TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Andrus Vaher 240684IAIB

Anna Gulova-Em 193687IAIB

Masinõppel põhineva pildianalüüsi tööriista loomine mikrofluidika tilkade uurimiseks

Bakalaureusetöö

Juhendaja: Evelin Halling

PhD

Tallinn 2026

Autorideklaratsioon

Kinnitame, et oleme 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.

Autorid: Andrus Vaher ja Anna Gulova-Em

01.06.2026

Annotatsioon

Tilkadel põhinevad mikrofluidika uuringud genereerivad suures koguses eksperimentaalseid andmeid, mille käsitsi analüüs on aeganõudev ja võib põhjustada inimlikke vigu. Olemasolevad pildianalüüsi tööriistad vajavad pidevat käsitsi parameetrite kohandamist ja ei suuda alati usaldusväärselt käsitleda ebakorrapärase kujuga objekte või visuaalset müra. Sarnase probleemiga puutus kokku ka Taltechi geenitehnoloogia ja biomeditsiini osakonna uurimisrühm, kes vajas protsesside kiirendamist. Käesoleva bakalaureusetöö eesmärk on arendada masinõppel põhinev pildianalüüsi rakendus tilga mikrofluidika eksperimentide automatiseerimiseks. Töö käigus analüüsiti varasemaid lahendusi ning sõnastati funktsionaalsed ja mittefunktsionaalsed nõuded, millele loodud rakendus peab vastama. Töö lõpuks valmis modulaarne töölauarakendus, mis integreerib masinõppemudelid ühtsesse analüüsitorustikku. Loodud rakendus võimaldab kasutajal analüüsida üksikfaile või terveid kaustu, visualiseerib tulemused ning ekspordib need Exceli ja PDF-vormingusse. Rakendus toetab analüüsiajaloo salvestamist ja sessioonide võrdlemist, mis võimaldab varasemaid tulemusi ilma uuesti töötlemiseta jälgida. Valideerimine kinnitas, et loodud lahendus vähendab oluliselt käsitsi analüüsi mahtu ning sobib praktiliseks kasutamiseks uurimistöös.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 46 leheküljel, 9 peatükki, 35 joonist, 8 tabelit.

Abstract

Development of Machine Learning-Based Image Analysis Tool for Microfluidic Droplet Research

Droplet-based microfluidics research generates large amounts of experimental data, the manual analysis of which is time-consuming and prone to human errors. Existing image analysis tools require constant manual parameter adjustments and cannot always reliably handle irregularly shaped objects or visual noise. A similar problem was encountered by the research group in Taltech's Department of Genetic Engineering and Biomedicine, which needed to accelerate their analysis workflow. The aim of this bachelor's thesis is to develop a machine learning-based image analysis application for automating droplet microfluidics experiments. This thesis analyses existing solutions and defines the functional and non-functional requirements that the developed application must meet. The outcome of this thesis is a modular desktop application that integrates machine learning models into a unified analysis pipeline. It allows the user to analyze individual files or entire folders, visualizes the results and exports them to Excel and PDF formats, as well as supports saving analysis history and comparing sessions, enabling previous results to be tracked without re-running the detection pipeline. Validation confirmed that the created solution significantly reduces manual workload and is suitable for practical use in research.

The thesis is in Estonian and contains 46 pages of text, 9 chapters, 35 figures, 8 tables.

Lühendite ja mõistete sõnastik

Blob detektor	pilditöötluse meetod, mida kasutatakse ühtlase intensiivsusega alade tuvastamiseks, võimaldades leida ebakorrapärase kujuga objekte
Cellpose	süvaõppel põhinev pildisegmenteerimise mudel, mis on treenitud bioloogiliste objektide tuvastamiseks mikroskoobipiltidel
CNN	konvolutsiooniline närvivõrk (*Convolutional Neural Network*)
DIC	diferentsiaalse interferentskontrasti mikroskoopia meetod (*Differential Interference Contrast*), mis võimaldab visualiseerida läbipaistvaid objekte kontrastsete struktuuridena
Excel	tabelarvutusprogramm ja failivorming struktureeritud andmete salvestamiseks ja töötlemiseks
F1-skoor	masinõppe mudeli hindamismõõdik, mis ühendab täpsuse (*precision*) ja saagise (*recall*) üheks näitajaks
Flet	Pythoni raamistik, mida kasutatakse platvormiüleste kasutajaliideste loomiseks
GPU	graafikaprotsessor (*Graphics Processing Unit*), mida kasutatakse suure arvutusmahuga ülesannete kiirendamiseks
Houghi teisendus	pilditöötluse meetod geomeetriliste kujundite, näiteks ringide, tuvastamiseks piltidel
JPEG	pakitud rastergraafika failivorming (*Joint Photographic Experts Group*), mida kasutatakse piltide salvestamiseks
JSON	kergekaaluline andmevahetusformaat (*JavaScript Object Notation*) struktureeritud andmete esitamiseks
mAP50	objekti tuvastamise mudelite hindamismõõdik, mis näitab keskmist täpsust tingimusel, et ennustatud ja tegelik objekt kattuvad vähemalt 50% ulatuses
MATLAB	tehniline arvutustarkvara ja programmeerimiskeskkond teaduslikeks arvutusteks ja pilditöötluseks
ONNX	avatud vorming masinõppemudelite esitamiseks ja vahetamiseks erinevate platvormide vahel (*Open Neural Network Exchange*)
OpenCV	avatud lähtekoodiga teek pilditöötluse ja arvutinägemise ülesannete lahendamiseks (*Open Source Computer Vision Library*)

Precision	mõõdik, mis näitab õigete positiivsete ennustuste osakaalu kõigi positiivsete ennustuste hulgas
PyInstaller	tööriist Pythoni rakenduste pakendamiseks iseseisvateks käivitatavateks failideks
RAM	muutmälu (*Random Access Memory*), mida kasutatakse andmete ajutiseks salvestamiseks
README	projekti kirjeldav juhendfail tarkvara kasutamise, paigaldamise ja arendamise kohta
Recall	mõõdik, mis näitab, kui suur osa tegelikest positiivsetest juhtudest mudeli poolt õigesti tuvastatakse
SQLite	kergekaaluline relatsiooniline andmebaasisüsteem, mis ei vaja eraldi serverit
TIFF	kõrgekvaliteediliste rasterpiltide failivorming (*Tagged Image File Format*)
U-Net	konvolutsiooniline närvivõrgu arhitektuur, mida kasutatakse peamiselt pildisegmenteerimiseks
uv	Pythoni projektide paketihaldur ja keskkonnahalduse tööriist
WTGFP	fluorestseeruva valguga märgistatud kanal, mida kasutatakse bakterite kasvu hindamiseks (*Green Fluorescent Protein*)
YOLOv8n	süvaõppel põhinev objekti tuvastusmudel, mis on optimeeritud kiiruse ja efektiivsuse jaoks

Sisukord

Jooniste loetelu

Tabelite loetelu

1 Sissejuhatus

Kaasaegses bio- ja geenitehnoloogias on üha olulisemaks muutunud meetodid, mis võimaldavad suuremahulist ja kiiret andmetöötlust rakutasandil. Üheks selliseks perspektiivseks suunaks on tilga mikrofluidika, mis võimaldab kontrollitud keskkonnas analüüsida samaaegselt sadu tuhandeid mikroobjekte. Kuigi antud metoodika pakub suurt läbilaskevõimet, on peamiseks kitsaskohaks kujunenud tohutute andmemahtude efektiivne ja täpne analüüs. Traditsioonilised pildianalüüsi meetodid jäävad sageli hätta bioloogiliste proovide varieeruvuse ja visuaalse müra töötlemisel, mistõttu on vajadus masinõppel põhinevate automatiseeritud lahenduste järele kriitilisem kui kunagi varem.

Lõputöö eesmärk on välja töötada masinõppel põhinev rakendus, mis automatiseerib tilkade tuvastamise ja analüüsi protsessi, pakkudes uurimisrühmale palju täpsemat ja kasutajasõbralikumat tööriista.

Rakenduse esialgne versioon on valmistatud tellimusprojekti raames kolmeliikmelise meeskonnatööna, mille liikmeteks olid ka antud bakalaureusetöö autorid, ning on mõeldud Taltechi geenitehnoloogia ja biomeditsiini osakonna uurimisrühma igapäevatöö toetamiseks. Käesoleva jätkuetapi eesmärk on rakendust täiendada, et see vastaks paremini uurimisrühma ootustele.

Eesmärgi saavutamiseks püstitati järgmised ülesanded:

- Analüüsida olemasolevaid pildianalüüsi lahendusi ja nende piiranguid;
- Täiustada mikroplasti tuvastamist;
- Lisada analüüsiajaloo ja sessioonide võrdlemise funktsionaalsus;
- Valideerida masinõppe mudelid reaalsete katseandmete peal;
- Viia läbi rakenduse testimine uurimisrühma liikmete osalusel, koguda nende tagasiside ja teha saadud ettepanekute põhjal rakendusse vajalikud täiendused.

Lisaks eelnevalt nimetatud ülesannetele esitatakse allpool põhjalik ülevaade rakenduse

tehnilisest teostusest ja funktsionaalsusest, mis võimaldab mõista süsteemi toimimist nii eelnevalt valminud versiooni kui ka lõputöös tehtud täienduste põhjal.

2 Analüüs

Selles peatükis kirjeldatakse täpsemalt lahendatavat probleemi ning antakse ülevaade olemasolevatest pildianalüüsi tööriistadest. Peatüki lõpus defineeritakse funktsionaalsed ja mittefunktsionaalsed nõuded loodavale rakendusele.

2.1 Probleemi kirjeldus

Praegu kasutab uurimisrühm mitmesuguseid tarkvaralahendusi. Küll aga eelkõige kasutatakse *CellProfiler*-it, et analüüsida erinevate mikroskoopiameetodite (helevälja-, fluorestsents- ja konfokaalmikroskoopia) abil saadud pilte. Fluorestsentsmikroskoopia pakub suurt kontrasti, kuna huvipakkuvad objektid on heledad ja taust tume, mis lihtsustab analüüsi. Samas on fluorestsentsil mitmeid puudusi: seda on keeruline integreerida bioloogilistesse katsetesse ning värvainete ja molekulaarsondide valik nõuab hoolikat kaalumist, kuna erinevate lainepikkuste signaalid võivad kattuda. Selle tõttu on uurimisrühm oma töös fookusesse võtnud DIC-mikroskoopia (diferentsiaalinterferentskontrast), sest seda on lihtsam rakendada kui fluorestsentsi. Kuid sellel kanalil on mitmeid puudusi, mis muudavad analüüsi keerukaks:

- objekt ja taust võib olla sama heledusega;
- objekt on must ning taust on valge;
- pildid sisaldavad sageli visuaalset müra ja kõrvalisi objekte;
- tilgad võivad olla ebakorrapärase kujuga või mõnega kattuda.

CellProfiler on aga loodud peamiselt fluorestsentspiltide analüüsimiseks, mitte DIC-piltide jaoks. Kuigi tarkvara suudab teatud määral töödelda ka erineva kvaliteediga DIC-pilte, nõuab protsess keerukat töövoogu, mis on aja- ja ressursimahukas ning võib põhjustada süsteemi tõrkeid. See on peamine põhjus, miks uurimisrühm algatas uue lahenduse väljatöötamise DIC-mikroskoopia piltide automatiseeritud analüüsiks.

2.2 Valmislahendused

Selles peatükis analüüsitakse olemasolevaid pildianalüüsi vahendeid, mida teadlased kasutavad katsete käigus saadud andmete töötlemiseks [1].

2.2.1 CellProfiler

CellProfiler on avatud lähtekoodiga tarkvara, mis on pildianalüüsis laialdaselt levinud. See võimaldab bioloogidel teostada kvantitatiivset ja usaldusväärset analüüsi ilma programmeerimisoskusteta. Tarkvara põhineb moodulitena korraldatud algoritmidel, mida saab järjestikku asetada tõhusate töötluskonveierite loomiseks.

Kuigi selline lähenemisviis säästab võrreldes käsitööga oluliselt aega, nõuab objektide tuvastamine parameetrite käsitsi seadistamist. See protsess võib olla pildianalüüsis väheste kogemustega teadlastele keeruline ja eksitav. Ka kogenud kasutajate jaoks on parameetrite, näiteks läviväärtuste korduv kohandamine sageli tüütu ja ebaefektiivne. Lisaks ei ole CellProfiler optimeeritud erineva kvaliteediga piltide töötlemiseks ühes töövoos. Seetõttu võib analüüs jätta osa objekte tuvastamata või tekitada andmetes liigset müra [2].

2.2.2 Cellpose

Cellpose on süvaõppel põhinev algoritm, mis on loodud spetsiaalselt objektide segmenteerimiseks. Tänu treenimisele suurel hulgal biopiltidel suudab see edukalt tuvastada ka kattuvaid objekte, mida teised tööriistad käsitleksid tõenäoliselt ühe tervikuna.

Selle peamiseks eeliseks on vajaduse puudumine heleduslävede käsitsi seadistamiseks ning võime töödelda erineva kvaliteediga pilte. Samas on tegemist ressursimahuka lahendusega, mistõttu töötab see ilma võimsa graafikakaardita (nt *NVIDIA CUDA* toega) märkimisväärselt aeglasemalt [3].

2.2.3 Ilastik

Ilastik põhineb juhendatud masinõppe kontseptsioonil, mis võimaldab kasutajatel märkida pildil käsitsi huvipakkuvad alad ja objektid (nt plekid või taust) nende klassifitseerimiseks. Tarkvara võimekust piirab aga arvuti operatiivmälu (RAM) maht. Seetõttu võib suure hulga tunnustega mudeli treenimine või mahukate piltide töötlemine süsteemi jõudlust

märgatavalt vähendada [4].

2.2.4 Fiji/ImageJ

ImageJ pakub pilditöötluse põhifunktsioone, mida saab keerukamate ülesannete lahendamiseks laiendada pistikprogrammidega, mis on koondatud *Fiji* tarkvarapaketti. Vajadus skriptide loomise järele võib aga osutuda takistuseks teadlastele, kes ei valda programmeerimiskeeli. Lisaks võib mitme pistikprogrammi paralleelne haldamine muuta tööprotsessi ebamugavaks [5].

2.2.5 MATLAB

Erinevalt ülaltoodud lahendustest antud tarkvaral pole avatud lähtekoodi ning selle kasutamiseks on vaja litsentsi. Segmenteerimiseks on võimalik kasutada sisseehitatud funktsiooni *imfindcircles(A, radius)*, kus A on lähtepilt ning radius on ringi raadius või tuvastatavate ringikujuliste objektide ligikaudne raadius, mis on määratud positiivse arvuna. Siiski on oluline arvestada asjaolu, et *Houghi* teisendus sobib ainult matemaatiliselt täpsete ringide leidmiseks. Seega on deformeerunud, kokku kleepunud või müraste objektide tuvastamine peaaegu võimatu [6].

2.2.6 Järeldused

Eelneva analüüsi põhjal selgub, et olemasolevad tööriistad ei paku DIC-piltide jaoks automatiseeritud ja usaldusväärset lahendust. Enamik neist nõuab, kas ulatuslikku käsitsi seadistamist, keerukat töövoogu või ei suuda ebaregulaarsete objektidega toime tulla. Seetõttu on vaja välja töötada kohandatud lahendus, mis vastaks uurimisrühma vajadustele. Järgnevalt on defineeritud funktsionaalsed ja mittefunktsionaalsed nõuded, millele arendatav lahendus peab vastama.

2.3 Funktsionaalsed nõuded

Funktsionaalsed nõuded on jaotatud kolme põhietappi: andmete sisestamine, nende analüüs ja tulemuste väljastamine. Järgnevad nõuded on sõnastatud tellija poolt.

Andmete sisestamine:

- Süsteem peab võimaldama .tif/.tiff formaadis failide ja kaustade lisamist analüüsi järjekorda;
- Kasutajal peab olema võimalus lisatud faile ja kaustasid nimekirjast eemaldada.

Analüüs:

- Rakendus peab tuvastama tilgad, klassifitseerima bakterite kasvu tüübi (homogeenne, agregeeritud või bakterite puudumine) ning leidma mikroplasti osakesed;
- Rakendus peaks kuvama analüüsi edenemist.

Tulemuste väljastamine:

- Rakenduses peab kuvama ülevaatlikud tulemused nii iga pildi kui ka üldiselt kõikide analüüsitud piltide kohta (tilkade, mikroplastide arv, bakterite kasv jne);
- Analüüsi tulemusena peab kasutajal olema võimalus genereerida iga pildi kohta 4 visuaalset väljundit:
    1. Tuvastatud tilgad
    2. Bakterite kasv (homogeenne ja agregeeritud) tilkade sees
    3. Mikroplastide arv tilkade sees
    4. Ülevaatlik pilt: kõik tilgad värviga tähistatud kasvu tüübi järgi (homogeenne, agregeeritud, baktereid mitte sisaldav);
- Süsteem peab koostama Exceli formaadis koondtabeli, mis sisaldab järgmisi andmevälju: kausta nimi, pildi nimi, tilga ID, mikroplastide arv, kasvu tüüp ja tilga diameeter.

Lisaks tellijate nõuetele on rakendusse lisatud mitu lisavõimalust, et parandada kasutajakogemust ja võimaldada analüüsitulemuste tõhusamat haldamist:

- Süsteem saab analüüsitulemusi salvestada andmebaasi ning võimaldab varasemaid sessioone vaadata ja võrrelda;
- Süsteem võimaldab analüüsitulemuste põhjal genereerida PDF-raporti.

2.4 Mittefunktsionaalsed nõuded

Mittefunktsionaalsete nõuete määratlemisel on lähtutud tarkvaraarenduse valdkonna parimatest praktikatest [7]. Käesoleva töö raames sõnastati järgmised mittefunktsionaalsed

nõuded:

- **Jõudlus.** Rakendus peab töötlema kõrge resolutsiooniga TIFF-faile mõistliku aja jooksul ning olema oluliselt kiirem kui varasemad käsitsi või poolautomaatsed analüüsimeetodid.
- **Platvormisõltumatus.** Serveripõhise lahenduse kõrgete ülalpidamiskulude tõttu tuleb süsteem realiseerida töölauarakendusena, mis töötab lokaalselt kasutaja arvutis. Rakendus peab olema ühilduv Windowsi, macOS-i ja Linuxi operatsioonisüsteemidega, kuna uurimisrühma liikmed kasutavad erinevaid seadmeid.
- **Paigaldatavus ja kättesaadavus.** Lõppkasutajale tuleb pakkuda vastava operatsioonisüsteemiga sobiv käivitatav fail. Rakendus peab olema lihtsalt paigaldatav ning arenduskeskkond kiiresti seadistatav.
- **Reprodutseeritavus.** Projektis kasutatavate teekide versioonid peavad olema üheselt määratletud ja lukustusfailiga fikseeritud, et tagada arenduskeskkonna reprodutseeritavus.
- **Arendusmugavus.** Projekt peab sisaldama selget juhendit (*README*), mis võimaldab uuel arendajal või teadlasel koodibaasist kiiresti aru saada ning seda vajaduse korral edasi arendada.
- **Kasutatavus.** Rakenduse sihtrühmaks on laboritöötajad, kellel ei pruugi olla põhjalikku IT-alast ettevalmistust. Seetõttu peab kasutajaliides olema intuitiivne ning tulemuste visualiseeringud vahetult loetavad ja tõlgendatavad ilma täiendava andmetöötluseta.
- **Usaldusväärsus ja täpsus.** Mikrofluidika andmete analüüsis peab valepositiivsete tulemuste hulk olema minimaalne. Lisaks peab rakendus automaatselt välistama tilgad, mis puudutavad pildi serva või ei ole ringikujulised, et tagada analüüsitavate andmete kvaliteet.
- **Turvalisus.** Pildianalüüs peab toimuma kasutaja lokaalses arvutis ning rakendus ei tohi edastada uurimisandmeid välisserveritesse. Arendusfaasis kasutatud Google Colab ja CVAT ei kuulu lõpliku rakenduse koosseisu.

3 Tehnoloogiline ülevaade

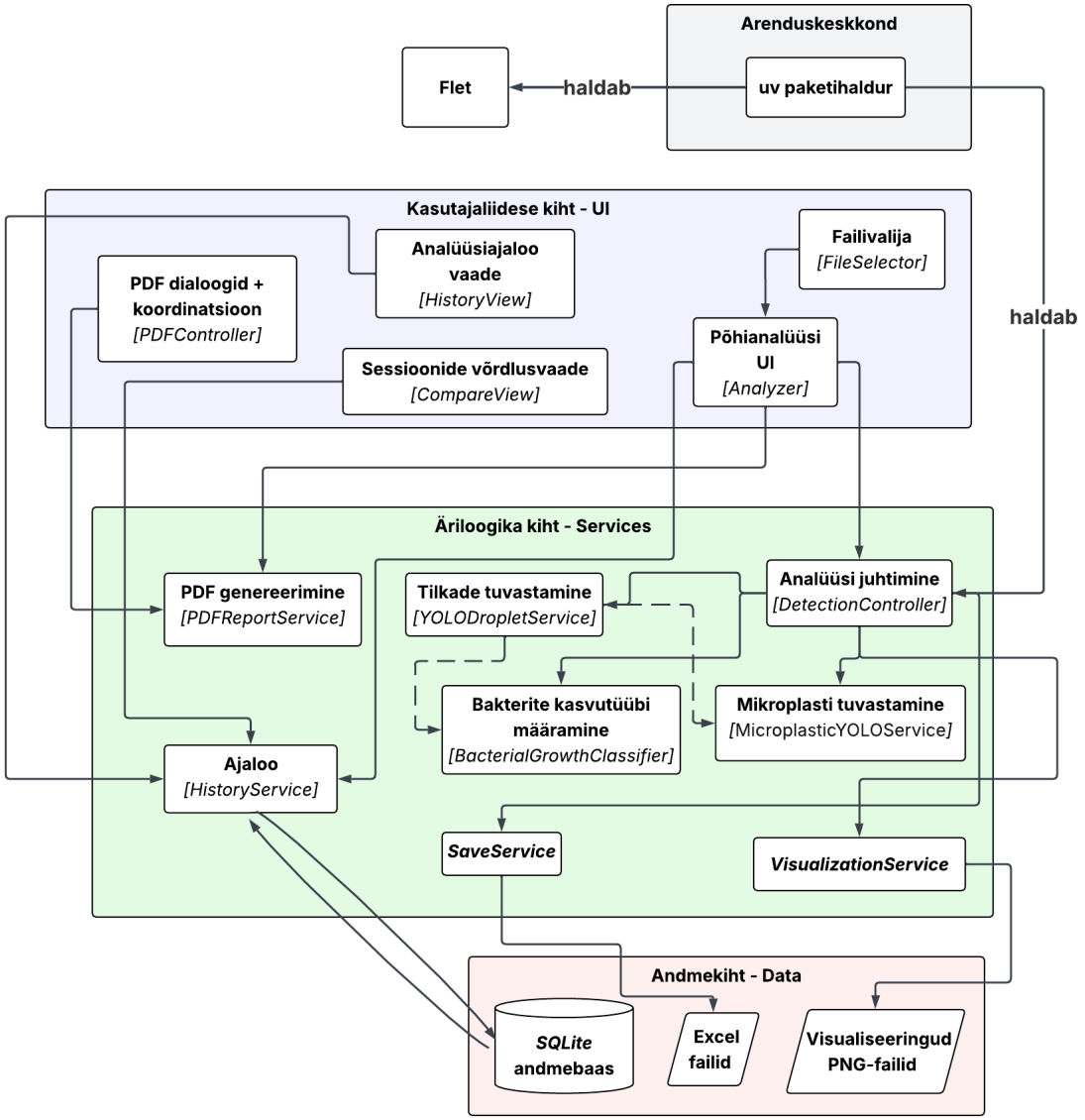
Selles peatükis kirjeldatakse rakenduse arhitektuuri, andmete töötlemise voogu, kasutatud tehnoloogiaid ja nende põhjendust.

3.1 Rakenduse arhitektuur

Rakenduse arhitektuur põhineb teenusorienteeritud lähenemisviisil, kus iga funktsionaalne üksus on realiseeritud eraldiseisva teenusmoodulina. Selline ülesehitus tagab süsteemi paindlikkuse ja võimaldab üksikuid komponente tulevikus täiendada või asendada ilma kogu süsteemi ümber kujundamata. Kuigi kasutajaliides on tihedalt integreeritud süsteemi äriloogikaga, on see arhitektuuriliselt sellest eraldatud.

3.2 Arhitektuuri ülesehitus

Rakendus koosneb kolmest põhilisest kihist, mis on omavahel selgelt eraldatud (joonis 1).



Joonis 1. Rakenduse kolmekihiline arhitektuur ja teenuste omavahelised seosed.

Kasutajaliidese kiht on realiseeritud Flet raamistiku abil, mis võimaldab luua töölauarakendusi Pythonis. See kiht vastutab visuaalse poole, kasutaja interaktsioonide ja analüüsi käivitamise eest. Selle kihi keskseks komponendiks on *Analyzer*, mis koordineerib analüüsi käivitamist, kuvab tulemused ning kutsub vajadusel välja teisi teenuseid. Samuti see kiht sisaldab ajaloo vaadet (*HistoryView*), sessioonide võrdlemise vaadet (*CompareView*) ja PDF-iga seotud kasutajaliidese komponente (dialoogid, faili salvestamine), mida haldab *PDFController*.

Äriloogika kiht sisaldab teenuseid, mis teostavad analüüsi keskseid samme. Kogu pildianalüüsi protsessi juhib *DetectionController*, mis tagab, et iga pilti töödeldakse alati

ühtemoodi. Peamised teenused on:

- *YOLODropletService* – pildi laadimine, eeltöötlus ja tilkade tuvastamine YOLOv8n mudeli põhjal;
- *BacterialGrowthClassifier* – bakterite kasvutüüpide klassifikatsioon konvolutsioonilise närvivõrgu abil;
- *MicroplasticYOLOService* – mikroplasti osakeste tuvastamine YOLOv8n mudeli põhjal;
- *VisualizationService* – visualiseeringute genereerimine;
- *SaveService* – tulemuste salvestamine Exceli formaati;
- *HistoryService* – analüüsisessioonide salvestamine ja päringud andmebaasist;
- *PDFReportService* – PDF-raportite loomine (graafikud, raporti struktuur).

Andmekiht vastutab analüüsitulemuste püsiva talletamise ja kättesaadavuse eest. Rakendusse on integreeritud SQLite andmebaas, mis võimaldab hallata analüüsisessioonide ajalugu. See lahendus tagab, et kasutaja saab varem tehtud analüüse vaadata ja võrrelda ilma pilte uuesti töötlemata. Lõplikud analüüsitulemused (tabelid ja visualiseeringud) väljastatakse siiski failipõhiselt, tagades nende lihtsa kasutatavuse väljaspool rakendust.

3.3 Andmete liikumine süsteemis

Sisendpildi töötlemine toimub kindlas järjekorras:

1. Pildi laadimine ja eeltöötlus [*YOLODropletService*]
   TIFF-pilt laaditakse ja teisendatakse hallskaalasse. Intensiivsus normaliseeritakse ja suurus muudetakse töötlemiseks optimaalseks.
2. Tilkade tuvastamine [*YOLODropletService*]
   Tuvastatakse kõik pildil olevad potentsiaalsed tilgad. Iga tuvastamise korral luuakse piirava kasti põhjal ringikujuline kontuur. Seejärel rakendatakse filtreerimist: eemaldatakse pildi servi puudutavad, liiga väikesed või mitte piisavalt ringikujulised tilgad. Ühe pildi analüüsimine võtab umbes 15-30 sekundit.
3. Bakterite kasvutüübi määramine [*BacterialGrowthClassifier*]
   Iga kehtiva tilga kohta eraldatakse huvipakkuv piirkond (ROI) ja suunatakse see konvolutsioonilisse närvivõrku (CNN). Mudel klassifitseerib iga tilga ühte kolmest

kategooriast: homogeenne kasv, agregeeritud kasv või bakterite puudumine.

4. Mikroplasti tuvastamine [*MicroplasticYOLOService*]

   Iga tilga kohta eraldatakse huvipakkuv piirkond (ROI) ning rakendatakse YOLOv8-seg mudelit. Mudel tuvastab tilga sees olevad mikroplastiosakesed. Tuvastatud osakeste keskpunktid kontrollitakse tilga kontuuri suhtes, et vältida valepositiivseid tulemusi. Mikroplasti tuvastamise üksikasjalik kirjeldus on toodud peatükis 5.

5. Tulemuste koondamine ja salvestamine

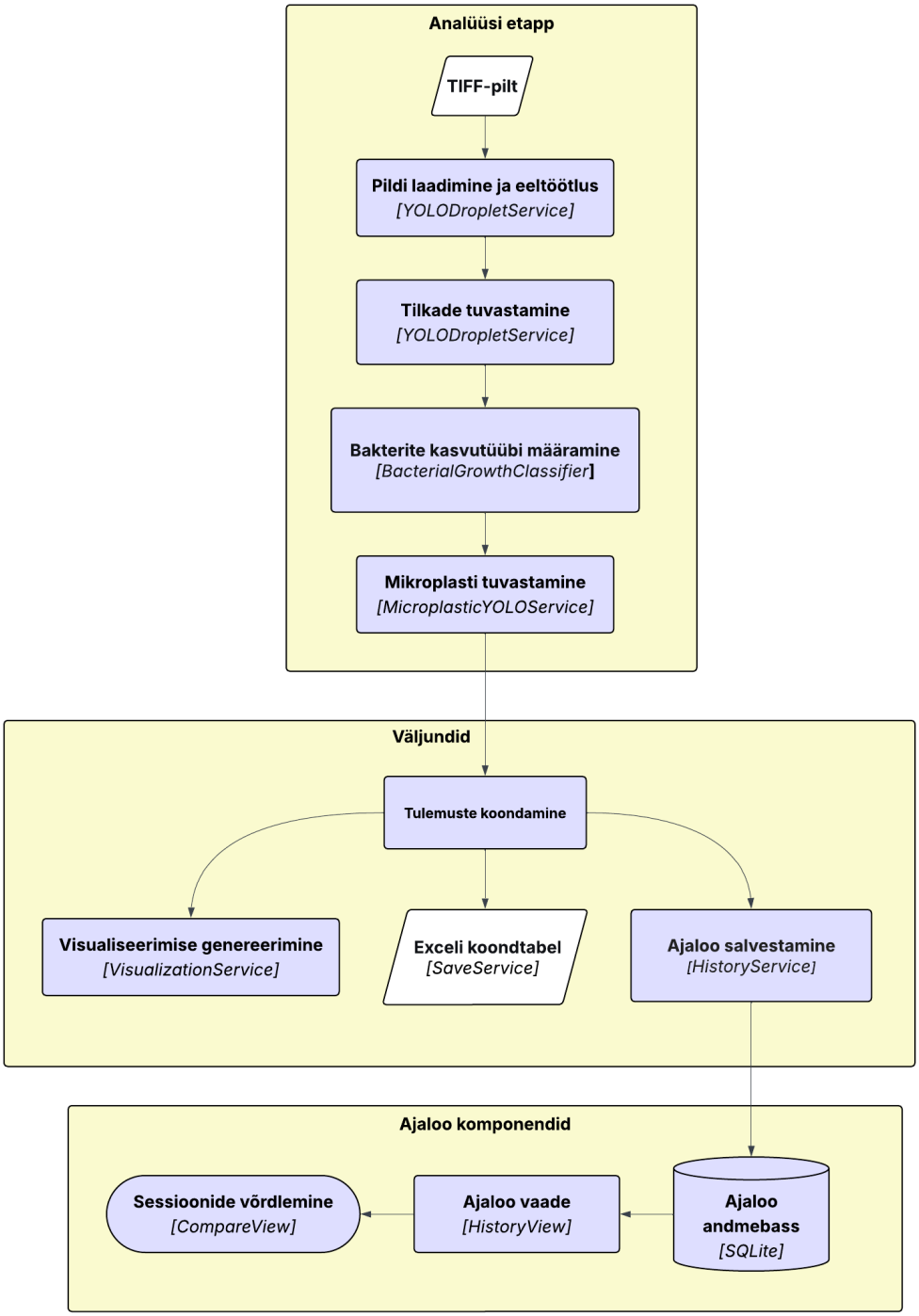
   SaveService koondab iga tilga andmed (kausta nimi ja faili nimi, tuvastatud tilkade arv, neis leitud mikroplastiosakeste arv, bakterite kasvu tüüp, tilga diameeter) Exceli tabelisse.

   HistoryService salvestab sessiooniandmed SQLite andmebaasi. Kasutaja saab neid andmeid hiljem vaadata ja võrrelda. Analüüsiajaloo ja sessioonide võrdlemise funktsionaalsusi on põhjalikumalt kirjeldatud peatükis 6.

6. Visualiseeringute genereerimine [*VisualizationService*]

   Luuakse nelja tüüpi visualiseeringuid: kõik tuvastatud tilgad, bakterite kasv, mikroplasti asukohad ning ülevaatlik pilt kõigist tilkadest värviga tähistatud kasvu tüübi järgi. Pildid salvestatakse eraldi kausta.

Kogu andmete liikumise voog on esitatud joonisel 2.



Joonis 2. Andmete liikumine läbi analüüsisüsteemi.

3.4 Sõltuvused ja kasutatud teegid

Arhitektuur põhineb järgmistel Pythoni teekidel:

- flet – töölauarakenduse kasutajaliidese loomine;
- ultralytics, torch, torchvision – masinõppemudelid (YOLOv8n, YOLOv8n-seg,

CNN);

- opencv-python-headless, scikit-image, tifffile, imageio – pilditöötlus (sh mikroplasti tuvastamise algoritm) ning TIFF-vormingu tugi;
- pandas, openpyxl – andmetöötlus ja Exceli tabelite loomine;
- matplotlib – visualiseerimine;
- reportlab – PDF-raportite genereerimine;
- pyinstaller – rakenduse pakkimine.

Valitud teegid on omavahel hästi ühilduvad ning võimaldavad tõhusalt realiseerida kõiki rakenduse funktsionaalsusi.

Sõltuvuste haldamiseks on kasutatud paketihaldurit uv. Erinevalt standardsest tööriistast pip tagab uv oluliselt kiirema arenduskeskkonna seadistamise ja teekide paigaldamise. Lisaks fikseerib lukustusfail uv.lock kõigi teekide täpsed versioonid, tagades reprodutseeritava keskkonna nii arendusfaasis kui ka lõppkasutaja seadmetes. Selline lähenemine välistab olukorra, kus erinevates keskkondades paigaldatakse teekide eri versioone, mis võiksid põhjustada vigu – seda eriti masinõppemudelite töös.

4 Masinõppe mudeli loomine

Sissejuhatuses on mainitud, et rakenduse esialgne versioon töötati välja tellimusprojekti osana, kus loodi masinõppe mudelid tilkade tuvastamiseks ja bakterite kasvu klassifitseerimiseks. Vaatamata sellele, et antud mudelid on juba varem loodud, mudeli edasise valideerimise mõistmiseks (alapeatükk 7.1) on vaja lugejat tutvustada mudeli loomise etappidega.

4.1 Tilkade tuvastamine

Tilkade segmenteerimiseks proovisid autorid esialgu kasutada *Houghi* algoritmi, kasutades OpenCV teegi *HoughCircles* funktsiooni, kuid see lähenemisviis ei sobi mitmel põhjusel:

- Pildistamistingimused võivad iga kord olla erinevad, mis nõuab eeltöötlusparameetrite pidevat kohandamist isegi adaptiivse eeltöötluse korral.
- *HoughCircles* tugineb objektide piiridel, püüdes leida täiuslikku ringi. See on mikrofluidikas sageli võimatu tilkade piiride tõttu, mis puutuvad tihedalt kokku, samuti õhumullide ja muude artefaktide tõttu.

Pärast võimalike lahenduste analüüsi koostati käsitsi märgendatud andmestik binaarseks segmenteerimiseks ning viidi läbi mudeli treenimine väiksel andmestikul. Katse osutus ebaõnnestunuks, kuna mudel hakkas ennustama üksnes tausta, jättes objektid tuvastamata. Probleem tulenes tõenäoliselt nii piiratud treeningandmete mahust kui ka ebatäpsetest maskidest. Lisaks võis tulemusi mõjutada valitud *U-Net* arhitektuur, mis ei sobi olukordadesse, kus tilgad puutuvad kokku või osaliselt kattuvad, käsitledes neid ühe pideva piirkonnana.

Edasises arenduses otsustati ajutiselt kasutada *Cellpose*'i mudelit. Kuigi see osutus väga tõhusaks, oli selle töökiirus märkimisväärselt madal: ilma graafikaprotsessorita oleks ühe pildi analüüs kestnud ligikaudu 10–15 minutit. Arvestades, et paljudel uurimisrühma tulevastel kasutajatel puudub sobiv riistvara, oli vajalik keskenduda lahenduse optimeerimisele.

Allpool on esitatud katsetatud optimeerimismeetodid, mille tulemused osutusid erineval määral mitterahuldavaks:

- *Cellpose*'i mudeli teisendamine ONNX-formaati ei andnud vajalikku kiirusekasvu;
- tilkade paralleelne töötlemine ei toonud märkimisväärset ajavõitu;
- pildi resolutsiooni vähendamine põhjustas olulise detailikadu, mis on edasiseks analüüsiks kriitiline;
- mudeli parameetrite optimeerimine viis analüüsi ebastabiilsuseni või täieliku peatumiseni.

Sellest tulenevalt osutus otstarbekaks naasta masinõppe lahenduste juurde, kasutades alusena eelnevalt treenitud YOLOv8n mudelit. Mudel omab juba võimet tuvastada põhiomadusi, nagu servad, tekstuurid ja objektide kujud, mistõttu vajab see täiendavat treenimist üksnes uurimisrühma esitatud andmete põhjal. Varasemate käsitsi märgendamisega seotud probleemide tõttu kasutati treeningandmestiku loomiseks *Cellpose*'i mudelit.

4.1.1 Treeningandmestiku loomine

Andmestiku loomise protsess automatiseeriti välja töötatud *DatasetCreator* klassi abil, mille implementatsiooni võib leida Google Colabi märkmikust lingilt[1]. Selle pilveplatvormi kasutamine andis juurdepääsu Tesla T4 GPU arvutusvõimsusele, mis vähendas närvivõrgu mudeli treeningaega mitmelt tunnilt 15-20 minutile.

Kvaliteetse andmestiku tagamiseks kasutati tugevate segmenteerimise artefaktidega, samuti erineva tilkade tiheduse ja erinevate valgustingimustega pilte. Algpiltide esialgseks segmenteerimiseks kasutati iga tilga piiride määramiseks *Cellpose* mudelit. Seejärel saadud maskide põhjal arvutati piiravad kastid, mis pärast seda teisendati nõutavasse YOLO-vormingusse: <*klassi_id*> <*x_keskus*> <*y_keskus*> <*laius*> <*kõrgus*>, kus on piirava kasti keskpunkti koordinaadid ning selle laius ja kõrgus on normaliseeritud vahemikus 0,0 kuni 1,0 pildi mõõtmete suhtes [8]. Kuna uuritakse üksnes piisku, on klassi identifikaator alati 0.

[1] `https://gitlab.cs.ttu.ee/andruv/iaib/-/blob/main/ml_resources/droplet%20detection/Dataset_for_droplet_segmentation_and_yolov8n_model_train.ipynb?ref_type=heads`

Lisaks on andmestikuga töötamiseks vaja säilitada paralleelne kataloogistruktuur, mis sisaldab JPEG-vormingus algpilte ning vastavaid tekstifaile koos iga tuvastatud objekti annotatsioonidega. Andmestik jaotati automaatselt treening- ja valideerimiskogumiteks suhtega 80/20.

4.1.2 YOLOv8n mudeli treenimine ettevalmistatud andmestikul

Masinõppe meetodite, näiteks siirdeõppe, kasutamisel soovitatakse üldjuhul kasutada vähemalt mitmesaja kuni tuhande märgistatud näite olemasolu iga klassi kohta, kuigi ranget miinimumi ei ole, kuna määravaks on objekti võimalike variatsioonide piisav esindatus treeningandmetes [9].

Mikrofluidika piltide suure tilgatiheduse tõttu oli võimalik koguda üle kahe tuhande märgistatud objekti. Treeningandmestiku mitmekesisust suurendati täiendavalt YOLOv8 sisseehitatud andmete augmentatsiooni abil, mis hõlmab värvikorrektsioone (tooni, küllastuse ja heleduse muutmine), geomeetrilisi teisendusi (skaleerimine, nihked, peegeldused) ning mitme pildi kombineerimist üheks [10].

Erinevad allikad soovitavad sõltuvalt andmestiku suurusest kasutada erinevat epohhide arvu [11, 12]. Ümberõppe vältimiseks valiti treenimiseks 50 epohhi. Lisaks rakendati varajase peatamise mehhanismi, mille kohaselt peatatakse treening juhul, kui valideerimiskogumi tulemused ei parane 10 järjestikuse epohhi jooksul.

Treeningprotsessi käigus mõõdetud näitajate dünaamika on esitatud tabelis 1. Tulemused näitavad, et kõige intensiivsem paranemine toimub ligikaudu epohhide 10–25 vahemikus, mille järel näitajad stabiliseeruvad. See viitab mudeli konvergentsile ning valepositiivsete tulemuste vähesele osakaalule, kinnitades valitud epohhide arvu sobivust

Tabel 1. YOLOv8n mudeli treeningmõõdikute dünaamika.

Epohh	**mAP50***	**Precision****	**Recall*****
1	0,167	0,203	0,643
5	0,400	0,295	0,934
10	0,419	0,299	0,947
15	0,729	0,966	0,427
20	0,938	0,960	0,810

Continues...

Tabel 1 – *Continues...*

Epohh	**mAP50***	**Precision****	**Recall*****
25	0,968	0,977	0,912
30	0,981	0,991	0,951
35	0,989	0,998	0,969
40	0,995	1,000	0,983
45	0,995	0,998	0,994
50	0,995	0,998	0,994

*mAP50 on standardmõõdik objekti detektorite kvaliteedi hindamiseks, mis näitab keskmist tuvastustäpsust, arvestades, et ennustatud piirav kast kattub tegeliku piirava kastiga vähemalt 50% ulatuses [13].

**Precision on näitaja, mis on suunatud positiivsete prognooside täpsusele [14].

***Recall on mõõdik, mis peegeldab mudeli võimet leida andmestikust kõik asjakohased eksemplarid [15].

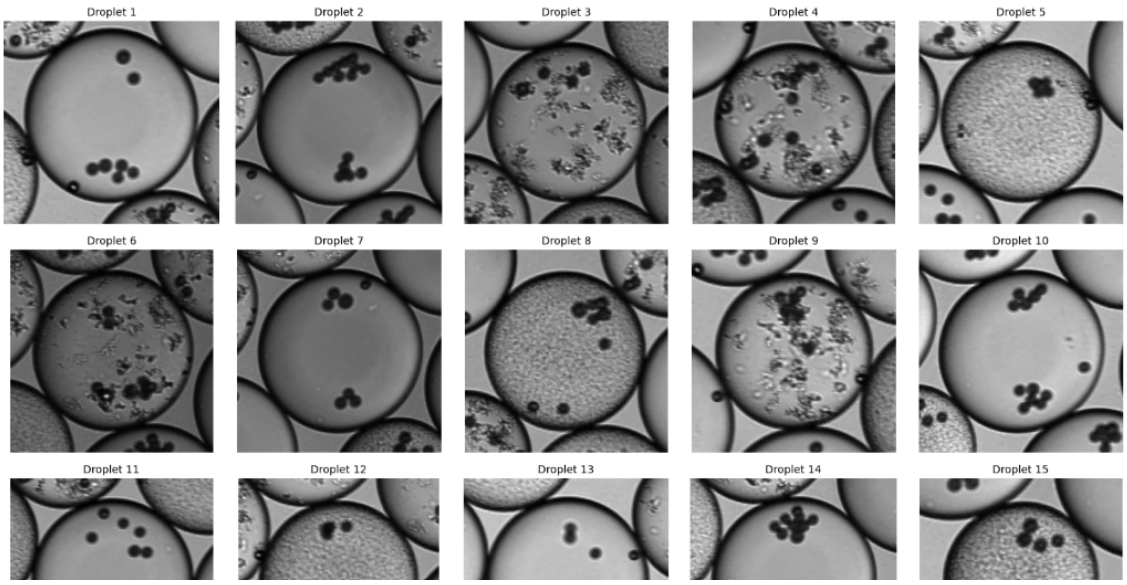
4.2 Bakterite kasvu klassifikatsioon

Selles peatükis kirjeldatakse masinõppe mudeli väljatöötamist bakterite kasvu klassifitseerimiseks tilkades. Selleks kasutatakse Google Colab keskkonda, mis kiirendab tänu Tesla T4 GPU-le oluliselt nii andmestiku loomist kui ka järgnevat närvivõrgu mudeli treenimist. Täielik kood on leitav järgmiselt lingilt[2].

4.2.1 Treeningandmestiku loomine

Andmestiku loomine automatiseeriti klassi *DatasetGenerator* abil, mis võimaldab iga tilka interaktiivselt ja eraldi klassifitseerida. Objektide tuvastamiseks lähtepiltidelt kasutati *Cellpose*'i mudelit, mis on spetsialiseerunud bioloogiliste objektide segmenteerimisele. *Cellpose*'i rakendati andmestiku loomise etapis, kuna lõplikku tarkvaralist lahendust ei olnud selleks hetkeks veel välja töötatud. Protsessi tulemusena genereeriti kõigist tuvastatud ning analüüsiks sobivatest tilkadest visualiseeringud (joonis 3).

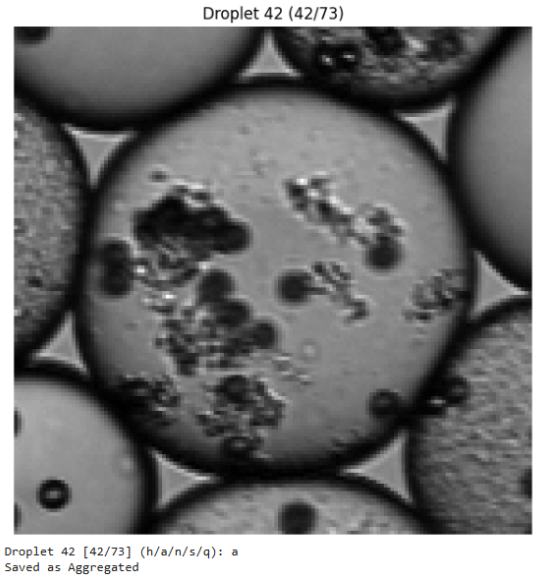
[2] `https://gitlab.cs.ttu.ee/andruv/iaib/-/tree/main/ml_resources/bacterial%20growth%20type%20classification`



Joonis 3. Andmestiku jaoks sobivate tilkade ruudustik.

Iga tilga jaoks on ette nähtud järgmised käsuvalikud (joonis 4):

- h [Homogenous] – bakterite ühtlane jaotumine;
- a [Aggregated] – bakterite klastrite moodustumine;
- n [No bacteria] – bakterite puudumine tilgas;
- s [Skip] – võimalus tilk vahele jätta, näiteks kui pildi kvaliteet on madal;
- q [Quit] – võimalus praeguse pildi töötlemine enneaegselt lõpetada.



Joonis 4. Valiku näide.

Pärast klassi valimist salvestatakse tilgast välja lõigatud pilt automaatselt vastavasse alamkataloogi (*/Homogenous*, */Aggregated* või */No_Bacteria*) unikaalse nimega, mis sisaldab ajatemplit ja tilga identifikaatorit.

Mitme mikrofluidika pildi töötlemise käigus loodud märgendatud andmestik sisaldab 509 üksiku tilga pilti. Kõik pildid on viidud ühtsesse vormingusse (PNG, halltoonides, 8 bitti piksli kohta) ning normaliseeritud. Klasside jaotust andmestikus on kirjeldatud tabelis 2.

Tabel 2. Klasside jaotus andmestikus.

Klass	**Piltide arv**	**Osakaal, %**
Homogenous	79	15,5
Aggregated	197	38,7
No Bacteria	233	45,8

Tabelist on näha, et andmestik on klasside lõikes tasakaalustamata: *No Bacteria* klass domineerib (45,8%), samas *Homogenous* klass on kõige vähem esindatud (15,5%). Selline jaotus peegeldab tegelikku bioloogilist jaotust eksperimentaalsetes andmetes, mida võeti mudeli treenimisel arvesse, kasutades andmete laiendamise meetodeid (augmentatsiooni) ja kaalutud kaotusfunktsiooni.

4.2.2 CNN-mudeli loomine bakterite kasvumustrite äratundmiseks

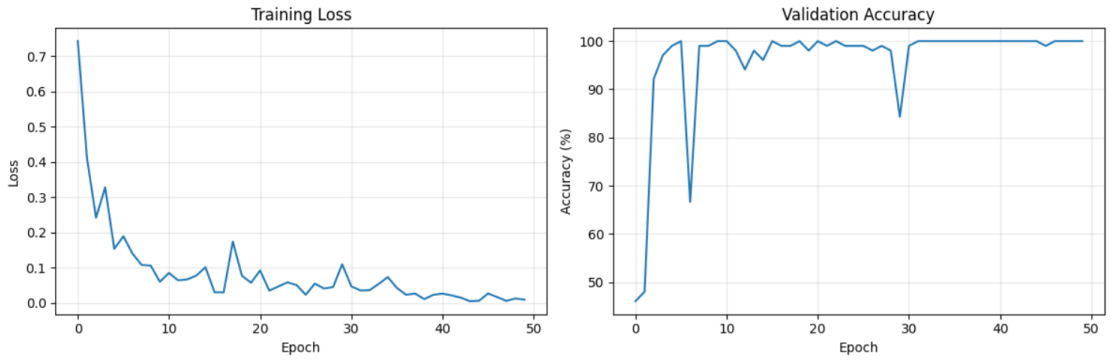
Eeltreenitud mudelid, nagu *ResNet* ja *EfficientNet*, on optimeeritud värvipiltide klassifitseerimiseks, millel on selgelt eristatavad objektipiirid. Käesoleva ülesande sisendandmed on aga madala kontrastsuse ja peeneteralise tekstuuriga monokroomsed mikrofluidika kujutised. Sellest tulenevalt loodi kerge konvolutsiooniline närvivõrk (CNN), mis on spetsiaalselt kohandatud peeneteraliste tekstuuritunnuste eraldamiseks: ühtlane teralisus (*Homogenous*), tumedad klastrid (*Aggregated*) ja sile tekstuur (*No Bacteria*).

Võrgu arhitektuur (koodis tähistatud kui *BacterialGrowthCNN*) põhineb klassikalisel ülesehitusel, kus vahelduvad konvolutsioonilised ja koondamiskihid. Filtrite arv suureneb järk-järgult (32 -> 64 -> 128 -> 256). Lisaks kasutatakse *Batch Normalization* ja *Dropout* mehhanisme üleõppimise vältimiseks. Olulisemate *PyTorch* realiseeringu fragmentidega saab tutvuda joonisel 29.

Mudeli treenimiseks jagati andmestik treening- (80%) ja valideerimisvalimiks (20%), säilitades klasside proportsioonid. Treeningandmestiku mitmekesisuse suurendamiseks rakendati järgmist andmete augmentatsiooni strateegiat: juhuslik horisontaalpeegeldus (p=0.5); juhuslik pööramine ±15° ulatuses; heleduse ja kontrastsuse muutmine (0,8–1,2); skaleerimine, 128×128 piksli suuruseks. Klasside tasakaalustamatuse kompenseerimiseks kasutati kaalutud *CrossEntropyLoss* kaofunktsiooni, kus klassikaalud arvutati esinemissagedusega pöördvõrdeliselt. Saadud kaalud olid: *Homogenous* – 2,15, *Aggregated* – 0,86, *No Bacteria* – 0,73.

Treening viidi läbi 50 epohhi jooksul. Õppimise kiiruse reguleerimiseks kasutati *ReduceLROnPlateau* mehhanismi: kui valideerimise kvaliteet ei paranenud 10 epohhi vältel, vähendati õppimise kiirust kahekordselt (factor=0,5). See aitas mudelil treeningu lõpufaasis kaale täpsemalt seadistada.

Nagu joonisel 5 esitatud graafikutelt näha, mudel demonstreerib kiiret konvergentsi: kolmandaks epohhiks on valideerimise täpsus 92% ja kuuendaks epohhiks 100%.



Joonis 5. Kaotusfunktsiooni ja valideerimise täpsus muutumise graafikud.

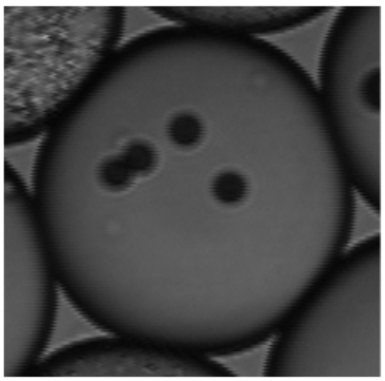
Saavutatud täpsust võib selgitada märgendatud andmestiku kõrge kvaliteediga ning klasside selge eristatavusega tekstuuri põhjal, mis lihtsustab klassifitseerimisülesannet. Samas koosneb valideerimisvalim vaid 102. pildist, mistõttu on võimalik, et mudel on õppinud spetsiifilisi mustreid, mitte üldistatavaid tunnuseid.

Kuigi treenitud mudel on juba integreeritud bakterianalüüsi moodulisse (*bacteria_service.py*) ning näitab reaalsetel mikrofluidika piltidel stabiilset toimimist, vajab see edasist valideerimist mitmekesisemal andmestikul.

5 Mikroplasti tuvastamine

5.1 Mikroplasti tuvastamise algoritm

Mikroplasti osakesed on tavaliselt väikesed, ümarad ja tumedad struktuurid, mis eristuvad heledamast taustast. Nende suurus jääb tavaliselt vahemikku 3–15 pikslit, olenevalt pildi skaalast (joonis 6).



Joonis 6. Näide mikroplasti osakesi sisaldavast tilgast.

Eesmärk on need osakesed tilkade sees automaatselt tuvastada, eristades neid bakteritest, mürast ja tilga servas tekkivatest varjudest. Seetõttu mikroplasti tuvastamise algoritm on töö käigus läbinud mitu arendusetappi.

Esialgne versioon kasutas ainult Houghi ringide detektorit adaptiivsete parameetritega, mis varieerusid sõltuvalt bakterite kasvu tüübist. Eelkõige muudeti dünaamiliselt järgmisi parameetreid:

- *minRadius* ja *maxRadius* – otsitavate ringide minimaalne ja maksimaalne raadius (bakteriteta tilkades kasutati kitsamat vahemikku, bakteritega tilkades laiemat);
- *param1* – Canny servade tuvastuse lävi (bakteritega tilkades tõsteti väärtust, et vähendada tundlikkust bakterite tekstuuri suhtes);
- *param2* – ringide tuvastuse lävi (bakteritega tilkades tõsteti väärtust valepositiivsete

tulemuste vältimiseks);

- *minDist* – minimaalne kaugus tuvastatud ringide vahel (bakteritega tilkades suurendati, et vältida kattuvaid tuvastamisi).

Lisaks parameetrite kohandamisele rakendati tuvastatud kandidaatidele mitmeastmelist filtreerimist:

- Intensiivsuse filter kõrvaldab liiga tumedad osakesed (*min_mean_intensity*) või liiga heledad osakesed (*max_mean_intensity*);
- Kuju filter nõuab piisavat ringikujulisust (*min_circularity*) ja lubab maksimaalset ekstsentrilisust (*max_eccentricity*);
- Kontrasti filter nõuab piisavat kontrasti osakese ja tausta vahel (*min_contrast_ratio*);
- Kauguse filter kõrvaldab tilga servale liiga lähedal asuvad osakesed (*min_distance_from_edge*).

Vaatamata rakendatud meetmetele ilmnesid mitmed piirangud. Kui osakesed paiknesid üksteisele väga lähedal või olid osaliselt ühinenud, osutus nende tuvastamine ebaühtlaseks, kuna Houghi ringtuvastus eeldab rangelt ümarat kuju. Kõige keerulisemaks osutusid bakteritega tilgad, kus peeneteraline tekstuur häiris filtreerimisprotsessi. See viis valepositiivsete tulemuste suurenemiseni, samas kui osa tegelikke mikroplasti osakesi jäi tuvastamata

Käesoleva bakalaureusetöö raames täiendati algoritmi, et oluliselt vähendada eelkirjeldatud piiranguid. Eelkõige on laiendatud filtrisüsteemi:

- Tekstuuri filtreerimine Laplace'i variatsiooni (*laplacian_var*) ja gradiendi standardhälbe (*gradient_std*) alusel eristab mikroplasti sileda tekstuuriga bakteritest, millel on kare ja peeneteraline tekstuur.
- Analüüs radiaalprofiilist arvutab, kui sujuvalt intensiivsus muutub osakese keskelt servani. See tähendab, et sileda pinnaga mikroplastil on madal väärtus (*profile_variance*), samas kui bakterite klastritel on ebaühtlane struktuur ning neil on kõrge väärtus. Lisaks sellele hinnatakse profiili sümmeetriat (*profile_asymmetry*). Madalad väärtused näitavad sümmeetrilise profiiliga mikroplasti, kõrged väärtused aga asümmeetrilise profiiliga tilkade servi.
- Samuti on olemas ka eraldi filtrite komplekt heledate taustade jaoks (*bright_background_filters*), mis kohandab intensiivsuse, kauguse ja asümmeetria piirväärtusi,

vähendades oluliselt valepositiivseid tulemusi heledal taustal.

Selleks et tuvastada ebakorrapärase kujuga või lähestikku asuvaid mikroplasti osakesi, on lisatud Blob-detektor, mis töötab paralleelselt Houghi ringtuvastusega. Mõlemad detektorid salvestavad oma tulemused ühisesse nimekirja (*all_particles*), mille järel dubleerivad tulemused eemaldatakse.

Blob-detektor kasutab järgmisi parameetreid:

- *minArea*, *maxArea* – tuvastatava ala minimaalne ja maksimaalne pindala arvutatakse põhifiltri *min_radius* ja *max_radius* väärtuste põhjal;
- *minCircularity* – minimaalne ringikujulisus;
- *minConvexity* – minimaalne kumerus välistab nõgusate või ebaühtlaste servadega objektid;
- *minInertiaRatio* – minimaalne inertsi suhe välistab tugevasti väljavenitatud objektid.

Kõik need täiendused on realiseeritud *MicroplasticService* klassis.

5.2 Mikroplasti tuvastamise mudeli loomine

Pärast valideerimist (alampeatükk 7.2.1) sai selgeks, et klassikaline Houghi ringidel ja Blob-detektoril põhinev algoritm ei suuda määratud ülesannet täielikult realiseerida, sest raskused ebakorrapärase kujuga osakeste ja bakteritega kattuvate mikroplastide tuvastamisel pole kuhugi kadunud. Eelkirjeldatud piirangute ületamiseks rakendati siirdeõpet, kasutades eeltreenitud YOLOv8n-seg mudelit.

Selleks loodi uus andmestik, milles märgendati väljalõigatud tilkade sees paiknevad mikroplasti osakesed, suunates mudeli tähelepanu asjakohasele piirkonnale. Treeningandmestik sisaldas 1564 pilti (sealhulgas 905 negatiivset näidet ilma mikroplastita), valideerimiskogum 392 pilti. Mudelit treeniti Google Colabis, kasutades Tesla T4 GPU-d 100 epohhi vältel. Siin[1] võib tutvuda treeningprotsessi ja andmestiku loomisega.

Väljatreenitud mudel saavutas treeningu valideerimiskomplektil tulemused, kus on täpsus 96,4%, tundlikkus 90,9% ja keskmine täpsus *(mAP50)* 93,9%. Nende tulemuste kinnita-

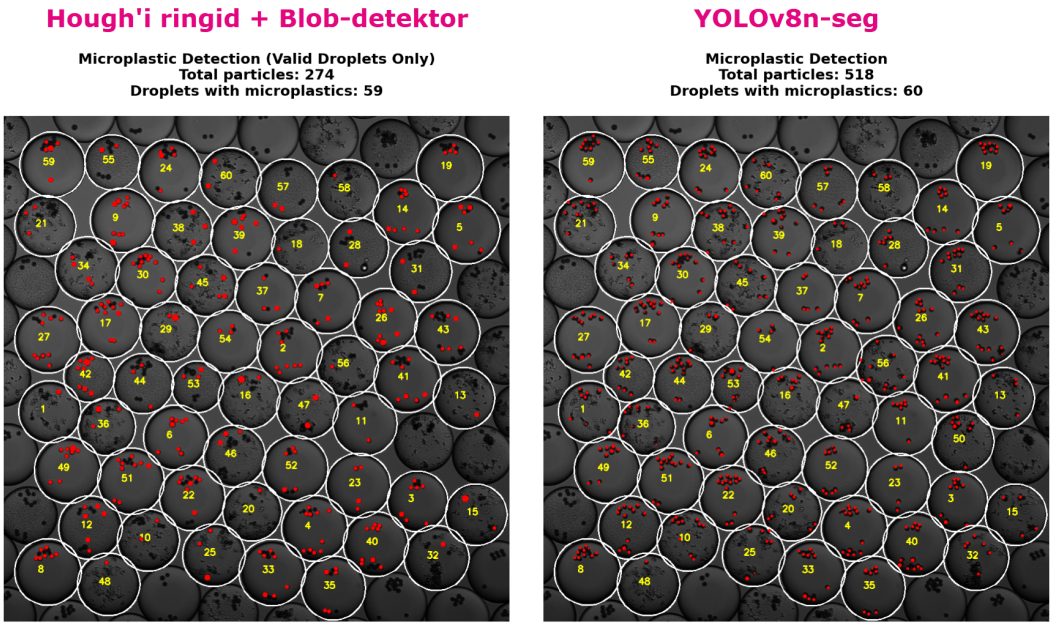
[1] `https://gitlab.cs.ttu.ee/andruv/iaib/-/blob/main/ml_resources/microplastic%20detection/DATASET_and_yolov8n_seg_training.ipynb?ref_type=heads`

miseks viidi läbi täiendav valideerimine iseseisval andmestikul, mille tulemused on toodud alampeatükis 7.2.2.

Väljatreenitud mudel integreeriti *MicroplasticYOLOService* klassi. Antud klass võtab sisendiks originaalpildi ja tilga info ning tagastab tuvastatud mikroplasti osakeste keskpunktid ja raadiused. Lisaks kontrollitakse automaatselt, et tuvastatud osakesed asuvad tilga piirkonnas. See on oluline, kuna YOLO-seg mudel töötab väljalõigatud tilga piirkonnal ning selle tulemused võivad sisaldada valepositiivseid tuvastusi, mis jäävad tilgast väljapoole.

Erinevalt klassist *MicroplasticService* ei vaja käesolev lahendus täiendavat filtreerimist (nt intensiivsuse, kuju, kontrasti või radiaalprofiili alusel) ning ei sõltu bakterite kasvutüübist, kuna mudel on õppinud mikroplasti osakesi eristama nende visuaalsete tunnuste põhjal.

Võrdluseks on allpool (joonis 7) näidatud sama pildi tuvastustulemused klassikalise algoritmi *(Hough'i ringid koos Blob-detektoriga)* ja *YOLOv8n-seg* mudeli abil. Näha on, et *YOLOv8n-seg* mudel toimib paremini kui klassikaline algoritm piirkondades, kus osakesed on üksteisele lähedal või kattuvad bakteritega.



Joonis 7. Tuvastustulemuste võrdlus.

6 Täiendavad analüüsi funktsionaalsused

6.1 Analüüsiajaloo ja sessioonide võrdlemine

Kasutajakogemuse parandamiseks lisati rakendusse ajaloo funktsionaalsus, mis võimaldab kasutajal vaadata varasemate analüüside tulemusi ilma pilte uuesti töötlemata. See on eriti oluline suurte andmemahtude korral, kus iga analüüs nõuab märkimisväärset aega sõltuvalt piltide arvust ja keerukusest.

Andmete salvestamiseks kaaluti algselt JSON-vormingu kasutamist, mille puhul hoitakse kõiki sessioone ühes failis. Sellisel juhul tuleks kogu fail iga kord mällu laadida, mis muutub andmemahu kasvades ebaefektiivseks. Seetõttu valiti andmebaasilahendusena SQLite, mis on serverivaba ja kergekaaluline ning sobib hästi töölauarakenduste jaoks. SQLite võimaldab teha optimeeritud päringuid ning toetab andmete filtreerimist ja indekseerimist.

Andmebaas koosneb kahest tabelist, mis on omavahel seotud *session_id* võtme kaudu:

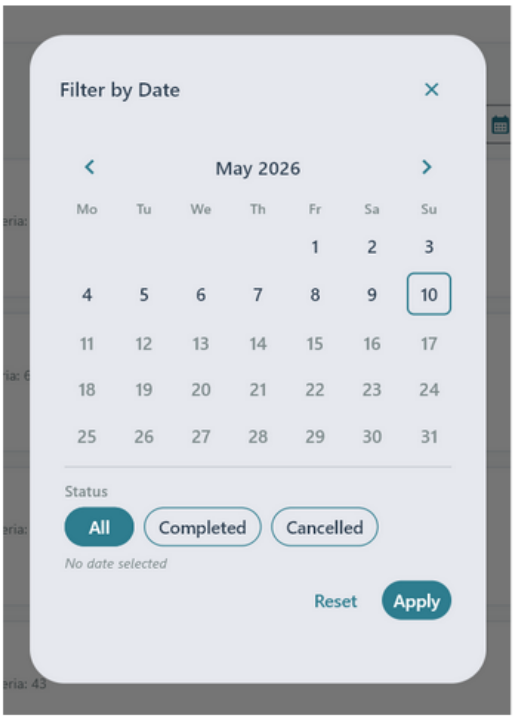
- *sessions* on põhitabel, kus *session_id* on primaarvõti. Tabel sisaldab iga sessiooni üldandmeid: algus- ja lõpuaega, staatust (lõpetatud või katkestatud) ning koondväärtusi (nt failide, tilkade, mikroplasti ja bakterite arv).
- *session_files* on seotud tabel, kus *id* on primaarvõti ja *session_id* on võõrvõti, mis viitab tabelile *sessions*. See sisaldab iga analüüsitud faili kohta üksikasjalikke tulemusi: faili nime, visualiseeringute salvestamise olekut ning vastavaid koondnäitajaid (tilkade, mikroplasti ja bakterite arv).

Tabelite täielik struktuur on esitatud joonisel 30 (vt lisa 3).

Andmebaasi struktuuri põhjal ajaloo vaade (*HistoryView*) pakub järgmisi võimalusi:

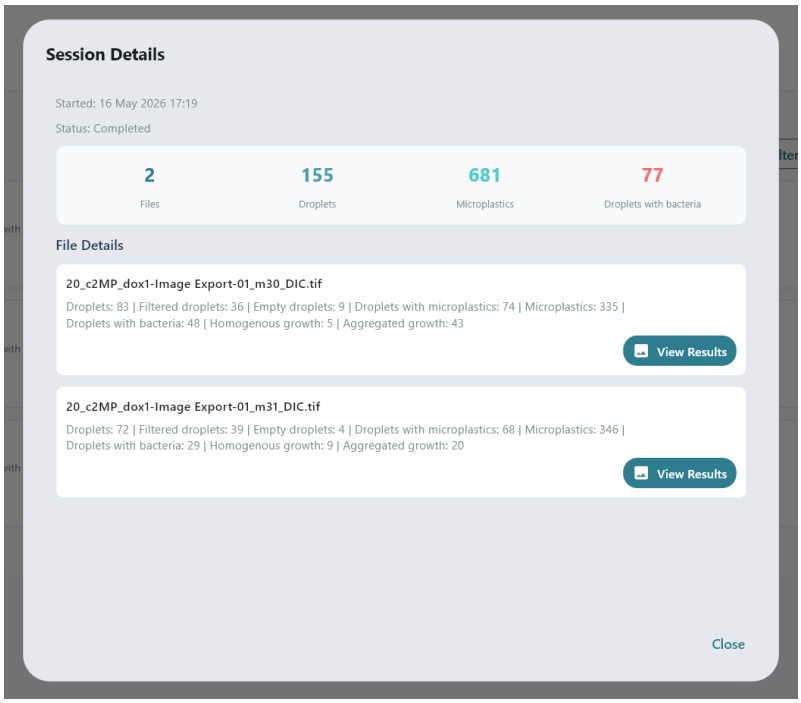
- vaadata kõiki varasemaid sessioone koos kokkuvõtlike väärtustega (failide arv, tilkade arv, mikroplastide arv, bakteritega tilkade arv);
- kustutada üksikuid sessioone või kogu ajalugu;
- filtreerida sessioone kuupäeva kalendrivaate ning analüüsi staatuse järgi (lõpetatud

või katkestatud) (joonis 8);



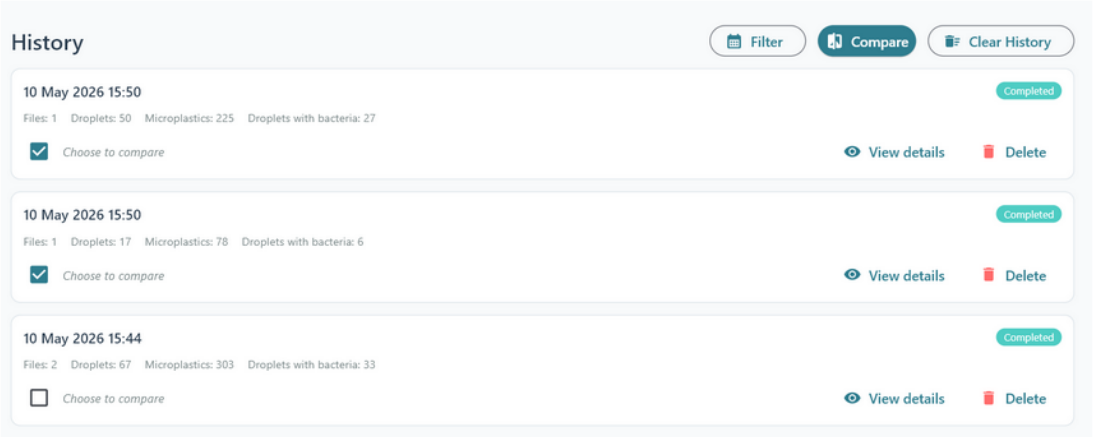
Joonis 8. Sessioonide filtreerimine kuupäeva ja staatuse alusel.

- vaadata iga sessiooni üksikasju, kus kuvatakse kõik analüüsitud failid koos nende kokkuvõtlike väärtustega (joonis 9);



Joonis 9. Sessiooni detailvaade.

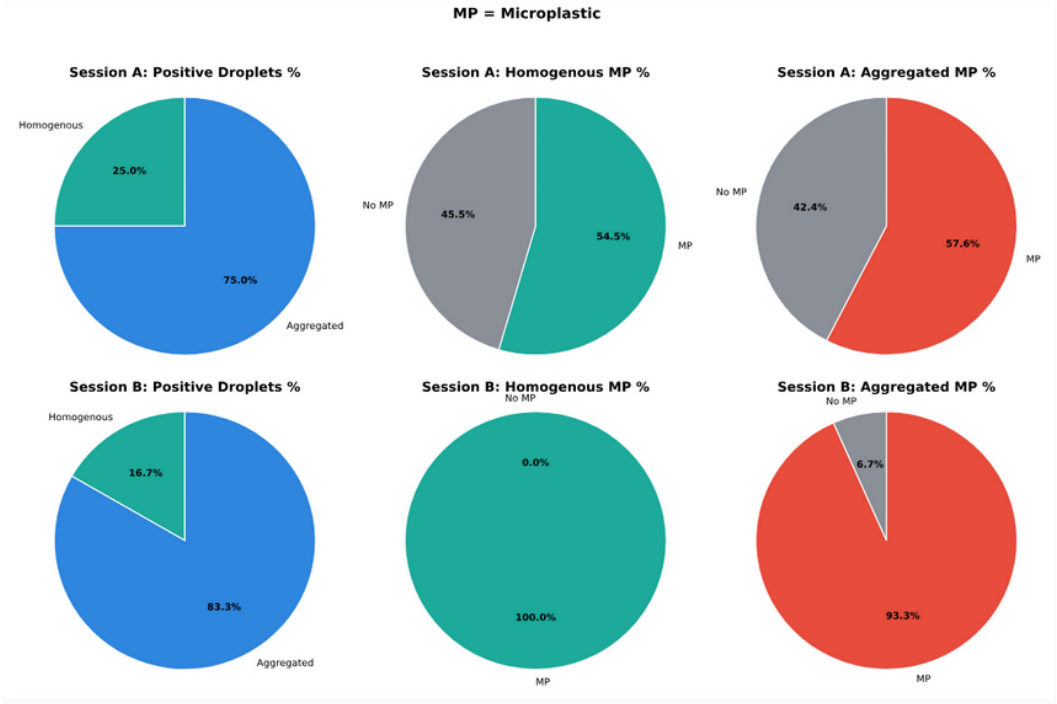
- valida sessioone võrdlemiseks (vahemikus 2-5), märkides need vastavate ruutudega ja vajutades nuppu *Compare* (joonis 10).



Joonis 10. Sessioonide valimine võrdlemiseks.

Pärast sessioonide valimist ja nupu *Compare* vajutamist avaneb võrdlemise vaade (*CompareView*). See kuvab tabelina valitud sessioonide peamised kokkuvõtlikud väärtused. Kahe sessiooni võrdlemise korral näidatakse ka absoluutset ja protsentuaalset muutust (*delta*). Lisaks genereeritakse *matplotlib* abil iga sessiooni kohta kolm sektordiagrammi (joonis 11):

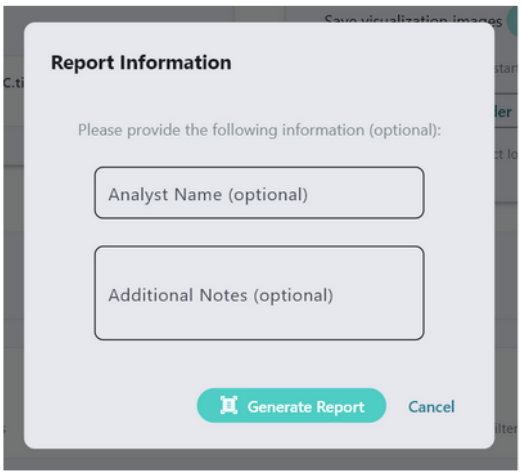
- baktereid sisaldavate tilkade jaotus homogeenseteks ja agregeerituks;
- mikroplasti osakaal homogeensetes tilkades;
- mikroplasti osakaal agregeeritud tilkades.



Joonis 11. Sektordiagrammide näited võrdlemise vaates.

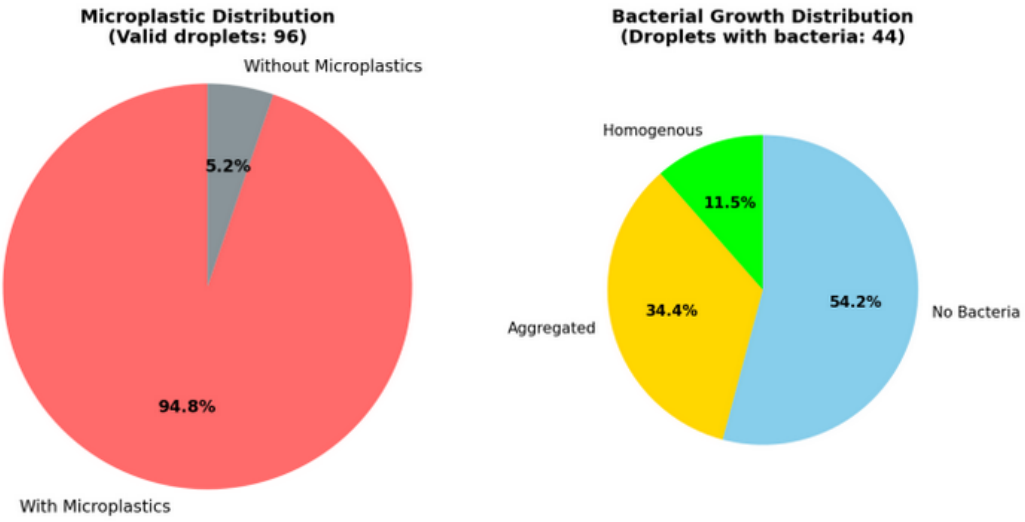
6.2 PDF-raporti genereerimine

Pärast analüüsi lõppu kasutaja saab analüüsi tulemuste põhjal genereerida PDF-raporti. Raporti loomisel on võimalik lisada analüütiku nimi ja täiendavad märkmed, näiteks analüüsitavate piltide üksikasjade kohta (joonis 12). Need väljad on vabatahtlikud, seetõttu kui neid ei täideta, jäetakse need raportist välja.



Joonis 12. Andmete sisestamise vorm raporti genereerimiseks.

Raport sisaldab üldist statistikat, kus kuvatakse kõigi analüüsitud failide koondtulemused. Seejärel esitatakse kaks sektordiagrammi: esimene näitab mikroplasti sisaldavate ja mikroplastita tilkade jaotust, teine bakterite kasvu jaotust (joonis 13).



Joonis 13. Sektordiagrammide näited PDF-raportis.

Seejärel kuvatakse iga faili kohta üksikasjalikud tulemused koos ülevaatliku pildiga, kus kõik tilgad on värvitud vastavalt bakterite kasvu tüübile. Kui kasutaja on analüüsi seadetes visuaalide salvestamise välja lülitanud, siis vastav visuaalne pilt raportist välja jäetakse.

7 Valideerimine

7.1 Masinõppe mudelite hindamine

Mudelite efektiivsust hinnati eraldiseisval andmestikul, mis koosnes uurimisrühma esitatud uutest mikrofluidika piltidest. Kuna andmestikku ei olnud treeningprotsessis kasutatud, võimaldas see objektiivselt hinnata mudelite suutlikkust tundmatute andmete töötlemisel.

7.1.1 Tilkade tuvastamise mudeli täpsus

Mudeli täpsuse hindamiseks kasutati 30 mikrofluidika pilti, mille jaoks loodi esmalt käsitsi võrdlusandmestik (*ground truth*) CVAT-i (*Computer Vision Annotation Tool*) keskkonnas. See tööriist võimaldab objektide täpset piiritlemist ning toetab andmete eksporti COCO 1.1 JSON-vormingusse, kus ellipsi parameetrid (keskpunkt, raadiused x- ja y-teljel ning pöördenurk) salvestatakse atribuutidena. See võimaldab vajaduse korral teostada täpsemat kujupõhist võrdlust.

Seejärel kasutati YOLOv8n mudelit samade piltide töötlemiseks. Selleks töötati välja automatiseeritud *Python*i skript[1], mis võrdleb YOLOv8n tuvastustulemusi käsitsi loodud võrdlusandmetega.

Iga tuvastatud tilga korral arvutatakse ühisosa ja ühendi suhe (*Intersection over Union*, *IoU*) vastava võrdlusobjektiga. Tuvastus loetakse tõepäraseks positiivseks (*True Positive*), kui $IoU \geq 0{,}5$ (50%). Vastasel juhul klassifitseeritakse tulemus valepositiivseks (*False Positive*) või loetakse tuvastamata tilgaks (*False Negative*). Skript arvutab kõigi 30 pildi põhjal täpsuse (*Precision*), saagise (*Recall*) ja F1-skoori ning genereerib tulemusi illustreerivad võrdluspildid.

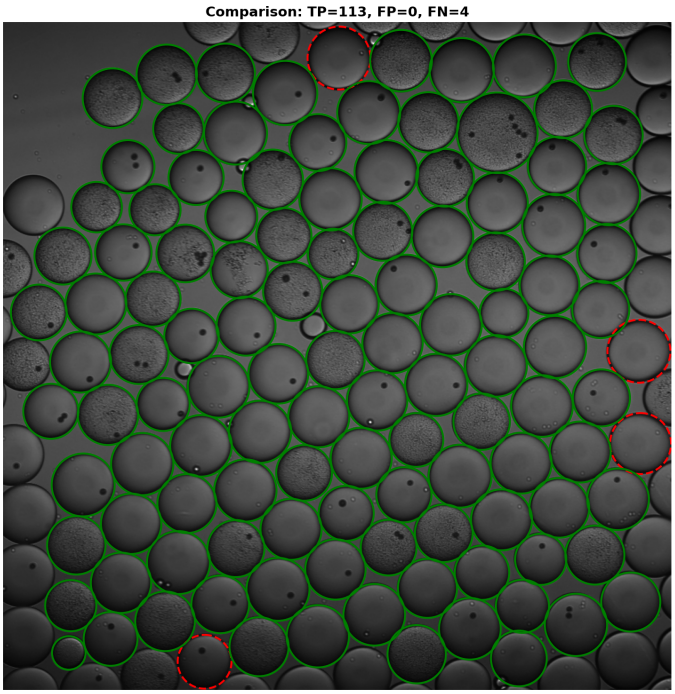
Täpsemad tulemused on esitatud tabelis 3.

[1] `https://gitlab.cs.ttu.ee/andruv/iaib/-/blob/main/validation/droplet%20detection/droplet_validation.py?ref_type=heads`

Tabel 3. Tilkade tuvastamise mudeli (YOLOv8n) statistilised näitajad testandmestikul.

Näitaja	**Väärtus**
Testpiltide arv	30
Tilkade arv (Ground Truth)	3329
Tuvastatud tilkade arv (YOLO)	3265
Õiged tuvastused (True Positives)	3265
Valepositiivsed tuvastused (False Positives)	0
Tuvastamata tilgad (False Negatives)	64
Täpsus (Precision)	1,000 (100,00%)
Saagis (Recall)	0,9808 (98.08%)
F1-skoor	0,9903

Tulemused näitavad, et YOLOv8n mudel saavutas tilkade lokaliseerimisel kõrge usaldusväärsuse. Valideerimise käigus valepositiivseid tuvastusi ei esinenud, sest kõik tuvastatud objektid vastasid tegelikele tilkadele. Tuvastamata tilkade osakaal on 1,9%. Enamik tuvastamata tilkadest paiknes pildi äärealadel või olid ebatüüpilise kujuga. Seda on visuaalsel võrdlusel selgelt näha joonisel 14. Joonisel on rohelisega tähistatud õiged tuvastused (*True Positives*) ja punasega tuvastamata tilgad (*False Negatives*).



Joonis 14. YOLOv8n mudeli tuvastustulemuste visualiseerimine testpiltidel.

Süsteemi on integreeritud filtreerimismehhanism, mis eemaldab automaatselt kõik pildi serva puudutavad tilgad (*margin* = 3). See tagab, et edasisse analüüsi kaasatakse üksnes täielikult vaateväljas paiknevad tilgad. Seetõttu kajastuvad servades asuvad objektid valideerimistulemustes tuvastamata tilkadena.

7.1.2 Bakterite kasvumustrite klassifitseerimise CNN-mudeli täpsus

CNN-mudeli täpsuse hindamiseks kasutati 17 mikrofluidika pilti, mis hõlmasid kokku 1469 individuaalset tilka. Esmalt tuvastati piltidel tilgad YOLOv8n mudeliga. Seejärel klassifitseeriti interaktiivselt iga tuvastatud tilga ühte kolmest klassist: homogeenne kasv, agregeeritud kasv või bakterite puudumine. Nende klasside põhjal loodi võrdlusmaskid (*ground truth*), kus maski väärtused: 1 – homogeenne, 2 – agregeeritud ning 3 – bakteriteta.

Tulemused on esitatud tabelis 4. Täpsemad hindamistulemused võib vaata siin[2].

Tabel 4. CNN-mudeli klassifitseerimise täpsusnäitajad klasside lõikes.

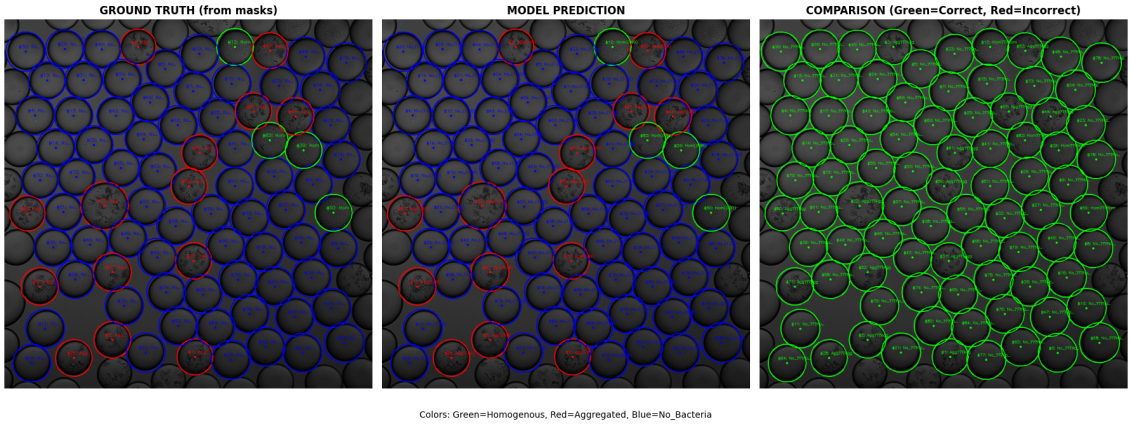
Klass	**Tilkade arv (Ground truth)**	**Täpsus (Precision)**	**Saagis (Recall)**	**F1-skoor**
Homogeenne kasv	111	0,863	0,910	0,886
Agregeeritud kasv	244	0,958	0,934	0,946
Bakterid puuduvad	1114	0,998	0,998	0,998
Koguvalim (kaalutud keskmine)	1469	0,981	0,981	0,981

Mudeli üldine täpsus oli 98,1%, saavutades tasakaalustatud täpsuse 94,8%. Kõige kõrgem täpsus saavutati bakterite puudumise tuvastamisel, kus mudel eksis vaid üksikutel juhtudel. Keerulisemaks osutus homogeense ja agregeeritud kasvu eristamine olukordades, kus sarnasemad tekstuurid ja madalam kontrastsus võisid põhjustada klassidevahelist ebaselgust. Siiski on mudeli usaldusväärsus piisavalt kõrge, et võimaldada mikroskoobipiltide automaatset analüüsi ja vähendada oluliselt käsitsi märgendamisele kuluvat aega.

Kvalitatiivsed näited klassifitseerimise tulemustest on esitatud joonisel 15. Joonisel on roheliste ringidega tähistatud korrektsed klassifikatsioonid ja punastega vead. Antud näitel

[2] `https://gitlab.cs.ttu.ee/andruv/iaib/-/blob/main/validation/bacterial%20growth%20type%20classification/new_bacterial_growth_cnn_evaluation.ipynb?ref_type=heads`

on kõik tuvastused korrektsed.



Joonis 15. CNN mudeli klassifitseerimistulemuste võrdlus maamärgistusega.

7.2 Mikroplasti tuvastamise lähenemisviisi hindamine

7.2.1 Mikroplasti tuvastamise algoritmi võrdlev analüüs

Testimiseks kasutati 17 pilti, millel oli kokku 2895 käsitsi märgendatud mikroplasti osakest. Märgendamine tehti CVAT (*Computer Vision Annotation Tool*) keskkonnas ning andmed on eksporditud COCO vormingus. Märgendamisel ümarate osakeste jaoks kasutati ellipsit, aga ebakorrapärase kujuga osakeste puhul erinevaid hulknurkasid, sõltuvalt mikroplasti kujust.

Pärast märgendamist eksporditi andmed JSON-failidena, millest eraldati mikroplasti osakeste koordinaadid. Nende põhjal loodi võrdlusmaskid (*ground truth*). Seejärel töödeldi samu pilte kahe algoritmi versiooniga: klassikaline Houghi ringide detektor ning täiustatud algoritm, kus mõlemad detektorid (Houghi ringid + Blob-detektor) töötavad paralleelselt. Mõlemal juhul saadi tuvastatud osakeste põhjal maskid.

Mikroplasti tuvastamise algoritmi täpsuse hindamiseks võrreldi võrdlusmaske (*ground truth*) algoritmi genereeritud maskidega. Vastavuse kriteeriumiks oli ühisosa ja ühendi suhe (*Intersection over Union*, *IoU*), kus $IoU \geq 0,3$. Väikeste objektide puhul langeb kattuvus juba väikese nihke korral kiiresti alla 0,5, mistõttu rangem lävi oleks liiga piirav. Seetõttu 0,3 on antud kontekstis sobiv kompromiss. Iga pildi kohta arvutati täpsusnäitajad: *precision*, *recall* ja F1-skoor.

Tulemused mõlema detektori versiooni kohta on esitatud tabelis 5.

Tabel 5. Mikroplasti tuvastamise tulemused (Houghi ringid + Blob-detektor).

Näitaja	**Väärtus**
Tuvastatud osakesi kokku	1687
Tuvastatud osakesi kokku käsitsi	2895
Õiged tuvastused (True Positives)	1647
Valepositiivsed tuvastused (False Positives)	40
Tuvastamata tilgad (False Negatives)	1248
Täpsus (Precision)	0,9763 (97,63%)
Saagis (Recall)	0,5689 (56,89%)
Üldine F1-skoor	0,7189
Üldine täpsus (accuracy)	0,6404 (64,04%)

Tulemused ainult Houghi ringidega versiooni kohta on esitatud tabelis 6.

Tabel 6. Mikroplasti tuvastamise tulemused (ainult Houghi ringid).

Näitaja	**Väärtus**
Tuvastatud osakesi kokku	1239
Tuvastatud osakesi kokku käsitsi	2895
Õiged tuvastused (True Positives)	1222
Valepositiivsed tuvastused (False Positives)	17
Tuvastamata tilgad (False Negatives)	1673
Täpsus (Precision)	0,9863 (98,63%)
Saagis (Recall)	0,4221 (42,21%)
Üldine F1-skoor	0,5912
Üldine täpsus (accuracy)	0,6172 (61,72%)

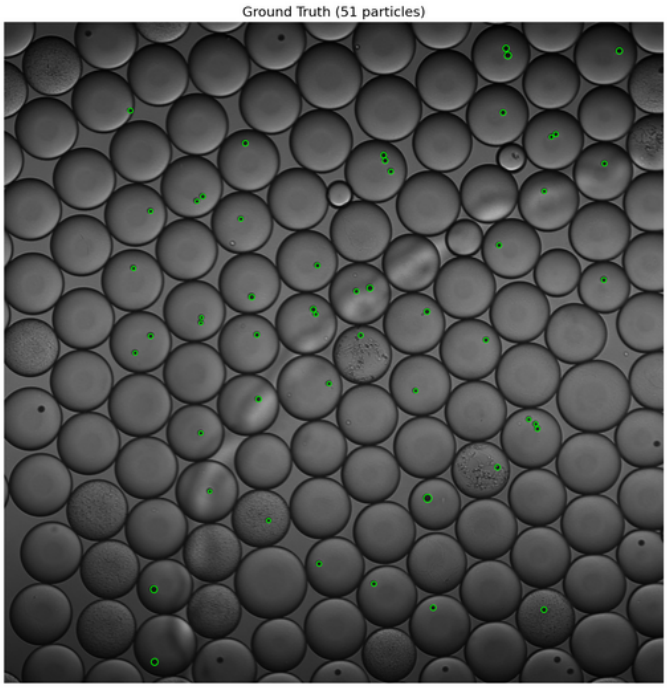
Tulemustest selgub, et tekstuuri filtreerimise ja radiaalanalüüsi kombinatsioon koos *Hough Circles* ja Blob detektoritega tõusis tundlikkus (recall) 42,21%-lt 56,89%-le. See tähendab, et uus algoritm suutis tuvastada rohkem tegelikke mikroplasti osakesi, eriti neid, mis on ebaregulaarse kujuga või asuvad tihedalt koos. Samal ajal täpsus (*precision*) langes vaid 98,63%-lt 97,63%-le (langes 1 protsendi võrra). Valepositiivsete tulemuste arv kasvas 17-lt 40-le, mis on aktsepteeritav kompromiss parema tundlikkuse saavutamiseks. F1-skoor paranes 0,591-lt 0,719-le, mis samuti näitab, et üldine tasakaal tundlikkuse ja täpsuse vahel on oluliselt parem.

Kuigi Blob-detektor parandas oluliselt bakteritega kattuvate või pildi servades asuva-

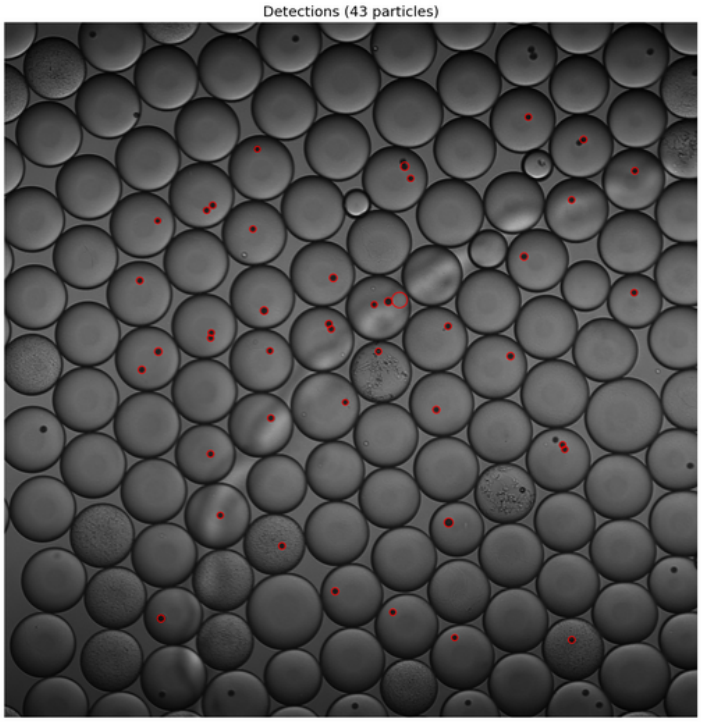
te mikroplasti osakeste tuvastamist, suurendas see ka veidi valepositiivsete tulemuste arvu. Kuna Blob-detektor on tundlikum, tuvastab see mõnikord müra või tilkade servi valepositiivsetena.

Algoritmi ja võrdlusandmete abil saadud tulemuste visuaalse võrdluse näide on toodud järgmistel joonistel:

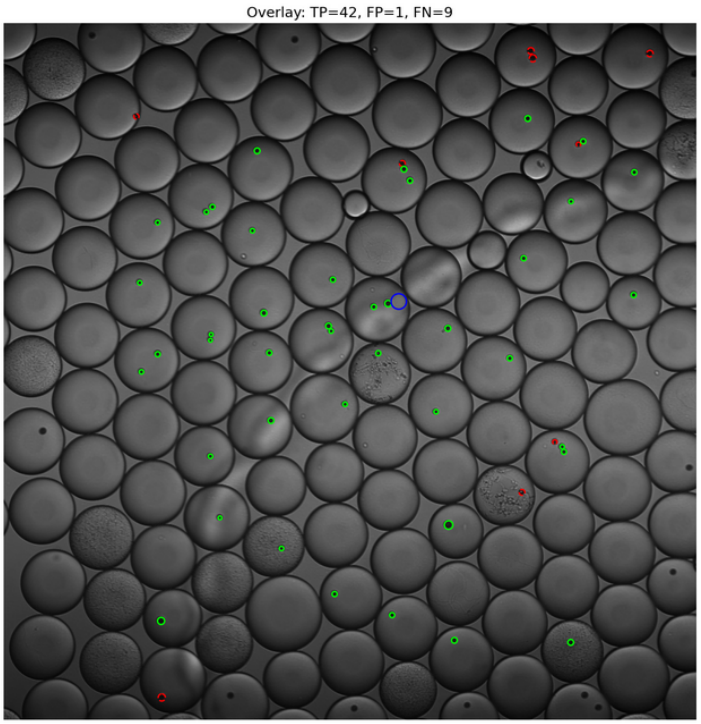
- võrdlusmask (ground truth), joonis 16;
- algoritmi mikroplasti tuvastused, joonis 17;
- TP/FP/FN ülekanne, joonis 18, kus rohelised ringid tähistavad õigeid tuvastusi (TP), sinised valepositiivseid (FP) ja punased tuvastamata osakesi (FN).



Joonis 16. Mikroplasti tuvastamise hindamine. Võrdlusmask (ground truth).



Joonis 17. Mikroplasti tuvastamise hindamine. Algoritmi mikroplasti tuvastused.



Joonis 18. Mikroplasti tuvastamise hindamine. TP/FP/FN ülekanne.

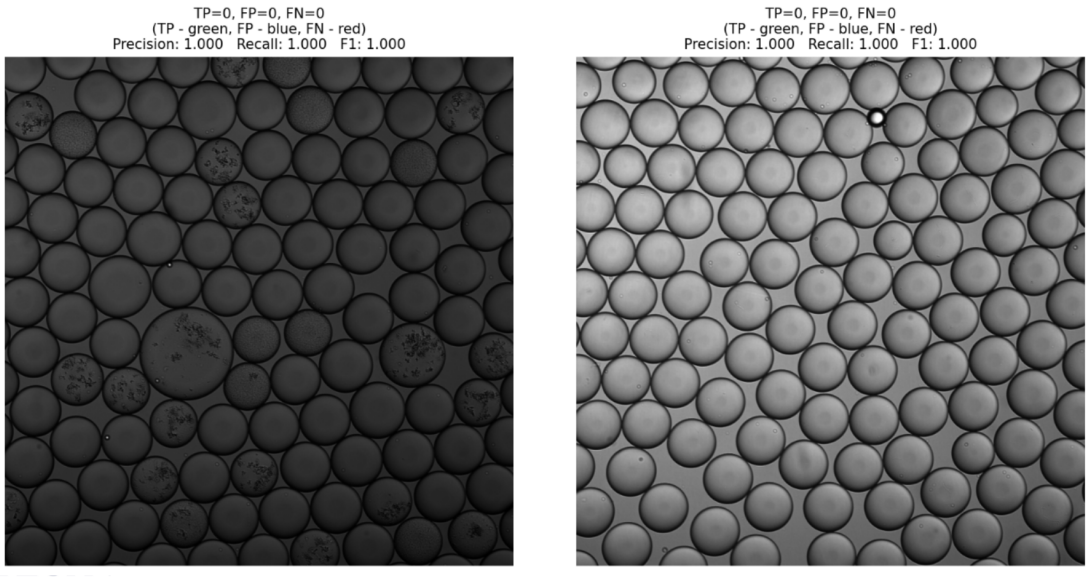
7.2.2 Mikroplasti tuvastamise YOLOv8n-seg mudeli hindamine

Varasemad hindamistulemused mõjutasid mudeli loomist, kasutades YOLOv8n-seg mudeli siirdeõppeks. Selle lähenemisviisi hindamine on toodud tabelis 7.

Tabel 7. Mikroplasti tuvastamise tulemused (YOLOv8n-seg mudel).

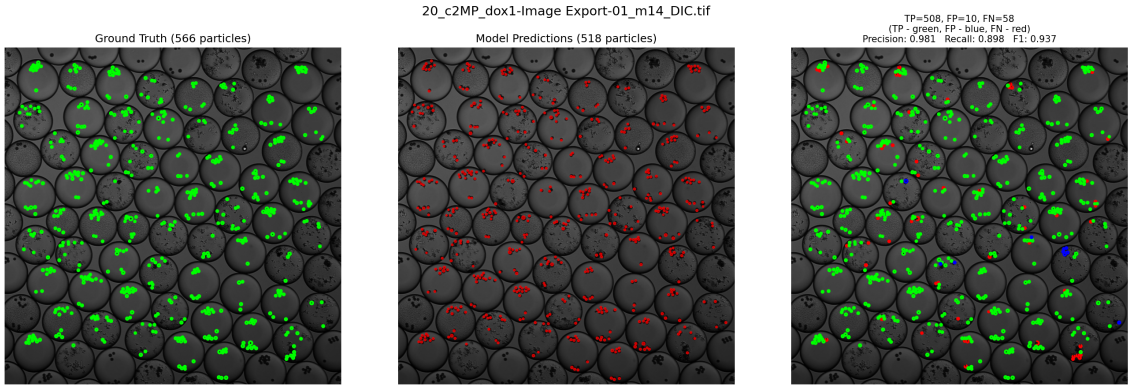
Näitaja	**Väärtus**
Testpiltide arv	17
Mikroplasti osakesed (Ground Truth)	2809
Mikroplasti osakesed (YOLOv8n-seg)	2551
Õiged tuvastused (True Positives)	2510
Valepositiivsed tuvastused (False Positives)	41
Tuvastamata osakesed (False Negatives)	385
Täpsus (Precision)	0,9839 (98,39%)
Saagis (Recall)	0,8936 (89,36%)
Üldine F1-skoor	0,9366

Visuaalne võrdlus (joonis 19) näitab, et valepositiivseid tulemusi ei esinenud proovidel, kus mikroplast puudus, näiteks bakterikolooniatega ja tühjades tilkades.



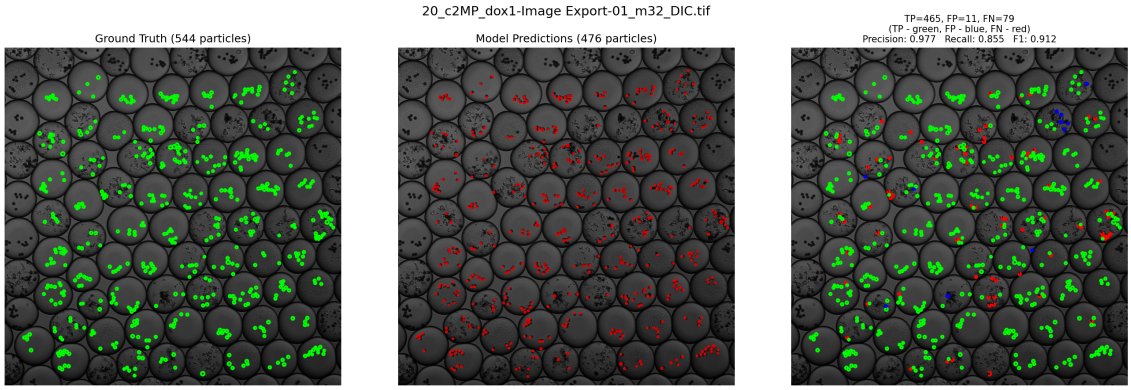
Joonis 19. YOLOv8n-seg mudeli tuvastustulemuste visualiseerimine testpiltidel. Näited ilma mikroplastita.

Teistel juhtudel on valepositiivsed tulemused sageli tingitud sellest, et mudel tunneb mikroplasti ära kohtades, mida inimesed ei pruugi näha või märkamata jätta (joonis 20).



Joonis 20. YOLOv8n-seg mudeli tuvastustulemuste visualiseerimine testpiltidel. Valepositiivsed tulemuste näited.

Puuduvad osakesed on tavaliselt seotud juhtudega, kus nad asuvad serval või kattuvad liiga palju teiste struktuuridega (joonis 21). Muud visuaalsed näited võib leida siit[3].



Joonis 21. YOLOv8n-seg mudeli tuvastustulemuste visualiseerimine testpiltidel. Valenegatiivsed tulemuste näited.

7.3 Tellijate tagasiside ja selle põhjal tehtud täiendused

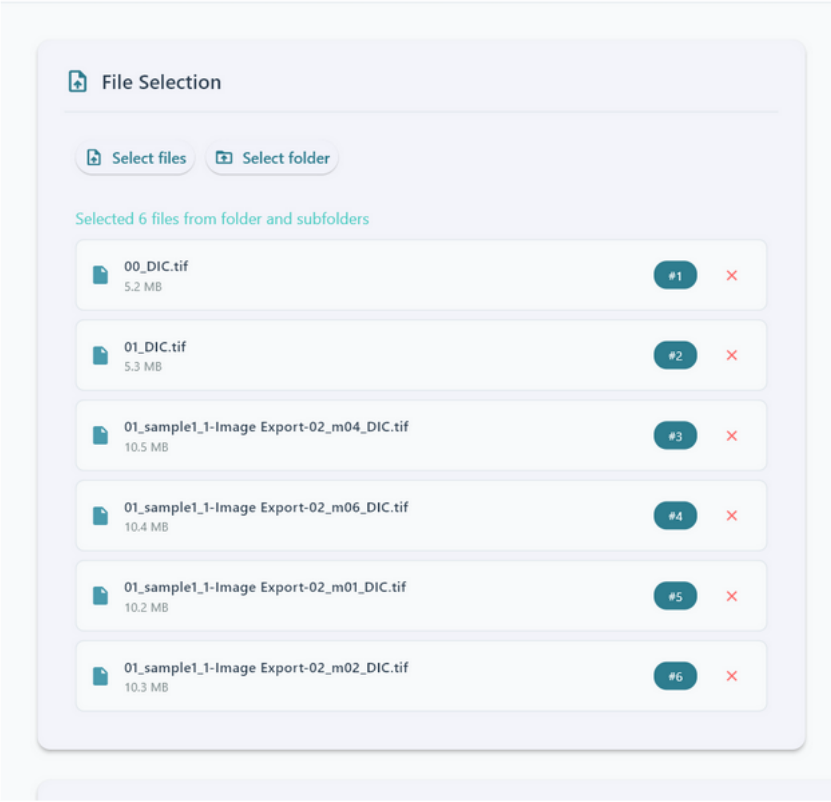
Töölauarakenduse arendus viidi läbi iteratiivsel viisil, mille käigus saadi tellijatelt mitu korda tagasisidet. Pärast kriitiliste kommentaaride hindamist tehti vajalikke täiustusi, et rakendus vastaks paremini kasutajate tegelikele vajadustele.

[3] https://gitlab.cs.ttu.ee/andruv/iaib/-/tree/main/validation/microplastic%20detection/yolo-seg%20model%20validation%20%5Bnew%20approach%5D/validation_results?ref_type=heads

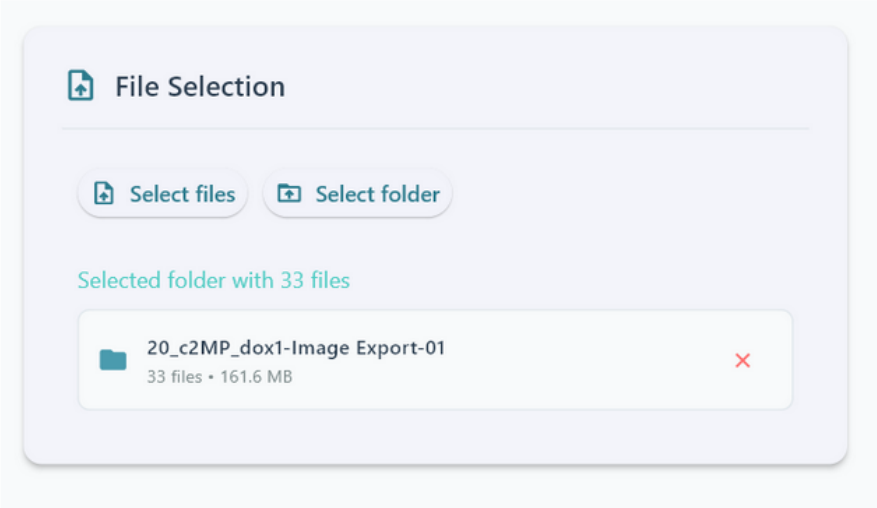
7.3.1 Esimene tagasiside ja tehtud muudatused

Esimese versiooni kohta andis tagasisidet uurimisrühma juht Simona Bartkova. Ta märkis, et kasutajaliides on lihtne ja mugav ning pildianalüüsi kiirus on suurenenud: „*Praeguseks on töötluskiirus märkimisväärselt paranenud, ulatudes ligikaudu 10–15 sekundini pildi kohta, mida võib pidada vastuvõetavaks.*" See näitab, et treenitud YOLOv8n mudel suudab oma ülesandega üldiselt hakkama saada. Varasemas etapis kasutati tilkade segmenteerimiseks Cellpose'i mudelit, kuid pärast kohtumist klientidega jõuti järeldusele, et selline lähenemisviis ei ole sobiv. Ilma graafikaprotsessorita võttis ühe pildi analüüs 10–15 minutit ning enamikul uurimisrühma liikmetel puudus selleks piisav riistvara.

Kausta laadimisel kuvas liides kõik selles olevad pildid, nagu on näidatud joonisel 22. See võib tohutu piltide arvu tõttu olla üsna ebamugav, seega teadlane palus kuvada ainult kausta nime. Parandatud versioon on näidatud joonisel 23.



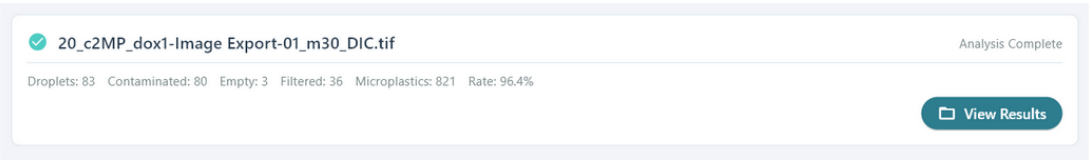
Joonis 22. Valitud kausta vana kuva.



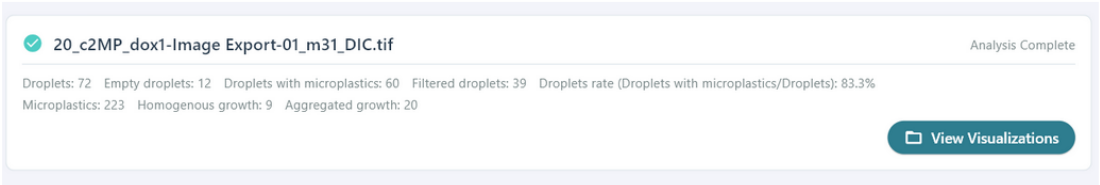
Joonis 23. Valitud kausta uus kuva.

Samuti toodi välja, et terminiga *contaminated* on seotud semantiline viga (vaata joonis 24): „*Contaminated tähendaks, et tilkades on midagi, mis seal ei peaks olema. Palun muutke seda ja selgitage iga tulemuse tähendust. Praegu pole selge, mida te näitate.*"

Tegelikkuses autorid tahtsid viidata sellele, et tilgast leiti baktereid või mikroplasti, kuid see oleks oluliselt seganud kasutajate tulemuste mõistmist, seega see termin eemaldati ja tulemuste väljundit täiendati detailsema teabega (vaata joonis 25).



Joonis 24. Tulemuste ebaselgus.



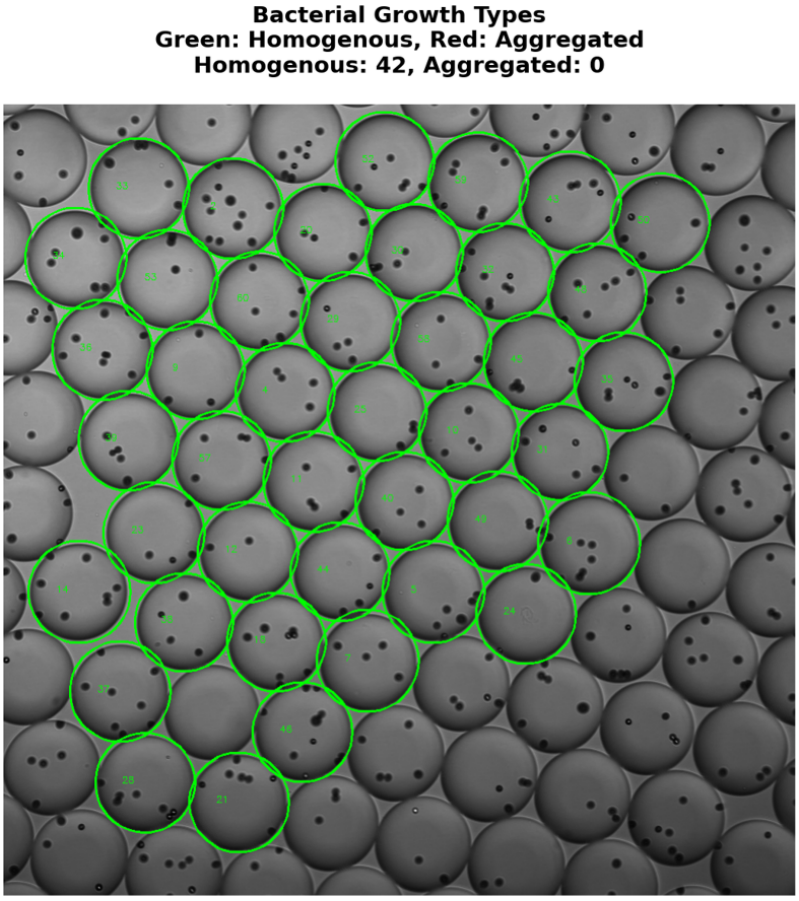
Joonis 25. Tulemused koos üksikasjalikuma teabega.

Üks olulisemaid parandusi tehti selle märkuse põhjal: „*Andmestiku analüüsil ilmneb puudus*

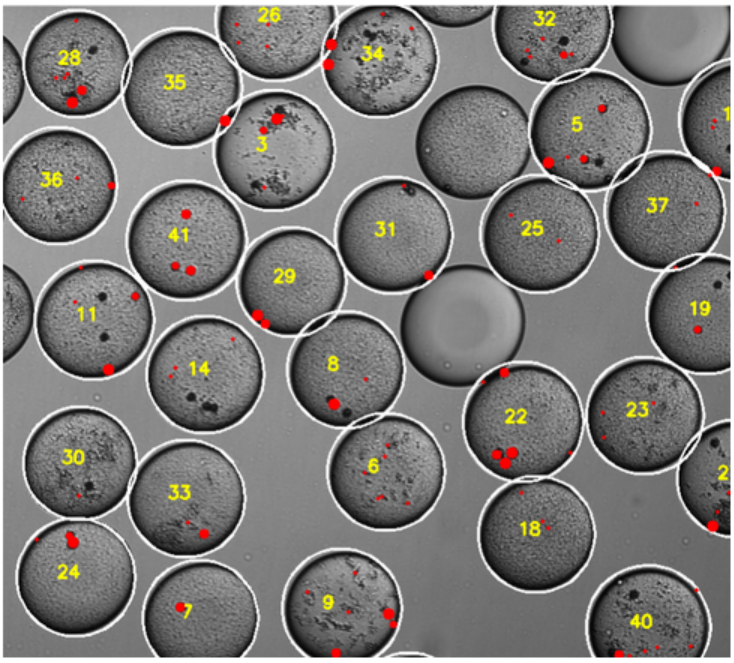
juhtudel, kui pildil pole baktereid. See leiab ekslikult baktereid tühjadest tilkadest. Vaata allpool (joonis 26). Sellel pildil polnud üheski tilgas baktereid, kuid mudel leidis baktereid ja tuvastas bakterid ekslikult enamikus tilkades."

Algselt treeniti CNN-mudelit tuvastama ainult kahte klassi: kasvavat (*Aggregated*) ja homogeenset (*Homogenous*). Enne klassifitseerimismudeli kasutuselevõttu teostati täiendav tekstuurianalüüs, et teha kindlaks, kas tilk sisaldab baktereid. Selle käigus filtreeriti välja mikroplast ja müra, mis mõjutasid tulemusi. See lähenemisviis osutus aga ebaefektiivseks, mistõttu otsustati lisasammud eemaldada ja mudel ümber treenida, lisades kolmanda klassi (*No_Bacteria*). See muudatus kõrvaldas valepositiivsed tulemused.

Esines väiksemaid probleeme paljude väikeste osakestega, mis ei ole mikroplast, kuid mida loeti mikroplastiks (joonis 27). See lahendati rangemate parameetrite kasutamisega.



Joonis 26. Mudeli väljund bakteriteta pildil.



Joonis 27. Valepositiivne mikroplasti tuvastamine.

7.3.2 Teine tagasiside ja tehtud muudatused

Teise versiooni kohta andsid tagasisidet kuus uurimisrühma liiget, kelle hinnangute koondülevaade on esitatud tabelis 8.

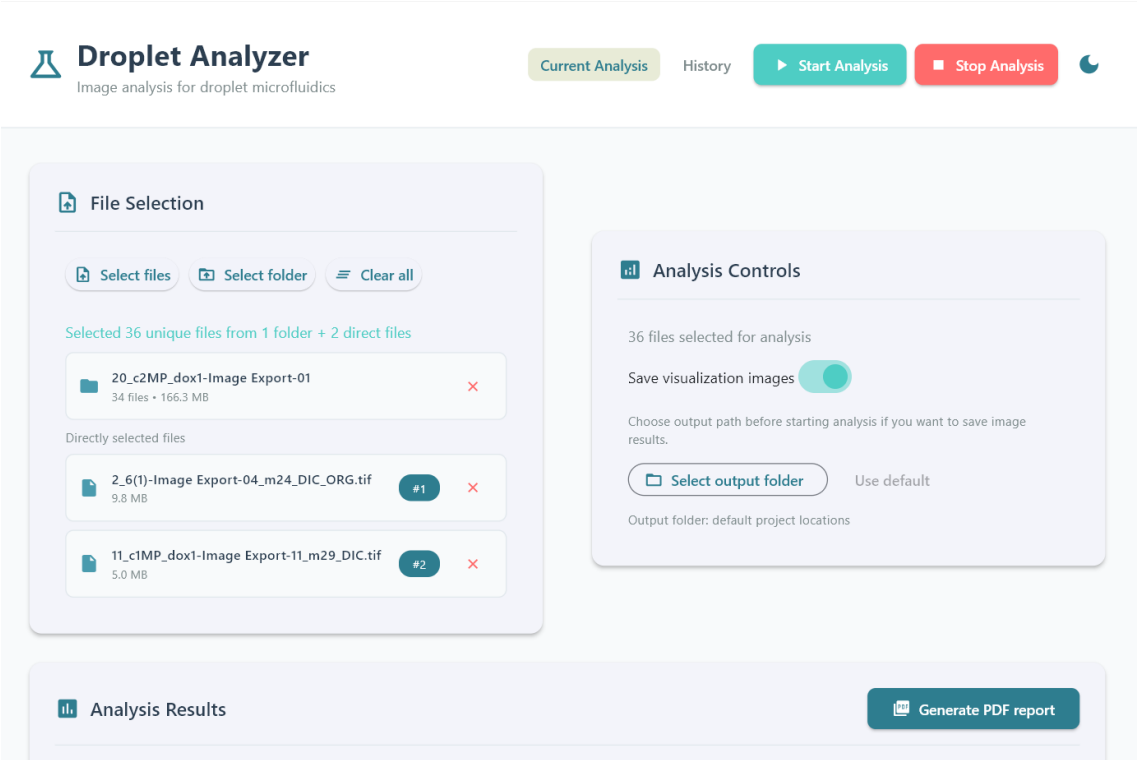
Tabel 8. Tagasiside teise versiooni kohta.

Kasutaja	**Tagasiside**
1	Lisada nupp praeguse analüüsi peatamiseks. Kui kasutaja avastab, et on midagi unustanud, ta saab analüüsi peatada ilma kogu analüüsi valmimist ootamata.
2	Kasutaja ei pakkunud midagi, kuid töölauarakendus töötab tema jaoks.
3	Töötles mikroplastiga ja ilma mikroplastita pilte eraldi ning märkas, et mikroplastiga pilte töödeldi kiiremini.
4	Sooviks valida kausta, kuhu programm oma tulemused salvestab.
5	Analüüsi lõppedes oleks väga kasulik heliteade või hüpikaken. See võimaldaks kasutajal teiste ülesannetega tegeleda ilma pidevalt kontrollimata, kas analüüs on valmis. Samuti soovitas lisada nupu „Stop"või „Cancel analysis"juhuks, kui midagi läheb valesti või valitakse vale andmekogum. Lisada võimalus analüüsida mitut kausta korraga, et töötada mitme katsega. Tulemused ei värskendata ega kirjutata üle.
6	Rakendus ei käivitunud macOS keskkonnas.

Teise tagasiside põhjal kujundati põhivaade ümber, võimaldades kasutajal lisada analüüsi samaaegselt mitu kausta ning üksikuid faile. Kattuvad failid, mis kuuluvad juba valitud kaustadesse, jäetakse analüüsist automaatselt välja. Lisaks sellele on võimalus valida kaust, kuhu programm tulemused salvestab. Kasutaja samuti märkas, et tulemused ei värskendata ega kirjutata üle. Selle probleemi lahendamiseks on rakenduses kolm salvestusrežiimi, mis on hetkel saadaval lähtekoodi muutmise või keskkonnamuutujate kaudu:

- *refresh* režiim tähendab piltide värskendamist. See kustutab esmalt kaustast kõik olemasolevad pildid ja salvestab seejärel alati viimase versiooni käivitatud failidest;
- *versioned* režiim tähendab uue versiooni loomist, mida loob iga käivitamise jaoks uue ajatempliga alamkausta;
- *overwrite* režiim tähendab Ülekirjutamist. See ei kustuta esmalt midagi, vaid salvestab lihtsalt uued failid, kirjutades üle kõik samanimelised failid.

Lisaks saab kasutaja analüüsi igal hetkel katkestada ning analüüsi lõppedes kuvatakse hüpikaken koos operatsioonisüsteemi teatega. Põhivaade pärast kõiki muudatusi on kujutatud joonisel 28.



Joonis 28. Põhivaade peale teist tagasisidet.

8 Edasiarendus

Käesolev töö keskendus DIC-mikroskoopia halltoonpiltidele. Uurimisrühm eelistab seda kanalit selle kasutusmugavuse tõttu, kuigi madal kontrast, müra ja ebakorrapärase tilgad muudavad analüüsi keerukaks. Loodud lahendus on optimeeritud just DIC-piltidele, kuid edaspidi on võimalik süsteemi analüüsiloogikat laiendada ka teistele laboris kasutatavatele kanalitele, näiteks fluorestsentsikanalid nagu WTGFP (*Green Fluorescent Protein*) ja AF647 (*Red Fluorescence Protein*). Fluorestsentsmikroskoopia pakub suurepärast kontrasti (hele objekt mustal taustal), kuid selle rakendamine on keerulisem. Tõenäoliselt on olemasolevaid masinõppe mudeleid võimalik kohandada uute andmetega ümberõppe abil. Täpset teostatavust ja vajalikke muudatusi saab hinnata ainult tegelike andmete peal testimisega. Lisaks võib tekkida vajadus kohandada eeltöötlust ja parameetreid uue kanali piltide järgi. Süsteemi praktilist väärtust saaks tõsta interaktiivsete funktsioonide lisamisega, mis võimaldaks kasutajal mudeli vigu käsitsi parandada, näiteks valesti klassifitseeritud bakterite kasvu tüüp või mikroplasti osakese lisamine ja eemaldamine. Need parandused salvestatakse andmebaasi, et tagada tulemuste täpsus ilma kohese mudeli ümberõppeta. Parandusi saaks kasutada ka mudelite perioodiliseks täiendõppeks, muutes süsteemi ajas täpsemaks ja paindlikumaks erinevate katsetingimuste suhtes. Kui töölauarakendus ühendada pilvepõhise teenusega, oleks võimalik koguda kasutajate parandusi tsentraalselt ja kasutada neid mudelite perioodiliseks täiendõppeks. See võimaldaks mudelitel õppida paljude kasutajate kogemustest. Tulevikus saaks rakendust täiendada reaalajas analüüsiga. Rakendus võtaks otse mikroskoobiga ühendatud kaamerast reaalajas pildivoo, töötleks iga kaadrit automaatselt ja kuvaks tulemused (tuvastatud tilgad, mikroplasti osakesed, bakterite kasvu tüübid) koheselt ekraanil. See võimaldab teadlastel tulemusi katseajal otse jälgida, ootamata hilisemat täielikku analüüsi.

9 Kokkuvõte

Käesoleva bakalaureusetöö eesmärk oli arendada masinõppel põhinev pildianalüüsi rakendus tilkade mikrofluidika eksperimentide automatiseerimiseks. Töö tulemusena valmis modulaarne töölauarakendus, mis saab tuvastada mikroskoobi piltidel tilgad, klassifitseerida bakterite kasvutüübid (homogeenne, agregeeritud, puudub) ning tuvastada mikroplasti osakesed. Lisaks sellele on rakendus optimeeritud halltoonides diferentsiaalinterferentsi mikroskoopia DIC-piltide jaoks.

Käesoleva töö raames viidi läbi olemasolevate masinõppe mudelite (tilkade tuvastamiseks ja bakterite kasvu klassifitseerimiseks) põhjalik valideerimine reaalsetel katseandmetel. Tulemused kinnitasid mudelite efektiivsust: tilkade tuvastamise täpsus (*precision*) oli 99,5% ja saagis (*recall*) 99,2%, bakterite klassifitseerimise üldine täpsus oli 98,1%.

Mikroplasti tuvastamise osas täiendati olemasolevat klassikalist algoritmi tekstuuri filtreerimise ja radiaalprofiili analüüsiga, mis parandas tundlikkust (*recall* tõusis 42,2%-lt 56,9%-le). Siiski ei võimaldanud see lähenemisviis täielikult ületada piiranguid keerukamate juhtumite korral. Seetõttu töötati välja *YOLOv8n-seg* mudelil põhinev lahendus, mis saavutas testandmestikul täpsuseks 98,4% ja tundlikkuseks 89,4%.

Loodud rakendus võimaldab kasutajal analüüsida üksikfaile või terveid kaustu, visualiseerib tulemused nelja pildina, ekspordib need Exceli ja PDF-vormingusse ning toetab analüüsiajaloo salvestamist ja sessioonide võrdlemist. Tuginedes lõppkasutajate tagasisidele, optimeeriti rakenduse kasutajaliidest. Need täiendused tagavad, et loodud tööriist vastab uurimisrühma vajadustele ning on valmis rakendamiseks reaalsetes töövoogudes. Rakenduse peamiste kasutajaliidese vaadete visuaalsed näited on toodud lisas 4.1 (peavaade enne analüüsi), 4.2 (peavaade pärast analüüsi), 4.3 (ajaloo vaade), 4.4 (kokkuvõtvate mõõdikute tabel sessioonide võrdlemise vaates), 4.5 (sektordiagrammid sessioonide võrdlemise vaates).

Rakenduse lähtekoodiga saab tutvuda repositooriumis[1].

Vaatamata sellele, et püstitatud eesmärgid said täidetud, on rakendusel veel võimalusi edasiarenduseks. Täpsem ülevaade nendest on toodud peatükis 8.

[1] `https://gitlab.cs.ttu.ee/andruv/iaib`

Kasutatud kirjandus

[1] Immanuel Sanka *et al.* „Investigation of Different Free Image Analysis Software for High-Throughput Droplet Detection“. *ACS Omega* 6.35 (2021), lk. 22625–22634. DOI: 10.1021/acsomega.1c02664. URL: https://pubs.acs.org/doi/full/10.1021/acsomega.1c02664 (vaadatud 13.03.2026).

[2] D. Kácsor *et al.* „Label-free droplet image analysis with CellProfiler“. *bioRxiv preprint* (2025). URL: https://www.biorxiv.org/content/10.1101/2025.04.04.647160v1 (vaadatud 04.03.2026).

[3] C. Stringer *et al.* „Cellpose: a generalist algorithm for cellular segmentation“. *Nature Methods* 18 (2021), lk. 100–106. URL: https://doi.org/10.1038/s41592-020-01018-x (vaadatud 04.03.2026).

[4] S. Berg *et al.* „ilastik: interactive machine learning for (bio)image analysis“. *Nature Methods* 16 (2019), lk. 1226–1232. URL: https://www.nature.com/articles/s41592-019-0582-9 (vaadatud 09.03.2026).

[5] J. Schindelin *et al.* „Fiji: an open-source platform for biological-image analysis“. *Nature Methods* 9 (2012), lk. 676–682. URL: https://www.nature.com/articles/nmeth.2019 (vaadatud 15.03.2026).

[6] MathWorks. *imfindcircles: Find circles using circular Hough transform*. 2025. URL: https://se.mathworks.com/help/images/ref/imfindcircles.html (vaadatud 15.03.2026).

[7] Perforce Software. *Non-Functional Requirements: Tips, Tools, and Examples*. 2025. URL: https://www.perforce.com/blog/alm/what-are-non-functional-requirements-examples (vaadatud 08.04.2026).

[8] Labelformat. *YOLOv8 Object Detection Format*. 2025. URL: https://labelformat.com/formats/object-detection/yolov8/ (vaadatud 17.04.2026).

[9] Joseph Redmon *et al.* „You Only Look Once: Unified, Real-Time Object Detection“. Teoses: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016. URL: https://openaccess.thecvf.com/content_cvpr_2016/html/Redmon_You_Only_Look_CVPR_2016_paper.html (vaadatud 17.04.2026).

[10] Ultralytics. *YOLO Data Augmentation*. 2025. URL: https://docs.ultralytics.com/guides/yolo-data-augmentation/#using-a-configuration-file (vaadatud 13.04.2026).

[11] Leslie N. Smith. „Cyclical Learning Rates for Training Neural Networks“. Teoses: *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2017. URL: https://arxiv.org/abs/1506.01186 (vaadatud 25.04.2026).

[12] Ultralytics. *Model Training Tips*. 2025. URL: https://docs.ultralytics.com/guides/model-training-tips/ (vaadatud 15.04.2026).

[13] Ultralytics. *Mean Average Precision (mAP)*. 2026. URL: https://www.ultralytics.com/ru/glossary/mean-average-precision-map (vaadatud 15.04.2026).

[14] Ultralytics. *Precision in Machine Learning*. 2026. URL: https://www.ultralytics.com/ru/glossary/precision (vaadatud 15.04.2026).

[15] Ultralytics. *Recall in Machine Learning*. 2026. URL: https://www.ultralytics.com/glossary/recall (vaadatud 15.04.2026).

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks[2]

Meie, Andrus Vaher ja Anna Gulova-Em

1. Anname Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "Masinõppel põhineva pildianalüüsi tööriista loomine mikrofluidika tilkade uurimiseks", mille juhendaja on Evelin Halling
    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. Oleme teadlikud, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autoritele.
3. Kinnitame, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

01.06.2026

[2]Lihtlitsents 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 loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 – Konvolutsioonilise närvivõrgu arhitektuur bakterite kasvu tüübi klassifitseerimise jaoks

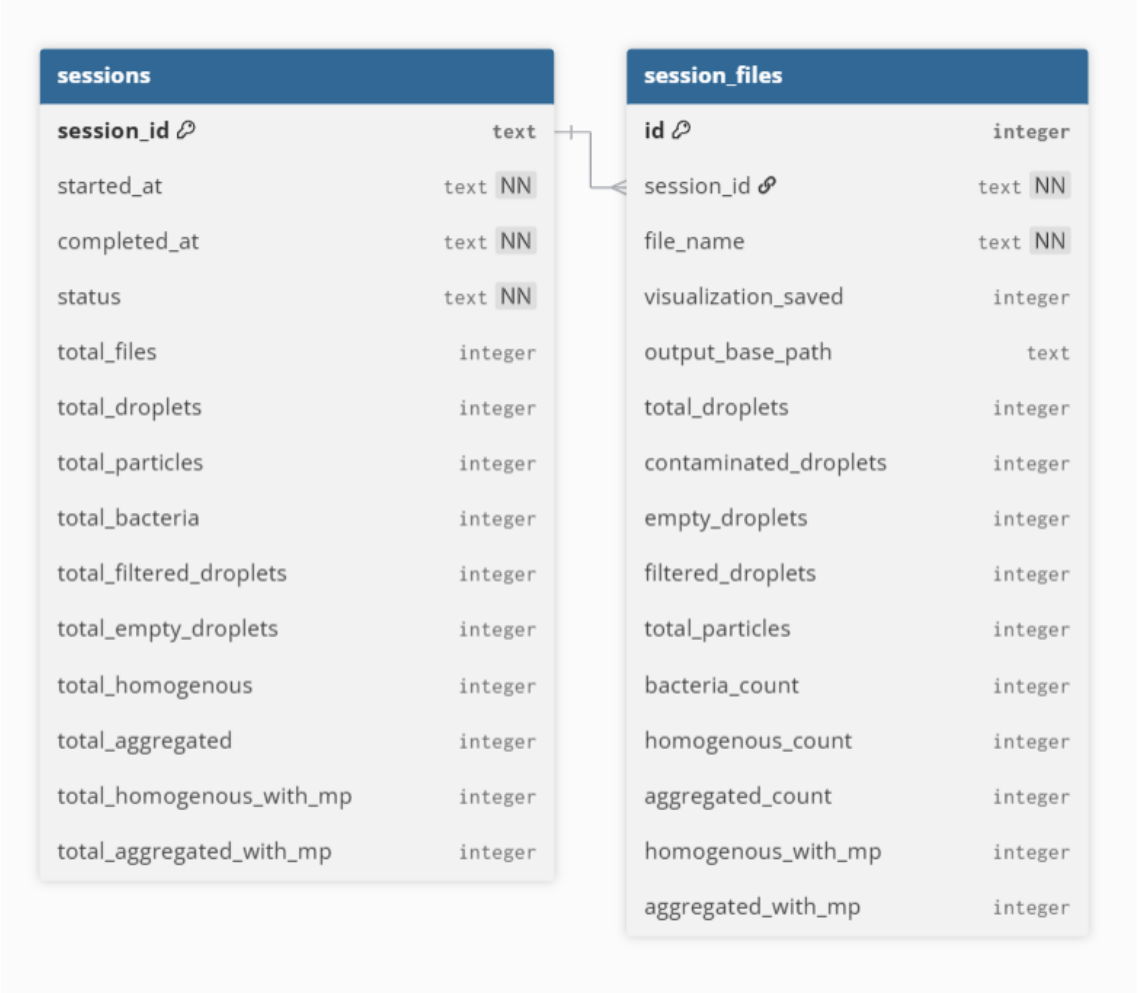
Bakterite kasvu tüübi klassifitseerimiseks arendatud CNN-i lähtekood:

```python
class BacterialGrowthCNN(nn.Module):
    """CNN model for classifying bacterial growth type (Homogenous, Aggregated, No_Bacteria)"""

    def __init__(self, num_classes=3):
        super(BacterialGrowthCNN, self).__init__()
        self.features = nn.Sequential(
            # Basic features
            nn.Conv2d(1, 32, 3, padding=1),
            nn.BatchNorm2d(32),
            nn.ReLU(inplace=True),
            nn.MaxPool2d(2),

            # Average features
            nn.Conv2d(32, 64, 3, padding=1),
            nn.BatchNorm2d(64),
            nn.ReLU(inplace=True),
            nn.MaxPool2d(2),

            # Complex features
            nn.Conv2d(64, 128, 3, padding=1),
            nn.BatchNorm2d(128),
            nn.ReLU(inplace=True),
            nn.MaxPool2d(2),

            # Global features
            nn.Conv2d(128, 256, 3, padding=1),
            nn.BatchNorm2d(256),
            nn.ReLU(inplace=True),
            nn.AdaptiveAvgPool2d((4, 4))
        )

        self.classifier = nn.Sequential(
            nn.Dropout(0.5),
            nn.Linear(256 * 4 * 4, 512),
            nn.ReLU(inplace=True),
            nn.Dropout(0.3),
            nn.Linear(512, 128),
            nn.ReLU(inplace=True),
            nn.Linear(128, num_classes)
        )

    def forward(self, x):
        x = self.features(x)
        x = x.view(x.size(0), -1)
        x = self.classifier(x)
        return x
```

Joonis 29. Konvolutsiooniline närvivõrk PyTorch raamistikus.

Lisa 3 – Andmebaasi skeem

Andmebaasi struktuur koosneb kahest seotud tabelist, kus on määratletud analüüsisessioonide andmeväljad ja nende andmetüübid:

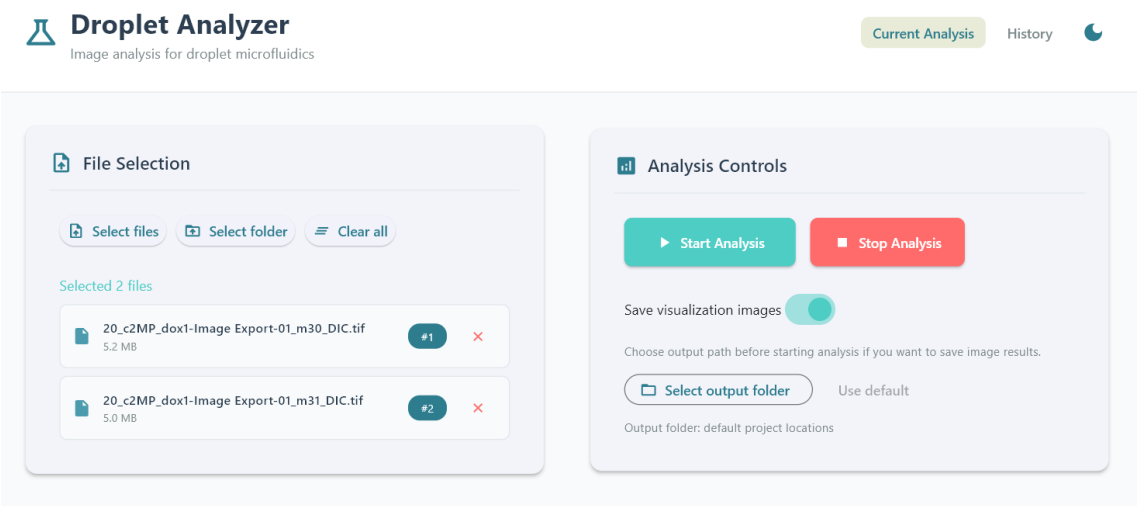


Joonis 30. Andmebaasi skeem: sessions ja session_files tabelid.

Lisa 4 – Rakenduse kasutajaliidese põhivaated

4.1 Rakenduse peavaade enne analüüsi

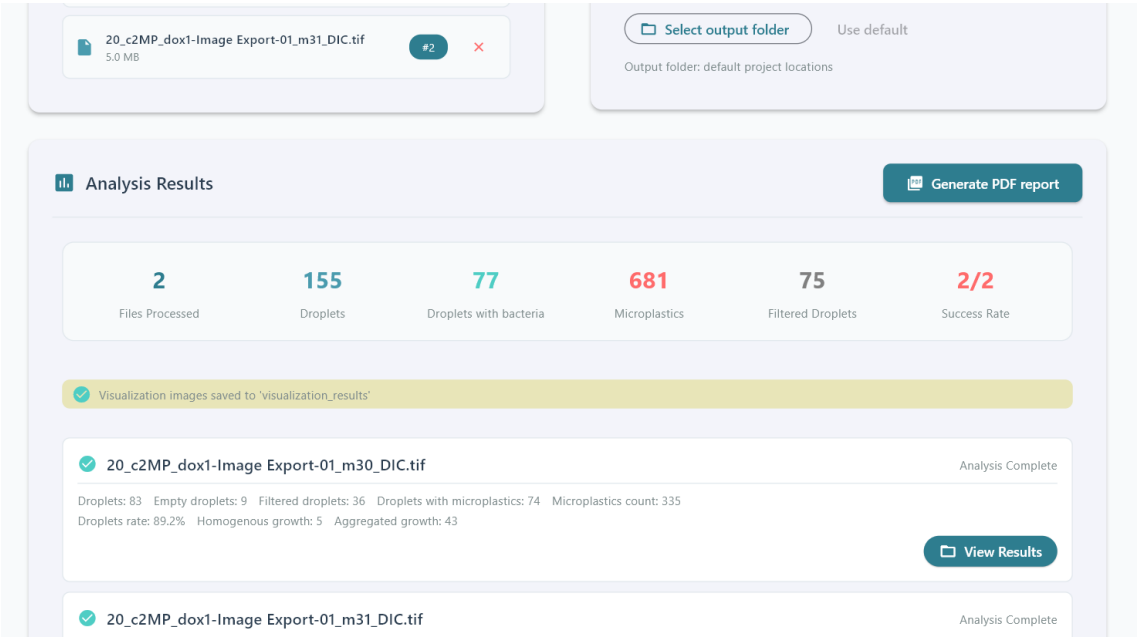
Rakenduse peavaade enne analüüsi koos failide valimise paneeli ja analüüsi juhtnuppudega:



Joonis 31. Peavaade enne analüüsi: failide valik ja analüüsi juhtnupud.

4.2 Rakenduse peavaade pärast analüüsi

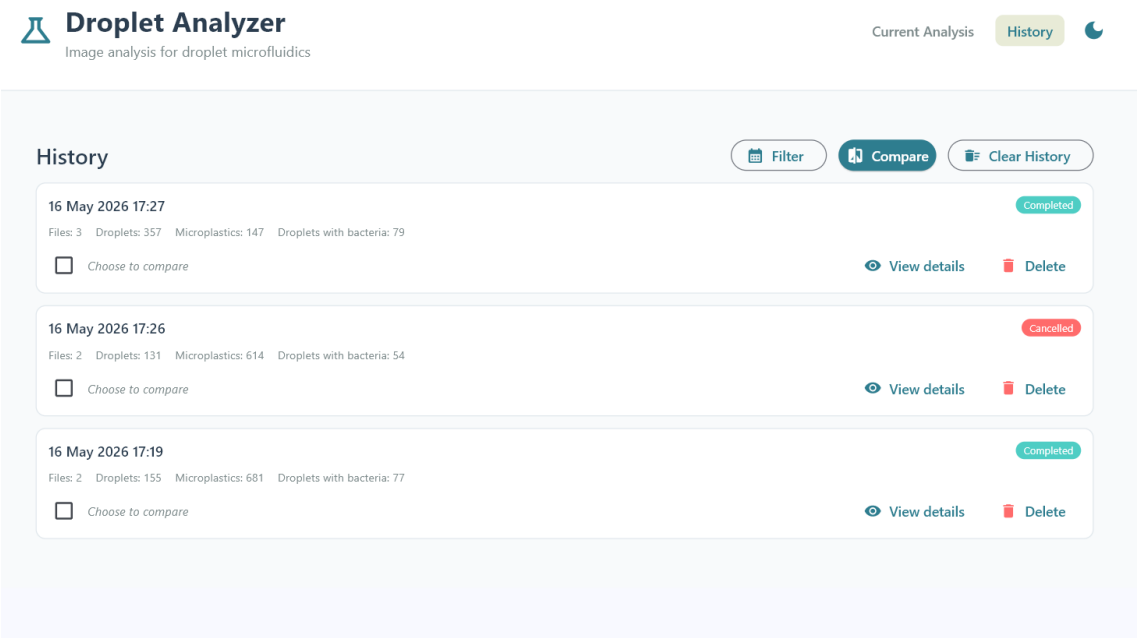
Analüüsijärgne peavaade koondstatistika, failipõhiste tulemuste ja PDF-raporti genereerimise võimalusega:



Joonis 32. Peavaade pärast analüüsi: analüüsitulemused ja PDF-raporti genereerimise nupp.

4.3 Rakenduse ajaloo vaade

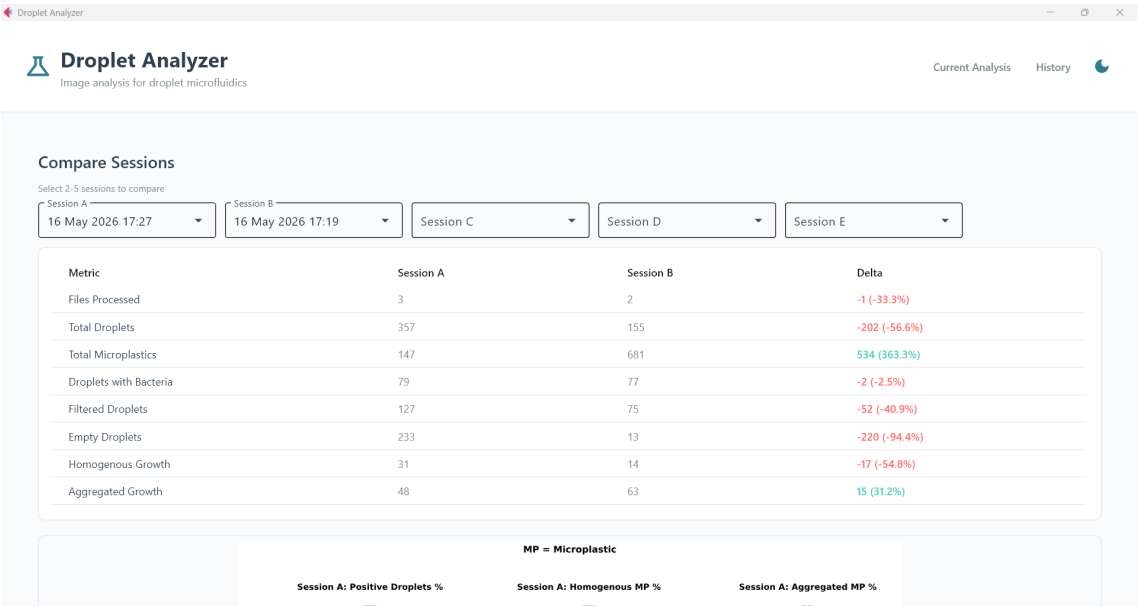
Rakenduse ajaloo vaade varasemate analüüsisessioonide loendi ning filtreerimis-, kustutamis- ja võrdlemisfunktsioonidega:



Joonis 33. Ajaloo vaade: varasemate sessioonide loend, filtreerimine ja võrdlemise võimalus.

4.4 Rakenduse sessioonide võrdlemise vaade: kokkuvõtvate mõõdikute tabel

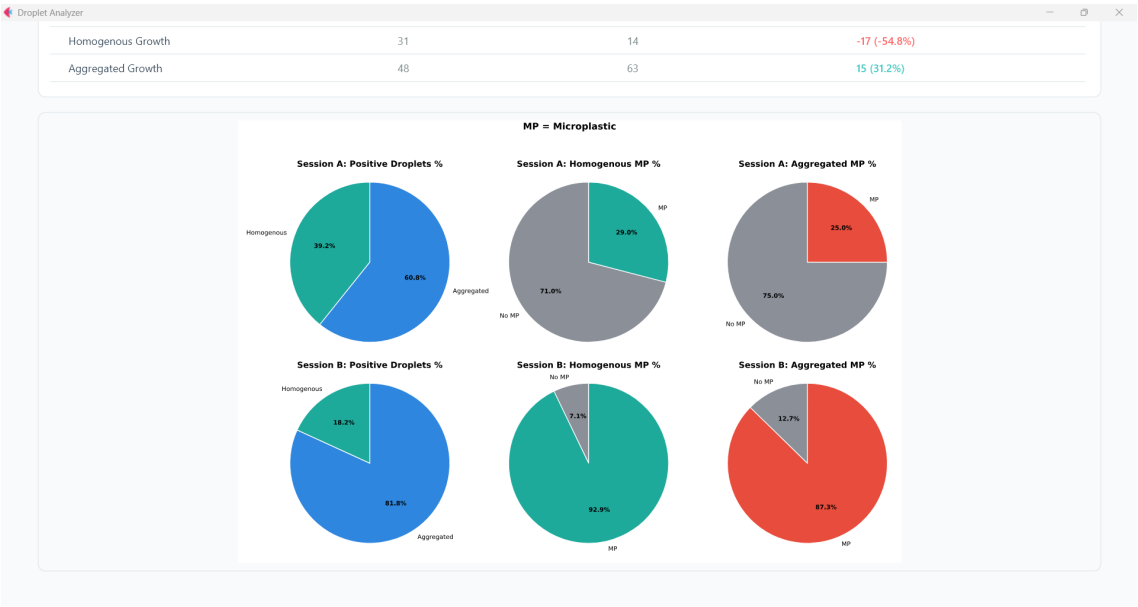
Sessioonide võrdlemise vaade valitud sessioonide statistiliste näitajate ja suhteliste muutuste (delta) arvutusega:



Joonis 34. Sessioonide võrdlemise vaade: kokkuvõtvate mõõdikute tabel.

4.5 Rakenduse sessioonide võrdlemise vaade: sektordiagrammid

Sessioonide visuaalne võrdlus sektordiagrammidena, mis illustreerivad mikroplasti ja bakterite kasvu jaotust võrreldavates andmekogumites:



Joonis 35. Sessioonide võrdlemise vaade: sektordiagrammid.