

TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Robin Lõhmus 222413IAIB

FISH DETECTION FROM UNDERWATER CAMERAS USING YOLOv8

Bachelor's Thesis

Supervisor: Jeffrey Andrew Tuhtan
Dr.-Eng.

Tallinn 2025

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Robin Lõhmus 222413IAIB

**KALADE TUVASTAMINE YOLOv8 MUDELIGA
VEEALUSTELT KAAMERATELT**

Bakalaureusetöö

Juhendaja: Jeffrey Andrew Tuhtan
Dr.-Eng.

Tallinn 2025

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Robin Lõhmus

10.06.2025

Abstract

This thesis focuses on the development and evaluation of an automated fish detection system using the YOLOv8 model on underwater cameras. Traditional manual methods are time-consuming and require the use of expert biologists to evaluate a large number of fish species. The first and most important step therefore, is to sort videos which have fish from those videos which have no fish in them.

The aim of the work is to create a solution for real-time detection of fish presence, while minimizing false negative results to ensure that fish are not missed, which is critical for ecological monitoring. The study utilized YOLOv8 nano and large models, which were trained on datasets from underwater cameras in Switzerland and Portugal, both separately and combined.

The methodology included video processing, semi-automated annotation, hyperparameter optimization, and model training on a high-performance computing cluster. Performance was evaluated using precision, recall, mean Average Precision (mAP@50 and mAP@50-95), and confusion matrices.

The results showed that the final YOLOv8n model, trained specifically on the Swiss dataset, achieved high performance (precision: 95,9%, recall: 97,2%, mAP@50: 99,0%, mAP@50-95: 88,1%), which has already found application in resource-constrained devices. The study confirmed the hypothesis that models trained solely on data from one location do not generalize well to other different locations, emphasizing the need for location-specific data or fine-tuning for optimal performance.

The thesis is written in Estonian and is 32 pages long, including 6 chapters, 20 figures and 4 tables.

Annotatsioon

Kalade tuvastamine YOLOv8 mudeliga veealustelt kaameratelt

Käesolev lõputöö keskendub automaatse kalade tuvastamise süsteemi arendamisele ja hindamisele, kasutades YOLOv8 mudelit veealustel kaameratel. Traditsioonilised manuaalsed meetodid on aeganõudvad ja nõuavad ekspertiisbioloogide kaasamist suure hulga kalaliikide tuvastamiseks. Seetõttu on esimene ja kõige olulisem samm sorteerida videod, milles esineb kalu, nendest videodest, milles kalu ei ole.

Töö eesmärgiks on luua lahendus reaalajas kalade olemasolu tuvastamiseks, minimeerides valenegatiivseid tulemusi, et tagada kalade mitte märkamata jäämine, mis on ökoloogilise seire jaoks kriitiline. Uuringus kasutati YOLOv8 nano ja large mudeleid, mida treeniti Šveitsi ja Portugali veealuste kaamerate andmestikel, nii eraldi kui kombineeritult.

Metoodika hõlmas videotöötlust, poolautomaatset annoteerimist, hüperparameetrite optimeerimist ning mudelite treenimist kõrgjõudlusega arvutusklastril. Jõudlust hinnati täpsuse, tundlikkuse, keskmise täpsuse (mAP@50 ja mAP@50-95) ning segadusmaatriksite abil.

Tulemused näitasid, et spetsiifiliselt Šveitsi andmestikul treenitud YOLOv8n lõppmudel saavutas kõrge jõudluse (täpsus: 95,9%, tundlikkus: 97,2%, mAP@50: 99,0%, mAP@50-95: 88,1%), mis leidis juba rakendust piiratud ressursidega seadmetes. Uuring kinnitas hüpoteesi, et vaid ühe asukoha andmetel treenitud mudelid ei üldistu hästi teistele erinevatele asukohtadele, rõhutades asukohapõhiste andmete või peenhäälestuse vajadust optimaalseks toimimiseks.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 32 leheküljel, 6 peatükki, 20 joonist, 4 tabelit.

Lühendite ja mõistete sõnastik

AP	<i>Average Precision</i> , keskmine täpsus
AP50	<i>Average Precision at IoU 50%</i> , keskmine täpsus, IoU 50%-lise läve järgi
AP95	<i>Average Precision at IoU 95%</i> , keskmine täpsus, IoU 50% kuni 95% vahemikus
BiFPN	<i>Bidirectional Feature Pyramid Network</i> , kahesuunaline omaduste püramiid
CNN	<i>Convolutional Neural Network</i> , konvolutsiooniline närvivõrk
COCO	<i>Common Objects in Context</i> , tehisnägemise andmestik
DETR	<i>DEtection TRansformer</i> , tuvastuse transformer - objekti tuvastuse mudelite perekond
DFL	<i>Distribution Focal Loss</i> , jaotuse fookuskaotus
GFLOP	<i>Giga Floating Point Operations</i> , gigaflop - miljard ujukomaarvutust
GPU	<i>Graphics Processing Unit</i> , graafikaprotsessor
HPC	<i>High Performance Computing</i> , suure jõudlusega arvutus
IoU	<i>Intesection over Union</i> , ühisosa ja ühendi jagatis
lr0	<i>learning rate 0</i> , algõppimiskiirus
lrf	<i>learning rate factor</i> , õppemäära lõppväärtuse kordaja
mAP	<i>Mean Average Precision</i> , keskmiste täpsuste keskmine
mAP@50	<i>Mean Average Precision at IoU 50%</i> , keskmiste täpsuste keskmine, IoU 50%-lise läve järgi
mAP@50-95	<i>Mean Average Precision at IoU 50%-95%</i> , keskmiste täpsuste keskmine, IoU 50% kuni 95% vahemikus
NMS	<i>Non-Maximum Srppression</i> , mittemaksimaalne maha surumine
PASCAL VOC	<i>PASCAL Visual Object Classes</i> , PASCAL'i visuaalsete objektide klasside andmestik
R-CNN	<i>Region-based Convolutional Neural Network</i> , regioonipõhine konvolutsiooniline närvivõrk - objekti tuvastuse mudelite perekond
SSD	<i>Single Shot MultiBox Detector</i> , üheetapiline mitmikaksti tuvasti - objekti tuvastuse mudelite perekond

SVM
YOLO

Support vector machine, tugivektormasin
You Only Look Once, sa vaatad vaid korra - objekti tuvastuse
mudelite perekond

Sisukord

1	Sissejuhatus	11
2	Objektide tuvastamine arvutinägemisega	14
2.1	Objektide tuvastamise mudelid	14
2.1.1	YOLO (You Only Look Once)	14
2.1.2	R-CNN (Regions with CNN features)	15
2.1.3	Single Shot MultiBox Detector (SSD)	15
2.1.4	EfficientDet	16
2.1.5	DETR ja Transformer-põhised mudelid	16
2.2	Mudelite võrdlus	16
2.3	Käesoleva töö mudeli valik	17
3	Metoodika	20
3.1	Hindamismeetodid	21
3.2	Andmestik ja eeltöötlus	22
3.2.1	Videofailide töötlemine ja kaadrite annoteerimine	22
3.2.2	Andmestike statistika	23
3.2.3	Andmestike võrdlus ja eripärad	25
3.3	Mudeli ülesehitus ja treenimine	26
3.3.1	Mudeli valik	27
3.3.2	Treeningprotsessi läbiviimine	27
3.3.3	Hüperparameetrite optimeerimine	28
4	Tulemuste analüüs	32
4.1	YOLOv8n mudeli tulemused	32
4.2	YOLOv8l mudelite tulemused ja võrdlus erinevates andmestikes	35
4.2.1	Šveitsi andmestiku tulemused	36
4.2.2	Portugali andmestiku tulemused	36
4.2.3	Kombineeritud andmestiku tulemused	37
4.2.4	Tulemuste võrdlus ja parima mudeli valik	37
5	Arutelu ja järeldused	39
5.1	Peamised järeldused	39
5.2	Töö piirangud	40
5.3	Tuleviku suunad ja edasised uuringud	40

6 Kokkuvõte	42
Kasutatud kirjandus	43
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	47
Lisa 2 – Šveitsi andmestikul treenitud mudeli tulemused	48
Lisa 3 – Portugali andmestikul treenitud mudeli tulemused	50
Lisa 4 – Kombineeritud andmestikul treenitud mudeli tulemused	52

Joonised

1	<i>YOLO objektituvastusmodelite areng aastatel 2015-2024. Allikas: Autori koostatud ajajoon teaduskirjanduse ja GitHubi väljalaskmiste põhjal . . .</i>	19
2	<i>Masinõppemudeli arendusprotsessi skeem. Allikas: Autori koostatud . . .</i>	21
3	<i>Annoteerimiskeskonna vaade. Allikas: Autori ekraanipilt märgistustark-varast</i>	23
4	<i>Šveitsi andmestiku annotatsioonide jaotus. Allikas: Autori koostatud (Šveitsi andmestiku põhjal)</i>	24
5	<i>Portugali andmestiku annotatsioonide jaotus. Allikas: Autori koostatud (Portugali andmestiku põhjal)</i>	25
6	<i>Šveitsi "Trout"kaamera näidiskaader, kus on üks kala nähtaval. Allikas: Šveitsi "Trout"kaamera</i>	26
7	<i>Portugali kaamera näidiskaader, kus on kolm kala nähtaval. Allikas: Portugali kaamera</i>	26
8	<i>Kombineeritud mudeli hüperparameetrite mõju mAP@50-le, absoluutne korrelatsioon. Allikas: Andmed on saadud Optuna katsetustest, mille tulemused on salvestatud. Korrelatsioonide arvutamine põhineb mAP@50 ja hüperparameetrite vahel</i>	31
9	<i>YOLOv8n mudeli treening- ja valideerimistulemuste ülevaade. Allikas: Autor, genereeritud Ultralytics YOLOv8 raamistiku abil</i>	33
10	<i>YOLOv8n mudeli objektide põhine segadusmaatriks Šveitsi andmestiku valideerimiskogumil. Allikas: Autor, genereeritud Ultralytics YOLOv8 raamistiku abil</i>	34
11	<i>YOLOv8n mudeli piltide põhine segadusmaatriks Šveitsi andmestiku valideerimiskogumil. Allikas: Autor, juhtumite kokkulugemisel Šveitsi andmestiku põhjal</i>	35
12	<i>Šveitsi andmestikul treenitud YOLOv8l mudeli objektide põhine segadusmaatriks Šveitsi andmestiku valideerimiskogumil. Allikas: Autor, genereeritud Ultralytics YOLOv8 raamistiku abil</i>	48
13	<i>Šveitsi andmestikul treenitud YOLOv8l mudeli treening- ja valideerimistulemuste ülevaade. Allikas: Autor, genereeritud Ultralytics YOLOv8 raamistiku abil</i>	49

14	<i>Šveitsi andmestikul treenitud YOLOv8l mudeli piltide põhine segadusmaatriks Šveitsi andmestiku valideerimiskogumil. Allikas: Autor, juhtumite kokkulugemisel Šveitsi andmestiku põhjal</i>	49
15	<i>Portugali andmestikul treenitud YOLOv8l mudeli objektide põhine segadusmaatriks Šveitsi andmestiku valideerimiskogumil. Allikas: Autor, genereeritud Ultralytics YOLOv8 raamistiku abil</i>	50
16	<i>Portugali andmestikul treenitud YOLOv8l mudeli treening- ja valideerimistulemuste ülevaade. Allikas: Autor, genereeritud Ultralytics YOLOv8 raamistiku abil</i>	51
17	<i>Portugali andmestikul treenitud YOLOv8l mudeli piltide põhine segadusmaatriks Šveitsi andmestiku valideerimiskogumil. Allikas: Autor, juhtumite kokkulugemisel Šveitsi andmestiku põhjal</i>	51
18	<i>Kombineeritud andmestikul treenitud YOLOv8l mudeli objektide põhine segadusmaatriks Šveitsi andmestiku valideerimiskogumil. Allikas: Autor, genereeritud Ultralytics YOLOv8 raamistiku abil</i>	52
19	<i>Kombineeritud andmestikul treenitud YOLOv8l mudeli treening- ja valideerimistulemuste ülevaade. Allikas: Autor, genereeritud Ultralytics YOLOv8 raamistiku abil</i>	53
20	<i>Kombineeritud andmestikul treenitud YOLOv8l mudeli piltide põhine segadusmaatriks Šveitsi andmestiku valideerimiskogumil. Allikas: Autor, juhtumite kokkulugemisel Šveitsi andmestiku põhjal</i>	53

Tabelid

1	<i>Objektiivastusmodelite võrdlus COCO test-dev andmestikul</i>	16
2	<i>Mudelitele leitud optimaalsed hüperparameetrid</i>	29
3	<i>Kombineeritud andmestiku katsete mAP@50 ning vastavad hüperparameetrid</i>	30
4	<i>YOLOv8l mudelite võrdlus erinevatel andmestikel</i>	36

1. Sissejuhatus

Automaatne kalade olemasolu tuvastamine veealustelt kaameratelt on muutunud tänapäeva ökoloogilistes ja keskkonnaseire projektides järjest olulisemaks [1]. Traditsioonilised meetodid, mis tuginevad käsitsi videolõikude läbivaatamisele, on aeglasemad, kulukamad, sageli subjektiivsed ning pole skaleeritavad. Masinõppepõhised objektituvastuse mudelid pakuvad võimalust sooritada reaajas kalade tuvastust piiratud riistvaratingimustes mitmetes asukohtades automaatselt [2], võimaldades bioloogidel ja keskkonnauurijatel kiiremini ja täpsemalt andmeid töödelda. Näiteks *Journal of Marine Science and Engineering* ajakirjas avaldatud uuringus saavutati YOLOv5s mudeli tuvastuskiiruseks 32.96 kaadrit sekundis, mis võimaldab reaajas töötlemist ka piiratud ressursiga seadmetel [3].

Käesoleva töö eesmärk on arendada ja hinnata YOLOv8 [4] mudelipõhist lahendust, mis tuvastab veealustest kaamerateist saadud piltidelt kalade olemasolu, rõhutades eelkõige valenegatiivsete juhtumite minimeerimist, ehk seda, et mudel ei jäta ühtegi reaalselt kala märkamata. Analüüs toimub kolmel erineval treeningandmestikul (Šveitsi, Portugali ja nende kombineeritud andmestikul), et uurida, kuidas mudel üldistub visuaalselt erinevates veealustes tingimustes.

Peamiste hindamismõõdikutena kasutati täpsust (*Precision*), tundlikkust (*Recall*), ja mudelite keskmist täpsust erinevatel piirjoonte kattuvuste lävedel (*mean Average Precision at IoU x*, edaspidi mAP@x, kus x tähistab piirjoonte kattuvust). Lisaks analüüsiti nii kaotufunktsioone ja objekti- kui ka pildipõhiseid segadusmaatrikseid, et saada terviklik ülevaade mudeli sooritustest ja käitumisest.

mAP on enimkasutatud täpsusnäitaja objektituvastuses, mis põhineb täpsuse (*precision*) ja tundlikkuse (*recall*) arvutamisel eri usalduslävete juures. Arvutatakse iga objektiklassi keskmine täpsus (AP) ning seejärel arvutatakse kõikide klasside AP keskmine, milleks on mAP. AP50 loeb kõik tuvastused õigeks, kus objekti pärisasukoht ja tuvastuse kattuvuse suhe (*Intersection over Union* ehk IoU) on vähemalt 50%. COCO kasutab oma andmestikus hindamiseks mAP-de keskmist väärtust vahemikus AP50 - AP95 sammuga 5%, mille tulemused on väga ranged ja tasakaalustatud, premeerides nii õigeid tuvastusi kui ka häid lokaliseerimisi. [5]

Parameetrite arv näitab, kui palju kaalutegureid mudel sisaldab. Üldiselt tähendab suurem parameetrite arv suuremat mudelit, millel on võimalik õppida andmestikust keerulisemaid

mustreid ja seeläbi täpsust tõsta. Selle kompromissiks on aga suurem mälu nõudlus ja aeglasem ennustuskiirus.

GFLOPs (giga floating point operations) näitab, mitu miljardit ujukomaarvutust on vaja ühe sisendi töötamiseks, mistõttu on see hea näitaja mudeli arvutuslikust keerukusest. Mida suurem on GFLOPs, seda rohkem arvutusressursse on vaja mudeli jooksumiseks.

Töö raames on püstitatud üks uurimisküsimus ning sellega seotud hüpotees, millele püütakse töö käigus vastused leida:

Uurimisküsimus: kas iga asukohapõhine veealune kaamera vajab mudeli jaoks eraldi ümberõppimist või peenhäälestamist, või saab sama mudeli edukalt rakendada mitmes erinevas asukohas?

Hüpotees: mudel, mis on treenitud eranditult ühe asukoha andmestikul, annab teise asukoha andmestiku valideerimisel madalama soorituse kui mudel, mis on treenitud kas osaliselt või täielikult valideerimisasukoha andmestiku peal.

Töös kasutati YOLOv8 large versioone (edaspidi YOLOv8l) mudelite võrdluseks, mis annab võimalikult täpse ülevaate andmestike mõjust mudelile, ilma et mudeli jõudlus jääks pudelikaelaks ning parima sooritusega andmestikul treenitud YOLOv8 nanoversiooni (edaspidi YOLOv8n) lõppseadmetesse paigalduseks, mis lubab piiratud arvutusressurssidega seadmetes rakendust reaalajas.

Töö süsteemne panus seisneb võrreldavate veealuste andmestike analüüsis ning nende mõju hindamises masinõppe mudelite üldistusvõimele binaarses kala olemasolu tuvastamise kontekstis. Erinevate asukohtade kaamerateest pärinevate andmestike kasutamine võimaldab hinnata, kas üks üldmudel suudab edukalt rakendada mitmes keskkonnas või on vajalik lokaalne kohandamine. Kuigi lõppmudeli rakendamine piiratud arvutusvõimekusega seadmes, nagu NVIDIA Jetson, ei ole töö keskne eesmärk, kujutab see endast olulist lisaväärtust praktikas rakendatava lahendusena. Käesolev töö ei käsitlenud kalaliikide tuvastamist ega nende suuruse hindamist, vaid keskendub üksnes binaarsele ülesandele - tuvastada, kas veealusel kaadril esineb kala või mitte.

Töö jaguneb viieks peatükiks. Teoreetiline taust (peatükk 2), kus antakse ülevaade seni enimkasutatud objektide tuvastusmudelitest ja käesoleva töö mudeli valikust, meetodika (peatükk 3), mis kirjeldab andmestike päritolu, eeltöötust ja annotatsiooniprotsessi koos mudelite hüperparameetrite optimeerimisega ja treenimisega, tulemuste analüüs (peatükk 4), kus võrreldakse mudelite sooritust Šveitsi andmestiku valideerimiskogumil

mAP, täpsuse, tundlikuse ja segadusmaatriksite kaudu, arutelu ja järeldused (peatükk 5), kus tõlgendatakse leide, hinnatakse mudeli praktilist rakendatavust ning käsitletakse piiranguid ja tuleviku võimalusi, ning lõpuks lisad, kust leiab kõik segadusmaatriksid ja kaotusfunktsiooni graafikud.

Autor tänab oma juhendajat Jefferey Andrew Tuhtanit abi ja kasulike nõuannete eest.

2. Objektide tuvastamine arvutinägemisega

Objektide tuvastamine (*Object Detection*) on arvutinägemise ülesanne, mille eesmärk on tuvastada, millised objektid pildil esinevad ja määrata nende täpsed asukohad. See ühendab klassifitseerimise (mis objekt see on) ja lokaliseerimise (kus see asub), kus objektide asukohad märgisatakse tavaliselt piirkastide (*Bounding Box*) abil.

Objektide tuvastamine on leidnud laialdast praktilist rakendust erinevates valdkondades, sealhulgas isesõitvates autodes, meditsiinilises diagnostikas, videovalves ja turvasüsteemides. Selles töös rakendatakse objektide tuvastamist veealuste kaamerate piltidel esinevate kalade automaatseks leidmiseks ja eristamiseks. Objektide tuvastamise muudab keeruliseks see, et objektid võivad varieeruda suuruses, kujus ja asukohas. Sageli esinevad pildil kattumised, taustamüra ja erinevad valgustustingimused, mis on oluliseks väljakutseks objektide tuvastamisel ning nõuab mudelilt head üldistusvõimet [6].

2.1 Objektide tuvastamise mudelid

Objekti tuvastamise mudelid on viimase kümnendi jooksul läbinud kiire arengu, liikudes keerukatest mitmeetapilistest lähenemistest lihtsamate ja tõhusamate reaalajasüsteemideni. Kuna lõputöö eesmärgiks on välja töötada ressursitõhus, kuid võimalikult täpne tuvastuslahendus piiratud riistvaraga seadmetes, on oluline mõista erinevate mudeliperekondade tugevusi ja piiranguid. Käesolevas alajaotuses käsitletakse valitud objekti tuvastamise lähenemisi, sealhulgas R-CNN [7], YOLO [8], EfficientDet [9] ja DETR [10], võrreldes nende arhitektuurilisi lahendusi, täpsust, ennustuskiirust ning sobivust rakendusvaldkonda.

2.1.1 YOLO (You Only Look Once)

2016 aastal arendati välja avatud lähtekoodiga YOLO (You Only Look Once) lähenemine, mis käsitleb objektide tuvastamist regressiooniprobleemina, ennustades objektide piirjooni ja klasside tõenäosusi otse piltidelt, kasutades selleks ainult ühte närvivõrku, mis võimaldab kogu tuvastusprotsessi optimeerida lõpptulemuse järgi. Siit tuleb ka selle nimi, YOLO “vaatab” pilti ainult ühe korra. Selle tulemusena on mudeli arhitektuur erakordselt kiire, suutes töödelda pilte reaalajas 45 kaadrit sekundis ning sellest väiksem versioon Fast YOLO 155 kaadrit sekundis. Lisaks saavutab see kaks korda suurema keskmise täpsuse (mAP) kui teised reaalajas tuvastavad objekti tuvastusmudelid. YOLO teeb küll rohkem vigu objektide täpsete asukohtadega, kuid on vähem tõenäoline tegema valepositiivseid

tuvastusi, mis on sobilik kompromiss töös lahendatava probleemi jaoks. [8]

2.1.2 R-CNN (Regions with CNN features)

Esimene objektide tuvastamisega tegelev meetod oli 2013 aastal välja tulnud R-CNN (Regions with CNN features), mis saavutas 30% parema keskmise täpsuse kui eelmised parimad pildi klassifikatsiooni meetodid. R-CNN genereerib esmalt piirkonnaettepanekud, seejärel tuvastab iga ettepaneku omadused CNN (*convolutional neural network*) abil ning seejärel klassifitseerib ettepanekud ja kohandab nende asukohti SVM-ide (*support vector machine*) ja lineaarsel regressioonil põhinevate mudelitega [7]. Sellele järgnesid 2015 aastal Fast R-CNN, mis parandas eelneva meetodi täpsust ja kiirust [11] ning 2016 aastal Faster R-CNN, mis arenas välja *Region Proposal Networki* (RPN), mis samaaegselt ennustab objektide piire ja nende tõenäosust igas positsioonis, tehes piirkonnaettepanekud, mis olid eelnevate meetodite pudelikaeladeks, ressursitõhusamaks [12]. Üldiselt on kaheastmelised mudelid nagu R-CNN täpsemad, kuid YOLO mudelid on mõnel juhul paremad nii kiiruse, täpsuse kui ka ressursitõhususe poolelt, mistõttu on YOLO eelistatavam mudel reaalse rakendustes [13].

2.1.3 Single Shot MultiBox Detector (SSD)

Single Shot MultiBox Detector (SSD) on ühe astme objekti tuvastusvõrk, mis ennustab objektide piirjooni ja klassitõenäosusi otse mitmelt konvolutsiooniliselt omaduskaardilt erinevatel skaaladel [14]. SSD512 mudel saavutas PASCAL VOC andmestikul keskmise täpsuse mAP@50 78,8%, mis on parem kui esialgsed YOLO mudelid nagu YOLOv1, mis saavutas 63,4% täpsuse, kuid kaasagsed YOLO mudelid nagu YOLOv8m ületavad selle 86,1% keskmise täpsusega [15, 8]. Lisaks ei ole käesolevas töös see sobilik mudeli valik veel kolmel põhjusel:

1. SSD arhitektuur ei skaleeri allapoole hästi, mistõttu väiksema arvutusvõimega lõppseadmetel (näiteks Jetson Nano) ei saavutata vajalikku ressursitõhusust.
2. SSD-l on halvem üldistusvõime, mis on oluline faktor, kuna valgustustingimused, kaugus kaamerast ja muud tingimused võivad varieeruda.
3. sellel on halvem kogukonna ja tööriistade tugi, mis teeb selle vähem kasutajasõbralikuks ning vähem tulevikukindlaks.

2.1.4 EfficientDet

EfficientDet mudeliperekond arendati välja 2019. aastal ning selle aluseks oli süsteemne arhitektuuriliste valikute uurimine koos mitmete optimeerivate komponentide lisamisega, et maksimeerida nii täpsust kui ressursitõhusust. Mudelite keskseks osaks on kaalutud kahesuunaline omaduste püramiid (BiFPN), mis võimaldab selle mudelitel skaleerida väikestest ja kiiretest versioonidest kuni suuremate, väga kõrgete täpsustega variantideni. [9] Sellel põhjusel on EfficientDet üheks heaks variandiks, mida kasutada, kuid YOLO mudeliperekonna eelisteks on suurem ennustuskiirus, kasutajasõbralikkus ja konkureeriv keskmine täpsus. [16]

2.1.5 DETR ja Transformer-põhised mudelid

DETR (DEtection TRansformer) mudelid on välja töötatud 2020. aastal ja lähenevad objekti tuvastusele komplekti ennustamisena, eemaldades vajaduse ankrustite ja NMS-i järele [10]. Võrreldes YOLO mudelitega, vajavad DETR mudelid pikemat treeningut ja suuremat arvutuskooormust, kuid selle tulemusel suudavad tuvastusi teostada kiiremini ja tagada parema üldistusvõime [17]. YOLO on sobilikum mudel käesolevas töös, kuna see pakub paremat tasakaalu täpsuse, kiiruse ja ressursikasutuse vahel piiratud riistvaraga süsteemi jaoks.

2.2 Mudelite võrdlus

Alljärgnev Tabel 1 toob välja valitud objektituvastusmudelite keskmise täpsuse (mAP), parameetrite arvu ja GFLOP-de hulga. Mudelid on järjestatud mAP väärtuse järgi kahanevalt, et näidata täpsuse ja arvutusliku keerukuse vahelist kompromissi. Tabelist on näha, et suurema täpsusega mudelid, nagu Co-DETR on sageli oluliselt ressursinõudlikumad kui väiksemad mudelid nagu YOLOv8n, mis sobivad paremini reaalajas piiratud ressursiga riistvaraga seadmetele.

Tabel 1. Objektituvastusmudelite võrdlus COCO test-dev andmestikul

Mudel	mAP	parameetrite arv (miljonites)	GFLOPs
Co-DETR	66,0 [18]	304 [19]	-
YOLOv8x	53,9 [20]	68 [21]	258 [21]
EfficientDet-D7	52,6 [18]	77 [9]	410 [9]
YOLOv7	51,4 [18]	36 [17]	104 [17]

Jätkub...

Tabel 1 – Jätkub...

Mudel	mAP	parameetrite arv (miljonites)	GFLOPs
YOLOv4	45,4 [18]	28 [22]	52 [22]
YOLOv8n	37,3 [20]	3 [21]	8 [21]
Faster R-CNN (ResNet-50 ja FPN)	35,9 [18]	42 [18]	447 [18]
SSD512	28,8 [18]	25 [23]	89 [23]
Fast R-CNN	19,7 [18]	-	-

Märkus: Mõõdikud GFLOPs ja parameetrite arv pärinevad osaliselt erinevatest allikatest ning võivad olla mõõdetud erinevate sisendresolutsioonide ja riistvarade puhul, mistõttu ei pruugi kõik väärtused olla võrreldavad üks ühele, vaid on suurusjärgu illustreerimiseks.

2.3 Käesoleva töö mudeli valik

Kuna YOLO on avatud koodibaasiga, on esialgsed autorid ja ka kõrvalisikud seda pidevalt arendanud ja täiustanud, pakkudes uusi omadusi ja parandusi täpsuses, kiiruses ja üldistes funktsioonides. Objektide tuvastamise valdkonnas on YOLO mudeliperekond muutunud üheks kõige laialdasemalt kasutatavaks ja tõhusamaks reaajas tuvastamise raamistikus. Siin vaadeldakse lühidalt erinevaid YOLO versioone ja selgitatakse, milline mudel kalade tuvastamise ülesandeks kõige paremini sobib.

Esimesed YOLO versioonid, YOLOv1 [8], YOLOv2 [24] ja YOLOv3 [25] on eeltreenitud mudelid andmestikel nagu PASCAL VOC, ImageNet ja COCO, mudelite kohendamine hõlmas peamiselt oma andmestiku peal peenhäälestamist [24, 25]. Käesoleva töö ülesannet on küll võimalik lahendada nende mudelitega, kuid hilisemad mudelid pakuvad rohkem paindlikkust, paremat täpsust ja kiirust.

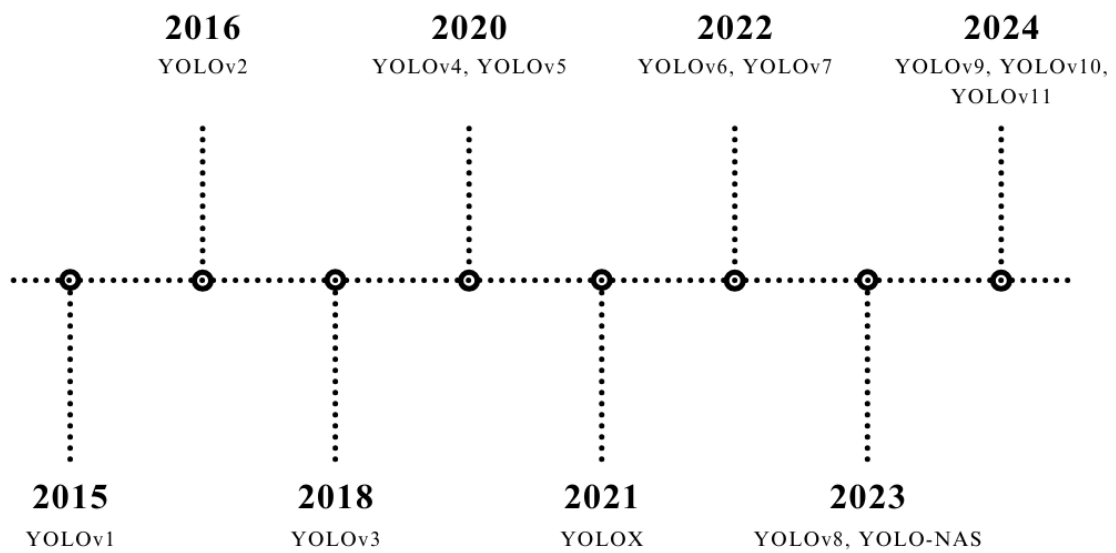
YOLOv4 [22] optimeeris eelnevat mudelit mitmel moel, kasutades CSPDarknet53 tuumvõrguna, mosaiik andmete juurdetekitamist ja enesevastast treenimist, mis tõstsid mudeli täpsust ja kiirust. YOLOv5 [26] on arendatud ja avalikustatud Ultralytics tiimi poolt, mis pakub kergemat kasutust, suuremat paindlikkust ja modulaarsust. Töö eesmärgiks on kalade tuvastamine reaajas piiratud ressursiga seadmetes, mistõttu on mõistlik kasutada veelgi tõhusamat ja kaasaegsemat mudelit.

YOLOv7 [27] on hilisem mudel YOLO mudeliperekonnas, mille on avalikustanud originaalsed autorid ning pakub optimeerimisi YOLOv4 mudelile veelgi täpsemaid ja oluliselt

kiiremaid tulemusi väiksemal hulgal treeningandmetel, ületades olemasolevate lahenduste kiiruse $\sim 500\%$ võrra [27]. Sellel põhjusel on see mudel üheks heaks variandiks käesolevas töös, kuid selle miinuseks on vähem paindlik süsteem ning see ei paku sama palju optimeerimisvõimalusi kui YOLOv8 [4].

YOLOv8 pakub kõrgemat täpsust ja suuremat kiirust kui YOLOv7 ning on hästi optimeeritud, modulaarne ja laialt aktsepteeritud teadusuuringutes ja tööstustes. Tänu nendele omadustele sobib see hästi rakendamiseks piiratud riistvaraga seadmetes, näiteks NVIDIA Jetsonites [28]. Lisaks, on sellel saadaval mitu erinevat mudeli varianti - nano, small, medium, large ja extra large [4]. Käesolevas töös kasutatakse YOLOv8 large (YOLOv8l) mudeleid erinevatel andmestikel treenitud mudelite võrdluseks teineteisega, kuna see võimaldab paremini hinnata andmestike kvaliteete, vältides seda, et pudelikaelaks jääb mudlite võimsus. Kui on selge, mis andmestik viib parimate tulemusteni, on võimalik treenida lõplik mudel sellel andmestikul kasutades YOLOv8 nano mudelit (YOLOv8n), mis on vähem ressursinõudlik ning sobilik projektis kasutatavate NVIDIA Jetson Origin Nanode peal jooksumiseks reaajas. See võimaldab esmalt hinnata andmestike kvaliteeti võimsamate mudelite abil ning seejärel juurutada parima tulemuse andnud andmestikul treenitud väike mudel lõppseadmele. YOLO mudelite arengut ja nende ilmumisaegu on visuaalselt kujutatud joonisel 1, mis aitab mõista versioonide ajalist järjestust.

Kuigi vahepeal on välja tulnud veel kõrvaliste isikute poolt arendatud mitmeid YOLO mudelite derivaate, nagu YOLOv6, YOLOX, YOLO-NAS, YOLOv9, YOLOv10 ja YOLOv11, ei ole need käesolevas töös käsitletud, kuna nende eesmärgid või optimeerimised kas erinevad YOLOv8 omast või on alles kiire arengu all eksperimentaalsed ja pole laialt kasutuses teadus- või rakenduskontekstis [29].



Joonis 1. *YOLO* objektituvastusmodelite areng aastatel 2015-2024.

Allikas: Autori koostatud ajajoon teaduskirjanduse ja GitHubi väljalaskmiste põhjal

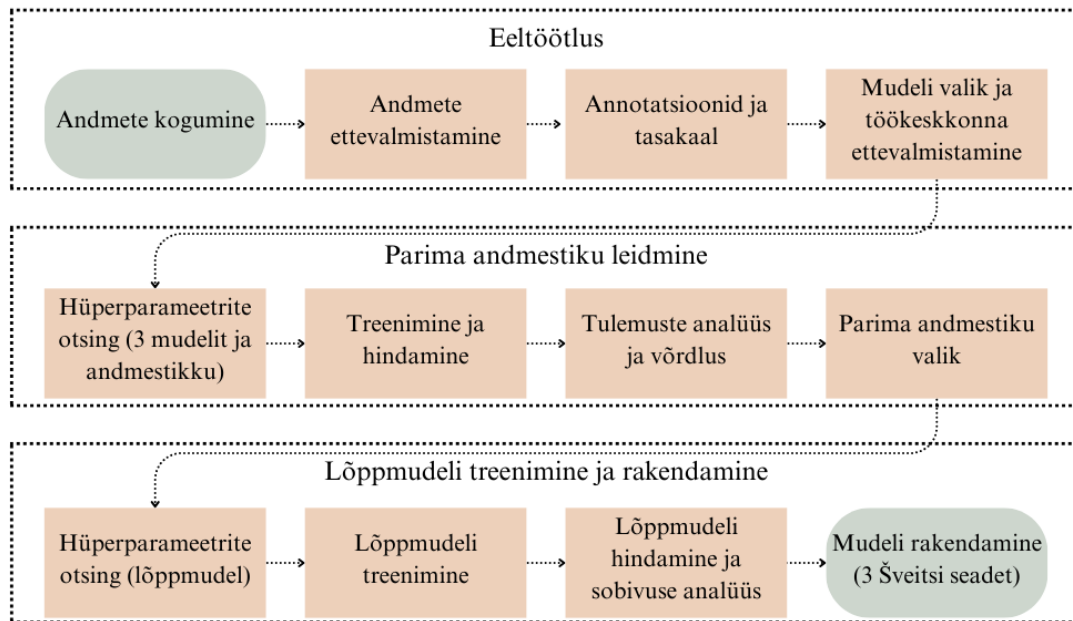
3. Metoodika

Käesolevas peatükis kirjeldatakse kasutatavaid meetodeid ja töövoogu, mille abil viidi läbi kalaliikide tuvastamine veealustelt kaamerapiltidelt. Esmalt tutvustatakse hindamismeetodeid, andmestikku ja selle eeltöötlust, seejärel kirjeldatakse masinõppemudelite ülesehitust ning treenimisprotsessi. Peatüki lõpuosas käsitletakse ka mudeli valideerimist ning tulemuste hindamismeetodeid. Metoodika valikul on arvesse võetud süsteemi hilisemat kasutamist ressursipiiratud keskkonnas ehk veealustes seadmetes.

Joonisel 2 on esitatud arendusprotsessi ülevaatlik skeem. Tööprotsess algas andmestiku kogumisega Šveitsi ja Portugali veealustest kaameratest, millele järgnes eeltöötlustetapp. Selles etapis filtreeriti videod, eraldati neist kaadrid, jaotati andmestik treeningu- ja testkomplektideks, koostati annotatsioonid ning kontrolliti klassitasakaalu.

Seejärel valiti sobiv masinõppemudel ning seadistati töökeskkond, kasutades kõrgtöövõimega arvutiklastrit. Iga andmestiku jaoks leiti esmalt optimaalsed hüperparameetrid, millega viidi läbi esmane treening. Järgmisena hinnati ja võrreldi erinevate mudelite tulemusi, mille põhjal valitati välja parimate sooritustega mudel. Sellele mudelile viidi läbi uus hüperparameetrite optimeerimine, et treenida väiksem ja tõhusam versioon, mis sobiks kasutamiseks piiratud ressursiga lõppseadmetes.

Lõpuks hinnati lõpliku mudeli täpsust ja sobivust reaalsetes tingimustes töötamiseks ning paigaldati see kolmele erinevale Šveitsis paiknevale veealusele seadmele. Kõiki mainitud etappe käsitletakse järgmistes alapeatükkides üksikasjalikumalt.



Joonis 2. Masinõppemudeli arendusprotsessi skeem.
Allikas: Autori koostatud

3.1 Hindamiseetodid

Kõigi mudelite puhul kasutati tulemuste hindamiseks järgmisi meetodeid:

- mAP@50 (*mean Average Precision* lävega $\text{IoU} = 0,5$), mis mõõdab, kui täpselt suudab mudel tuvastada objekte õigetes asukohtades. $\text{IoU} = 0,5$ tähendab, et tuvastatud objekti piirjoon peab vähemalt 50% kattuma tegeliku objekti piirjoonega. mAP annab keskmise täpsuse kõigi piltide ja klasside suhtes ning on üks enimkasutatavaid objekti tuvastuse täpsusnäitajaid.
- Segadusmaatriksid (*confusion matrices*), mis näitavad, kui palju esines õigeid ja vale ennustusi kala olemasolu kohta. Need aitavad analüüsida, kas mudel kipub üle- või alatuvaastama kala olemasolu.
- Täpsus (*precision*) näitab, kui suur osa mudeli poolt tuvastatud kaladest olid ka päriselt kalad. Kõrge täpsus tähendab vähe valepositiivseid.
- Tundlikkus (*recall*) näitab, kui suurt osa tegelikest kaladest suutis mudel õigesti tuvastada. Kõrge tundlikkus tähendab vähe valenegatiivseid.
- Kaotuse areng ajas (*loss curve*) on graafik, mis näitab treeningu jooksul mudeli

kaotuse (*loss*) väärtuse muutumist. See võimaldab jälgida mudeli õppimisprotsessi ning aitab tuvastada üleõppimist.

3.2 Andmestik ja eeltöötlus

Töös kasutatud andmestik pärineb kahest allikast: Šveitsis paiknevast veealusest kaamerast nimega *Trout* ning Taltechi / Portugali uurimisrühma varasemast kalade tuvastamise projektist [30]. Portugali andmed olid eelneva töö raames juba eeltöödeldud kujul, sisaldades üksikuid kaadreid, millele olid lisatud kalade paiknemist tähistatavad märgistusandmed (*Bounding Box* annotatsioonid).

Šveitsi kaamera salvestas andmed ajavahemikus 24. kuni 30. November 2024, kasutades olemasolevat reaaliajast töötavat mudelit, mis salvestas lühikesi (tavaliselt alla 3 minuti pikkuseid) videoklippe iga kord, kui kaadris tuvastati kala olemasolu. Kuna olemasolev mudel ei olnud eksimatu, tuli kõik Šveitsi videod manuaalselt üle kontrollida autori poolt, et kinnitada, kas videos oli tegelikult kala või oli tegemist valepositiivse tuvastusega. Andmete valideerimise tulemusena tuvastati 731 juhtumit, kus kala oli tõepoolest videos olemas, ning 914 juhtumit, kus tuvastus osutus valepositiivseks. Seega oli olemasoleva mudeli tõeliste positiivsete tulemuste määr ligikaudu 44,4%.

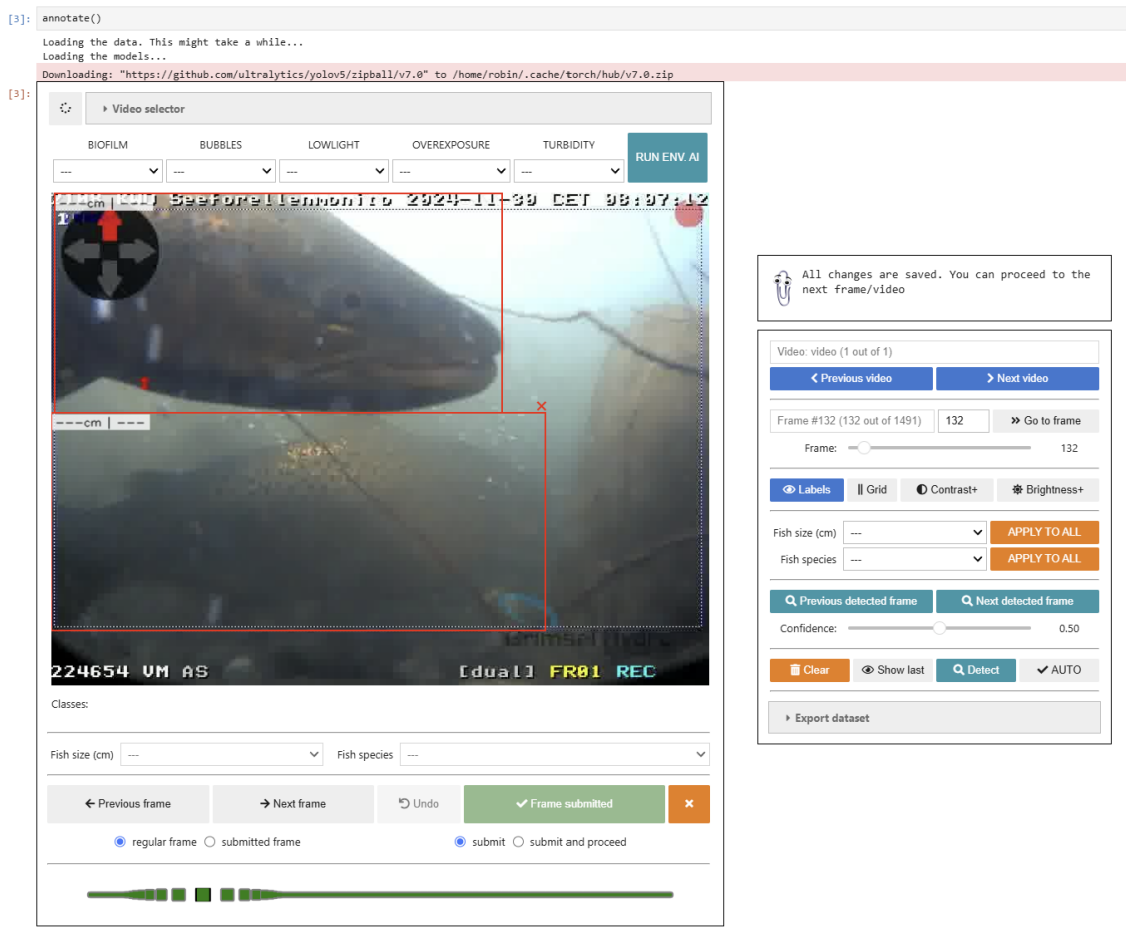
3.2.1 Videofailide töötlemine ja kaadrite annoteerimine

Kaadrite kättesaamiseks videofailist kasutati käsureapõhist tööriista *ffmpeg* [31], mille abil loodi iga videoklipi põhjal järjestatud pildikaadrid. Kuigi sisendiks olnud videod olid tuvastuspõhiselt filtreeritud ning kestuselt kokku vaid paarkümmend tundi, kujunes kaadrite eraldamine siiski ajamahukaks ülesandeks ning kokku genereeriti edasiseks analüüsiks üle 100 000 kaadri. Kuna Šveitsi kaamera tarkvaras oli rakendatud loogika, mille kohaselt pärast viimast kala tuvastust salvestati veel 3-sekundiline puhver, jäeti need viimased 3 sekundit kaadrite eraldamisel teadlikult vahele. Selle eesmärk oli tagada, et igast videost eraldatud kaader sisaldaks kindlasti kala. Protsess teostati automatiseeritud Python skriptiga, mis tuvastas iga .mp4 faili kestuse ja genereeris kaadrid sobivasse kausta.

Videod, kus kala puudus, kasutati tühjade kaadrite loomiseks, sealt saadi kokku 37 865 kaadrit. Iga tühja kaadri juurde loodi vastava nimega tühi tekstifail, mis treeingprotsessis toimis sildina, kuid ilma kastide asukohtade ja suuruste informatsioonita. Videoid, kus kala oli olemas, töödeldi samuti kaadriteks, sealt saadi kokku 97 936 kalaga kaadrit ning neile teostati käsitsi märgistus olemasoleva bounding box annotatsiooni tarkvara [32] abil. Nii Portugali ja Šveitsi andmestik jagati 85/15 suhtega treening- ja valideerimiskogumiteks,

et maksimeerida mudeli kvaliteeti, kuid siiski saada võimalikult hea ülevaade mudelite täpsusest.

Annotatsioonide kiirendamiseks treeniti esmalt lihtsustatud objektituvastusmodel, kasutades olemasolevaid eelnevalt märgistatud andmeid. Seda mudelit rakendati uute kaadrite esmasel tuvastusel, mille tulemusi kontrolliti ja korrigeeriti käsitsi. Selline poolautomaatne lähenemine vähendas märgistusaega märgatavalt. Annoteerimiseks kasutati tööriista, mis põhineb *jupyter-bbox-widget* paketil [32], kuid mida on projekti vajadustest lähtuvalt täiendatud. Täiendused võimaldavad paremat integreeritust andmetöötlusvooga ning parandavad kasutusmugavust märgistamisel (joonis 3).

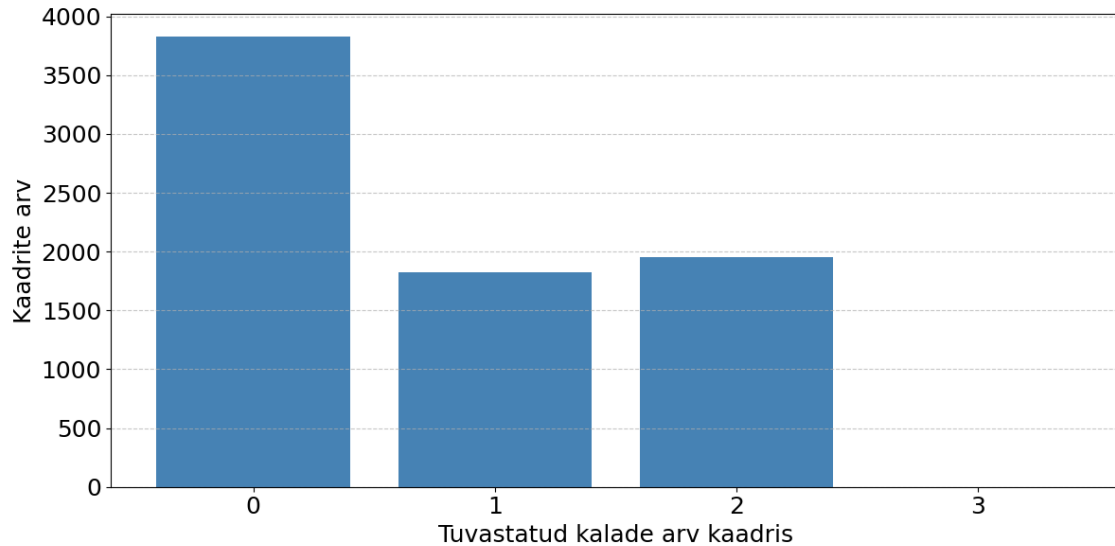


Joonis 3. Annoteerimiskeskonna vaade.
Allikas: Autori ekraanipilt märgistustarkvarast

3.2.2 Andmestike statistika

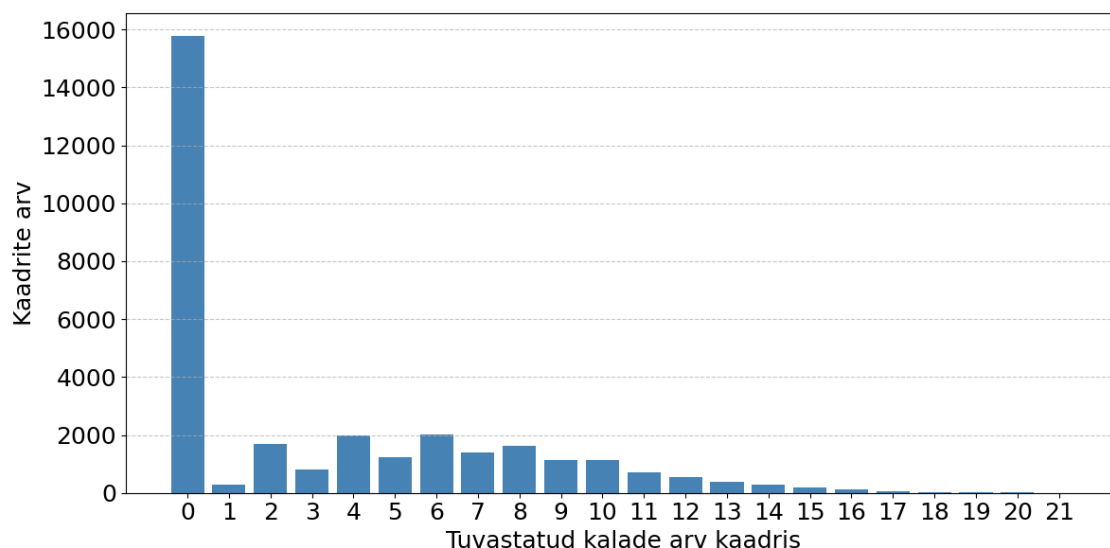
Pärast kaadrite annoteerimist koostati järgmine statistika Šveitsi andmestiku kohta, põhinedes kalade arvule (joonis 4). Kuna kõik kaadrid pärinesid manuaalselt valideeritud juhtumitest, sisaldas igauks neist vähemalt ühte kala. Ilma kaladeta kaadrid lisati hiljem

valepositiivsete tuvastuste videodest, et tasakaalustada treeningandmestik. Graafikust selgub, et ligikaudu pooltes kaadrites esines üks kala ja ülejäänud pooles esines kaks või kolm kala. Statistika koostamiseks kasutati annoteerimistulemuste tekstiridade arvu kokku lugemist, kuna iga rida vastab ühele annotatsioonile.



Joonis 4. Šveitsi andmestiku annotatsioonide jaotus.
Allikas: Autori koostatud (Šveitsi andmestiku põhjal)

Portugali andmestiku kohta koostati samuti statistika kaadrites tuvastatud kalade arvu kohta (joonis 5). Kuna kõik Portugali kaaadrid olid manuaalselt märgistatud, sisaldas igauks vähemalt ühte kala. Tulpdiaagramm näitab, et võrreldes Šveitsi andmestikuga on Portugali kaadrites kalade arv palju varieeruvam ja tihedam. Tuvastuste arv ühes kaadris ulatus üle kahekümne, ning enamus kaadreid sisaldas rohkem kui kolm kala. Levinud olid kaadrid, kus esines kümme või enam kala, mis viitab kalaparvede sagedasele esinemisele.



Joonis 5. Portugali andmestiku annotatsioonide jaotus.
Allikas: Autori koostatud (Portugali andmestiku põhjal)

3.2.3 Andmestike võrdlus ja eripärad

Allpool olevatel pildidel on näha näidiskaadrid Šveitsi ja Portugali kaamerateist. Šveitsi kaamerast (joonis 6) on näha suuremaid kalu, näiteks forelle, kes esinevad tavaliselt ühekaupa või väikestes gruppides. Pildikvaliteet on keskpärane ning kalade eristamist raskendavad mitmed keskkonnatingimused, selhulgas kehv valgustus, madal vee läbipaistvus ja vaatevälja takistavad objektid.

Portugali kaamera (joonis 7) töötab dual-kaamera süsteemina, kus iga kaader sisaldab kahte sünkroniseeritud vaadet. Sealsed keskkonnatingimused on paremad: vee läbipaistvus on parem, valgustus on ühtlasem. Siiski on sealsed kalad väiksemad ning liiguvad sageli suurtes parvedes, mis toob kaasa tuvastusprobleeme.

Nende erinevuste tõttu on oluline kaaluda, milline lähenemine mudeli arendamiseks on sobivam: kas arendada igale kaamerale kohandatud spetsialiseeritud mudelid või luua üks üldine mudel, mis suudab toime tulla võimalikult paljude erinevate keskkonna tingimustega. Spetsialiseeritud mudelid võimaldavad saavutada kõrgema täpsuse, kuna neid saab optimeerida konkreetse kaamera ja keskkonna eripärade järgi. Samas kaasneb sellega suurem arendus- ja hoolduskulu iga kaamera lisandumisel.

Üldine, kaamerateist sõltumatu mudel pakub paremat skaleeritavust ja lihtsamat kasutuselevõttu mitmes riigis ja asukohas. Samas eeldab see, et mudel suudaks saavutada piisavalt hea üldistusvõime, mis toimib hästi mitmesugustes tingimustes, mis võib kaasa

tuua kompromisse mudeli täpsuses.

Üheks potentsiaalseks kompromisslahenduseks on liitõpe (*federated learning*), mille puhul treenitakse üks üldine baasandmestikule tuginev mudel ning seejärel kohandatakse see mudel igasse asukohta selle asukoha andmete põhjal. Selline lähenemine ühendab üldise mudeli skaleeritavuse lokaalse kohandamise täpsusega, võimaldades süsteemi paremat kohanemisvõimet erinevates asukohtades. [33]



Joonis 6. Šveitsi "Trout"kaamera näidiskaader, kus on üks kala nähtaval.
Allikas: Šveitsi "Trout"kaamera



Joonis 7. Portugali kaamera näidiskaader, kus on kolm kala nähtaval.
Allikas: Portugali kaamera

3.3 Mudeli ülesehitus ja treenimine

Käesolevas peatükis kirjeldatakse masinõppemudelite arhitektuuri, treenimisekeskkonda ning andmestiku ettevalmistust. Lähtudes eelnevast jaotusest, kus tutvustati andmestiku

eripärasid ja annoteerimisprotsessi, keskendub see peatükk konkreetsetele tehnilistele otsustele, mis mõjutavad mudeli võimekust veealustes tingimustes kalu tuvastada.

Arvestades lõppeesmärki, kasutada mudelit ressursipiiratud keskkonnas, valiti sobiv mudeli arhitektuur kompromissina täpsuse ja arvutusliku efektiivsuse vahel. Samuti kirjeldatakse, kuidas andmestik jagati treeninguks ja valideerimiseks. Peatüki käigus tutvustatakse ka kasutatud hüperparameetreid, treeningprotsessi kulgu ja treenimiskeskonna konfiguratsiooni. Kokkuvõttes annab see ülevaate, kuidas andmete ja mudeli koostoimel kujunes välja süsteem, mis suudab veealustelt kaadritelt tõhusalt kalade esinemist tuvastada.

3.3.1 Mudeli valik

Mudeli valikul oli kaks peamist eesmärki: kõrge tuvastustäpsus ja tagada töövõime piiratud riistvaral, nagu projektis kasutatavad veealused kaamerad. YOLOv8 mudeliperekond on hästi skaleeritav ning võimaldab valida sobiva mudelivariandi vastavalt seadme võimekusele. Lõppseadme jaoks otsustati kasutada YOLOv8n mudelit, mis on perekonna kõige väiksem ja vähem ressursinõudvam variant, säilitades samas konkurentsivõimelise täpsuse (vt tabel 1).

Selleks, et hinnata erinevate andmestike mõjusid mudeli täpsusele ja üldistusvõimele, treeniti võrdluseks kolm YOLOv8l mudelit eraldi Portugali, Šveitsi ning kombineeritud (Portugal + Šveits) andmestikel. YOLOv8 mudeli valik selleks etapiks on tingitud selle suuremast täpsusest, mis võimaldab tundlikumalt mõõta erinevate andmestike mõju.

Lõplik mudel treeniti YOLOv8n arhitektuuril, et võimaldada selle rakendamist NVIDIA Jetson Orin Nano seadmel [34], millel on piiratud GPU ja mäluressurss. See treeniti Šveitsi andmestiku peal, kasutades eelnevalt optimeeritud hüperparameetreid, et tagada parim võimalik tulemus sihtplatvormil. Mudelite täpne treeningprotsess ja hüperparameetrite optimeerimine on kirjeldatud järgnevas jaotises.

3.3.2 Treeningprotsessi läbiviimine

Mudelite treening viidi läbi TalTechi kõrgtöövõimega arvutusklastri (HPC) [35], mis võimaldab kasutada suure jõudlusega graafikakaarte ja mitmetuumalisi protsessoreid. Treeningprotsessis kasutati Ultralyticsi YOLOv8 raamistiku Pythoni põhiseadme versiooni koos PyTorch masinõppe teegiga.

Treening viidi läbi Jupyter Labi virtuaalses treeningkeskkonnas. Tarkvaraversioonid, mis

olid kasutusel on järgmised: Python 3.10.16, Pytorch 2.7.0, Ultralytics YOLOv8 8.3.119, CUDA 11.8 ja Optuna 4.3.0.

Riistvara, mis oli kasutusel, oli amp2 arvutiklaster, kus saadaval olevast kaheksast NVIDIA A100 GPU-st kasutati kahte, 24 lõime ja 576 GB muutmälu. Treeningskriptid käivitati Slurm tööhalduri kaudu. Tulemused ja mudeli vahepealsed kontrollpunktid salvestati TalTechi failiserverisse.

Hüperparameetrite optimeerimine kestis ligikaudu 20 tundi iga mudeli kohta, kasutades 15 epohhi pikkust treeningtsükli 20 katse peal Optuna raamistiku kaudu. Optimeeritud parameetritega lõpptreening kestis mudelist ja andmestikust sõltuvalt 8 kuni 16 tundi, kasutades 100 epohhi pikkust treeningtsükli ning selle kestel salvestati parima valideerimistulemusega mudel. Treeningu edenemist jälgiti epohhi kaupa ning hinnati valideerimiskogumi soorituse põhjal.

Treeningprotsessi käigus ilmnasid mitmel korral GPU mälupiirangutest tingitud tõrked, täpsemalt hüperparameetrite optimeerimisel. Lahendusena suurendati muutmälu kasutuse hulka ning vähendati optimeerimise etapis sisendpiltide maksimaalset suurust, et vähendada ressursitarbimist ja tagada stabiilne töövoog.

Lisaks kasutati treeningprotsessi usaldusväärsuse ja tulemuste analüüside huvides logimist ja visualiseerimistööriistu, nagu Ultralytics sisseehitatud statistikamoodul. See võimaldas jälgida kaotuse (*loss*) ja täpsuse (*accuracy*) arengut kogu treeningu vältel.

3.3.3 Hüperparameetrite optimeerimine

Et tagada mudelite võrdlemisel parim võimalik täpsus ja ausad tingimused, viidi iga andmestiku jaoks läbi hüperparameetrite optimeerimine. Kõigi kolme mudeli treenimiseks kasutati eelnevalt optimeeritud hüperparameetreid, mille leidmiseks rakendati hüperparameetrite otsingut Optuna [36] raamistikus. Hüperparameetriteks, mille väärtusi otsiti, olid: algne õppemäär (*initial learning rate*, *lr0*), õppemäära lõppväärtuse kordaja (*learning rate final factor*, *lrf*), partiisuurus (*batch size*) ja sisendpildi suurus (*image size*). Leitud optimaalseid väärtusi kasutati lõplikuks mudeli treenimiseks, et oleks võimalik võrrelda erinevaid mudeleid võimalikult võrdsetes tingimustes.

Hüperparameetri otsing viidi läbi kõigi kolme mudeli puhul samades piirides, et tagada tulemuste võrreldatavus. Otsinguruum oli järgmine:

- algne õppemäär (*lr0*) logaritmiliselt jaotatud vahemikus 0,00001 kuni 0,01

- õppemäära lõppväärtuse kordaja (*lrf*) ühtlaselt jaotatud vahemikus 0,1 kuni 1
- partiisuurus (*batch size*) väärtustega vahemikus 8 kuni 64 sammuga 8
- sisendpildi suurus (*image size*) väärtustega vahemikus 640x640 kuni 960x960 sammuga 160 pikslit

Tabelist 2 on näha, et iga andmestik nõudis oma spetsiifilise struktuuri ja varieeruvuse tõttu erinevat lähenemist treenimisele. Kõikide mudelite puhul osutus optimaalseks üsna madal algne õppemäär, mis viitab sellele, et stabiilne ja aeglasem õppimine andis paremaid tulemusi. Mudelile YOLOv8l_CH leiti väikseim sisendpildi suurus ja väikseim partiisuurus, mis viitab sellele, et Šveitsi andmestiku omadustele sobis paremini detailsem töö väiksema andmemahuga. Kombineeritud andmestikul treenitud mudelil oli suurim partiisuurus ja suur sisendpilt, mis viitab sellele, et mitmekesine andmestik kasutas efektiivsemalt suuremat treeningmahtu.

Tabel 2. *Mudelitele leitud optimaalsed hüperparameetrid*

Mudel	lr0	lrf	batch size	image size
Yolov8l_PT	0,0001	0,2	16	960
Yolov8l_CH	0,0001	0,11	8	640
Yolov8l_PT+CH	0,0001	0,4	32	960

Optimaalsete hüperparameetrite valik põhines Optuna [36] raamistiku abil läbi viidud hüperparameetrite otsingul, kus katsetati 20 erinevat hüperparameetrite kombinatsiooni, mille iga katse kestis 15 epohhi (*epoch*). Katsetes mõõdeti iga kombinatsiooni mõju täpsusele, kasutades mAP@50 mõõdikut.

Näiteks kombineeritud andmestiku puhul YOLOv8l mudelile läbi viidud Optuna otsingus leiti 20 erinevat hüperparameetrite kombinatsiooni. Alltoodud tabelis 3 on toodud iga katse (*trial*) tulemusena saavutatud mAP@50 ning vastavad hüperparameetrid.

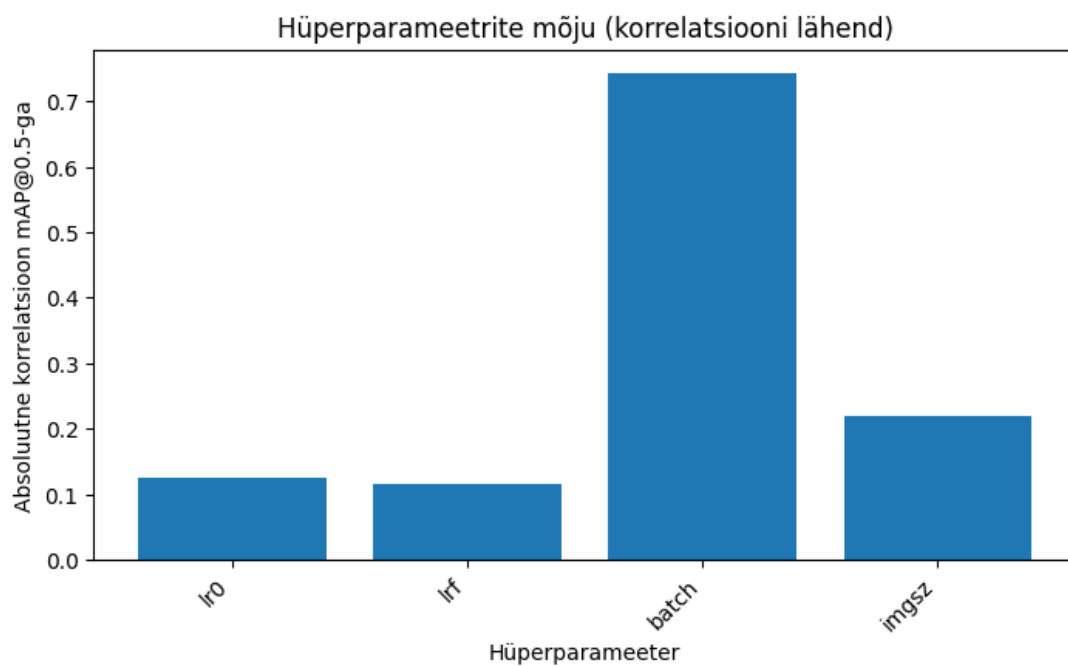
Optimaalsete hüperparameetrite valikul oli oluline tasakaalu leidmine õppimiskiiruse ja piltide suuruse vahel, kuna piltide ja partiisuuruste suurendamine võib kaasa tuua suurema täpsuse, kuid ka süsteemi arvutuslikud nõudmised tõusevad märgatavalt. Kõrgemad partiiisuuruse väärtused, nagu 32, koos suuremate piltidega, nagu 960x960, võimaldasid mudelil saavutada suurema täpsuse. Samuti oli oluline ka õppemäära ja õppemäära lõppkordaja väärtuste valik, kuna need mõjutavad mudeli õppimisprotsessi kaalude muutmise kiirust ja seeläbi ka efektiivsust ja täpsust.

Lisaks tabelis 3 esitatud katsetulemustele koostati joonis 8, mis illustreerib erinevate hü-

perparameetrite absoluutset korrelatsiooni mAP@50-ga, andes ülevaate sellest, millised parameetrid mõjutasid mudeli täpsust kõige enam. Selle põhjal oli võimalik teha teadlikke valikuid, suunates optimeerimist enim mõju avaldanud parameetritele. Näiteks on selgelt näha, et partiisuurusel (*batch size*) oli kombineeritud andmestiku puhul väga suur mõju mAP@50-le, võrreldes teiste parameetritega nagu algne õppemäär (*lr0*) ja õppemäära lõppväärtuse kordaja (*lrf*). Seetõttu pöörati parimate parameetrite valimisel sellele hüperparameetrile rohkem tähelepanu. Samuti on oluline märkida, et suuremate partiisuuruste kasutamisel kasvavad ka süsteemi arvutuslikud nõudmised, tehes treeningu aeglasemaks ja arvutuslikult kulukamaks.

Tabel 3. *Kombineeritud andmestiku katsete mAP@50 ning vastavad hüperparameetrid*

Trial	mAP@50	lr0	lrf	batch size	image size
13	0,973	0,000078	0,341	32	960
6	0,973	0,00312	0,174	32	640
16	0,972	0,000110	0,617	32	960
2	0,972	0,000063	0,334	32	960
1	0,972	0,000383	0,733	32	960
3	0,971	0,000769	0,136	32	640
19	0,970	0,000033	0,193	24	960
15	0,969	0,000010	0,433	24	640
10	0,968	0,00690	0,479	24	640
17	0,966	0,00141	0,222	32	800
7	0,966	0,000335	0,869	24	960
12	0,966	0,000059	0,326	24	640
11	0,966	0,000020	0,327	32	800
8	0,965	0,000686	0,493	8	800
4	0,960	0,000411	0,651	8	800
18	0,958	0,00808	0,358	16	640
5	0,949	0,000392	0,875	8	960
14	0,948	0,00364	0,243	16	800
9	0,944	0,00259	0,119	16	800
0	0,927	0,000190	0,225	8	640



Joonis 8. *Kombineeritud mudeli hüperparameetrite mõju mAP@50-le, absoluutne korrelatsioon.*

Allikas: Andmed on saadud Optuna katsetustest, mille tulemused on salvestatud. Korrelatsioonide arvutamine põhineb mAP@50 ja hüperparameetrite vahel

4. Tulemuste analüüs

Tulemuste analüüsi peatükk keskendub treenitud modelite täpsusele, üldistusvõimele ja toimivusele eri andmestike puhul. Hinnati nii treening- kui ka valideerimistulemusi, kasutades mitmeid mõõdikuid ja visuaale, sealhulgas segadusmaatrikseid ja täpsuse arengukõveraid.

4.1 YOLOv8n mudeli tulemused

YOLOv8n mudel treeniti hüperparameetrite optimeerimise tulemusena leitud parameetritega Šveitsi andmestikul, selle tulemused on järgmised:

- Täpsus (*precision*): 0,959 - 95,9% kõigist mudeli tuvastustest osutusid ka tegelikult õigeks.
- Tundlikkus (*recall*): 0,972 - mudel suudab õigesti tuvastada 97,2% kõigist juhtumitest, kus päriselt on tegemist kalaga.
- mAP@50: 0,99 - keskmine täpsus, kus objekti ja kala piirjoonte kattuvus on vähemalt 50%, on 99%.
- mAP@50-95: 0,881 - keskmine täpsus erinevate piirjoonte kattuvuse lävedes vahemikus 50% kuni 95% on 88,1%.

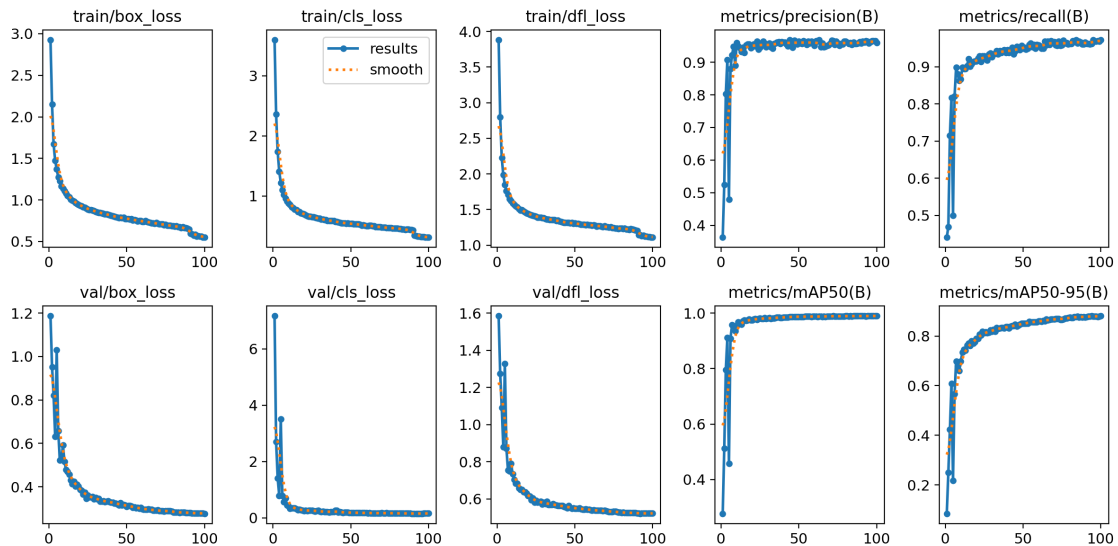
Joonisel 9 on kujutatud mudeli treeningu jooksul mõõdetud kaotusfunktsioonid ja hindamismõõdikud. Graafikud on YOLOv8 raamistikust genereeritud tulemused. Kuvatud on:

- *train/box_loss* - kastide asukoha kaotusfunktsioon (*bounding box regression loss*)
- *train/cls_loss* - klassifitseerimiskaotus (*classification loss*)
- *dfl_loss* - jaotuse fookuskaotus (*distribution focal loss*)
- *metrics/precision(B)* - täpsus valideerimiskogumil
- *metrics/recall(B)* - tundlikkus valideerimiskogumil
- *metrics/mAP50(B)* - mAP@50 valideerimiskogumil
- *metrics/mAP50-95(B)* - mAP@50-95 valideerimiskogumil
- *val/* eesliide tähistab vastava mõõdiku väärtust valideerimiskogumil

Kaotusfunktsioonide analüüsil on võimalik hinnata, kas mudel on üleõppinud (*overfitting*) või mitte. Üleõppimise tunnusteks võivad olla näiteks: valideerimiskogumi kaotusfunktsioonid (*val/loss*) ei vähene samas tempos kui treenimiskogumi kaotusfunktsioonid (*train/*

loss), vaid jäävad kõrgele või isegi tõusevad [37]. Jooniselt 9 on näha, et *train/box_loss* ja *val/box_loss* kaotusfunktsioonide väärtused langevad sarnaselt, mis viitab üleõppimise puudumisele. Samal ajal saavutab *metrics/mAP50(B)* kiiresti kõrge väärtuse ja stabiliseerub, mis näitab mudeli efektiivset õppimisvõimet. *metrics/mAP50-95(B)* väärtus tõuseb kõrgele tasemele aeglasemalt, peegeldades mudeli lokaliseerimistäpsuse jätkuvat paranemist treeningu käigus.

Antud mudeli tulemuste põhjal võib järeldada, et treening- ja valideerimiskaotused vähenevad stabiilselt ning hindamismõõdikud paranevad mõlemas komplektis, viidates heale üldistusvõimele ja sellele, et üleõppimist ei esine. Lisaks *mAP@50* väärtus üle 0,9 ja *mAP@50-95* väärtus üle 0,85 viitavad sellele, et mudel suudab kalade esinemise ja asukohta väga täpselt tuvastada erinevates olukordades. Samuti on mudeli täpsus ja tundlikkus kõrged, mis tähendab, et valetuvastusi ja tuvastamata kalu on vähe ning enamus kalu tuvastatakse.



Joonis 9. YOLOv8n mudeli treening- ja valideerimistulemuste ülevaade.

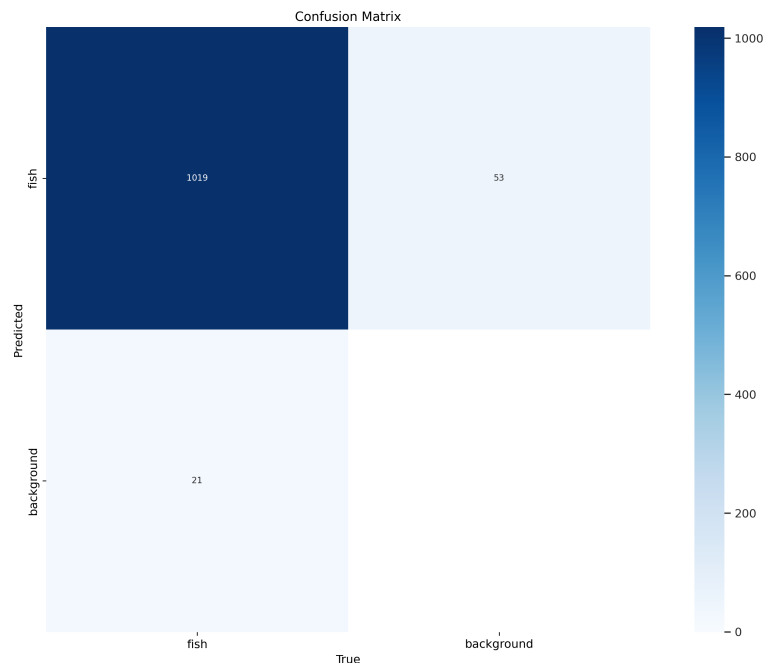
Allikas: Autor, genereeritud Ultralytics YOLOv8 raamistiku abil

Segadusmaatriksis (joonis 10) on kujutatud mudeli objekti-põhine klassifikatsioon:

- Tõelised positiivid: 1019
- Vale positiivid: 53
- Vale negatiivid: 21
- Tõelised negatiivid: ei rakendata YOLO objekti ennustuse kontekstis, kuna negatiivseid piirkondi ehk taustapilte või alampiirkondi, kus ühtegi objekti ei ole, on igal pildil pea lõpmatu hulk ning ei pakuks objekti tuvastuse kvaliteedi hindamiseks lisaväärtust [38].

Maatriksi selgitus:

- Tõeline positiiv (*true positive*) tähendab, et mudel tuvastas kala ja kattuvus tegeliku piirjoonega ületas IoU läve.
- Vale positiiv (*false positive*) tähendab, et mudel tuvastas kala, kuid seda tegelikult ei olnud või kattuvus tegeliku piirjoonega jäi alla IoU läve.
- Vale negatiiv (*false negative*) tähendab, et mudel ei tuvastanud olemasolevat kala.



Joonis 10. YOLOv8n mudeli objektide põhine segadusmaatriks Šveitsi andmestiku valideerimiskogumil.

Allikas: Autor, genereeritud Ultralytics YOLOv8 raamistiku abil

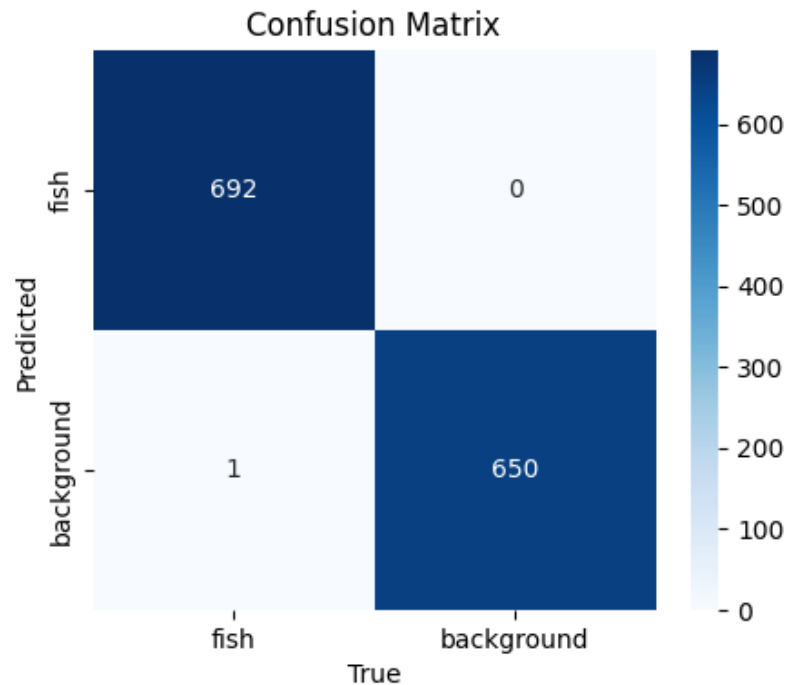
Teises segadusmaatriksis (joonis 11), mis on piltide-põhine, on esitatud:

- Tõelised positiivid: 692
- Vale positiivid: 0
- Vale negatiivid: 1
- Tõelised negatiivid: 650

See näitab mudeli võimet tuvastada kogu pildi tasandil ilma IoU piiranguta.

Objektide põhine segadusmaatriks on rangem, kuna see keskendub üksikutele tuvastatud

objektidele ja nõuab mudelilt piisavalt suurt piirjoonte kattumist (IoU) tõelise objektiga, kuid piltide põhine segadusmaatriks arvestab kogu pildi analüüsi, pakkudes rohkem üldteavet mudeli töötlustest.



Joonis 11. YOLOv8n mudeli piltide põhine segadusmaatriks Šveitsi andmestiku valideerimiskogumil.

Allikas: Autor, juhtumite kokkulugemisel Šveitsi andmestiku põhjal

YOLOv8n, olles YOLOv8 arhitektuuri kõige väiksem, nn "nano"versioon, saavutas Šveitsi andmestiku puhul väga head tulemused, hoolimata väiksemast arvutusressursi nõudlusest. Mudel suudab üldiselt kalu täpselt tuvastada erinevates olukordades ning on ka hea lokaliseerimisvõimega. Kaotusfunktsiooni analüüs ei viidanud üleõppimisele ning segadusmaatriksid näitavad head toimimist nii objektide kui ka piltide põhisel tasandil. Kokkuvõtteks näitavad tulemused seda, et mudel on usaldusväärne ja suudab täpselt tuvastada kalade esinemise, pakkudes efektiivset lahendust kalade jälgimiseks ja tuvastamiseks väiksema arvutusvõimekusega seadmetel.

4.2 YOLOv8l mudelite tulemused ja võrdlus erinevates andmestikes

Selle alapeatüki eesmärk on võrrelda kolme erinevas andmestikus treenitud YOLOv8l mudeli tulemusi, et hinnata nende töövõimet ja üldistusvõimet Šveitsi valideerimisandmestikul. YOLOv8l on suurendatud versioon YOLOv8 mudelist, mis suudab pakkuda kõrgemat täpsust ja paremat sooritust, kuid nõuab rohkem arvutusressursse. Selle analüüsi

raames hinnatakse, kuidas mudel toimib Šveitsi, Portugali ja kombineeritud andmestike puhul.

Tulemuste hindamiseks on kasutatud samu meetodeid, nagu YOLOv8n mudeli puhul. Kõik kolm mudelit hinnatakse järgmiste mõõdikutega: täpsus, tundlikkus, mAP@50, mAP@50-95, segadusmaatriksid ja kaotusfunktsioonid.

Tabel 4. *YOLOv8l mudelite võrdlus erinevatel andmestikel*

Andmestik	Täpsus	Tundlikkus	mAP@50	mAP@50-95
Šveits	0,965	0,978	0,991	0,901
Kombineeritud	0,962	0,974	0,99	0,89
Portugal	0	0	0	0

4.2.1 Šveitsi andmestiku tulemused

Šveitsi andmestikul treenitud mudel saavutas väga kõrged tulemused, mida on näha lisast 6: Õigesti tuvastati 1022 kalaobjekti, tuvastamata jäi 18 ning valepositiivseid tuvastusi esines 57.

Valideerimistulemused kinnitavad mudeli head toimimist: täpsus oli 0,965, tundlikkus 0,978, ning mAP50 ulatus 0,991 peale, mAP50-95 väärtuseks jäi 0,901. Sellest võib järeldada, et mudel suudab usaldusväärselt ja hea lokaliseerimisega kalu tuvastada Šveitsi tingimustes.

Lisaks jälgiti treeningprotsessi vältel kaotusfunktsiooni muutumist ning selle alusel ei ilmnenud üleõppimist.

4.2.2 Portugali andmestiku tulemused

Portugali andmestiku puhul mudeli sooritus langes oluliselt, tulemusi on näha lisast 6: mitte ühtegi kalaobjekti ei tuvastatud õigesti, samal ajal jäi 1040 objekti tuvastamata ning 12 objekti klassifitseeriti eksikult kaladena.

Valideerimistulemused kinnitasid seda tulemust, nii täpsus, tundlikkus kui ka mõlemad mAP väärtused olid nullilähedased, kuid piltidepõhise segadusmaatriksi põhjal suutis mudel ära tuvastada 13 pilti, kus on kala ning edukalt määras ära 650 pilti, kus kala puudub, kuid 680 kala jäi tuvastamata.

Kaotusfunktsioon ei viidanud siiski üleõppimisele, mistõttu võib järeldada, et mudeli halvad tulemused on tingitud andmestiku olemuslikust erinevusest võrreldes Šveitsi andmestikuga. Portugali veealused tingimused, valgustus, taust, kalaliigid kui ka kaamera liides erinesid valideerimisandmetest märkimisväärselt. Võib eeldada, et kuna tühjade kaadritena kasutati sellel andmestikul ainult Šveitsi kaamerast pärinevaid pilte, õppis mudel kalduma nendes mitte kalu tuvastama.

4.2.3 Kombineeritud andmestiku tulemused

Mudel, mis treeniti nii Šveitsi kui Portugali andmestikku kombineerides, saavutas 1024 tõset positiivi, 16 kala jäi tuvastamata ning valepositiivseid tulemusi esines 64.

Valideerimistulemused olid samuti tugevad: täpsus 0,962, tundlikkus 0,974, mAP50 0,990 ning mAP50-95 0,89. Need tulemused on sarnased Šveitsi andmestikuga treenitud mudeli tulemustele, kuid väga natukene aga järjepidevalt kehvemad. Erinevus on väike ja üldiselt võib järeldada, et Portugali andmestiku lisamine ei kahjustanud märgatavalt mudeli üldist tuvastusvõimet. Samuti kaotusfunktsioon ei viidanud üleõppimisele.

4.2.4 Tulemuste võrdlus ja parima mudeli valik

Tulemuste põhjal (tabel 4) võib parimaks pidada Šveitsi andmestiku põhjal treenitud mudelit, kuid erinevus on statistiliselt ebaoluline: see saavutas 0,3% võrra kõrgema täpsuse, 0,4% parema tundlikkuse, 0,1% parema mAP50 väärtuse kui ka 1,2% parema mAP50-95 väärtuse. Kuigi kombineeritud andmestiku põhjal treenitud mudel oli väga lähedal, näitavad kergelt langenud tulemused seda, et Portugali andmestiku lisamine suurendas mudeli üldistusvõimet, kuid samas tõi kaasa väikese täpsuskao Šveitsi andmestikul.

Portugali andmestiku põhjal treenitud mudel oli märgatavalt halvem ning seetõttu seda mudelit üksi ei saa praktiliseks kasutuseks Šveitsis pidada.

Kokkuvõtteks võib öelda, et:

- Šveitsi andmestikul treenitud mudel on kõige täpsem ja sobivam kasutamiseks sarnastes tingimustes, kus andmestik koguti.
- Kombineeritud andmestikul treenitud mudel pakub suuremat üldistusvõimet, säilitades samas väga hea täpsuse, mistõttu sobib paremini mitmekesistesse või ettearvamatusesse keskkondadesse.
- Portugali andmestikul treenitud mudel on antud juhul ebapiisav, kuna ei suuda

sarnaseid objekte teistsugustes tingimustes usaldusväärselt tuvastada.

Kuigi statistiliselt võivad Šveitsi ja kombineeritud mudeli tulemused tunduda väga sarnased, annab iga protsendipunkti paranemine kalade tuvastusel tegelikus kasutuses märkimisväärselt eelise, kus iga tuvastamata kala võib tähendada olulise info kaotust. Seetõttu valiti lõppseadmes kasutusele võtuks YOLOv8n mudelit treenitud ainult Šveitsi andmestikul. Treeningprotsessi täpsem kirjeldus on alapeatükis 4.1. Kui aga eesmärgiks on tulevikus arendada mitmekesisemates keskkondades töötav süsteem, siis on kombineeritud mudel tugevaks aluseks.

5. Arutelu ja järeldused

Käesolevas osas seotakse eelnevas peatükis (peatükk 4) esitatud tulemid laiemasse konteksti. Analüüsitakse mudelite praktilist kasutatavust, arutatakse töö piiranguid, valitakse lõppseadmetes rakendamiseks sobiv mudel ning edastatakse soovitusi edasisteks arenguteks.

5.1 Peamised järeldused

YOLOv8 arhitektuuri nanoversioon (YOLOv8n) suudab Šveitsi veekogude põhjas paiknevate kaameratelt tuvastada kalade olemasolu väga kõrge täpsusega piiratud arvutusressursside juures. Šveitsi andmestikul saavutati täpsus 95,9% ja tundlikkus 97,2%, mis näitab mudeli stabiilset toimimist varieeruvates valgustus- ja taustatingimustes.

Projekti praktiline eesmärk on tagada, et kalad ei jääks märkamata, ehk valenegatiivide arv oleks võimalikult väike, isegi kui kaasneb mõnevõrra suurem valepositiivsete tuvastuste osakaal. Kuna lõppeesmärgiks on aidata automaatselt filtreerida ja edasi suunata kaladega kaadreid bioloogilise järelanalüüsi jaoks, on suurem valepositiivsete osakaal aktsepteeritav kompromiss. Selle tasakaalu saavutamiseks saab mudeli enesekindlusläve (*confidence threshold*) väärtust kohandada: madalam lävi vähendab valenegatiivsete juhtumite arvu, kuid suurendab valepositiivsete tuvastuste hulka, võimaldades süsteemi vastavalt vajadustele tundlikumaks või täpsemaks seadistada.

Kombineeritud treeningandmestik (Portugali ja Šveitsi andmed) parandas mudeli üldistusvõimet, säilitades samal ajal kõrged, kuid veidi madalamad põhinäitajad (täpsus 96,2% ja tundlikkus 97,4%) kui ainult Šveitsi andmestik. Portugali andmestiku eripära tõttu eraldiseisvalt treenitud mudel ebaõnnestus, rõhutades kohapõhiste andmete tähtsust mudeli töökindlusele.

Kaotusfunktsioonide analüüs ei näidanud üleõppimise märke, kinnitades hüperparameetrite optimeerimise ja treeningprotsessi kvaliteedi õnnestumist.

Lisaks mudelite täpsuse ja üldistusvõime hindamisele toetab käesolev töö TalTechi olemasoleva seiresüsteemi arendamist ning kinnitab praktilist rakendusväärtust. Keskuses teostatav projekt tegeleb veealuste kaamerapiltide põhjal kalade ja muude veealuste objektide jälgimisega, sealhulgas rändekäitumise analüüsi, kaitsealuste või invasiivsete liikide seire

ning ökosüsteemide dünaamilise muuuste hindamisega. Käesolev töö pakub lisandväärtust selle kaudu, et loob aluse töökindlama ja skaleeritavama automaatse tuvastussüsteemi väljatöötamiseks.

5.2 Töö piirangud

Töö ulatus piirdus binaarse kalatuvastusega (olemasolu või puudumine). Olukorras, kus on vajalik kalaliikide täpne liigitamine või kalade mõõtmete hindamine, oleks vaja keerulisemat annotatsiooni ja ressursse nõudvamat mudelit.

Portugali andmestiku puhul näitas mudeli ebaõnnestumine, et mudel vajab visuaalselt mitmekesisemat andmestikku. Samuti mõjutab mudeli jõudlust oluliselt annotatsioonide kvaliteet ja valgustusolude varieeruvus.

Edge-platvormidel, näiteks projektis kasutatavatel NVIDIA Jetson Orin Nanodel töötamisel tuleb arvestada mudeli arvuts- ja energiavajadustega, kuigi YOLOv8n on optimeeritud piiratud ressursidele, on potentsiaalselt võimalik kasutusele võtta ka suurem mudel, nagu YOLOv8 small või medium, kui langetada pildivärskenduse sagedust või pildi kvaliteeti.

5.3 Tuleviku suunad ja edasised uuringud

Mudeli võimekuse suurendamiseks soovitatakse järgnevaid samme:

- Andmestiku laiendamine: koguda lisaandmeid erinevatest veealustest keskkondadest, valgustustingimustest ja kaamerasuundadest, et parandada mudeli üldistusvõimet.
- Domeenispetsiifiline peenhäälestus: kohandada juba treenitud mudel konkreetse keskkonna andmetega, et vähendada valenegatiivsete tuvastuste arvu ja suurendada täpsust.
- Mitmeklassiline tuvastus: laiendada mudelit kalaliikide klassifitseerimiseks või ligikaudsete mõõtmete hindamiseks.
- Reaalajas katsetused: hinnata mudeli stabiilsust ja jõudlust pikema tööperioodi jooksul lõppseadme peal ning optimeerida pildivärskenduse sagedust ja suurust ning energiatarbimist.
- Suuremad YOLOv8-mudelid: uurida, kas YOLOv8n asemele YOLOv8 small või medium versioon suudab anda parema täpsuse ilma ülemäärase ressursikuluta.
- Riistvara täiendus: valida sobivaim veealune kaamera, paigaldada fikseeritud kaugusele ja taustaga seadistus, näiteks taustakardin või kontrastne koridor, tagades ühtlase valgustuse ja optimaalse kaamerasihi, et vähendada müra ja parandada tuvastuste

usaldusväärsust.

Kokkuvõttes näitab töö, et YOLOv8n pakub tugevat alust kalade olemasolu tuvastamiseks veealustelt kaameratelt, kombineerides kiiret töötlust ja piiratud riistvaranõudeid. Tugevusteks on kõrged valideerimistulemused ja kohandatav enesekindluselävi ning väljakutsed seisnevad mitmekesisema andmestiku ja riistvaralise optimeerimise vajaduses. Lõppseadmetes kasutamiseks on eelistatud mudel, mis tasakaalustab tundlikkuse ja täpsuse praktiliste piirangutega, ning edasised uuringud peaksid keskenduma nii eri asukohtade mudelite peenhäälestamisele kui ka andme- ja riistvaratäiendusele, et saavutada süsteemne ja stabiilne automaatne seire.

6. Kokkuvõte

Töö käsitles automaatset kalade olemasolu tuvastamist veealustelt kaameratelt, kasutades YOLOv8 arhitektuuri nano- ja large-versioone. Metoodika hõlmas kahte sisult erinevat andmestikku (Šveits ja Portugal), mille peal mudelid treeniti nii eraldi kui kombineeritult, ning põhjalikku eeltöötlust ja annotatsiooni Jypiter-keskkonnas. Mudeli sooritust hinnati täpsuse, tundlikkuse, mAP@50, mAP@50-95 ning nii objekti- kui ka pildipõhiste segadusmaatriksite abil.

Tulemustest selgus, et Šveitsi andmestikul treenitud mudel pakub kõige kõrgemat täpsust, tundlikkust kui ka mAP väärtusi erinevatel piirjoonte kattuvuste lävedel, mistõttu treeniti lõppmudel tuginedes sellele andmestikule. YOLOv8n saavutas kõrge täpsuse (95,9%), tundlikkuse (97,2%), mAP@50 (99,0%) ja mAP@50-95 (88,1%). Lisaks ei järeldunud mitte ühestki mudeli kaotusfunktsioonide analüüsist üleõppimist.

Vastavalt püstitatud uurimisküsimusele kinnitasid tulemused hüpoteesi: mudelid, mis on treenitud ainult ühe asukoha andmestikul, ei üldistu hästi teise asukoha andmestikule, mistõttu on iga keskkonna jaoks vajalik vähemalt peenhäälestus või ümberõpe

Mudeli praktilises kasutuses on kriitiline minimeerida valenegatiivseid juhtumeid, mida saab reguleerida mudeli enesekindluslävega. Samuti aitavad veelgi stabiilsust ja tulemusi parandada keskkonna-, riistvara- ja andmetäiendused, nagu kvaliteetne kaamera, ühtlane valgustus, kontrastne taust ja suurem andmekogum.

Töö piirangud on seotud binaarse lähenemisega. Tulevikus on projekti raames võimalik tegeleda mitmeklassilise liigitusega, domeenispetsiifilise peenhäälestusega, kalade mõõtmete hindamisega ja suuremate või uuemate YOLO mudelite rakendamisega lõppseadmetel. Kokkuvõttes näitab töö, et Šveitsi andmestikul treenitud YOLOv8n mudel on ressursikohane, tugev ja lubav lahendus kalade automaatseks seireks ning et iga asukoha spetsiifiline peenhäälestamine on tõepoolest vajalik.

Kasutatud kirjandus

- [1] S. Lopez-Marcano *et al.* „Automatic detection of fish and tracking of movement for ecology“. *Ecology and Evolution* 11.12 (2021). [Accessed: 16-05-2025], lk. 8254–8263. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC8216886/>.
- [2] E. M. Ditria *et al.* „Deep learning for automated analysis of fish abundance: the benefits of training across multiple habitats“. *Environmental Monitoring and Assessment* 192.11 (2020). [Accessed: 18-05-2025], lk. 698. URL: <https://pubmed.ncbi.nlm.nih.gov/33044609/>.
- [3] W. Li *et al.* „A Real-Time Fish Target Detection Algorithm Based on Improved YOLOv5“. *Marine Science and Engineering* 11.3 (2023). [Accessed: 18-05-2025], lk. 572. URL: <https://www.mdpi.com/2077-1312/11/3/572>.
- [4] Glenn Jocher, Ayush Chaurasia ja Jing Qiu. *Ultralytics YOLOv8*. Versioon 8.0.0. [Accessed: 02-05-2025]. 2023. URL: <https://github.com/ultralytics/ultralytics>.
- [5] COCO. *Detection Evaluation*. [Accessed: 07-05-2025]. URL: <https://cocodataset.org/#detection-eval>.
- [6] J. A. Rodríguez-Rodríguez *et al.* „The Impact of Noise and Brightness on Object Detection Methods“. *Sensors* 24.3 (2024). [Accessed: 02-05-2025], lk. 821. URL: <https://www.mdpi.com/1424-8220/24/3/821>.
- [7] R. Girshick *et al.* *Rich feature hierarchies for accurate object detection and semantic segmentation*. [Accessed: 29-04-2025]. URL: <https://arxiv.org/abs/1311.2524>.
- [8] J. Redmon *et al.* *You Only Look Once: Unified, Real-Time Object Detection*. [Accessed: 01-05-2025]. URL: <https://arxiv.org/abs/1506.02640>.
- [9] M. Tan, R. Pang ja Q. V. Le. *EfficientDet: Scalable and Efficient Object Detection*. [Accessed: 06-05-2025]. URL: <https://arxiv.org/abs/1911.09070>.
- [10] N. Carion *et al.* *End-to-End Object Detection with Transformers*. [Accessed: 06-05-2025]. URL: <https://arxiv.org/abs/2005.12872>.
- [11] R. Girshick. *Fast R-CNN*. [Accessed: 01-05-2025]. URL: <https://arxiv.org/abs/1504.08083>.

- [12] S. Ren *et al.* *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. [Accessed: 01-05-2025]. URL: <https://arxiv.org/abs/1506.01497>.
- [13] F. Joiya. „OBJECT DETECTION: YOLO VS FASTER R-CNN“. *International Research Journal of Modernization in Engineering Technology and Science* 4.9 (2022). [Accessed: 06-05-2025]. URL: <https://www.doi.org/10.56726/IRJMETs30226>.
- [14] W. Liu *et al.* *SSD: Single Shot MultiBox Detector*. [Accessed: 06-05-2025]. URL: <https://arxiv.org/abs/1512.02325>.
- [15] M. Zhang *et al.* „Efficient Small-Object Detection in Underwater Images Using the Enhanced YOLOv8 Network“. *Applied Sciences* 14.3 (2024). [Accessed: 06-05-2025], lk. 1095. URL: https://www.researchgate.net/publication/377796046_Efficient_Small-Object_Detection_in_Underwater_Images_Using_the_Enhanced_YOLOv8_Network.
- [16] Ultralytics. *YOLOv5 vs. EfficientDet: A Detailed Comparison for Object Detection*. [Accessed: 06-05-2025]. URL: <https://docs.ultralytics.com/compare/yolov5-vs-efficientdet/>.
- [17] Y. Zhao *et al.* *DETRs Beat YOLOs on Real-time Object Detection*. [Accessed: 06-05-2025]. URL: <https://arxiv.org/abs/2304.08069>.
- [18] Papers with Code. *Object detection on COCO test-dev leaderboard*. [Accessed: 07-05-2025]. URL: <https://paperswithcode.com/sota/object-detection-on-coco>.
- [19] Z. Zong, G. Song ja Y. Liu. „DETRs with Collaborative Hybrid Assignments Training“. Teoses: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. [Accessed: 07-05-2025]. 2023. URL: https://openaccess.thecvf.com/content/ICCV2023/papers/Zong_DETRs_with_Collaborative_Hybrid_Assignments_Training_ICCV_2023_paper.pdf.
- [20] Ultralytics. *Explore Ultralytics YOLOv8*. [Accessed: 07-05-2025]. URL: <https://docs.ultralytics.com/models/yolov8/>.
- [21] G. A. Pereira. *Fall Detection for Industrial Setups using YOLOv8 variants*. [Accessed: 07-05-2025]. URL: <https://arxiv.org/html/2408.04605>.
- [22] A. Bochkovskiy, C. Wang ja H. M. Liao. *YOLOv4: Optimal Speed and Accuracy of Object Detection*. [Accessed: 01-05-2025]. URL: <https://arxiv.org/abs/2004.10934>.

- [23] S. Wu *et al.* „Enhanced SSD framework for detecting defects in cigarette appearance using variational Bayesian inference under limited sample conditions“. *Mathematical Biosciences and Engineering* 21.2 (2024). [Accessed: 07-05-2025]. URL: <https://doi.org/10.3934/mbe.2024145>.
- [24] J. Redmon ja A. Farhadi. *YOLO9000: Better, Faster, Stronger*. [Accessed: 01-05-2025]. URL: <https://arxiv.org/abs/1612.08242>.
- [25] J. Redmon ja A. Farhadi. *YOLOv3: An Incremental Improvement*. [Accessed: 01-05-2025]. URL: <https://arxiv.org/abs/1804.02767>.
- [26] Glenn Jocher. *Ultralytics YOLOv5*. Versioon 7.0. [Accessed: 01-05-2025]. 2020. DOI: 10.5281/zenodo.3908559. URL: <https://github.com/ultralytics/yolov5>.
- [27] C. Wang ang A. Bochkovskiy ja H. M. Liao. *YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*. [Accessed: 01-05-2025]. URL: <https://arxiv.org/abs/2207.02696>.
- [28] D. K. Alqahtani, A. Cheema ja A. N. Toosi. *Benchmarking Deep Learning Models for Object Detection on Edge Computing Devices*. [Accessed: 02-05-2025]. URL: <https://arxiv.org/html/2409.16808v1>.
- [29] G. Jocher. *Ultralytics*. [Accessed: 02-05-2025]. URL: <https://github.com/ultralytics/ultralytics>.
- [30] J. Soom *et al.* *Real-time, non-invasive fish detection and size estimation utilizing binocular camera system in a Portuguese river affected by hydropneaking*. [Accessed: 09-05-2025]. URL: <https://doi.org/10.2139/ssrn.5074643>.
- [31] FFmpeg Developers. *FFmpeg*. [Accessed: 09-05-2025]. URL: <https://www.ffmpeg.org/>.
- [32] gereleth. *jupyter-bbox-widget*. [Accessed: 09-05-2025]. URL: <https://github.com/gereleth/jupyter-bbox-widget>.
- [33] P. Kairouz *et al.* „Advances and Open Problems in Federated Learning“. *Foundations and Trends in Machine Learning* 14.1–2 (2021). [Accessed: 11-05-2025]. URL: <https://doi.org/10.1561/22000000083>.
- [34] NVIDIA Corporation. *Jetson Orin Nano Developer kit*. [Accessed: 12-05-2025]. URL: <https://www.nvidia.com/en-eu/autonomous-machines/embedded-systems/jetson-orin/nano-super-developer-kit/>.
- [35] Tallinn University of Technology. *TalTech High Performance Computing Center*. [Accessed: 11-05-2025]. URL: <https://hpc.taltech.ee/>.

- [36] Takuya Akiba *et al.* „Optuna: A Next-generation Hyperparameter Optimization Framework“. Teoses: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. [Accessed: 11-05-2025]. 2019, lk. 2623–2613. URL: <https://doi.org/10.1145/3292500.3330701>.
- [37] S. Salman ja X. Liu. *Overfitting Mechanism and Avoidance in Deep Neural Networks*. [Accessed: 19-05-2025]. URL: <https://arxiv.org/abs/1901.06566>.
- [38] A. Sharma. *Mean Average Precision (mAP) Using the COCO Evaluator*. [Accessed: 11-05-2025]. URL: <https://pyimagesearch.com/2022/05/02/mean-average-precision-map-using-the-coco-evaluator/>.

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

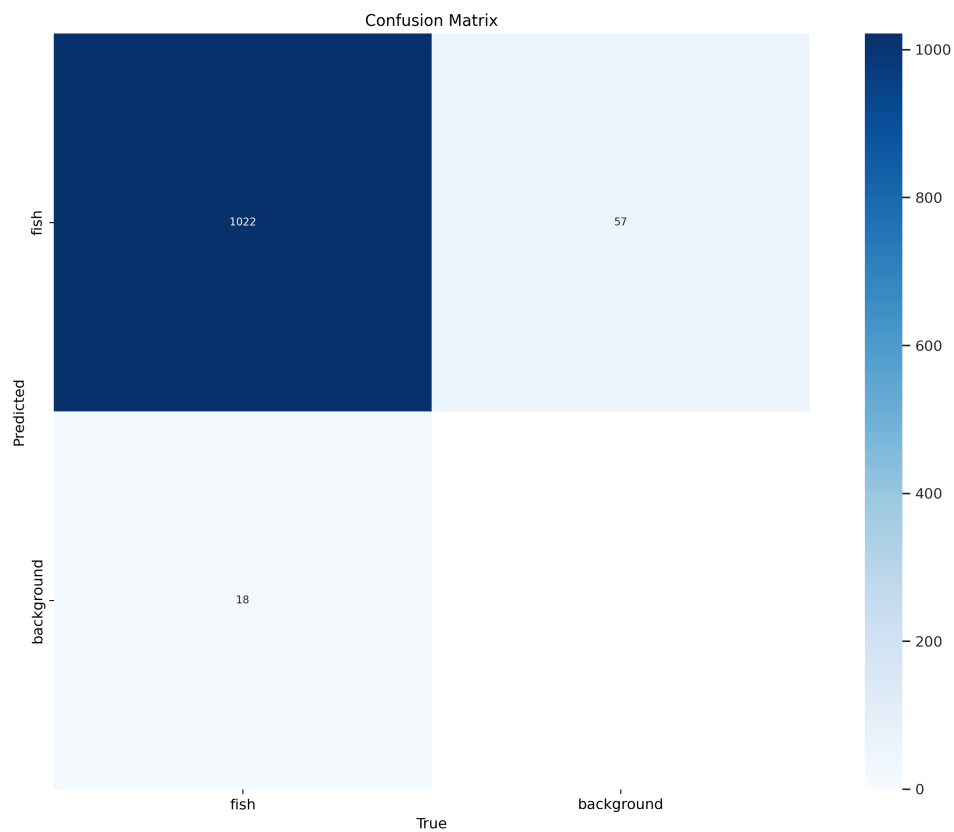
Mina, Robin Lõhmus

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose “Kallade tuvastamine YOLOv8 mudeliga veealustelt kaameratelt” mille juhendaja on Jeffrey Andrew Tuhtan
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

10.06.2025

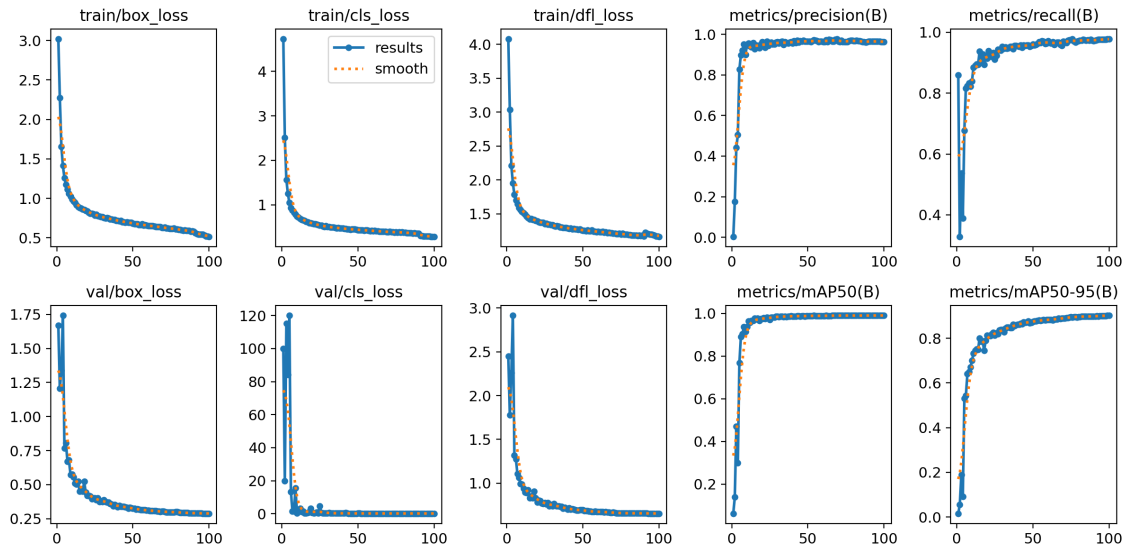
¹Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtjaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 - Šveitsi andmestikul treenitud mudeli tulemused



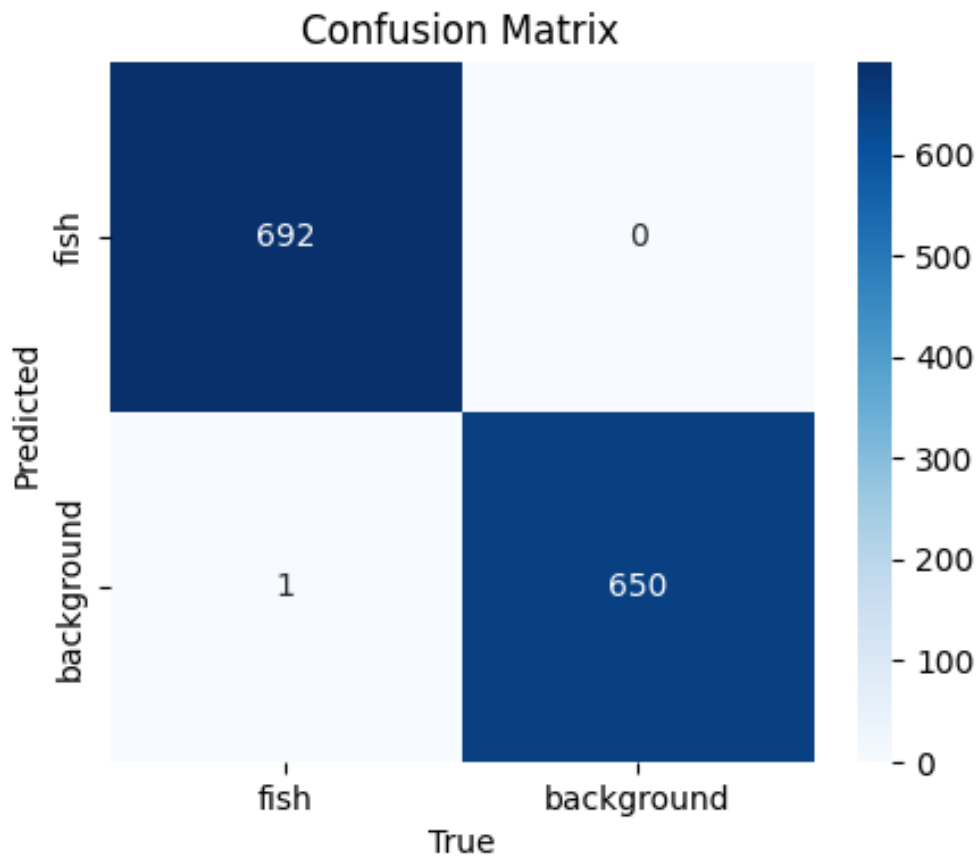
Joonis 12. Šveitsi andmestikul treenitud YOLOv8l mudeli objektide põhine segadusmaatriks Šveitsi andmestiku valideerimiskogumil.

Allikas: Autor, genereeritud Ultralytics YOLOv8 raamistiku abil



Joonis 13. Šveitsi andmestikul treenitud YOLOv8l mudeli treening- ja valideerimistulemuste ülevaade.

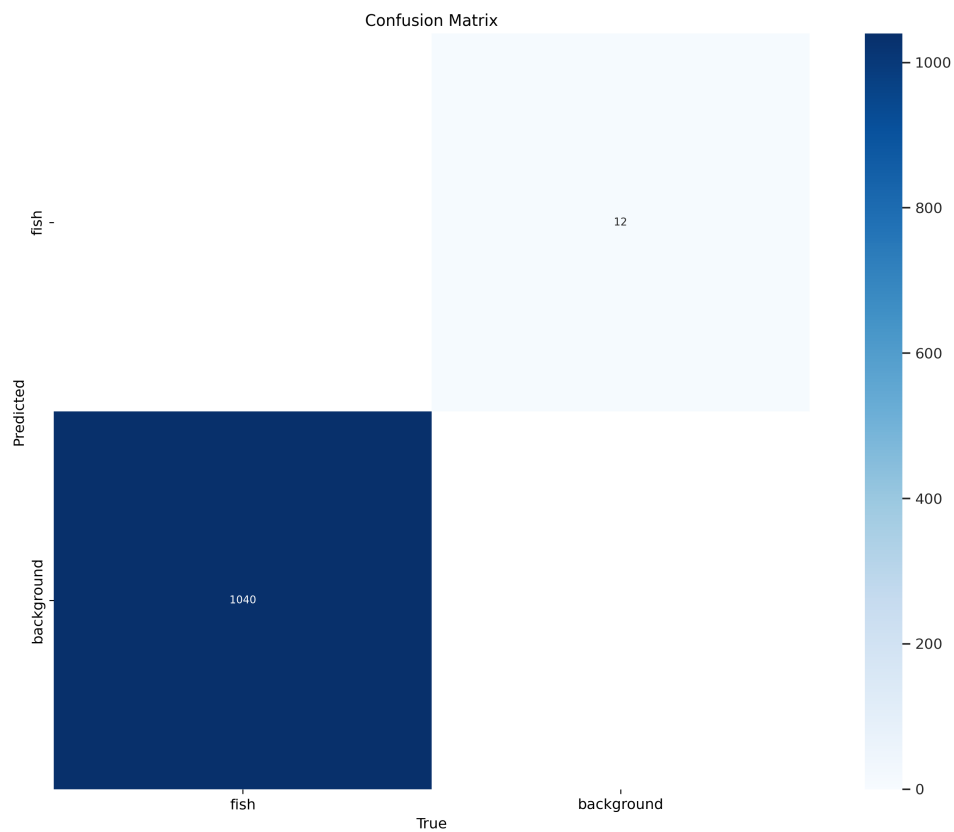
Allikas: Autor, genereeritud Ultralytics YOLOv8 raamistiku abil



Joonis 14. Šveitsi andmestikul treenitud YOLOv8l mudeli piltide põhine segadusmaatriks Šveitsi andmestiku valideerimiskogumil.

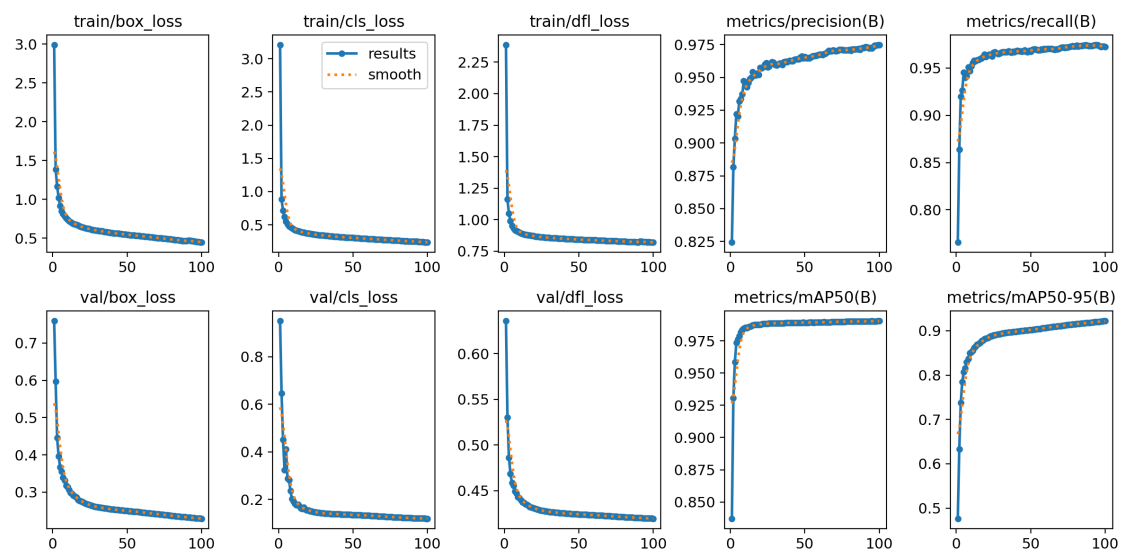
Allikas: Autor, juhtumite kokkulugemisel Šveitsi andmestiku põhjal

Lisa 3 – Portugali andmestikul treenitud mudeli tulemused



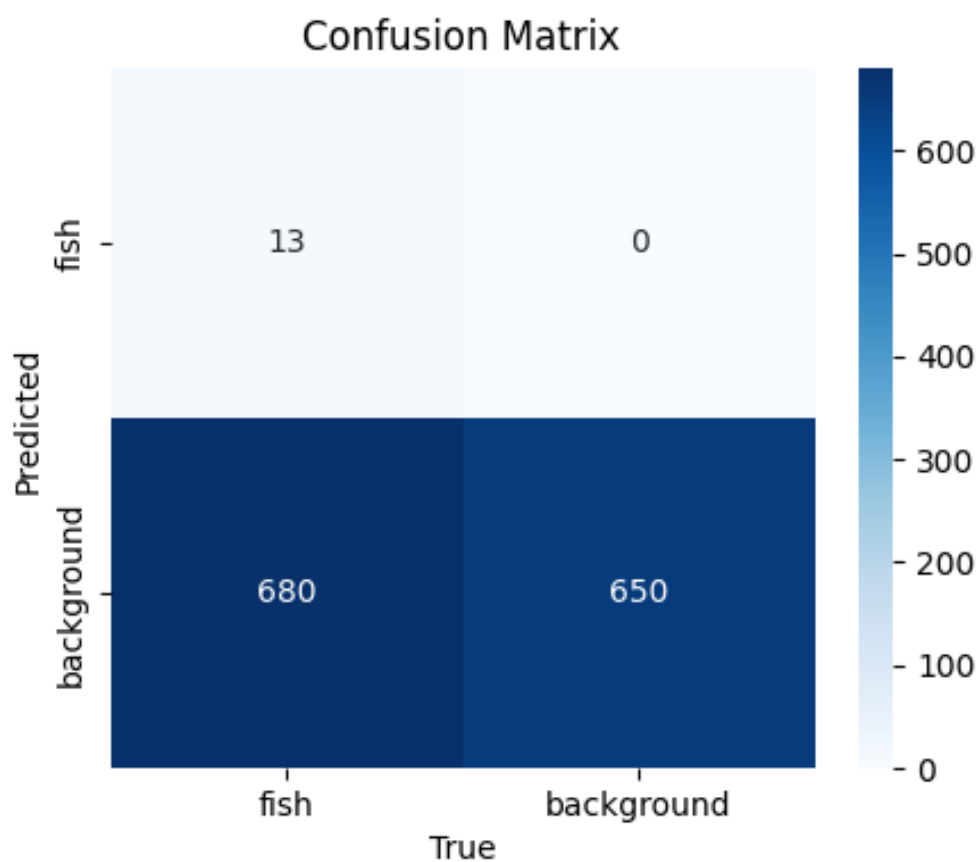
Joonis 15. Portugali andmestikul treenitud YOLOv8l mudeli objektide põhine segadusmaatriks Šveitsi andmestiku valideerimiskogumil.

Allikas: Autor, genereeritud Ultralytics YOLOv8 raamistiku abil



Joonis 16. Portugali andmestikul treenitud YOLOv8l mudeli treening- ja valideerimistulemuste ülevaade.

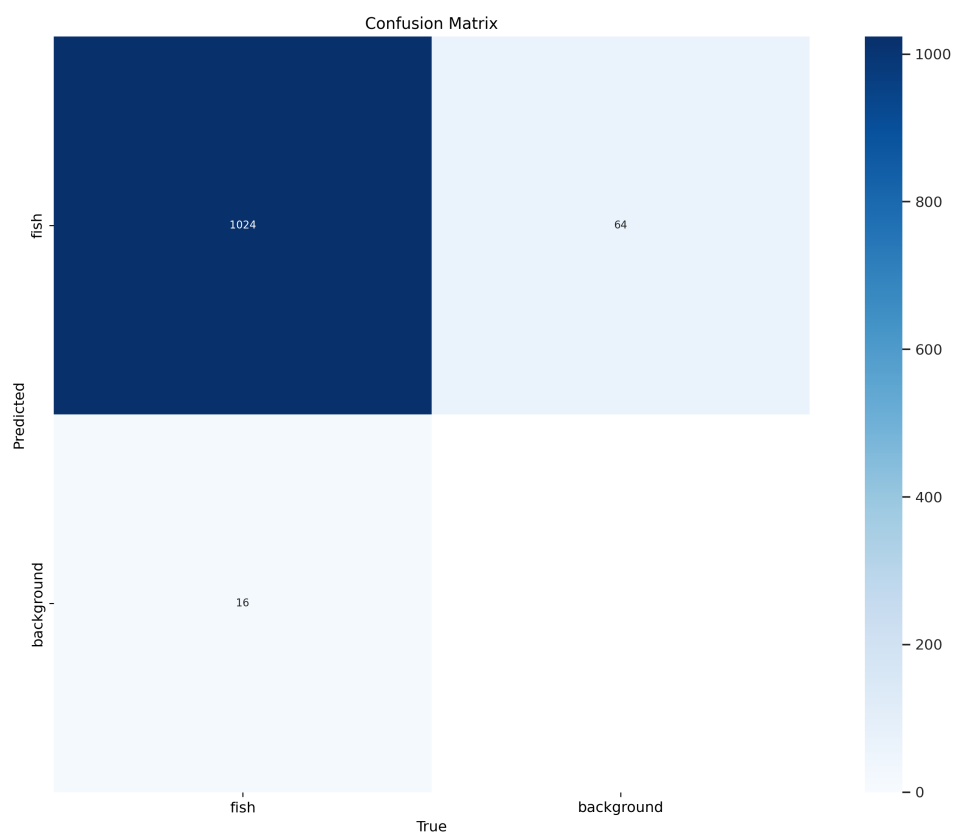
Allikas: Autor, genereeritud Ultralytics YOLOv8 raamistiku abil



Joonis 17. Portugali andmestikul treenitud YOLOv8l mudeli piltide põhine segadusmaatriks Šveitsi andmestiku valideerimiskogumil.

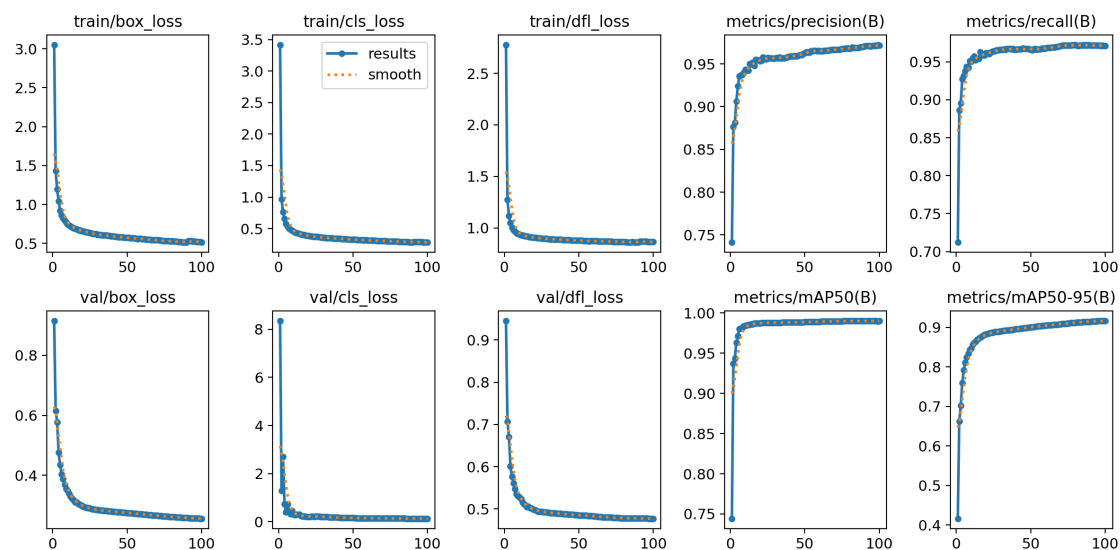
Allikas: Autor, juhtumite kokkulugemisel Šveitsi andmestiku põhjal

Lisa 4 – Kombineeritud andmestikul treenitud mudeli tulemused



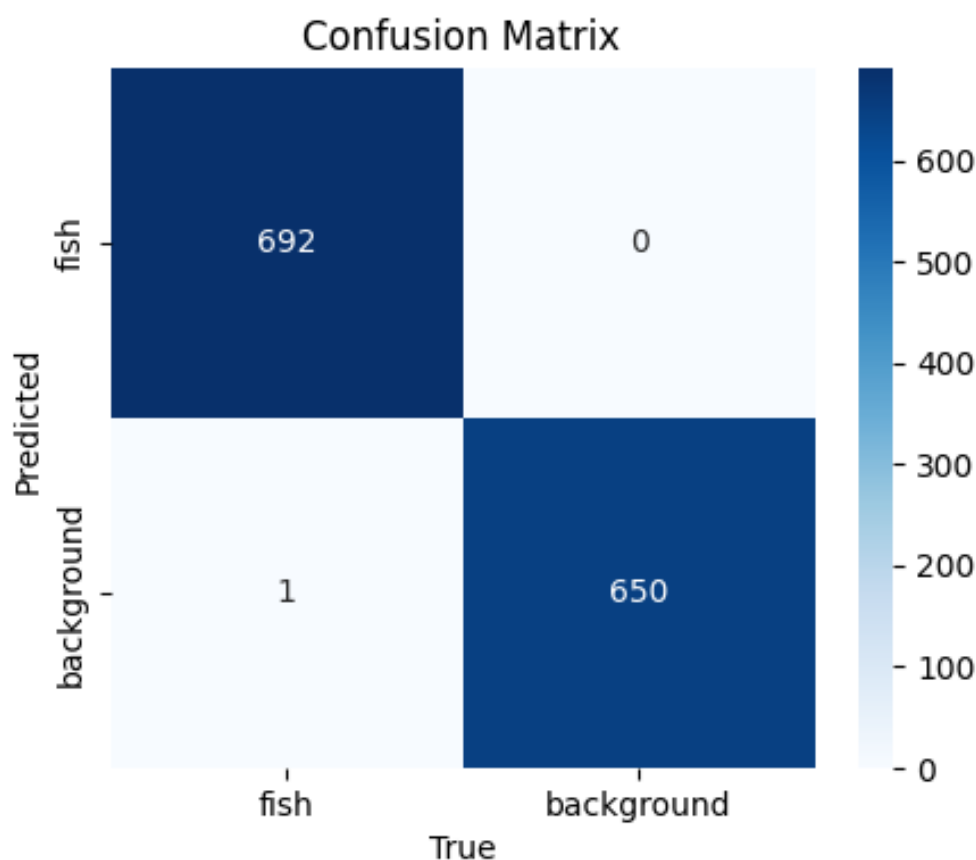
Joonis 18. Kombineeritud andmestikul treenitud YOLOv8l mudeli objektide põhine segadusmaatriks Šveitsi andmestiku valideerimiskogumil.

Allikas: Autor, genereeritud Ultralytics YOLOv8 raamistiku abil



Joonis 19. *Kombineeritud andmestikul treenitud YOLOv8l mudeli treening- ja valideerimistulemuste ülevaade.*

Allikas: Autor, genereeritud Ultralytics YOLOv8 raamistiku abil



Joonis 20. *Kombineeritud andmestikul treenitud YOLOv8l mudeli piltide põhine segadusmaatriks Šveitsi andmestiku valideerimiskogumil.*

Allikas: Autor, juhtumite kokkulugemisel Šveitsi andmestiku põhjal