“ diagramose yra svyravimų, tačiau kreivė yra virš 90 %, o tai parodo, kad modelio mokymas buvo sėkmingas. „Metrics/mAP50(B)“ diagramoje matoma, kad reikšmės svyruoja nuo 90 % iki 95 %. „Metrics/mAP50–95(B)“ diagramoje matoma didėjimo tendencija – modelis vis tiksliau aptinka objektus.

Išmokyto modelio veikimui patikrinti taip pat buvo panaudotas testavimui skirtas duomenų rinkinys (3.27 pav.).

```
# Drono aptikimas vaizduose
!yolo task=detect mode=predict model=/content/runs/detect/train7/weights/best.pt conf=0.5 source=/content/Drones_test/images/ save=True
```

```
Ultralytics YOLOv8.0.20 Python-3.9.16 torch-1.13.1+cu116 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11125971 parameters, 0 gradients, 28.4 GfLOPs
image 1/20 /content/Drones_test/images/422021.jpg.rf.8cd1761e223f8b341c56a25f78cd870e.jpg: 608x800 1 drone, 20.7ms
image 2/20 /content/Drones_test/images/images-10.jpeg.rf.46261397b8f5f64e81764bf36dd4cb41.jpg: 416x800 1 drone, 20.7ms
image 3/20 /content/Drones_test/images/images-11.jpeg.rf.65062b93fc2c4c8b022bcc34e0927ba5.jpg: 480x800 1 drone, 20.1ms
image 4/20 /content/Drones_test/images/images-12.jpeg.rf.e8ec775c73a81eef88626455a1a3c43f.jpg: 544x800 3 drones, 21.0ms
image 5/20 /content/Drones_test/images/images-13.jpeg.rf.233c7610006aa704f5de9cb69e99f678.jpg: 544x800 1 drone, 18.3ms
image 6/20 /content/Drones_test/images/images-14.jpeg.rf.dc1b02a01606fe6588333dcfcd3258d4.jpg: 544x800 18.2ms
image 7/20 /content/Drones_test/images/images-15.jpeg.rf.4e270804df50f4d1631b3cbf97337973.jpg: 544x800 16.2ms
image 8/20 /content/Drones_test/images/images-16.jpeg.rf.7c52aae1e02a24671ece8f9e9c19eadcb.jpg: 448x800 1 drone, 20.9ms
image 9/20 /content/Drones_test/images/images-17.jpeg.rf.bde2378c6d0c3fb0126bfd95a2f18ba.jpg: 544x800 1 drone, 14.1ms
image 10/20 /content/Drones_test/images/images-18.jpeg.rf.aa254a358ec4c13b65ba4453153971b5.jpg: 544x800 13.3ms
image 11/20 /content/Drones_test/images/images-19.jpeg.rf.a3c0ad6659ea1978d0ed339b456665cc.jpg: 544x800 1 drone, 13.2ms
image 12/20 /content/Drones_test/images/images-2.jpeg.rf.821cfcf732f83008fe1a6ae0b1c54e59.jpg: 544x800 1 drone, 13.3ms
image 13/20 /content/Drones_test/images/images-3.jpeg.rf.e186c5c8b9beb21cea8183aef16bf503.jpg: 480x800 1 drone, 12.8ms
image 14/20 /content/Drones_test/images/images-4.jpeg.rf.4fae04243cbb1494e4a352aff2efb5bb.jpg: 512x800 23.0ms
image 15/20 /content/Drones_test/images/images-5.jpeg.rf.535c0069cc53207ff3a7050dd99c70a3.jpg: 448x800 1 drone, 12.5ms
image 16/20 /content/Drones_test/images/images-6.jpeg.rf.0083c55d18db3f6abfa1fa0b1b930a2d.jpg: 576x800 20.0ms
image 17/20 /content/Drones_test/images/images-7.jpeg.rf.fff3640779cdf2d84e767d20daa1375ed.jpg: 544x800 1 drone, 14.3ms
image 18/20 /content/Drones_test/images/images-8.jpeg.rf.83b49a44b8af9283c75fc5ddb8e5e1e8.jpg: 384x800 1 drone, 27.2ms
image 19/20 /content/Drones_test/images/images-9.jpeg.rf.8445a96472da6f083c758d4fe8bb0f09.jpg: 544x800 1 drone, 11.4ms
image 20/20 /content/Drones_test/images/images.jpeg.rf.03b8fc245bfe2755f5599f9f2942e9c2.jpg: 448x800 1 drone, 14.2ms
Speed: 0.5ms pre-process, 17.3ms inference, 1.5ms postprocess per image at shape (1, 3, 800, 800)
Results saved to runs/detect/predicts
```

3.27 pav. Trečiojo bandymo bepiločio orlaivio aptikimo rezultatai, naudojant testavimo duomenų rinkinį (sudaryta autorės)

Kaip matoma iš gautų rezultatų, pateiktų 3.27 paveiksle, bepiločiai orlaiviai buvo atpažinti 15 vaizdų iš 20. Šis modelis veikia geriau nei pirmojo eksperimento metu gautas tinklas, tačiau antruoju bandymu buvo pasiektas geresnis rezultatas.

Taip pat gautų modelių veikimas buvo patikrintas naudojant vaizdo įrašą. Gauti rezultatai yra panašūs į aptikimą vaizduose: pirmasis ir trečiasis tinklai dažniau neaptinka bepiločio orlaivio, keičiantis fonui; antrasis tinklas aptinka bepilotį orlaivį ir pasikeitus fonui, be to, šio tinklo aptikimo pasiklivimo įvertis yra aukštesnis. Trijų bandymų rezultatai pateikti 3.1 lentelėje.

3.1 lentelė. Tinklo mokymo rezultatai (sudaryta autorės)

Modelio versija	Mokymo laikas	Mokymui naudoti duomenų rinkiniai	Tinklo tikslumas po mokymo	Aptikta bepiločių orlaivių testavimo metu (iš 20)	Vieno kadro apdorojimo laikas, ms
I	80,1 min. (50 epochų)	„Drone_dataset“ (50 epochų)	94,5 % mAP50 66,3 % mAP50–95	10	11,9
II	92,7 min. (200 epochų)	„Drone_dataset“ (50 epochų) „Drones“ (150 epochų)	99,5 % mAP50 53,3 % mAP50–95	18	13
III	81,84 min. (155 epochų)	„Drones“ (100 epochų) „Drone_dataset“ (55 epochos)	94,3 % mAP50 66 % mAP50–95	15	12,7

Kaip matyti iš bandymų rezultatų, nors antrojo tinklo mAP50–95 yra žemiausias, šio modelio testavimo metu pavyko pasiekti geriausią rezultatą. Visų trijų modelių gauti geriausi svoriai buvo išsaugoti, ir jų veikimas toliau bus patikrintas kartu su sekikliu.

3.2 lentelėje pateiktas baigiamajame darbe siūlomo „YOLOv8“ modelio (antrojo išmokyto tinklo) palyginimas su Aydin ir Singha (2023) siūlomais „YOLOv4“ ir „YOLOv5“ modeliais.

3.2 lentelė. YOLO tinklų palyginimas, aptinkant bepiločius orlaivius (sudaryta autorės, remiantis Aydin ir Singha (2023))

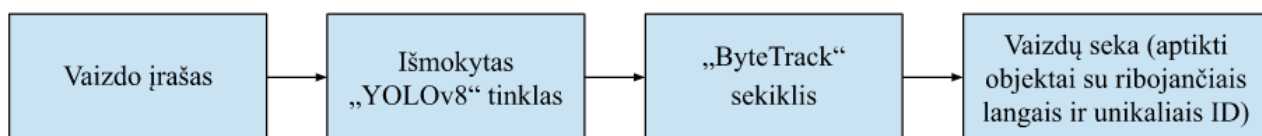
Modelis	Preciziškumas	Jautrumas	mAP50	Vieno kadro apdorojimo laikas, ms
„YOLOv4“	0,95	0,68	0,7436	19,75
„YOLOv5“	0,918	0,875	0,904	27,55
„YOLOv8“	1	0,944	0,995	13

Kaip galima matyti iš 3.2 lentelėje pateiktų duomenų, naudojant „YOLOv8“ modelį bepiločiams orlaiviams aptikti, buvo pasiektas geresnis rezultatas: tinklas tiksliau aptinka dronus ir vienas kadras yra apdorojamas greičiau.

3.4. Bepiločių orlaivių sekimas

Bepiločiui orlaiviui aptikti naudojami prieš tai išmokyti „YOLOv8“ tinklai. Kitas žingsnis yra sekimas, kuriam baigiamajame darbe naudojamas „ByteTrack“ algoritmas. Sekami objektai yra anotuojami naudojant „Roboflow Supervision“ paketą.

Bepiločių orlaivių sekimo, naudojant „YOLOv8“ tinklą ir „ByteTrack“ sekiklį, programos kodas pateiktas C priede. Bepiločių orlaivių sekimo žingsniai pateikti 3.28 paveiksle.



3.28 pav. Bepiločių orlaivių sekimo schema (sudaryta autorės)

Įkėlus išmokytą tinklą ir vaizdo įrašą bei paleidus sekiklio kodą, buvo patikrintas trijų paruoštų modelių veikimas. Visiems eksperimentams buvo naudojamas tas pats vaizdo įrašas, kad būtų paprasčiau palyginti rezultatus. Vaizdo įrašas atsiųstas iš duomenų rinkinio, pateikto „GitHub“ platformoje (Svanström et al., 2021).

Bandymui pasirinktas 10-ties sekundžių vaizdo įrašas, kurį modelis išskaido į 301 kadrą. Vaizdo įrašė užfiksuotas vienas skraidantis bepilotis orlaivis (3.29 pav.). Gauti rezultatai pateikti 3.3 lentelėje.

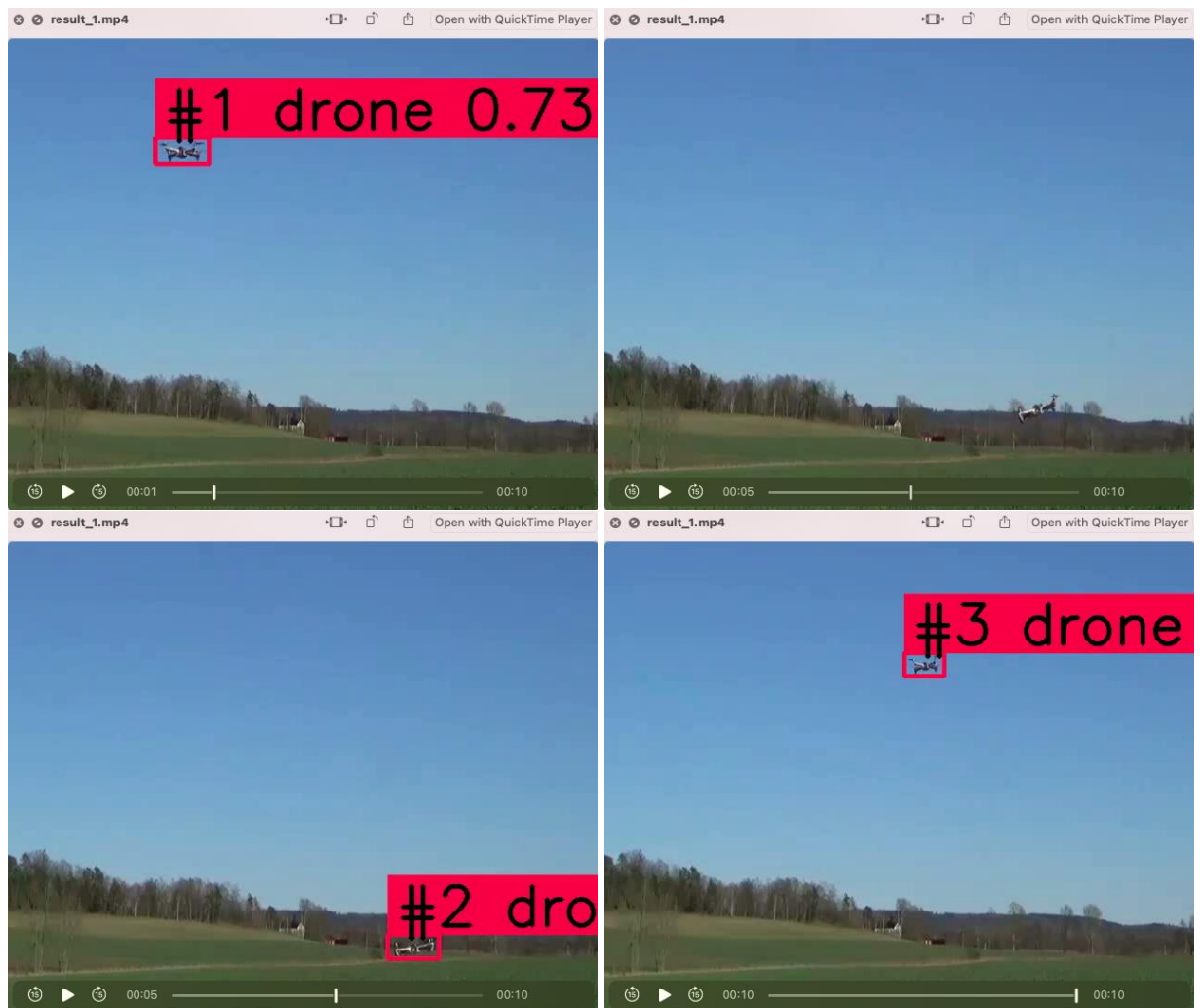


3.29 pav. Vaizdo įrašo, naudojamo eksperimentui, kadras (sudaryta autorės)

3.3 lentelė. Sekimo rezultatai (sudaryta autorės)

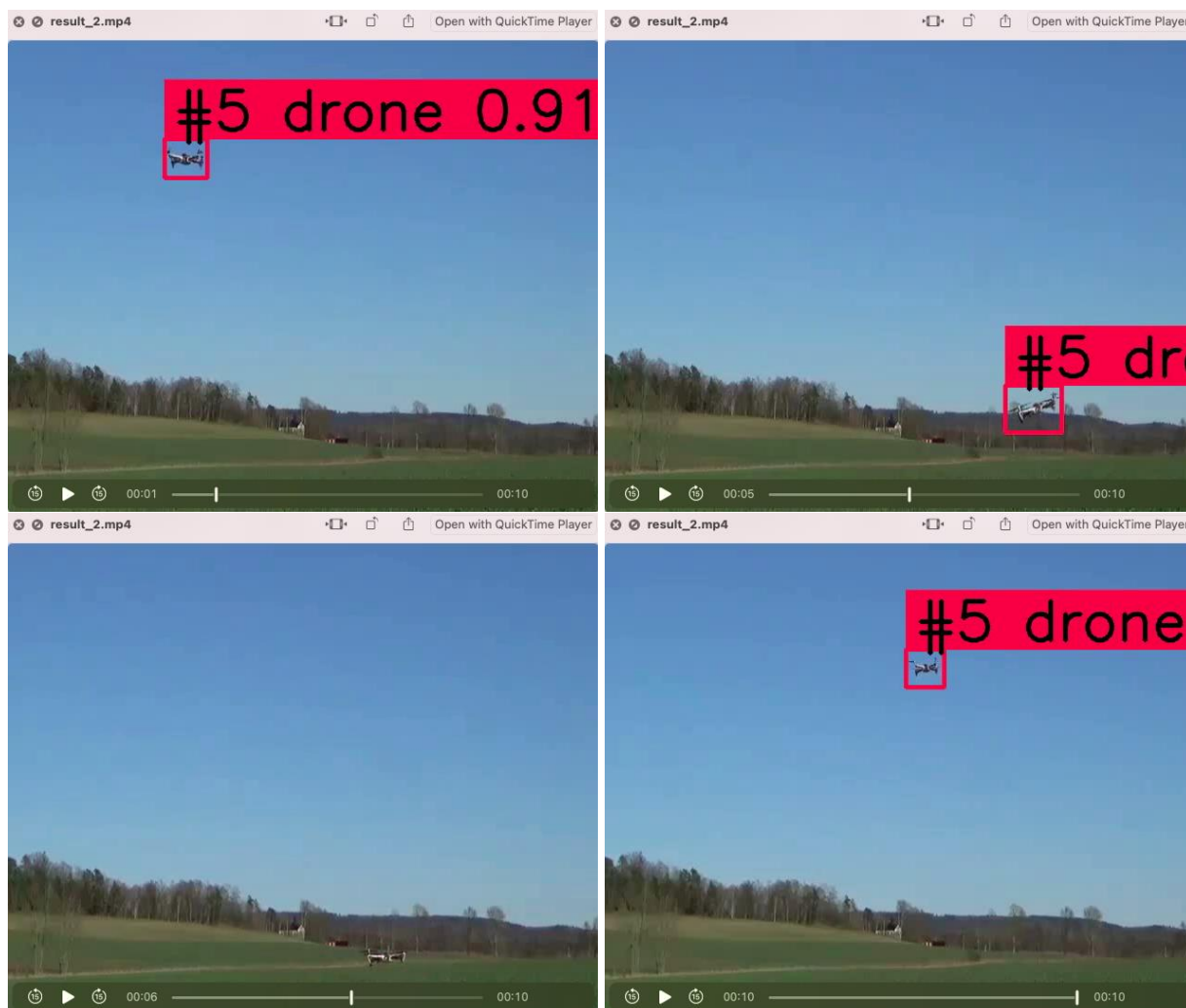
Modelio versija	Tinklo tikslumas po mokymo	Kadrai, kuriuose aptiktas bepilotis orlaivis (iš 301 kadro)	Atpažinta skirtingų bepilotių orlaivių
I	94,5 % mAP50 66,3 % mAP50–95	244	3
II	99,5 % mAP50 53,3 % mAP50–95	292	1
III	94,3 % mAP50 66 % mAP50–95	244	3 (indeksas pasikeitė per 5 žingsnius)

Kaip matyti iš 3.30 paveiksle pateiktų kadrų, sekant bepilotį orlaivį, kai taikiniui aptikti yra naudojamas pirmasis išmokytas tinklas, pasikeitus fonui jis buvo pamestas. Be to, aptikus droną iš naujo, jis buvo atpažintas kaip kitas objektas. Ilgesniam laikui šis tinklas pameta sekamą bepilotį orlaivį, kai jis yra miško fone.



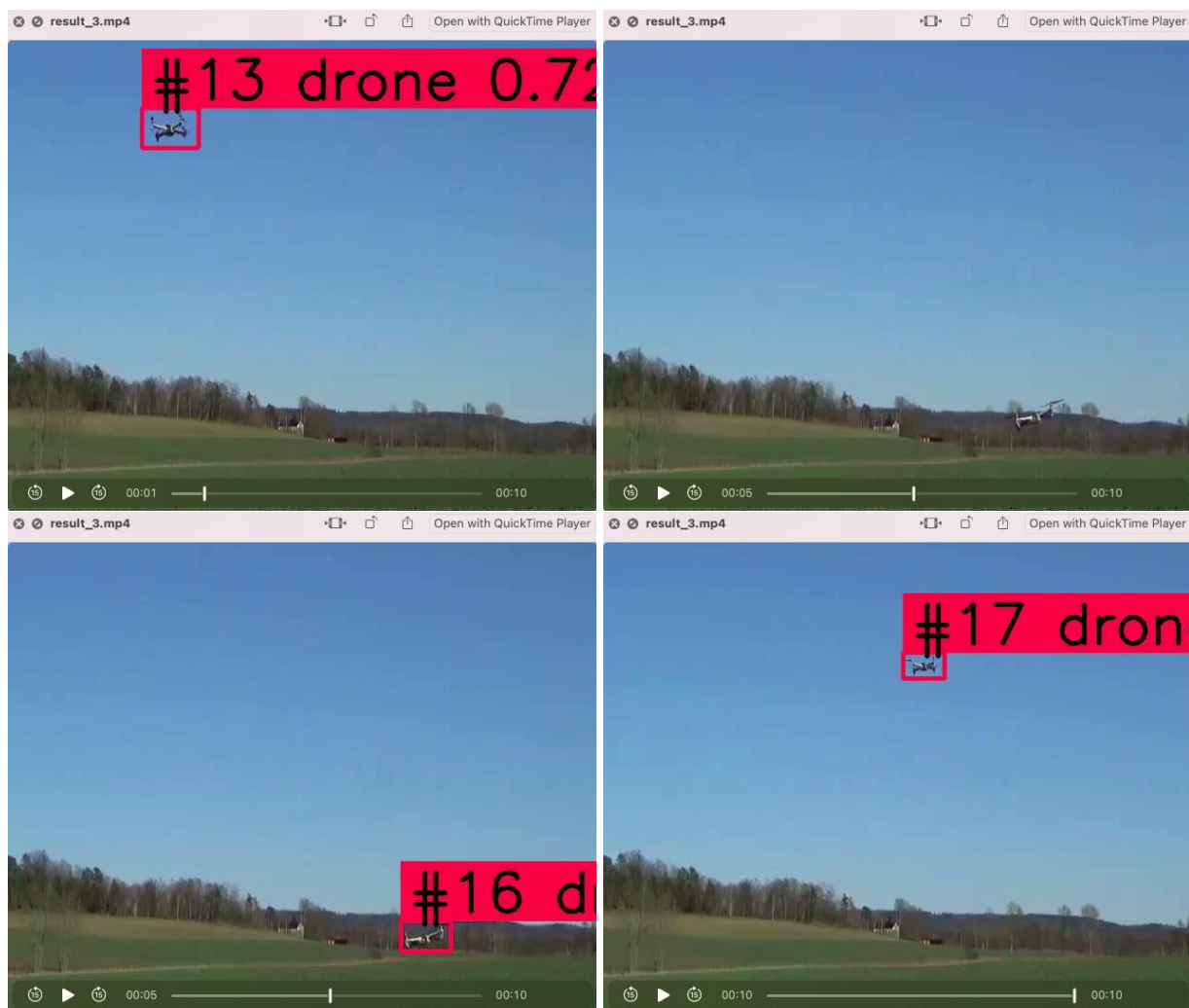
3.30 pav. Pirmojo išmokyto tinklo sekimo rezultatai (sudaryta autorės)

Kaip matyti iš 3.31 paveiksle pateiktų kadrų, sekant bepilotį orlaivį, kai taikiniui aptikti yra naudojamas antrasis išmokytas tinklas, pasikeitus fonui jis trumpam buvo pamestas, bet vėl aptiktas kaip tas pats sekamas taikiny. Iš 301 kadro modelis veikė tinkamai 292 kadruose (tai sudaro 97 %).



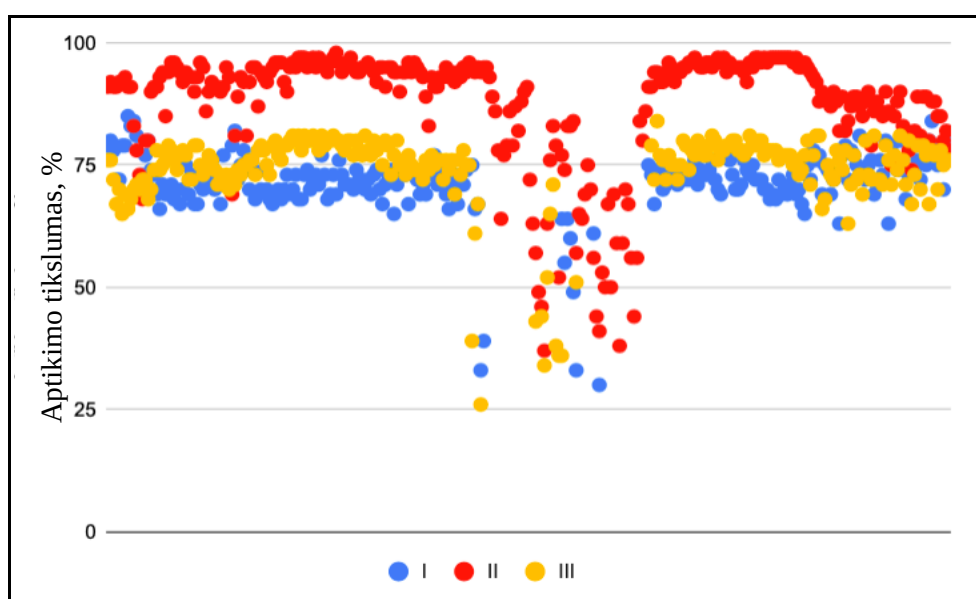
3.31 pav. Antrojo išmokyto tinklo sekimo rezultatai (sudaryta autorės)

Kaip matyti iš 3.32 paveiksle pateiktų kadrų, sekant bepilotį orlaivį, kai taikiniui aptikti yra naudojamas trečiasis išmokytas tinklas, pasikeitus fonui dronas buvo pamestas, ir, aptikus jį iš naujo, buvo atpažintas kaip kitas objektas. Be to, indeksavimo metu buvo praleisti du žingsniai. Kaip ir pirmojo bandymo metu, ilgesniam laikui trečiasis tinklas pameta sekamą droną, kai jis yra miško fone.



3.32 pav. Trečiojo išmokyto tinklo sekimo rezultatai (sudaryta autorės)

Bepiločio orlaivio aptikimo vaizdo įrašo kadruose tikslumo pasiskirstymas pateiktas 3.33 paveiksle.



3.33 pav. Išmokytų tinklų bepiločio orlaivio aptikimo tikslumo pasiskirstymas sekimo metu (sudaryta autorės)

Kaip matyti iš grafiko (3.33 pav.), antruoju bandymu buvo pasiektas tikslesnis rezultatas. Taip pat matyti, kad visų trijų tinklų veikimas yra prastesnis, kai vaizdo įrašė esančio bepiločio orlaivio padėtis keičiasi fono atžvilgiu (vaizdo įrašo 5–6 sekundės atitinka miško/žolės foną).

Taip pat buvo patikrintas siūlomo metodo veikimas, kai kadre yra daugiau nei vienas bepilotis orlaivis. Gautas rezultatas pateiktas 3.34 paveiksle (dronams aptikti buvo naudojamas antrasis išmokytas YOLO tinklas).



3.34 pav. Kelių objektų aptikimo ir anotavimo rezultatas (sudaryta autorės)

Visi trys bepiločiai orlaiviai buvo aptikti (3.34 pav.), ribojančių langų pasiklivimo įverčiai yra virš 0,95. Taigi, siūlomas metodas gali tinkamai veikti esant vienam ir daugiau bepiločių orlaivių.

3.5. Skyriaus išvados

1. Vienas iš svarbiausių žingsnių, kuriant sekimo sistemą, yra tinkamo duomenų rinkinio paruošimas. Duomenų rinkinio kokybė ir įvairovė yra labai svarbūs, norint tiksliai sekti objektą ir pagerinti sistemos veikimą.

2. „YOLOv8“ algoritmas yra efektyvus bepiločių orlaivių aptikimo įrankis, nes gali tiksliai aptikti bepiločius orlaivius realiuoju laiku. YOLO tinklas buvo pasirinktas baigiamajame darbe kaip bepiločių orlaivių aptikimo modelis dėl jo tikslumo, aptinkant objektus ir sukuriant ribojančius langus, kartu išlaikant našumą realiuoju laiku.

3. „ByteTrack“ algoritmas yra efektyvus ir patikimas bepiločių orlaivių sekimo metodas, pagal ankstesnį judėjimą numatantis būsimą drono padėtį. BYTE metodas geba atkurti objektą iš žemą

įvertį turinčių aptikimo langų, o tai leidžia sumažinti trūkstamų aptikimų skaičių ir išlaikyti trajektorijų vientisumą, todėl buvo nuspręsta naudoti „ByteTrack“ algoritmą baigiamajame darbe.

4. Atlikus bandymą buvo nustatyta, kad „YOLOv8“ ir „ByteTrack“ algoritmų derinys suteikia patvarų ir efektyvų sprendimą bepiločiams orlaiviams sekti. Siūlomo metodo veikimo įvertinimo rezultatai parodė, kad sistema gali aptikti ir sekti dronus, esant aukštiems preciziškumo ir jautrumo įverčiams. Sistemos lankstumas buvo patvirtintas įvairiais scenarijais, įskaitant fono sudėtingumą ir drono uždengimą, o tai parodo algoritmo praktinio pritaikymo galimybes įvairiose srityse.

IŠVADOS IR REKOMENDACIJOS

1. Bepiločiai orlaiviai pastaruoju metu įgijo didžiulį populiarumą dėl technologijų pažangos ir naudojimo sričių padidėjimo, įskaitant eismo valdymą, saugumą ir stebėjimą, būtinųjų atsargų tiekimą, ekstremaliųjų įvykių valdymą ir kt. Ankstyvosios drono versijos buvo daug didesnės, tačiau laikui bėgant jos mažėjo ir tapo išmanesnės. Dėl to, atsiradus mažiems ir judriems dronams, jų taikymai ne kartą kėlė susirūpinimą dėl saugumo. Tai paskatino mokslinių tyrimų kryptį, skirtą dinaminiam kelių dronų aptikimui ir lokalizavimui sistemose.

2. Literatūroje yra pateikti skirtingi metodai, naudojami objektams aptikti, pavyzdžiui, gali būti naudojami įvairūs jutikliai (radarai, LiDAR ir kt.). Tačiau proveržis įvyko, kai objektams aptikti ir sekti pradėta taikyti gilųjų mokymąsi. Taikant šiuos metodus, svarbu atsižvelgti į dvi pagrindines kliūtis – aparatinės įrangos poreikį skaičiavimams atlikti ir veikimo realiuoju laiku užtikrinimą.

3. Pagrindinė sekimo algoritmo užduotis yra užtikrinti tikslų taikinio sekimą, atsižvelgiant į algoritmo realizavimo laiką. Dėl nenuspėjamumo sekimo algoritmas, paremtas konvoliuciniu neuroniniu tinklu (CNN), gali būti paveikiamas. Pavyzdžiui, kai objektas yra uždengtas, taikiny s gali būti prarastas ir sekimas nepavyks. Vyksta tyrimai ir ieškoma būdų, kaip padaryti CNN modelį tinkamesnį sudėtingiems taikiniams aptikti, esant įvairiems scenarijams.

4. Kuriant sekimo sistemą, vienas svarbiausių žingsnių yra tinkamo duomenų rinkinio paruošimas. Duomenų rinkinio, kuris naudojamas tinklui mokytis, kokybė ir įvairovė yra labai svarbūs, norint tiksliai sekti objektą ir pagerinti sistemos veikimą. Todėl baigiamajame darbe buvo naudojami bepiločių orlaivių vaizdai, daryti iš skirtingų aukščių, kampų ir esant skirtingiems fonams, taip pat buvo taikomi duomenų papildymo metodai.

5. Bepiločių orlaivių sekimas kelia iššūkių (pavyzdžiui, fono pasikeitimas, mažas dronų dydis vaizdo kadruose). Judančių objektų aptikimas yra pradinis vaizdo sekos analizės etapas. Dronai gali skristi dideliu greičiu, todėl taikomų metodų aptikimo greitis taip pat turi būti didelis. Pastaruoju metu mokslininkai renka YOLO algoritmus, pasižyminčius dideliu našumu ir greitu objektų aptikimu bei veikiančius realiuoju laiku. Baigiamajame darbe „YOLOv8“ tinklas buvo pasirinktas kaip bepiločių orlaivių aptikimo modelis dėl jo tikslumo, aptinkant objektus ir sukuriant ribojančius langus, kartu išlaikant našumą realiuoju laiku.

6. Sekimas pagal aptikimą yra efektyviausia šiuo metu prieinama kelių objektų sekimo paradigma. Dėl sudėtingų vaizdo įrašų scenarijų detektoriai yra linkę daryti netobulas prognozes. Todėl, sekant objektus, yra svarbu sumažinti trūkstamų aptikimų skaičių ir išlaikyti trajektorijų vientisumą. Šiam tikslui baigiamajame darbe buvo pasirinktas BYTE metodas, gebantis atkurti objektą iš žemą įvertį turinčių aptikimo langų.

7. Atlikus bandymą buvo nustatyta, kad pažangių kompiuterinės regos technologijų – „YOLOv8“ ir „ByteTrack“ algoritmų – derinys suteikia patvarų ir efektyvų sprendimą sekant

bepiločius orlaivius. Siūlomas metodas sujungia abiejų algoritmų pranašumus, kad būtų pasiektas geresnis veikimas, siekiant didesnio tikslumo, greičio ir efektyvumo. Siūlomo metodo veikimo įvertinimo rezultatai parodė, kad sistema gali aptikti ir sekti dronus, esant aukštiems preciziškumo ir jautrumo įverčiams. Sistemos lankstumas buvo patvirtintas įvairiais scenarijais, įskaitant fono sudėtingumą ir drono uždengimą, o tai parodo algoritmo praktinio pritaikymo galimybes įvairiose srityse.

8. Atliktas tyrimas pabrėžia giliojo mokymosi metodų derinimo potencialą sprendžiant realaus pasaulio problemas. Būsimi tyrimai gali būti pritaikyti naudojant pažangesnes architektūras ir duomenų papildymo metodus, siekiant pagerinti siūlomos sistemos našumą.

Baigiamojo darbo taikinių sekimo sistemos tyrimas viešai pristatytas pranešime, skaitytame 26-ojoje Lietuvos jaunųjų mokslininkų konferencijoje „Mokslas – Lietuvos ateitis. Aviacijos technologijos“. Pranešimo medžiaga pateikta D priede.

LITERATŪROS SĄRAŠAS

- Abhinu, C., Aswin, P., Kiran, K., Bonymol, B. & Viji, K. (2021). Multiple Object Tracking using Deep Learning with YOLO V5. *International Journal of Engineering Research & Technology*, 9(13), 47–51.
- Agarwal, V., Naidu, V. & Sivakumaran, N. (2016). Six Object Tracking Algorithms: A Comparative Study. In *International Conference on Innovations in information Embedded and Communication Systems*. <https://doi.org/10.13140/RG.2.1.5115.4803>.
- Aydin, B. & Singha, S. (2023). Drone Detection Using YOLOv5. *Eng*, 4(1), 416–433. <https://doi.org/10.3390/eng4010025>.
- Alsanad, H. R., Sadik, A. Z., Ucan, O. N., Ilyas, M. & Bayat, O. (2022). YOLO-V3 based real-time drone detection algorithm. *Multimedia Tools and Applications* 81, 26185–26198. <https://doi.org/10.1007/s11042-022-12939-4>.
- Appiah, O., Asante, M. & Hayfron-Acquah, J. (2020). Improved approximated median filter algorithm for real-time computer vision applications. *Journal of King Saud University – Computer and Information Sciences*, 34(3), 782–792. <https://doi.org/10.1016/j.jksuci.2020.04.005>.
- Arasaratnam, I., Haykin, S. & Hurd, T. (2010). Cubature Kalman filtering for continuous discrete systems: theory and simulations. *IEEE Transactions on Signal Processing*, 10, 4977–4993. <https://doi.org/10.1109/TSP.2010.2056923>.
- Athanesious, J. & Suresh, P. (2012), Systematic Survey on Object Tracking Methods in Video. *International Journal of Advanced Research in Computer Engineering & Technology*, 1(8), 242–247. <http://ijarcet.org/wp-content/uploads/IJARCET-VOL-1-ISSUE-8-242-247.pdf>.
- Avidan, S. (2004). Support Vector Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26(8), 1064–1072. <https://doi.org/10.1109/TPAMI.2004.53>.
- Azari, M. M., Sallouha, H., Chiumento, A., Rajendran, S., Vinogradov, E. & Pollin, S. (2018). Key Technologies and System Trade-offs for Detection and Localization of Amateur Drones. *IEEE Communications Magazine*, 56(1), 51–57. <https://doi.org/10.1109/MCOM.2017.1700442>.
- Babenko, B., Yang, M. & Belongie, S. (2009). Visual tracking with online multiple instance learning. In *Conference on Computer Vision and Pattern Recognition* (pp. 983–990). IEEE.
- Balaji, S. & Karthikeyan, S. (2017). A survey on moving object tracking using image processing. In *11th International Conference on Intelligent Systems and Control* (pp. 469–474). IEEE.
- Bathija, A. & Sharma, G. (2019). Visual Object Detection and Tracking using YOLO and SORT. *International Journal of Engineering Research & Technology*, 8(11), 705–708. <https://doi.org/10.17577/IJERTV8IS110343>.
- Blackledge, J. (2005). *Digital Image processing*. Horwood Publishing.

- Bochkovskiy, A., Wang, C-Y. & Liao, H-Y. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv*. <https://doi.org/10.48550/arXiv.2004.10934>.
- Bolme, D., Beveridge, J., Draper, B. & Lui, Y. (2010). Visual object tracking using adaptive correlation filters. In *Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 2544–2550). IEEE.
- Candes, E. & Wakin, M. (2008). An Introduction To Compressive Sampling. *IEEE Signal Processing Magazine*, 25(2), 21–30. <https://doi.org/10.1109/MSP.2007.914731>.
- Canny, J. (1986). A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6), 679–698. <https://doi.org/10.1109/TPAMI.1986.4767851>.
- Chen, Y., Han, C., Wang, C., Jeng, B. & Fan, K. (2006). The Application of a Convolution Neural Network on Face and License Plate Detection. In *18th International Conference on Pattern Recognition* (pp. 552–555). IEEE.
- Chen, J., Zeng, Q. & Qin, X. (2020). Target Tracking Algorithm Based on Density Clustering. In *International Conference on Swarm Intelligence* (pp. 556–563). Springer.
- Chernyak, V. (2002). Multisite Radar Systems with Information Fusion: A Technology of XXI Century. *Computer Science*, 85442334.
- Cheung, S. & Kamath, C. (2004). Robust techniques for background subtraction in urban traffic video. *Visual Communications and Image Processing*, 5308. <https://doi.org/10.1117/12.526886>.
- Dagienė, V., Grigas, G. & Jevsikova, T. (2014). *Enciklopedinis kompiuterijos žodynas* (IV leidimas). <http://www.ims.mii.lt/EK%C5%BD/index.html>.
- Elgammal, A., Harwood, D. & Davis, L. (2000) Non-parametric Model for Background Subtraction. In *European Conference on Computer Vision* (pp. 751–767). Springer.
- Farlik, J., Kratky, M., Casar, J. & Sary, V. (2017). Radar cross section and detection of small unmanned aerial vehicles. In *17th International Conference on Mechatronics* (pp. 1–7). IEEE.
- Gillani, I., Munawar, M., Talha, M., Azhar, S., Mashkoor, Y., Sami-uddin, M. & Zafar, U. (2022). Yolo5, Yolo-x, Yolo-r, Yolo7 Performance Comparison: A Survey. In *8th International Conference on Artificial Intelligence and Fuzzy Logic System* (pp. 17–28). AIRCC.
- Hao, W., Zong-ping, C., Bin, T. & Ze-xiang, Y. (2017). Review of the algorithms for radar single target tracking. In *IOP Conference Series: Earth and Environmental Science* (pp. 1–8). IOP Publishing Ltd.
- Henriques, J., Caseiro, R., Martins, P. & Batista, J. (2015). High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3), 583–596. <https://doi.org/10.1109/TPAMI.2014.2345390>.

- Hong, T., Yang, Q., Wang, P., Zhang, J., Sun, W., Tao, L., Fang, C. & Cao, J. (2021). Multitarget Real-Time Tracking Algorithm for UAV IoT. *Wireless Communications and Mobile Computing*, 2021. <https://doi.org/10.1155/2021/9999596>.
- Hu, Y-W., Zhang, H-C., Liu, S-J. & Lu, H-T. (2018). Sequential Monte Carlo Method Toward Online RUL Assessment with Applications. *Chinese Journal of Mechanical Engineering* 31(5).
- Hu, W., Li, X., Luo, W., Zhang, X., Maybank, S. & Zhang, Z. (2012). Single and Multiple Object Tracking Using Log-Euclidean Riemannian Subspace and Block-Division Appearance Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(12), 2420–2440. <https://doi.org/10.1109/TPAMI.2012.42>.
- Yang, C., Chen, B. & Liao, F. (2010). Mobile Location Estimation Using Fuzzy-Based IMM and Data Fusion. *IEEE Transactions on Mobile Computing*, 9(10), 1424–1436. <https://doi.org/10.1109/TMC.2010.105>.
- Yang, C., Wu, Z., Chang, X., Shi, X., Wo, J. & Shi, Z. (2018). DOA Estimation using amateur drones harmonic acoustic signals. In *Proceedings of the 10th Sensor Array and Multichannel Signal Processing Workshop*, (pp. 587–591). IEEE. <https://doi.org/10.1109/SAM.2018.8448797>.
- IEEE. (2017). IEEE Standard for Radar Definitions. In *Std 686-2017 (Revision of IEEE Std 686-2008)* (pp. 1–54). <https://doi.org/10.1109/IEEESTD.2017.8048479>.
- Yilmaz, A., Xin, L. & Shah, M. (2004). Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11), 1531–1536. <https://doi.org/10.1109/TPAMI.2004.96>.
- Youngjoo, K. & Hyochoong, B. (2018). *Introduction to Kalman Filter and Its Applications*. IntechOpen.
- Jalal, A. & Singh, V. (2012). The State-of-the-Art in Visual Object Tracking. *Informatica*, 36, 227–248. <http://www.informatica.si/index.php/informatica/article/view/405/408>.
- Jia, X., Lu, H. & Yang, M. (2012). Visual tracking via adaptive structural local sparse appearance model. In *Conference on Computer Vision and Pattern Recognition* (pp. 1822–1829). IEEE.
- Jing, L., Tingting, D., Chunyuan, Z., Chunbo, X., Huixin, T. & Jiangtao, X. (2015). Target Tracking Algorithm Using Angular Point Matching Combined with Compressive Tracking. *Advances in OptoElectronics*, 2015. <https://doi.org/10.1155/2015/420183>.
- Joshi, K. & Thakore, D. (2012). A Survey on Moving Object Detection and Tracking in Video Surveillance System. *International Journal of Soft Computing and Engineering*, 2(3), 44–48. <https://www.ijscce.org/wp-content/uploads/papers/v2i3/C0675052312.pdf>.
- Julier, J. & Uhlmann, J. (2002). Reduced Sigma Point Filters for the Propagation of Means and Covariances through Nonlinear Transformations. In *Proceedings of the 2002 American Control Conference*, 2 (pp. 887–892). IEEE.

- Julier, J. & Uhlmann, J. (1997). A New extension of the Kalman filter to nonlinear systems. *Signal Processing, Sensor Fusion, and Target Recognition VI*, 3068, 182–183. <https://doi.org/10.1117/12.280797>.
- Kaur, N. & Kaur, Y. (2014). Object classification Techniques using Machine Learning Model. *International Journal of Computer Trends and Technology*, 18(4), 170–174. <https://doi.org/10.14445/22312803/IJCTT-V18P140>.
- Kong, L. Wang, J. & Zhao, P. (2022). YOLO-G: A Lightweight Network Model for Improving the Performance of Military Targets Detection. *IEEE Access*, 10, 55546–55564. <https://doi.org/10.1109/ACCESS.2022.3177628>.
- Kwon, J. & Lee, K. (2010). Visual tracking decomposition. In *Conference on Computer Vision and Pattern Recognition* (pp. 1269–1276). IEEE.
- Liu, B., Huang, J., Yang, L. & Kulikowski, C. (2011). Robust tracking using local sparse appearance model and k-selection. In *Conference on Computer Vision and Pattern Recognition* (pp. 1313–1320). IEEE.
- Liu, D., Ma, X., Zhu, Y., Chen, P. & Yang, X. (2010). Algorithm Based on Combined Iterated UKF for SINS Alignment. *Journal of System Simulation*, 10, 2404–2406.
- Liu, H., Fan, K., Ouyang, Q. & Li, N. (2021). Real-Time Small Drones Detection Based on Pruned YOLOv4. *Sensors*, 21(10). <https://doi.org/10.3390/s21103374>.
- Liu, Q., Lu, X., He, Z., Zhang, C. & Chen, W. (2017). Deep convolutional neural networks for thermal infrared object tracking. *Knowledge-Based Systems*, 134, 189–198. <https://doi.org/10.1016/j.knosys.2017.07.032>.
- Matsuzaki, T., Kameda, H., Tsujimichi, S. & Kosuge, K. (2000). Maneuvering target tracking using constant velocity and constant angular velocity model. In *IEEE International Conference on Systems, Man and Cybernetics* (pp. 3230–3234). IEEE.
- Mu, J. & Cai, Y. (2011). Iterated cubature Kalman filter and its application. In *International Conference on Cyber Technology in Automation, Control, and Intelligent Systems* (pp. 33–37). IEEE.
- Mu, J., Cai, Y. & Zhang, J. (2011). Cubature Particle Filter and Its Application. *Journal of Xi'an Jiaotong University*, 45(8), 13–17. <https://doi.org/10.7652/xjtuxb201108003>.
- Nyström, A. (2019). *Evaluation of Multiple Object Tracking in Surveillance Video*. Linköping: Linköping University.
- Özel, M. (n. d.). *Drone Dataset (UAV): Unmanned Aerial Vehicle (Rotary Wing Unmanned Aerial Vehicles)*. <https://www.kaggle.com/datasets/dasmehdixtr/drone-dataset-uav>.
- Panchal, P., Prajapati, G., Patel, S., Shah, H. & Nasriwala, J. (2015). A Review on Object Detection and Tracking Methods. *International Journal for*

Research in Emerging Science and Technology, 2(1), 7–12.
<https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.680.6920&rep=rep1&type=pdf>.

Pang, S., del Coz, J., Yu, Z., Luaces, O. & Diez, J. (2017). Deep learning to frame objects for visual target tracking. *Engineering Applications of Artificial Intelligence*, 65, 406–420.
<https://doi.org/10.1016/j.engappai.2017.08.010>.

Parekh, H., Thakore, D. & Jaliya, U. (2014). A Survey on Object Detection and Tracking Methods. *International Journal of Innovative Research in Computer and Communication Engineering*, 2(2), 2970–2978. <https://www.rroij.com/open-access/a-survey-on-object-detection-and-tracking-methods.pdf>.

Qu, Y., Zhang, G., Zou, Z., Liu, Z. & Mao, J. (2017). Active Multimodal Sensor System for Target Recognition and Tracking. *Sensors* 17(7), 1518. <https://doi.org/10.3390/s17071518>.

Rakibe, R. & Patil, B. (2013). Background Subtraction Algorithm Based Human Motion Detection. *International Journal of Scientific and Research Publications*, 3(5), 1–4.
<http://www.ijsrp.org/research-paper-0513/ijsrp-p17106.pdf>.

Rashmi, S., Kumar, M. & Saxena, R. (2013). Algorithm and Technique on Various Edge Detection: A Survey. *Signal & Image Processing: An International Journal*, 4(3), 65–75.
<https://doi.org/10.5121/sipij.2013.4306>.

Reddy, P., Nagaraju, C., & Reddy, R. (2016). Canny Scale Edge Detection. *International Journal of Engineering Trends and Technology*. <https://doi.org/10.14445/22315381/IJETT-ICGTETM-N3/ICGTETM-P121>.

Ross, D., Lim, J., Lin, R. & Yang, M. (2008). Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77, 125–141. <https://doi.org/10.1007/s11263-007-0075-7>.

Rudinskas, D. (2011). *Bepiločių orlaivių skrydžio parametrų matavimų duomenų perdavimo saugos metodikos sukūrimas: daktaro disertacija*. Technika. <https://doi.org/10.20334/1975-M>.

Sharma, A., Jain, N. & Kothari, M. (2022). Lightweight Multi-Drone Detection and 3D-Localization via YOLO. *arXiv*. <https://doi.org/10.48550/arXiv.2202.09097>.

Singh, A. & Sharma, A. (2007). A Survey on Object Detection, Classification and Tracking in Video. *International Journal of Progresses in Engineering, Management, Science and Humanities*, 3(2), 14–19. <http://troindia.in/journal/IJPESH/vol3iss2/14-19.pdf>.

Singha, S. & Aydin, B. (2021). Automated Drone Detection Using YOLOv4. *Drones*, 5(3), 95. <https://doi.org/10.3390/drones5030095>.

Svanström, F., Alonso-Fernandez, F. & Englund, C. (2021). A dataset for multi-sensor drone detection. *Data in Brief*, 39. <https://doi.org/10.1016/j.dib.2021.107521>.

Svilainis, L. (2019). *Radiolokacija*. <https://www.vle.lt/straipsnis/radiolokacija>.

- Tan, L., Huangfu, T., Wu, L. & Chen, W. (2021). Comparison of RetinaNet, SSD, and YOLO v3 for real-time pill identification. *BMC Medical Informatics and Decision Making* 21(324). <https://doi.org/10.1186/s12911-021-01691-8>.
- Tiwari, S. (2022). YOLOv5 Based Drone Detection and Identification. *Neuro Quantology* 20(13), 2559–2563. <https://doi.org/10.14704/nq.2022.20.13.NQ88318>.
- Toyama, K., Krumm, J., Brumitt, B. & Meyers, B. (1999). Wallflower: principles and practice of background maintenance. In *The Seventh IEEE International Conference on Computer Vision* (pp. 255–261). IEEE.
- Vo, B., Mallick, M., Bar-Shalom, Y., Coraluppi, S., Osborne, R., Mahler, R. & Vo, B. (2015) Multitarget Tracking. *Wiley Encyclopedia of Electrical and Electronics Engineering*. <https://doi.org/10.1002/047134608X.W8275>.
- Wang, N. & Yeung, D. (2013). Learning a deep compact image representation for visual tracking. In *26th International Conference on Neural Information Processing Systems* (pp. 809–817). Curran Associates Inc.
- White, R. (1996). *Object Classification in Astronomical Images: Steps in Developing a Classifier*. <http://sundog.stsci.edu/rick/SCMA/node3.html>.
- Wu, Y., Lim, J. & Yang, M. (2015). Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9), 1834–1848. <https://doi.org/10.1109/TPAMI.2014.2388226>.
- Xinlong, L. & Hamdulla, A. (2020). Research on infrared small target tracking method. In *12th International Conference on Measuring Technology and Mechatronics Automation*. IEEE.
- Zaman, F., Abdullah, S., Razak, N., Johari, J., Pasya, I. & Kassim, K. (2021). Visual-Based Motorcycle Detection using You Only Look Once (YOLO) Deep Network. In *IOP Conference Series: Materials Science and Engineering* (pp. 1–11). IOP Publishing Ltd.
- Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W. & Wang, X. (2022). Multi-Object Tracking by Associating Every Detection Box. *arXiv*. <https://doi.org/10.48550/arXiv.2110.06864>.
- Zhang, K., Zhang, L. & Yang, M. (2012). Real-Time Compressive Tracking. In *12th European Conference on Computer Vision, Lecture Notes in Computer Science* (pp. 864–877). Springer.
- Zhang, L. & Fang, Q. (2020). Multi-target tracking based on target detection and mutual information. In *Chinese Control and Decision Conference* (pp. 1242–1245). IEEE.
- Zhang, L., Wang, C. & Jin, X. (2018). Research and Implementation of Target Tracking Algorithm Based on Convolution Neural Network. In *37th Chinese Control Conference* (pp. 9573–9577). IEEE.

Zhang, S., Li, J. & Wu, L. (2013a). A novel multiple maneuvering targets tracking algorithm with data association and track management. *International Journal of Control, Automation and Systems*, 11, 947–956. <https://doi.org/10.1007/s12555-012-0177-z>.

Zhang, T., Liu, S., Xu, C. & Lu, H. (2013b). Mining Semantic Context Information for Intelligent Video Surveillance of Traffic Scenes. *IEEE Transactions on Industrial Informatics*, 9(1), 149–160. <https://doi.org/10.1109/TII.2012.2218251>.

Бакулев, П. (2004). *Радиолокационные системы*. Радиотехника.

PRIEDAI

A priedas. Bepiločių orlaivių sekimo sistemos kūrimo algoritmas

B priedas. „YOLOv8“ tinklo mokymo programos kodas

C priedas. Bepiločių orlaivių sekimo, naudojant „YOLOv8“ tinklą ir „ByteTrack“ sekiklį, programos kodas

D priedas. Pranešimo 26-ojoje Lietuvos jaunųjų mokslininkų konferencijoje medžiaga