TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Marion Claudia Striž 206260IADB

Generatiivsete meetodite kasutamine masinõppe mudeli treeningandmete koguse suurendamiseks juuksejuure mikrofotode näitel

Bakalaureusetöö

Juhendaja: Toomas Lepikult
PhD

Tallinn 2024

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: Marion Claudia Strižˇ

04.01.2024

Annotatsioon

Lõputöös käsitletakse treeningandmete koguse ja tasakaalustatuse probleemi masinõppe mudelite loomisel, kui andmete kogumine on keeruline või aeglane.

Üheks võimalikuks lahenduseks probleemile on kasutusele võtta generatiivsed meetodid. See hõlmab nii piltide klassikalisi transformatsioone nagu pööramine, peegeldamine, objekti liigutamine, kui ka masinõppe generatiivseid mudeleid.

Lõputöö eesmärgiks on analüüsida, kas ja kuidas generatiivsed meetodid parandavad konkreetse masinõppe mudeli täpsust, milleks on mikrofotodelt biomarkerite tuvastamise mudel. Seejuures tehakse andmetega eeltööd, seadistatakse märgenduskeskkond, teostatakse piltide transformatsioone ning luuakse generatiivne vastandvõrk.

Töö käigus selgub, et generatiivsete meetodite kasutamine treeningandmestiku koguse suurendamiseks võib olla tõhusaks viisiks, kuidas masinõppe mudeli täpsust suurendada.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 34 leheküljel, 8 peatükki, 12 joonist, 3 tabelit.

Abstract

Usage of Generative Methods to Increase the Quantity of Machine Learning Training Data by Example of Hair Root Micrographs

The thesis aims to study the problem of quantity and balance of training data in the development of machine learning models, in cases where data collection in acceptable quantities is difficult or slow.

A possible solution to this problem is to utilize generative methods. This includes both typical image transformations such as rotation, mirroring and translation, as well as generative machine learning models.

The aim of the thesis is to analyze whether generative methods improve the accuracy of a specific machine learning model – an object detection model aimed at detecting biomarkers from hair root micrographs. In doing so, preliminary work is performed on the data, a labelling environment is set up, classical image transformations are performed and a generative adversarial network is trained.

The findings show that using generative methods to increase the quantity of training data can indeed be an effective way to improve the accuracy of a machine learning model.

The thesis is in Estonian and contains 34 pages of text, 8 chapters, 12 figures, 3 tables.

Lühendite ja mõistete sõnastik

Atribuut	Andmetele omastatav tunnus
AWS	*Amazon Web Services.* Amazoni pilvetaristuteenus
Biomarker	Tervisenäitaja füüsiline vorm juuksejuure mikrofotol
Boto3	Pythoni teek, mida kasutatakse AWS ressursihalduseks
cGAN	*Conditional GAN.* GAN, mis võimaldab genereerida etteantud tingimusele vastavaid andmeid
CSV	Tekstifailivorming, mis kasutab väärtuste eraldamiseks komasid
Docker	Konteineriseerimisteenus
EC2	*Elastic Compute Cloud.* Virtuaalarvutite rentimise teenus
Elastic Beanstalk	AWS veebirakenduste orkestreerimisteenus
ETag	*Entity Tag*. S3 olemi unikaalne MD5 räsi
GAN	*Generative Adversarial Network.* Generatiivne vastandvõrk, üks levinumatest generatiivsetest sügavõppemudelitest
Generatsioon	Loomine, tekitamine
IaC	*Infrastructure-as-Code.* Taristu-kui-kood
IAM	*Identity and Access Management.* AWS identiteedi- ja juurdepääsuhaldussüsteem.
ID	Identifikaator
ImageAI	Pythoni teek, võimaldab luua YOLOv3 objektituvastusmudelit
imgaug	Pythoni teek, võimaldab märgendatud andmeid transformeerida
IoU	*Intersection over union.* Objektituvastusmudeli analüüsil märgenduskastide ühisosa suhe ühendist
json	Andmevahetusvorming, mis põhineb JavaScriptil
Label Studio	Andmemärgendustarkvara
Masinõpe	Masina võime teha järeldusi, leida mustreid jm. ilma etteantud reegliteta
mAP	*Mean average precision*. Keskmine täpsus, objektituvastus-mudelite täpsuse mõõtmise ühik
MATLAB	Andmetöötluskeskkond
MD5	*Message-digest algorithm*. Levinud räsifunktsioon

Mode collapse	Fenomen, kus GAN generaator õpib tootma ainult ühte tüüpi andmeid
NoSQL	Termin omane mitte-relatsioonilistele andmebaasidele
PostgreSQL	Relatsiooniliste andmebaaside haldamise süsteem
Python	Programmeerimiskeel, mis on laialdaselt kasutusel masinõppe rakendustes
PyTorch	Pythoni masinõppe teek
RDS	*Relational Database Service.* AWS relatsiooniliste andmebaaside teenus.
ReLU	*Rectified Linear Unit.* Masinõppes alaldi aktiveerimisfunktsioon
S3	*Simple Storage Service*. AWS hoidla
SQL	*Structured Query Language*. Struktuurpäringukeel
Tensorflow	Pythoni masinõppe teek
Treeningandmed	Andmed, mida kasutatakse masinõppe mudeli treenimiseks
VAE	*Variational autoencoder.* Variatsiooniline autokooder, üks generatiivsetest masinõppemudelitest
VPC	*Virtual Private Cloud*. AWS teenuste omavaheliste ühenduste orkestreerija
Ülesproovimine	*Upsampling.* Protsess, mille käigus suurendatakse andmete mõõtmeid
XML	*Extensible Markup Language.* Üldotstarbeline märgistuskeel
YOLO	*You Only Look Once.* Objektituvastusalgoritm

Sisukord

Jooniste loetelu

Tabelite loetelu

1 Sissejuhatus

Masinõppe mudelite treenimisel on olulisemateks faktoriteks treeningandmete kogus, kvaliteet ja tasakaalustatus. Esineb olukordi, kus reaalsete andmete kogumine soovitud aja jooksul on raskendatud või lausa võimatu. Teisalt, isegi kui andmeid on piisavas koguses, võivad need katta vaid väikest osa kõikvõimalikest juhtudest, mispuhul mudeli reaalne täpsus jääb oluliselt madalamaks treenimisel saavutatud täpsusest.

Probleemi lahendamise üheks võimalikuks lähenemiseks on kasutusele võtta generatiivsed meetodid. See tähendab olemasoleva andmestiku põhjal uute andmete loomist, olgu selleks olemasolevate piltide muutmine transformeerimise kaudu või generatiivsete masinõppemudelite kasutamine.

Juuksejuure mikrofotodelt nähtavate biomarkerite põhjal on võimalik tuletada olulist teavet inimese tervise kohta. Protsessi automatiseerimiseks saab luua masinõppel põhineva süsteemi, mis tuvastaks pildil esinevad biomarkerid ehk tervisenäitajate füüsilised omadused ning koostaks suurema arvu piltide kokkuvõttes raporti inimese tervise kohta. Kirjeldatud mudel vajab aga suurtes kogustes treeningandmeid, mis kataksid piisaval määral kõikvõimalikke markereid ning nende mitmekülgseid esinemisviise.

Käesoleva lõputöö eesmärgiks on uurida võimalusi juuksejuure biomarkerite tuvastusmudeli jaoks treeningandmete juurde genereerimiseks, valida probleemi raames asjakohasemad, teostada need ning analüüsida saavutatud tulemusi.

Eeldatud on algsete juuksejuure mikrofotode olemasolu ja autoril neile juurdepääs. Lisaks on tingimuseks AWS (*Amazon Web Services*) pilveteenuste kasutusvõimalus märgendamisteenuse ettevalmistamiseks. Andmete märgendamiseks on vaja juuksejuure spetsialiste, kes oskaksid ja tahaksid abistada märgendustööga.

Töö on jaotatud teoreetiliseks ning praktiliseks osaks. Analüüsi käigus kirjeldatakse olemasolevaid generatiivseid meetodeid, põhjendatakse nende asjakohasust või selle puudumist treeningandmete koguse suurendamise vaatest ning tehakse valikud meetodite

teostuse osas käesoleva eesmärgi lahendamiseks. Kirjeldatakse olemasolevaid andmeid, mis on aluseks genereeritavatele.

Praktiline osa kirjeldab toorete piltandmete märgendamiseks kasutatava keskkonna ülesseadmist, analüüsi käigus valitud generatiivsete meetodite teostust ning saavutatud vahe- ja lõpptulemuste kirjeldust. Lisaks analüüsitakse töö käigus tekkinud probleeme ja takistusi, nende võimalikke lahendusi ja meetodite edasiarendusi vaadeldava projekti raames.

2 Probleemi analüüs

Masinõpe on toiminud katalüsaatorina väga paljude erinevate tööstusharudega seonduvate protsesside automatiseerimisel. Meditsiinitööstuses on leidnud kasutust näiteks piltide klassifitseerimis- ja tuvastusmudelid, eriti süvaõppemudelid, mis on komplekssed kunstlikest ajurakkudest koosnevad närvivõrgud. Neid on muuhulgas treenitud mikrofotodelt kõrge täpsusega keerulisi diagnoose määrama. Näiteks on diabeetilist retinopaatiat määrav mudel, mis on näidanud inimestest oftalmoloogidega võrdset või madalamat eksimismäära [1].

Kuigi sellised mudelid saavutavad teatud tingimustel inimese vahelesegamiseta väga häid tulemusi ja on suureks abiks protsesside kiirendamisel, on nende treenimiseks kasutatavatel andmetel määrav tähtsus. Sageli on vaja sadu tuhandeid kvaliteetseid ja mitmekülgseid treeningeksemplare, et mudel õpiks eesmärgile vastavaid olulisi nüansse piltidelt järjepidevalt tuvastama [2].

Andmete kogumise probleem on süvaõppes üheks suurimaks pudelikaelaks. Esiteks on esile tõusnud valdkonnad, kus kvaliteetsete andmete kogumine piisavas koguses on keeruline ja aeganõudev. Teiseks on süvaõppemudelid, vastupidiselt traditsioonilistele masinõppemudelitele, võimelised atribuutide ehk andmeid kirjeldavate tunnuste valikut tegema automaatselt, mis vajab aga ka rohkemat arvu näiteid lähteandmete kujul [3].

Järgnevalt analüüsitakse treeningandmete koguse, kvaliteedi ja tasakaalustatuse probleemi masinõppe mudelite loomisel, kuidas generatiivsed meetodid võivad olla seejuures abiks ning seatakse lõputöö skoop ja tingimused.

2.1 Andmete kogus

Andmete kogumise keerukus tõstatab vajaduse leida minimaalne treeningandmete hulk, et saavutada masinõppemudeli soovitud täpsus. Konkreetsete probleemide nüansid, nagu probleemi keerukus, nõuded mudeli täpsusele ning andmetes esinev müra, muudavad aga

üldistuse pea võimatuks, sest need võivad märkimisväärselt mõjutada andmete vajalikku kogust [4].

Siiski on üldlevinud arvamus, et komplekssete mudelite puhul mida rohkem on treeningandmeid, seda efektiivsem mudel. Suured ja mitmekesised andmekogumid on masinõppes tihti väärtuslikumad, kui keerukad algoritmid [5]. Näiteks histoloogilistelt mikrofotodelt kudede määramise mudeli treeningandmete arvu 684-lt 2280-le suurendades paranes uuringualuse mudeli täpsus 16.51-32.79%, olenevalt kas andmed pärinesid ühest või mitmest erinevast algallikast [6].

Analüüsides käesolevat probleemistikku, on selge, et treeningandmete kogus peab olema suur. Nõuded juuksejuure tuvastusmudeli minimaalsele täpsusele on kõrged, kuna soov on seda hakata kasutama spetsialistiga koostöös, et klientidele kõrgkvaliteetset teenust pakkuda. Seejuures on palju erinevate biomarkerite klasse, mida tuvastada vaja.

Andmete juurde genereerimine on sellise probleemi loomulik lahendus. Samas ei ole kasu kuitahes suurest kogusest treeningandmetest, kui samaaegselt ei pöörata tähelepanu ka andmete kvaliteedile ja tasakaalustatusele.

2.2 Andmete kvaliteet

Andmete kvaliteet näitab andmete vastavust etteantud eesmärgile. See mängib olulist rolli pea kõigis arvutirakendustes, olgu selleks masinõppega seotud või mitte, ning on ärilistel eesmärkidel väärtuslik tööriist informeeritud otsuste tegemisel. Kvaliteedi alla kuuluvad näiteks andmete täielikkus, järjepidevus, mittekorduvus, täpsus ja asjakohasus. [7]

Ajalooliselt on andmete kvaliteedi kontrolli läbi viidud manuaalselt või lihtsakoeliste tööriistadega. Selline lähenemine on aga aeganõudev ja kaldub sisaldama inimvigu. Ka hiljutiselt esiletõusnud automaatsed lahendused on keskendunud rohkem relatsioonilistele SQL (struktuurpäringukeel, ing. k. *Structured Query Language*) andmebaasidele, mis nii aina rohkem kasutust leidvate NoSQL andmebaaside kui ka muus vormis andmete (pildid, videod, heli) puhul ei leia kasutust [7]. Piltide kvaliteedi hindamiseks on loodud konkreetselt selleks eesmärgiks eraldiseisvaid masinõppe mudeleid, millest mõned [8] proovivad jäljendada ka inimese poolt pakutavaid hindeid piltide üldisele kvaliteedile.

Probleem tekib, kui ebakvaliteetseid andmeid ei ole eesmärgile vastavalt võimalik olemasolevate tööriistadega välja filtreerida või parandada. Näiteks juuksejuure mikrofotode hulka on võinud sattuda ülesvõtteid, kus juuksejuure paigutus on vale, juuksejuurt pole üldse näha, või ekraanitõmmised töölauast on kogemata teiste piltide vahele ära kadunud. Seega on treeningandmeid märgendades või töödeldes siiski vaja manuaalselt tuvastada ja eemaldada andmed, mis eesmärki ei rahulda. Samas on oluline arvesse võtta, et mõnede mudelite, nt. pildituvastusmudeli, treenimisel on tähtis jätta sisse ka näiteid, kus otsitavaid elemente üldse ei esine.

Kasutades treeningandmete juurde genereerimisel kvaliteetseid lähteandmeid, võib see aidata vähendada ebakvaliteetsete andmete osakaalu lõppmudelis. Mida rohkem reaalsele töökeskkonnale (ing. k. *production environment*) vastavaid treeningandmeid mudeli treenimisel kasutusele võtta ning ebakvaliteetseid vähendada, seda suurem on tõenäosus, et õnnestub saavutada mudeli soovitud sooritustäpsus.

2.3 Andmete tasakaalustatus

Masinõppe kasutamine praktikas on näidanud, et märkimisväärseks pudelikaelaks on olukorrad, kus andmete jaotus klassidesse pole tasakaalustatud, s.t. andmeid on ühe klassi kohta palju rohkem kui teiste [9]. Üldlevinud näide reaalsest olukorrast on haiguseid tuvastavad mudelid, kus enamus lähteandmetest katab haiguseta juhte ning eesmärgi vaates pigem olulisemaid, haigust kujutavaid andmeid jääb väheks [10].

Eriti kerkib tasakaalustatuse probleem esile olukordades, kus domeen on keerukas (klassidesse jaotumine pole lineaarne), olenemata treeningandmete kogusest. Vähemuses esineva klassi valimi suurendamine andmeid suvaliselt korrates või vastupidi, enamuse suvaline vähendamine, on mõlemad olnud efektiivsed meetodid probleemiga tegelemiseks, suurendades mudeli saavutatud täpsust [9]. Lisaks valimi muutmisele on edu saavutatud ka klassifikaatori enda kaalude või künniste parandamisega, liigutades sel viisil tugifunktsioone valimi vähemuse poole [11].

Andmete tasakaalustatus võib mõjuda mudeli täpsusele, seega oleks mõistlik treeningandmete juurde genereerimisel keskenduda vähemuses olevatele klassidele. Seeläbi saab klassijaotust kunstlikult tasakaalustada, hoidudes samaaegselt andmete otsesest kordumisest.

2.4 Lõputöö skoop

Lõputöö raames on peamiseks eesmärgiks katsetada juuksejuure mikrofotode kui masinõppe mudeli treeningandmete juurde genereerimist, et parandada biomarkerite tuvastusmudeli treeningandmestiku kogust ja tasakaalustatust. Analüüsitakse ja võrreldakse erinevaid generatiivseid meetodeid ning valitakse käesoleva eesmärgi vaates kaks parimat, mida praktiliselt teostada.

Skoobist jääb välja andmete kvaliteedi parendus generatiivsetel meetoditel, sest lähteandmed tulenevad kõik samast allikast (mikroskoobist) ning seega võib olla võrdlemisi kindel nende kvaliteedis.

Generatiivsete meetodite eeltöö ja analüüsi raames seatakse üles märgenduskeskkond, andmete hoidla, teisendatakse lähteandmed sobivale kujule ning luuakse esialgne objektituvastusmudel, et võimalusel võrrelda algandmetega ja genereeritud andmetega saavutatud täpsust. Objektituvastusmudel luuakse võimalikult lihtne ning treenimisel kasutatakse kõige rohkem kolme erinevat märgendiklassi, et lihtsustada treeningprotsessi.

Lõputöö skoopi ei kuulu objektituvastusmudeli kvaliteedi analüüs, sest seda kasutatakse vaid generatiivse protsessi tulemuste määratlemiseks. Samuti ei mahu skoopi märgendusandmete ja -keskkonna halduse detailne analüüs – kirjeldatakse lühidalt, miks ja mida valiti, et reaalse eesmärgi kallale asuda.

Oluline on mainida, et rakendatavad generatiivsed meetodid ei lisa juurde, mida lähteandmestikus üldse ei esine. Pigem saab nende abil esitada olemasolevaid andmeid teistsugustes vormides, mis algse treeningandmestiku poolt kaetud pole.

2.5 Nõuded probleemi lahendusele

Lõputöö raames luuakse objektituvastusmudeli loome- ja treeningandmete genereerimisprotsessi esialgne lahendus, eesmärgiga parandada juuksejuure piltidelt biomarkerite tuvastamise mudeli treeningandmete kogust ja tasakaalustatust. Lahendus hõlmab mudeli loomeprotsessi alates andmete märgendamiskeskkonna seadistamisest kuni generatiivsete meetodite rakendamiseni ja objektituvastusmudeli peal testimiseni.

Praktilise töö funktsionaalsed nõuded on järgmised:

- Võimaldab **lisada ja salvestada objektituvastuse märgendeid.**
- Võimaldab **eksportida** märgendeid **masinloetaval kujul.**
- Olemasolevate treeningandmete põhjal saab genereerida uusi ehk **suurendada treeningandmete kogust.**
- Võimaldab genereerida treeningandmeid kindlate biomarkerite kohta, mida esineb vähem kui teisi, ehk **parandada andmete tasakaalustatust.**
- Kus sobilik, võimaldab **tõlkida olemasolevaid märgendeid genereeritud treeningandmetele**, mille märgendid on teada (nt. pärinevad ühest kindlast pildist).

Mittefunktsionaalsed nõuded on järgmised:

- Algsete ja genereeritud treeningandmete märgendid on **turvaliselt säilitatud.**
- Genereeritud ja algsetele pilt- ning märgendusandmetele **pääsevad ligi vaid autoriseeritud isikud.**

3 Generatiivsete meetodite analüüs

Masinõppe generatiivsed mudelid on leidmas aina rohkem kasutust treeningandmete koguse suurendamisel, kui reaalsete andmete kogumine on raskendatud. Tänapäeval üheks piltide genereerimiseks kasutatavateks enimlevinud mudeliteks on generatiivsed vastandvõrgud, variatiivsed autokoodrid ja difusioonmudelid.

Siiski pole masinõppe rakendamine ainuke viis, kuidas pilttreeningandmeid juurde genereerida. Piltide klassikaline transformeerimine annab lihtsakoelise võimaluse mitmekordistada algsete andmete arvu, tehes muudatusi näiteks pildi heleduses, pilti pöörates, objekti nihutades jne.

Järgnevalt kirjeldatakse lähemalt piltandmete transformatsioone ja kolme erinevat generatiivset masinõppemudelit. Seejuures analüüsitakse nende erinevusi ning sobivust lõputöös käsitletava probleemi raames ehk juuksejuure mikrofotode arvu suurendamiseks pildituvastusmudeli treenimise eesmärgil.

3.1 Piltide transformatsioonid

Pildi transformeerimine kujutab endast nii objekti paigutuse kui ka valgustuse, teravuse jm. töötlemist. See on saanud levinud tavaks treeningandmestiku ettevalmistamisel, kui algseid andmeid jääb väheks, ning võimaldab andmestikku suurendada kuni tuhandetekordselt [12].

Inimese jaoks on enamasti lihtne tuvastada objekti erinevates keskkondades ja paigutustes, kuid masina puhul võib väheste andmete puhul tekkida probleem ülesobitusega ehk mudeli võimetusega üldistada õpitut väljaspool treeningandmeid. Seega on kasulik objekte esitada võimalikult paljudes erinevates suurustes, nurkades, peegeldustes ja valgustes, mis võivad reaalses töökeskkonnas esineda.

Juuksejuure mikrofotode puhul on selline lähenemine põhjendatud, sest juuksejuured võivad olla jäädvustatud igat pidi – olla keerdus või sirged, teiste juurtega sassis, suurema või väiksema suurendusega ning jääda pildile ainult otsapidi või terves pikkuses. Mida

rohkem erinevaid võimalusi on kaetud, seda paremini õpib mudel markereid tuvastama atribuutide järgi, mis päriselt loevad, ning jätma kõrvale neid, mis markerite tuvastamisega seotud pole. Samuti on võimalik parandada andmete tasakaalustatust, lisades transformatsioone vähemuses olevatele klassidele.

Nii klassifikaatorite kui pildituvastusmudelite puhul on transformatsioonidega saavutatud mudelite täpsuses parandusi. Näiteks L. Perez ja kaasautorite [13] läbi viidud uuringus tõusis koerte ja kasside klassifikaatori täpsus treeningandmeid sel viisil kahekordistades 70.5%-lt 77.5%-le. Ka tehisavaradari sihtmärgituvastusmudeli treenimisel saavutati suurim täpsus, kui treeningandmeid eelnevalt transformeeriti [14].

Transformatsioonide eelisteks on masinõppe mudelite ees protsessi automatiseerimise lihtsus, kiirus ning võimalus säilitada varasemalt lisatud märgendeid. Samas tuleb meeles pidada, et tegemist on siiski mingil määral dubleeritud andmetega, ehk kasutegur on praktikas piiratud. [15]

Piltandmete transformatsioonid on ka kasulikuks eeltööks järgnevalt kirjeldatud generatiivsete masinõppemudelite treeningandmete koguse suurendamisele. Lisaks on võimalik mudelite õnnestunult genereeritud andmeid omakorda transformeerida. Kokkuvõttes on see tänu oma lihtsusele, tõestatud kasutegurile ja kohaldatavusele loomulikuks esimeseks sammuks treeningandmete genereerimise protsessis.

3.2 Generatiivsed sügavõppemudelid

Masinõppe mudelid jagunevad diskriminatiivseteks ja generatiivseteks. Diskriminatiivsed mudelid tegelevad klassidevaheliste piiride leidmisega, näiteks klassifikatsioon ja regressioon. Generatiivsed mudelid on viimaste aastate jooksul kerkinud juhendamata sügavõppe rakendustes esile kui tõhusad meetodid analüüsimaks ja mõistmaks märgendamata andmeid. Nende üldine eesmärk on leida andmetes peituv tõenäosusjaotus, mida kasutada sarnaste andmete genereerimiseks. [16]

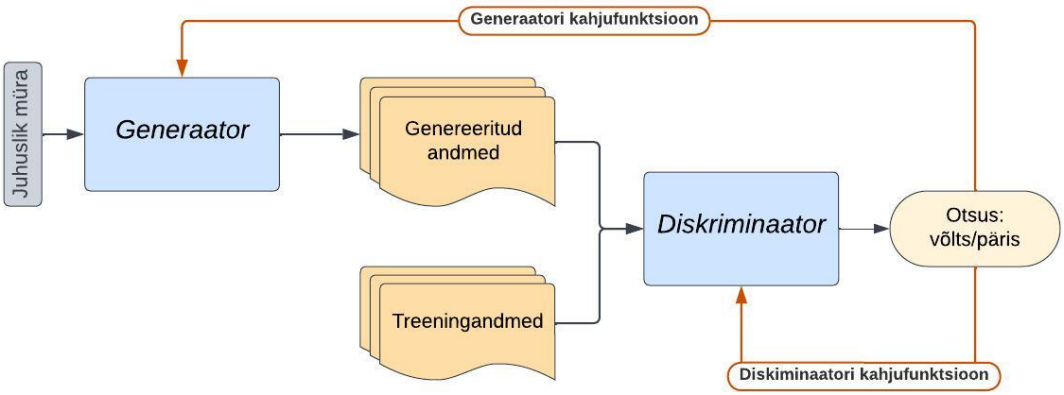
Kui treening sujub, siis on masinõppega treeningandmete koguse suurendamisel piltide traditsiooniliste transformatsioonide ees eelis – see võtab inimeselt ära vajaduse teha otsuseid, sarnaselt nagu süvaõpe eemaldab vajaduse määrata andmestikule tunnuseid. Lisades treeningandmestikku näiteid kõikvõimalikest erinevates paigutustest, oskab mudel ideaalis ka genereerida pilte objektist mistahes võimalikus olukorras. [17]

Järgnevalt kirjeldatakse lähemalt generatiivseid vastandvõrke, mis on generatiivsete mudelite hulgas tempokalt populaarsust ja edasiarendusi kogumas.

3.2.1 Generatiivsed vastandvõrgud

Generatiivsed vastandvõrgud (GAN, ing. k. *generative adversarial network*) pakuti esmalt välja I. Goodfellow ja kaasautorite [18] poolt 2014. aastal. Neid võib kirjeldada kui kahte omavahel konkureerivat süsteemi: **generaator** ja **diskriminaator**. Tüüpiliselt põhinevad need tehisnärvivõrkudel, kuid võivad olla mistahes diferentseeruvad süsteemid, mis kaardistavad andmeid ühest ruumist teise [19].

Generaatori ülesandeks on õppida looma võltsandmeid, mis sarnaneks võimalikult palju reaalsete andmetega. Diskriminaator on klassifitseerija ning käitub justkui ’kriitik’, proovides saada eksperdiks võltsingute tuvastamises. Generaatoril ei ole tegelikku ligipääsu reaalsetele andmetele – ainuke viis, kuidas saada tagasisidet oma töö kohta, on diskriminaatori tulemused ja tagasilevi. Diskriminaatori treeningandmeteks on nii reaalsed andmed kui generaatori loodud andmed. Valesti klassifitseeritud andmed mõjutavad kahjufunktsioone, mille tagasilevi aitab parandab süsteemide parameetreid ehk kaalusid [19], [20]. Joonisel 1 on esitatud GAN-i töövoog.



Joonis 1. Generatiivne vastandvõrk.

Kuigi GAN-id on pildigeneratsiooni ülesannetes ühed edukamatest mudelitest [16], on nende treeningprotsessil levinud puuduseid, mida aktiivselt uuritakse. Esiteks, kui diskriminaator on liiga osav, võib tekkida hääbuva gradiendi probleem (ing. k. *vanishing gradient problem*) – diskriminaator ei paku generaatorile piisavalt informatsiooni, et areneda. Teiseks, üldiselt on soov, et GAN genereeriks laia valikut andmeid. Võib aga juhtuda, et generaator leiab ühte tüüpi väljundi, millega diskriminaatori alati ära petab

(ing. k. *mode collapse*). Kolmandaks, kui generaator on piisavalt osav, et diskriminaator ei tee enam reaalsetel ja genereeritud andmetel vahet, siis tekib koonduvuse probleem (ing. k. *failure to converge*). Diskriminaator hakkab suvaliselt pakkuma ja generaatorile valet tagasisidet andma, mille tagajärjeks on generaatori kvaliteedi halvenemine [20].

Kui treening sujub, on GAN-id aga näidanud lubavaid tulemusi nii treeningandmestiku kvaliteedi parandamisel kui uute treeningandmete genereerimisel. Näiteks [21] kasutab GAN-e sünteetiliste andmete kvaliteedi parandamiseks, et nendega edasises treeningprotsessis paremaid tulemusi saavutada. Meditsiinilistest rakendustest on näiteks maksakahjustuse klassifitseerija treenimisel võrreldud treeningandmete traditsioonilisi transformatsioone ja GAN-i genereeritud andmeid, ning GAN-idega saavutatud keskmiselt 7% kõrgem täpsus [22].

GAN on laialdaselt uuritud ja kasutatud mudel, seega on sellele välja pakutud palju parendusi ja spetsiifilistele ülesannetele sobivaid muudatusi. Edasiarenduste hulka kuuluvad:

- *Conditional GAN* (cGAN) [23]. Treenides cGAN-i märgendatud andmetel, saab sellega genereerida andmeid kindlatest klassidest.
- *Progressive growing GAN* [24]. Võimaldab tavalisest GAN-ist kiiremini ja efektiivsemalt genereerida kõrgema kvaliteediga pilte.
- *Image-to-image translation GAN* [25]. Võtab sisendiks pildi ja genereerib sellest sarnaste omadustega uue pildi, näiteks hobusekujuga sketšist võib saada pildi sama kujuga realistlikumast hobusest.

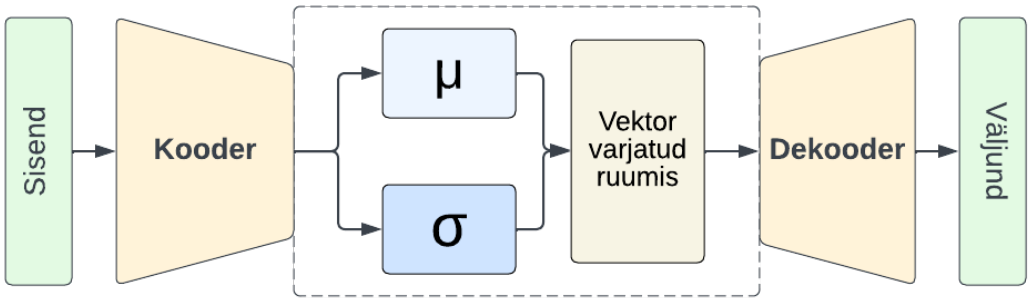
3.2.2 Variatsioonilised autokoodrid

Variatsioonilised autokoodrid ehk VAE-d (ing. k. *variational autoencoders*) on generatiivsed mudelid, mis on eriti palju kasutust leidnud olemasolevate andmete muutmisel või arendamisel spetsiifilises suunas. Lihtsaks näiteks on inimesele pildil prillide lisamine või numbrimärkide laiemaks tegemine [26]. Lisaks on VAE-d kasutatud ka päris uute andmete genereerimiseks, näiteks kõrgresolutsiooniga digitaalse kunsti loomisel ning väljamõeldud tegelaskujude nägude genereerimisel [27].

Tüüpilise autokoodri baaskomponentideks on **kooder** ja **dekooder**. Kooder on närvivõrk, mis teisendab sisendi palju kompaktsemaks esituseks, eesmärgiga säilitada võimalikult palju asjakohast infot ning vabaneda ebaolulisest. Dekoodri ülesandeks on kodeering võimalikult sarnasel kujul uuesti üles ehitada. Kaofunktsiooniks on tavaliselt n-ö rekonstrueerimiskadu, mis karistab võrku sisendist erineva väljundi loomisel. [26]

Kuna autokooder õpib tootma sisendile väga sarnaseid andmeid, on selle kasutusalad üpris piiratud, näiteks müra vähendamine pildil. Variatsioonilised autokoodrid aga lisavad koodri töösse oma nimele kohaselt sisendandmete variatsioone. Lühidalt tähendab see seda, et ühe kodeeringvektori asemel väljastab see kaks – keskmiste vektor ja standardhälvete vektor. Nende kahe vektori kui parameetri põhjal luuakse lõplik kodeering, mis samade sisendandmete puhul sisaldab iga kord väikseid variatsioone. [26]

Joonisel 2 on esitatud variatsioonilise autokoodri diagramm.



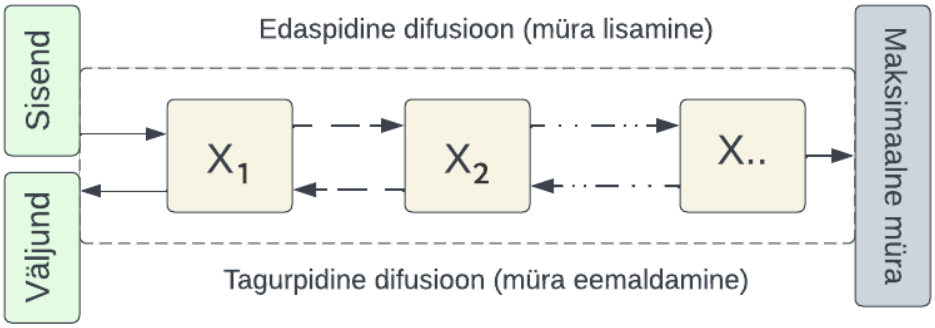
Joonis 2. Variatsiooniline autokooder.

Sisendandmeid, mis on treeningandmetest erinevad, ei suuda mudel eriti hästi dekodeerida – seega on normist erinevate andmete ehk anomaaliate leidmine teiste generatiivsete meetodite kõrval VAE üheks tugevamaks küljeks. Samuti on VAE lihtsakoeline ning seda on kerge treenida. Samas on täheldatud, et VAE genereeritud piltandmed on tüüpiliselt hägusemad, kui teiste mudelite toodetud andmed, seega ei sobi see hästi kasutuseks rakendustes, kus detailid on olulised. [28]

3.2.3 Difusioonmudelid

Difusioonmudelid on hiljuti esile tõusnud generatiivsed mudelid ning on praeguse tehisintellektibuumi esirinnas. Need suudavad GAN-ist ja VAE-st palju kõrgema täpsusega genereerida suure eraldusvõimega ja mitmekesiseid pilte [29]. Enimlevinud difusioonmudelite hulka kuuluvad OpenAI toodetud GLIDE ja DALL.E-3. [30]

Difusioonmudelit kirjeldab edaspidine ja tagurpidine difusiooniprotsess (Joonis 3). Algandmetele lisatakse järkjärgult müra, mida seejärel võimalikult sarnaselt uuesti eemaldada üritatakse. Nagu VAE, on tegu varjatud ruumi muutujate peal treenitud mudeliga, kusjuures madalamal tasemel rakendatakse ka mittetasakaalulise termodünaamika põhimõtteid. [29]



Joonis 3. Difusioonmudel.

Difusioonmudelite peamiseks eeliseks teiste generatiivsete mudelite ees on pildikvaliteet. Esiteks ei vähendata treeningprotsessi raames algandmete mõõtmeid, nagu seda teeb VAE. Teiseks ei kasutata kergesti petetavat vastandlikku protsessi nagu GAN. Näidiste genereerimiseks on vaja aga närvivõrku mitu korda käitada, et järkjärguliselt algandmetest uued andmed genereerida. Sel põhjusel võib mudeli treenimine võtta eespool kirjeldatutest kordades rohkem aega. [31]

3.3 Meetodite valik

Käesoleva lõputöö raames mahub skoopi kaks generatiivset meetodit, kusjuures valiku all on piltide transformatsioonid ning üks kolmest generatiivsest masinõppemudelist.

Piltide transformatsioonide näol on võrreldes generatiivsete mudelitega tegemist väga lihtsakoelise lahendusega, mis ei nõua arendamiseks palju ressurssi, on skaleeritav ning võimaldab genereeritud andmeid koheselt kasutada treeningprotsessis, sest märgendusandmed on tõlgendatavad. Kuigi midagi treeningandmetest väga erinevat sel meetodil genereerida ei saa, on varasemad uuringud siiski näidanud, et pilditransformatsioonidest võib olla masinõppe mudeli treenimise vaates väga palju abi. Seega on see esimeseks lõputöös rakendatavaks generatiivseks meetodiks.

Tabelis 1 võrreldakse generatiivsete masinõppemudelite erinevaid omadusi: pildikvaliteeti, treeningprotsessi, edasiarendusi ning genereeritud andmete mitmekesisust. Selle abil tehakse valik lõputöö praktilises osas loodava mudeli osas.

Tabel 1. Generatiivsete masinõppemudelite võrdlus. [31]

GAN	**VAE**	**Difusioonmudel**
Pildikvaliteet on hea – diskriminaator peab õppima generaatorit ära petma.	Pildikvaliteet on keskmine – kipub jääma hägune või ebadetailne.	Pildikvaliteet on väga hea – järkjärguline protsess võimaldab keskenduda detailidele.
Treeningprotsess jooksul võib kahe kaofunktsiooni tõttu olla raske määrata, kui mudel on koondunud.	Treeningprotsess on lihtsakoeline, kasutusel on üksainus kaofunktsioon.	Treeningprotsess on järkjärguline, võib võtta rohkem aega kui teised mudelid ning nõuab palju arvutiressurssi.
Edasiarendusi ja uuringuid on palju ning mudel areneb kiiresti edasi.	Edasiarendusi ei ole teoreetilise keerukuse tõttu nii palju kui. GAN-il ning mudel areneb aeglasemalt.	Mudel areneb suure kiirusega ning on kasutusel kõige uudsemates tehnoloogiates.
Mitmekesisusega on vahel probleeme – generaator võib õppida tootma diskriminaatori petmiseks ainult ühte tüüpi väljundit.	Kõrge mitmekesisus – tõenäosus katta treeningandmestiku kõiksugused esinemised on suurem.	Kõrge mitmekesisus – tõenäosus katta treeningandmestiku kõikvõimalikud esinemised on suurem.

Juuksejuure mikrofotodelt biomarkerite tuvastamiseks on pildikvaliteet oluline. Biomarkerid võivad piltidel olla raskelt märgatavad, eriti kui neid on väikestes koguses või algusfaasis – seetõttu loeb mudeli treeningul iga detail. Kuna VAE jääb GAN-i ja difusioonmudeli kõrval selles vallas alla, ei ole see käsitletava probleemi ehk juuksejuure mikrofotode genereerimise raames parim valik. Lisaks on VAE madala taseme teostus teoreetiliselt võrdlemisi keerukas ning seega ei ole sellele loodud nii laialdaselt edasiarendusi, kui näiteks GAN-ile. Seega oleks käesoleva lõputöö edasiarendused piiratud.

Difusioonmudelite tulemused on kolmest mudelist kõige parema kvaliteediga ning kõige mitmekesisemad. Lisaks on tegu uudsema ning tulemuste põhjal laiema rakendatavusega mudeliga, mida hoogsas tempos edasi arendatakse. Kahjuks on mudeli treeningaeg võrreldes teise kahega väga kõrge, sest kasutab järkjärgulist arengut. Lõputöö teostuse

vaates võib see osutuda suureks pudelikaelaks, sest autori kasutuses ei ole graafikakaarti, mis ühilduks levinumate masinõppeteekidega ning kiirendaks oluliselt mudeli treenimist.

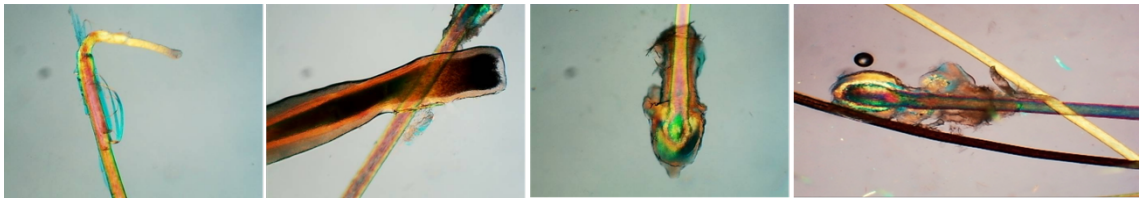
GAN-i nõrgaks küljeks on *mode collapse* ehk fenomen, kus generaator õpib tootma vaid ühte tüüpi andmeid, millega diskriminaatori iga kord ära petab. Veel võib mudeli treenimise jooksul olla raske jälgida kahte kaofunktsiooni korraga ning määrata, millal mudel on koondunud ja treeningu võiks lõpetada. Kui aga olla treeningprotsessi jooksul hoolikas, tähelepanelik ning katsetada erinevaid treeningparameetreid, on mõlemad nendest muredest kontrollitavad. GAN on suuteline genereerima hea kvaliteediga pilte, mis juuksejuure mikrofotode vaates on oluline. Kuigi difusioonmudel on hetkel generatiivsete mudelite hulgas väga populaarne, on ka GAN laialdaselt generatiivsetes rakendustes kasutusel ja sellega on saavutatud väga häid tulemusi. Sel põhjusel on GAN ressursi, kvaliteedi, ning edasiarenduste poolest lõputöö eesmärgi ja skoobi vaates kõige parem valik.

Kokkuvõttes valitakse generatiivsetest meetoditest teostuseks piltide klassikalised transformatsioonid ning generatiivne masinõppemudel GAN.

4 Lähteandmestik

Lõputöö lähteandmeteks on ligikaudu 14500 eksemplari juuksejuurte mikroskoopilistest ülesvõtetest (mikrofotodest). Tervisenäitajad esinevad juuksejuurtel kindlate tunnuste ehk biomarkerite kujul. Juuksejuure preparaatide valmistamisel on kasutatud spetsiaalseid tinte, et tuleks esile biomarkeritele omased värvikavad, mida spetsialist oskab seejärel mikrofotolt määrata.

Tervisenäitajate hulka kuuluvad näiteks immuunpuudulikkus, hapnikupuudus, stress ja paistetus. Mikrofotol võib lisaks juuksejuurele esineda teiste juurte osi, õhumulle või omaette klassimärgendi alla kuuluvaid parasiite. Joonisel 4 on esitatud juuksejuure mikrofotodest näiteid.



Joonis 4. Näiteid juuksejuurte mikroskoopilistest ülesvõtetest.

Enamus piltidest on toorel kujul 640x480 pikslit või 1280x720 pikslit. Autorile on nendest mõned saadetud tervisenäitajate kaupa kaustadesse jagatult, kuid märgendamata kujul. Ühte pilti võib esineda mitmes kaustas, olenevalt kui mitu biomarkerit sellel leidub. Teised on kuupäevade kaupa kokku kogutud, kuid sorteerimata. See tähendab suurt eeltööd eelkõige märgendamisega ehk piltidele tuleb määrata biomarkerite asukohad sellises vormis, et need oleksid kasutatavad masinõppe mudelite treeningandmetena.

Projekti lõppeesmärgiks (mis jääb lõputöö skoobist välja) on luua võimalikult kõrge täpsusega biomarkerite tuvastuse mudel, mida saaks kasutada reaalses keskkonnas protsessi automatiseerimiseks. Selleks, et lõputöö raames saaks testandmete juurde genereerimise kasulikkust mõõta, on vaja luua tuvastusmudeli algne testversioon, mille täpsust originaalsete testandmetega ja juurde lisatud, genereeritud testandmetega võrrelda.

4.1 Andmete jaotus ja valik

Tabelis 2 on esitatud kõigi 9 teadaoleva biomarkeri kohta esinemiste ennustatav arv esmase kaustadesse jagunemise põhjal ning selle arvu osakaalu 14407-st pildist. Biomarkerite kirjeldavad tunnused ja tähendused on projektiliikmete soovil peidetud ning ei oma lõputöö kontekstis tähtsust.

Tabel 2. Biomarkerite esinemiste jaotus ja lõputöö skoopi kuuluvus.

Biomarkeri identifikaator	**Esinemiste ennustatav arv**	**Esinemiste osakaal kõikidest andmetest**	**Sisaldub lõputöö skoobis**
MARKER1	2874	19,9%	Jah
MARKER2	1893	13,1%	Jah
MARKER3	1782	12,4%	Jah
MARKER4	2195	15,2%	Ei
MARKER5	863	6,0%	Ei
MARKER6	614	4,3%	Ei
MARKER7	1312	9,1%	Ei
MARKER8	194	1,3%	Ei
MARKER9	701	4,9%	Ei

Erinevaid võimalikke biomarkereid on palju ja treeningandmeid võrdlemisi vähe, mis teeb tuvastusmudeli töö raskeks. Selleks, et mahutada mudelite loomine, võrdlus ja analüüs lõputöö skoopi ning lihtsustada tulemusi, on autor valinud kolm levinumat ja suuremate omavaheliste erinevustega markerit, millega esmast objektituvastusmudelit treenida ning mille märgendite põhjal andmeid juurde genereerida. Tabeli 2 viimane veerg esitab otsustuse tulemusi.

Kuigi MARKER4 esinemiste arv on suurem kui MARKER2 ja MARKER3, on selle esinemisviisid sarnased teistele valitud markeritele. Arvestades andmete võrdlemisi väikest kogust, on otsustatud see marker lõputöö raames välja jätta, et valitud markerid oleksid paremini eristatavad ja lihtsustaksid tuvastusmudeli tööd.

5 Metoodika

Järgnevalt kirjeldatakse, mis meetoditel planeeritakse teostada märgenduskeskkonna ülesseadmist, andmetöötlust, transformatsioone, generatiivset masinõppemudelit GAN ning objektituvastusmudelit. Seejuures põhjendatakse lühidalt tehtud valikuid.

5.1 Märgendushaldus

Juhendatud masinõppes on märgendamisel määrav tähtsus. Kuna andmeid on sageli suurtes kogustes, võiks märgendamine toimida võimalikult kiirelt ja mugavalt. Selleks otstarbeks on kasulik üles seada märgenduskeskkond, kuhu on ligipääs domeeniteadlikel isikutel.

Palju leidub mugavusteenustega märgendustarkvara, mille kasutamiseks on vaja litsentsi kas kohe alguses, teatud aja pärast või ületades etteantud kasutuspiirid. Nende alla kuuluvad näiteks *Labelbox*, *Amazon SageMaker Ground Truth* ja *Superannotate*. Olemas on ka vabatarkvaralisi lahendusi nagu *Label Studio*, *CVAT* ja *Sloth*. *Label Studio* on eelnimetatutest üheks levinumaks ning ametlikus dokumentatsioonis on kasulikku infot, kuidas seda privaatselt hostida [32]. Valides *Label Studio*, saab seega vähendada üldkulusid ning pakkuda ööpäevaringset ligipääsu kuitahes paljudele märgendajatele.

Treeningandmete hoiustamine, *Label Studio* abiteenused ja hostimine tuleb üles seada nii, et taristuhaldusele oleks kiire ligipääs ka teistel, juhul kui selleks vajadus tekib. Autoril ei ole isiklikku füüsilist serverit, seega kõige mugavam lahendus on kasutada olemasolevaid pilvetaristuteenuseid nagu AWS (*Amazon Web Services)*. AWS teenused on üle teatud piiri tasulised, kuid autoril on projekti raames ligipääs kõigile vajalikele AWS ressurssidele.

5.2 Andmetöötlus ja masinõpe

Masinõppesüsteemide ja nendega lähedalt seotud andmetöötluse rakendustes on tüüpiliselt kasutusel kas programmeerimiskeelel Python põhinevad teegid või

andmetöötluskeskkond MATLAB. Kumba neist kasutada, oleneb ülesande spetsiifikast ning kasutaja soovidest ja kogemusest.

Esiteks on oluline, et kuigi Tallinna Tehnikaülikooli tudengina on autoril litsents, siis MATLAB on tasuline teenus. Python on igaühele tasuta saadaval, seega seda kasutatakse laialdasemalt, eriti suurte masinõppe projektide arendamiseks. Rohkemate kasutajatega kaasneb rohkem teekide edasiarendusi, võimalusi ja painduvust. MATLAB on mugavama kasutajaliidesega ning sobib hästi väiksemateks projektideks. [33]

Võttes arvesse, et lõputöö raames loodavaid lahendusi võib tulevikus vaja minna kas toorel kujul või edasiarendustes, tundub Python paindlikum ja kättesaadavam valik. Lisaks on autoril Pythoniga rohkem kogemust.

Järgnevalt on nimetatud olulisemad lõputöös kasutatud Pythoni teegid.

- Märgendatud andmete transformeerimiseks saab kasutada *imgaug* teeki, mille suureks eeliseks on võimalus pilte transformeerides samaaegselt muuta märgenduskastide koordinaate [34]. Nii kaob vajadus käsitsi pildi suuruse ja objektipaigutuse muutusi uuteks koordinaatideks tõlgendada.

- Testimiseks kasutatava objektituvastusmudeli treenimiseks kasutatakse *ImageAI* teeki, mis pakub lihtsat, kuid võimsat lähenemist YOLOv3 arhitektuuril põhinevate objektituvastusmudelite loomisel [35].

- GAN loomisel kasutatakse *PyTorch* sügavõppe teeki, sest võrreldes teise populaarse sügavõppe teegi *TensorFlow*-ga on see dünaamilises modelleerimises ja generatiivsetes rakendustes paremaid tulemusi näidanud [36].

- AWS ressursihalduseks ehk peamiselt S3 hoidla andmehalduseks kasutatakse AWS ametlikku teeki *Boto3*, mis võimaldab madalatasemelist juurdepääsu AWS teenustele [37].

6 Töö käik

Järgnevalt on kirjeldatud lõputöö praktilist tööd. See hõlmab andmete ettevalmistust, märgenduskeskkonna ülesseadmist, andmete transformeerimist, GAN-i treenimist ning testimiseks algsete objektituvastusmudelite loomist. Koodinäited on saadaval GitHub-is (Lisa 2).

6.1 Andmete ettevalmistus

Selleks, et masinõppe mudelit oleks võimalik treenida, tuleb treeningandmed ette valmistada, viies need sobivale kujule ja lisades märgendid.

Andmeid on kasulik hoiustada, kus neile oleks kõigil osapooltel mugav ligipääs. Seega teisaldati andmed esimese sammuna AWS S3 hoidlasse, kus andmetele määrati vastavalt nende allikale eesliited (kaustad):

- **MAIN** – algsed andmed, millel esinevad biomarkerid olid juba teada algse kaustadesse jagunemise alusel.
- **RAW** – hiljem lisanduvad andmed, mis on toorel kujul ehk teadmata markeritega piltandmed. Neid koguneb aja jooksul aina rohkem juurde, sest spetsialistid koguvad järjest juuksejuurte proove ning saadavad tuvastusmudeli treenimiseks.
- **GEN** – genereeritud piltandmed. Siia pannakse nii piltide traditsiooniliste transformatsioonide tulemused kui ka masinõppe generatiivsete mudelite toodetud pildid.

MAIN kausta piltide lisamisel pandi nimele nende algsete kaustade alusel eesliide, mis tuvastaks nendel esineva biomarkeri. Näiteks algse ’stress’ kausta pildile nimega *158.jpg* sai uueks nimeks *stress-158.jpg*. Nii suudeti säilitada infot piltidel esinevate biomarkerite kohta, eemaldades vajaduse jaotada pildid markeripõhistesse kaustadesse.

6.1.1 Korduvate andmete eemaldamine

Algsed piltandmed esitati autorile biomarkerite kaupa kaustadesse jaotatult, kusjuures ühte pilti võis olenevalt biomarkerite arvust esineda mitmes kaustas. Et vältida piltide kordumist S3 hoidlas, lahendati see korduvate olemite eemaldamisega.

Esialgu seadistati AWS IAM (*Identity & Access Management*) olemid, millel oleks ligipääs ja luba vastavast S3 hoidlast esemeid loendada, lisada ja kustutada. Selleks loodi uue kasutajaga grupp, millega seostati poliis, mis annab vajalikud load grupi liikmetele vastava S3 hoidlaga vajalikke tegevusi läbi viia (Lisa 2). Kasutajale loodi ligipääsuvõti, mille saab kohalikus masinas üles seada nii, et *boto3* leiaks selle automaatselt üles.

Igal S3 olemil on ETag (*Entity Tag*), mis on objekti MD5 räsi ning mida saab kasutada olemite samasuse võrdlemiseks. Korduvate piltide korral ühendati MAIN kausta olemite puhul nimed nii, et säiliks informatsioon piltidel esinevate biomarkerite kohta. Näiteks kui pilt, kus on biomarkerid MARKER1 ja MARKER2, esineks hoidlas korduvate olemitena *marker1-3.jpg* ja *marker2-3.jpg*, siis saaks uue olemi nimeks *marker1-marker2-3.jpg*. Kuna S3 olemit muuta ei saa, kustutatakse mõlemad eelmised hoidlast ning lisatakse pilt uue nimega.

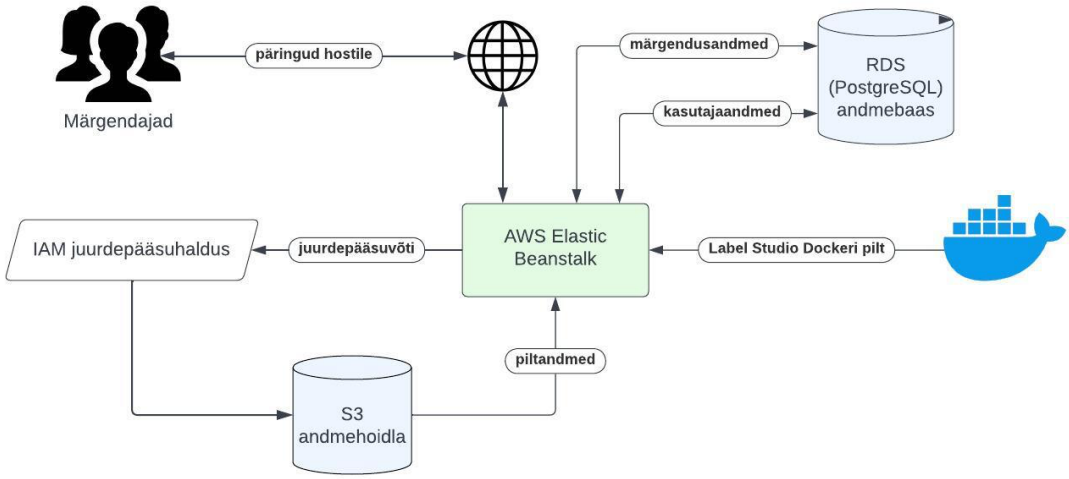
Skripti oli vaja käitada nii mitu korda, kui esines ühe pildi kohta kõige rohkem korduseid. Parendusena saaks otsida ühe käitamise jooksul piltide kõik kordused ning teha töö korraga ära. Nii saab hoiduda ka piltide mitmekordsest kustutamisest ja uuesti lisamisest.

Skripti käitamiste käigus vähenes piltide arv MAIN kaustas 17769-lt 11227-le ehk eemaldati 6542 korduvat pilti.

6.1.2 Märgenduskeskkonna seadistamine

Andmete märgendamiseks seati üles märgendustarkvara *Label Studio* privaatselt hostitud eksemplar. Selleks kasutati AWS *Elastic Beanstalk* orkestreerimisteenust koos EC2 (*Elatic Compute Cloud*) serveri ja RDS (*Relational Database Service*) PostgreSQL andmebaasiga. Seadistamisel on tuginetud artiklile [38].

Joonisel 5 on esitatud ülevaatlik skeem taristu komponentide omavahelistest seostest.



Joonis 5. Ülevaade märgenduskeskkonna taristust.

Esmalt loodi märgendusandmete turvalise säilitamise eesmärgil PostgreSQL andmebaas, sest see ühildub *Label Studio*ga. Selleks kasutati AWS RDS teenust. Seejärel seadistati *Label Studio* rakendus, kasutades AWS orkestreerimisteenust *Elastic Beanstalk,* mis teeb veebirakenduse ülesseadmise Dockeri konteinerisüsteemi baasil väga mugavaks. Rakenduse loomisel tuli üles laadida Dockerfile – failiformaat, mis sisaldab Dockerile spetsiifilist rakenduse loomise juhendit. Valides serveri käitamiseks sama VPC (*Virtual Private Cloud*), mis andmebaasil, seadistati nende vahel võrguühendus. Lisaks käivitati serveri seiresüsteem ja logid.

*Label Studio*le pääseb ligi *Elastic Beanstalk*is rakendusele seatud hostinimel. *Label Studio* esmasel avamisel tuli luua uus projekt ning seadistata vajaduspõhine keskkond koos asjakohaste märgendusklasside ning märgendustüübiga, mis objektituvastusmudeli puhul on kastide joonestamine. S3 hoidla piltidega tuli projektis luua ühendus, kasutades peatükis 6.1.1 loodud IAM ligipääsuvõtme ID-d ning ligipääsuvõtit ja täpsustades soovitud piltide asukohtadele hoidlas vastav regulaaravaldis.

Label Studio andmehalduse kitsakohaks on asjaolu, et märgendusandmed on seotud piltandmete nimedega S3 hoidlas. Kui pilt näiteks eemaldatakse ja lisatakse uue nimega, siis kaob andmebaasis seos pildi ja selle märgendite vahel. Seega tuleb olla väga ettevaatlik, kui märgendatud piltide nimesid hoidlas uuendada ning vajadusel tuleb manuaalselt vastavad muudatused teha ka andmebaasis.

6.2 Generatiivsete meetodite rakendamine

Generatiivsetest meetoditest rakendati algandmetele piltide klassikalisi transformatsioone ning treeniti generatiivne masinõppemudel GAN. Järgnevalt on kirjeldatud nende teostust, esinenud probleeme ja leitud lahendusi.

6.2.1 Piltide transformeerimine

Piltide transformeerimiseks kasutatav *imgaug* teek eeldab, et piltide märgendusandmed on esitatud märgenduskastide vasaku alumise ja parema ülemise nurga pikselkoordinaatidena. Kuigi *Label Studio* eksportfunktsioon võimaldab märgendusandmeid CSV-formaadis eksportida (näide Lisas 4), tuli neid transformatsioonidel kasutamiseks sobivale kujule teisendada.

Algselt eksporditud märgendusandmed on esitatud protsentväärtustena pildi maksimaalsetest mõõdetest. Seejuures on kasutusel vasaku alumise nurga pikselkoordinaatide suhe pildi laiusest ja kõrgusest ning kasti enda laius ja kõrgus. Kuna antud on ka pildi laius ja kõrgus, sai selle kaudu võrdlemisi kergelt teisendada andmed protsendilistest väärtustest sobival kujul pikselkoordinaatideks.

Masinõppe mudelite treenimiseks piltandmetel on kasulik teisendada pildid ruudukujule ning teha mõõtmed nii väikseks, kui võimalik, et kiirendada treeningprotsessi ja eemaldada müra tekitavaid parameetreid. Algsed juuksejuure mikrofotod polnud ei ruudukujulised ega kuigi väiksed, seega tuli esimese sammuna need muutused sisse viia. Piltide ruuduks teisendamisel tuli teha valik, kas pilti lõigata või lisada n-ö täitematerjali. Et säilitada kogu algses pildis edastatav informatsioon, valiti täitematerjali lisamine, kusjuures algne pilt jääb alati võimalikult suurena ruudu keskele.

Transformatsioonide osas tuli otsustada, milliseid transformatsioonitüüpe valikusse arvata, mitu transformatsiooni korraga ühele pildile rakendada ning mitu genereeritud pilti ühest algsest pildist toota. Valik langes umbes võrdsele osale geomeetrilistele muudatustele (pööramine, peegeldus, transpositsioon) ja visuaalsetele muudatustele (valgus, hägusus), et andmestikku sisse viia võimalikult palju erinevaid versioone algsetest andmetest. Sobiv tundus korraga rakendada kaks transformatsiooni ja mitte rohkem, et säilitada teatud määral realistlikkust. Iga algse pildi kohta genereeritakse kaks transformeeritud pilti. Nii saab esitada sama juuksejuurt mitmel viisil, kuid samas piisavalt vähe, et vältida andmestiku üleküllastumist väga sarnaste andmetega.

Et nii genereeritud pildid kui ka nende märgendusandmed oleksid kõigile osapooltele kättesaadavad, laetakse genereerimise käigus pildid järjest S3 hoidla GEN kausta. Protsessi lõppedes laetakse uued märgendusandmed CSV failina nii S3 hoidlasse kui ka kohalikule kettale (näide Lisas 5).

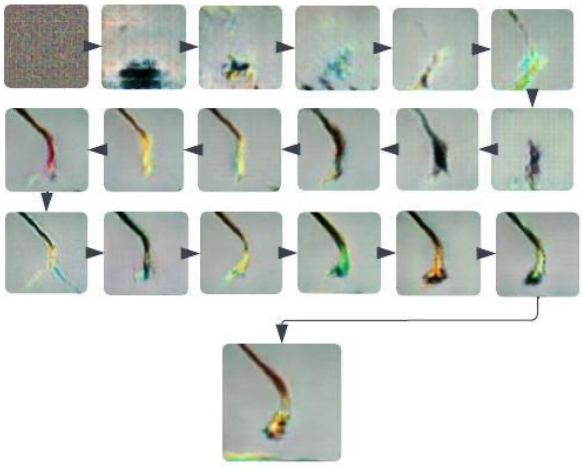
6.2.2 Generatiivse vastandvõrgu loomine

GAN-i loomisel kasutatati Pythoni masinõppe teeki *PyTorch* ning tugineti *PyTorch* avaldatud juhendile [39]. Esmase GAN-i treenimise lihtsustamiseks vähendati treeningandmete mõõtmeid 128x128 pikslile.

Selleks, et generaator oleks võimeline teisendama varjatud ruumi vektori 128x128 mõõtmeliseks värviliseks pildiks, tuli luua mitmeastmeline konvolutsiooniliste kihtide seeria. Igale kihile lisati alaldi (ReLU, *Rectified Linear Unit*) aktiveerimisfunktsioon ning normaliseerimiskiht ning rakendati ülesproovimist (ing. k. *upsampling*), mille raames väljundobjekti ruumilisi mõõtmeid järkjärgulist suurendatakse.

Diskriminaatori ülesandeks on võtta sisendiks pilt ning väljastada skalaarne tõenäosus, et tegu on reaalse pildiga. Ka selle jaoks loodi konvolutsiooniliste kihtide seeria koos normaliseerimiskihi ja ReLU aktiveerimisfunktsiooniga. Lõplik tõenäosus saavutati läbi sigmoid-aktiveerimisfunktsiooni.

Joonisel 6 on esitatud generaatori erinevate epohhide ehk treeningtsüklite jooksul genereeritud vahetulemused ühe pildi kohta. Alustatakse suvalisest mürast, kuid mida rohkem generaator diskriminaatorilt kahjufunktsiooni kaudu tagasisidet saab, seda realistlikumaks muutub genereeritud pilt.

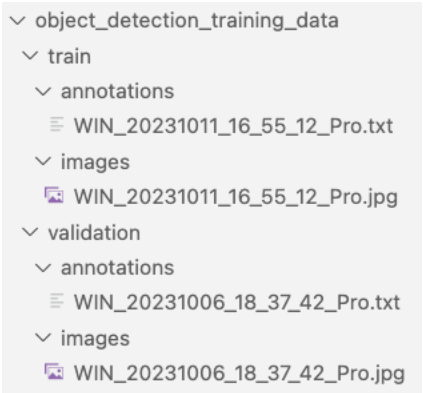


Joonis 6. Generatiivse vastandvõrgu generaatori töö arenemine ajas.

GAN-i treenimine isiklikul arvutil osutus keerukaks, sest nõuab palju graafilist ressurssi, mida autori 2020 aasta Apple Macbook Air M1 oma integreeritud 7-tuumalise graafikakaardiga eriti pakkuda ei suutnud. Seega võttis igakordne programmikoodi käivitamine ja lõpptulemuseni jõudmine keskmiselt 14 tundi. Kui arvuti vahepeal mingil põhjusel välja või puhkerežiimile lülitus, tuli algusest alustada. Seega on protsess väga aeganõudev, vajab palju kannatust ning loomulikult oleks etem hankida ligipääs võimsamale, spetsiaalselt masinõppe jaoks mõeldud graafikakaardile. Seda eriti, kuna andmeid sooviks genereerida suurema eraldusvõimega.

6.3 Biomarkerite objektituvastusmudeli loomine

Objektituvastusmudeli treeningandmete ettevalmistuseks võeti kasutusele peatükis 6.2.1 kirjeldatud transformatsioonide (ka algse pilditöötluse) käigus loodud CSV failid piltide märgendusandmetega, mida treenimisel kasutada soovitakse. Seejärel laeti S3 hoidlast kettale kõik failis esindatud pildid, kusjuures 80% eraldati suvaliselt treeningandmete kausta (*train*) ning 20% kontrollandmete kausta (*validation*). Ka siin pidi märgendusandmed teisendama YOLO arhitektuuriga ühilduvale kujule. Iga pildifaili kohta tuli luua samanimeline tekstifail, kus iga rida esindab ühe märgenduskasti andmeid (Lisa 6). Lõplik treeningandmete kataloogistruktuur koos näiteandmetega on esitatud Joonisel 7.



Joonis 7. Objektituvastusmudeli treeningandmete kataloogistruktuur.

Nagu objektituvastusmudelite treenimisele omane, võeti kasutusele n-ö ülekantav õpe (ing. k. *transfer learning*). See tähendab, et mudel jätkab treenimist teise mudeli seatud kaalude järgi. Algne mudel pärines *ImageAI* teegi dokumentatsioonist [35].

Iga epohhi ehk mudeli treeningtsükli tagant teostab *ImageAI* mudeli täpsuse analüüsi. Kui keskmine täpsus (mAP, ing. k. *mean average precision*) on kõrgem kui varasemate epohhide puhul, asendatakse salvestatud mudel uuega. Tsükkel kordub, kuni etteantud arv epohhe on täidetud või programm katkestatakse. Salvestatud mudelit saab otse kasutada piltidelt treenitud objektide tuvastamiseks.

Mudeli treeningu jooksul tuli silma peal hoida iga epohhi järgsel keskmisel täpsusel ning kaofunktsiooni väärtustel. Kui hakkas tunduma, et täpsus väheneb või kadu suureneb, oli mõttekas treeningprotsess lõpetada, sest see on märk ülesobitumisest. Keskmiselt saavutasid mudelid lühiajaliselt kõrgeima täpsuse ligikaudu 10 epohhiga.

Nagu GAN puhul, võttis ka objektituvastusmudeli treenimine väga palju aega, sest kasutatud arvutil sai treening toimuda ainult protsessori-, mitte graafikakaardipõhiselt. See oli takistuseks kõrgema täpsusega mudelite saavutamisel. Lisaks sai kasutada ainult märgendatud andmeid, seega sõltus andmestiku suurus sellest, kui palju spetsialistid olid selleks ajaks jõudnud märgendada. Nagu mainitud peatükis 2.4, siis pole tuvastusmudeli kõrge täpsuse saavutamine osa lõputöö otsesest eesmärgist ning seega ei mõjuta väiksem arv epohhe ja vähem algseid treeningandmeid lõputöö tulemusi.

7 Tulemused

Lõputöö eesmärgiks oli uurida, kuidas generatiivsed meetodid aitavad lahendada treeningandmete koguse ja tasakaalustatuse probleemi masinõppe mudeli, täpsemalt juuksejuure mikrofotodelt biomarkerite tuvastuse mudeli loomisel.

Eeltööna paigaldati treeningandmete märgenduskeskkond ning seadistati märgendusprojekt. Spetsialistide poolt märgendatud treeningandmete põhjal loodi esialgne objektituvastusmudel, et genereeritud andmete lisatud väärtust oleks hiljem võimalik kvantifitseerida. Generatiivsetes meetoditest võeti kasutusele piltide transformatsioonid ning loodi algne generatiivne sügavõppemudel GAN.

Märgenduskeskkond *Label Studio* seati üles privaatselt hostitud veebirakendusena ning sellele on juuksejuure mikrofotode märgendamiseks spetsialistidel ööpäevaringne ligipääs. Igal märgendajal on oma kasutaja ning uusi kasutajaid saab luua vaid ainulaadse ligipääsulingiga. Märgendatavaid ja märgendatud pilte saab filtreerida erinevate atribuutide, näiteks märgendite, märgendanud kasutajate, pildinime jpm. põhjal.

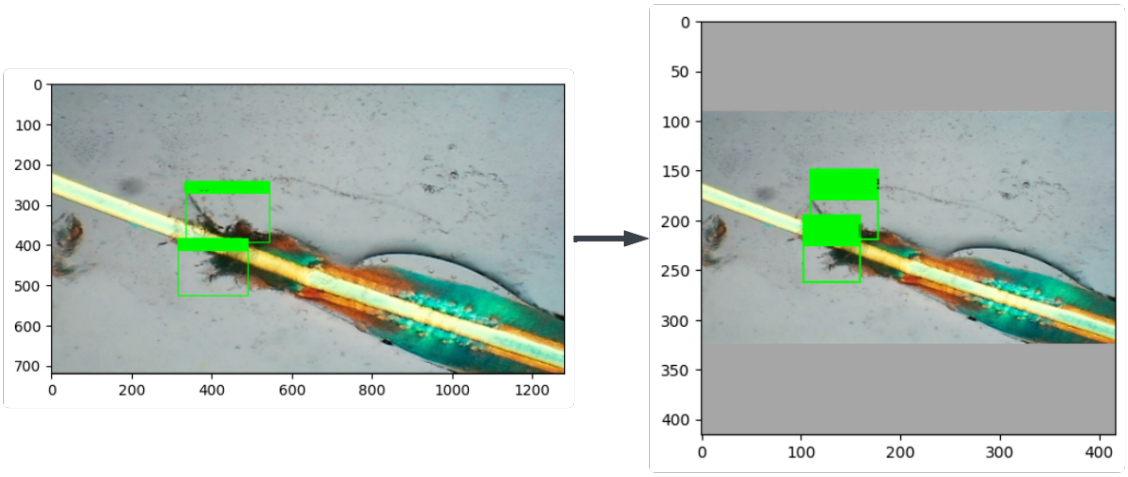
Joonisel 8 on näide sellest, milline näeb välja üksiku pildi märgendamise vaade *Label Studio*s. Märgendusklasside nimed on nende varjamiseks joonisel kaetud. Igaühe kohta on klaviatuuril otsetee, mida saab kasutada kiireks ja mugavaks märgendamiseks.

Joonis 8. *Label Studio* pildimärgendusvaade.

Piltandmeid hoiustatakse S3 hoidlas ning kasutaja- ja märgendusandmeid säilitatakse *Label Studio*ga ühendatud PostgreSQL andmebaasis. Märgendusandmeid saab sobivas formaadis paari klikiga failina eksportida kas edasiseks töötluseks või otse mudelite treenimisel kasutamiseks. Lisas 3 on toodud näide CSV-formaadis eksporditud märgendusandmetest. See sisaldab pildifaili asukohta S3 hoidlas, pildile vastavate märgenduskastide vasaku alumise nurga koordinaatide suhet pildi enda laiuse ja kõrgusega, kasti laiust ja kõrgust, kasti märgendusklassi ning muid vähemtähtsaid metaandmeid.

Objektituvastusmudeli treeningprotsessi lihtsustamiseks eksporditud andmed esmalt töödeldakse. Skript laeb algsed piltandmed mälusse, vähendab nende mõõtmeid nii, et pildi maksimaalne laius ja kõrgus oleksid seadistatav arv piksleid (autor kasutas väärtuseks 416) ning polsterdab vajadusel pildi ruudukujuliseks. Pildituvastusmudelil on sellisel kujul pilte lihtsam töödelda. Töödeldud pilt laetakse uuesti S3 hoidlasse GEN kausta üles. Seejuures on *imgaug* teegi abil muudetud vastavalt ka piltide märgenduskastide koordinaadid. Töödeldud andmestik salvestatakse uue CSV failina, kus on piltide asukohad S3 hoidlas ning neile vastavate märgenduskastide uued koordinaadid.

Joonisel 9 on näide pildi algsest ja töödeldud kujust koos uuendatud märgenduskastidega. Telgedel on märgitud piltide mõõtmed pikslites. Kastide klassimärgendid on näite puhul ebaolulised ja on seetõttu eemaldatud.

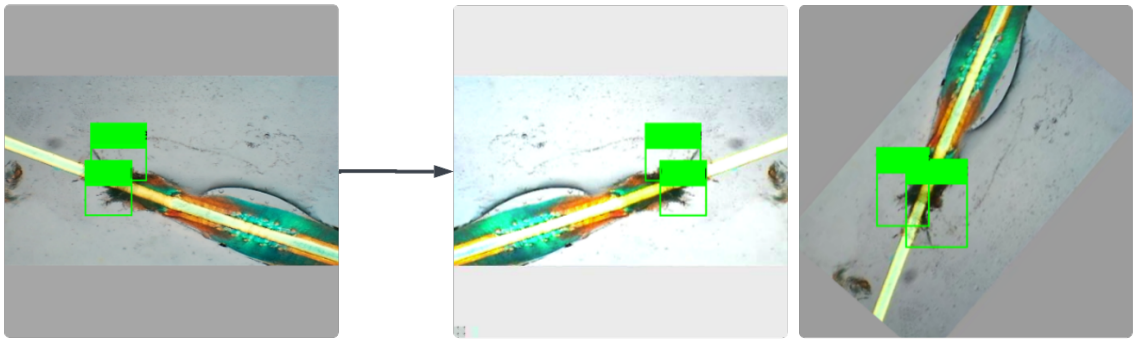


Joonis 9. Näide algsest pildist (vasakul) ning töödeldud pildist (paremal) koos märgenduskastidega.

Treeningandmete juurde genereerimisel transformatsioonide teel saab määrata, kui mitu transformatsiooni korraga ühele pildile rakendatakse ning mitu uut eksemplari ühe

lähtepildi kohta genereeritakse. Transformatsioonide valikus on suuruse muutmine, pööramine, transleerimine (külgedele või üles-alla liigutamine), horisontaalsed ja vertikaalsed peegeldused, heleduse muutmine ning hägustamine. Nende hulgast tehakse suvaline valik määratud arvu transformatsioonide rakendamiseks ning protsessi korratakse määratud arv kordi, et genereerida soovitud arv uusi pilte. Samaaegselt uuendatakse vajadusel märgenduskastide koordinaate, kui need on transformatsiooni käigus liikunud. Nagu pildi algsel töötlusel, salvestatakse transformeeritud pildid S3 hoidlasse ning luuakse CSV fail uute märgendusandmetega (näide Lisas 4).

Joonisel 10 on näide algsest pildist ja märgendusandmetest transformatsioonide teel genereeritud tulemused. Ka siin on märgendusklassid ebaolulised ning on eemaldatud.



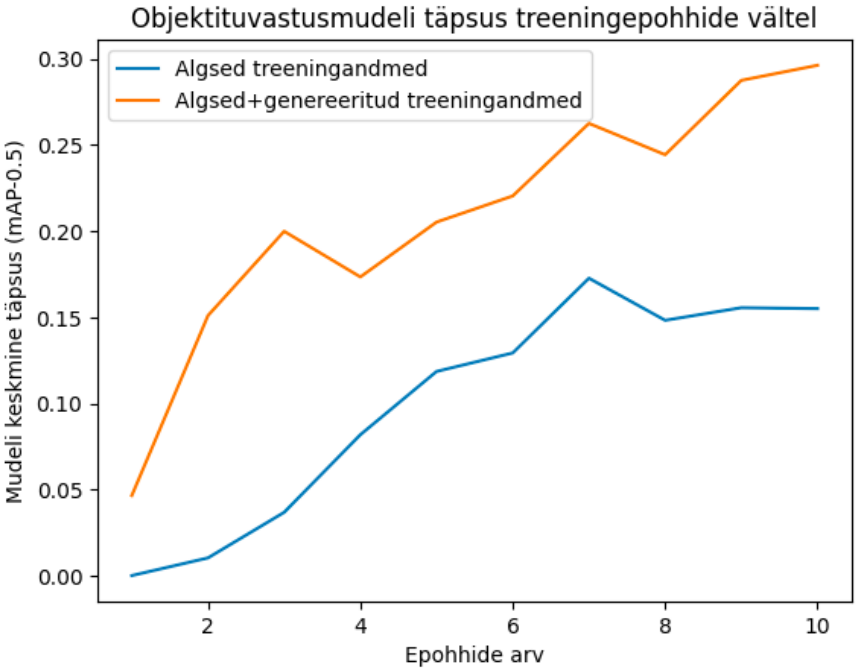
Joonis 10. Näide algsest pildist (vasakul) transformatsioonide teel genereeritud piltidest (keskel ja paremal) koos märgenduskastidega.

YOLO tüüpi objektituvastusmudelit treeniti kolme erineva märgendiklassi peal. Mudeli loomise hetkeks oli *Label Studio*s märgendatud 3088 pilti. Märgendatud andmed jaotati suvaliselt 80% treeningandmeteks ning 20% kontrollandmeteks, mille peal mudeli täpsust valideerida. Andmete genereerimisel transformatsioonide teel on suureks plussiks, et saab säilitada märgendusandmeid. Seega saab sel meetodil genereeritud andmeid lisatööta kaasata objektituvastusmudeli treeningandmestikku.

Lõputöö raames loodava objektituvastusmudeli puhul ei omanud tähtsust, et tegemist oleks kõige efektiivsema või täpsema mudeliga. Selle peamiseks eesmärgiks oli võrrelda, kuidas genereeritud treeningandmed mõjutavad mudeli treeningprotsessi ning saavutatud täpsust, kusjuures olulisem on algandmete ja genereeritud andmete peal treenitud mudelite võrdlus kui saavutatud mudel ise.

Joonisel 11 on esitatud treeningepohhide vältel objektituvastusmudeli täpsuse graafik. Sinisega on esitatud algse töödeldud andmestiku ehk 2450 treeningeksemplari ja 638

kontrolleksemplari peal treenitud mudel. Oranžiga on näidatud algandmete ja transformatsioonide teel genereeritud andmete peal treenitud mudel, kusjuures iga algse eksemplari kohta genereeriti kaks tükki juurde. Sel mudelil oli ligikaudu kolm korda rohkem, ehk 7377 treeningeksemplari ja 1874 kontrolleksemplari.



Joonis 11. Objektituvastusmudeli täpsuse (mAP) areng epohhide vältel algsete ja genereeritud treeningandmetega.

Mudeli täpsuse ühikuna on kasutatud mAP-d. See põhineb märgenduskastide ühisosa suhtel ühendist (IoU, ing. k. *intersection over union*), täpsusele seatud IoU künnisel (nt. 0.5 või 50%) ning sellega lähedalt seotud kolmel mõõdikul: segadusmaatriks (ing. k. *confusion matrix*), tõeste positiivsete oletuste osakaal kõigist positiivsetest oletustest (ing. k. *precision*) ning tõeste positiivsete oletuste osakaal kõigist oletustest (ing. k. *recall*). [40]

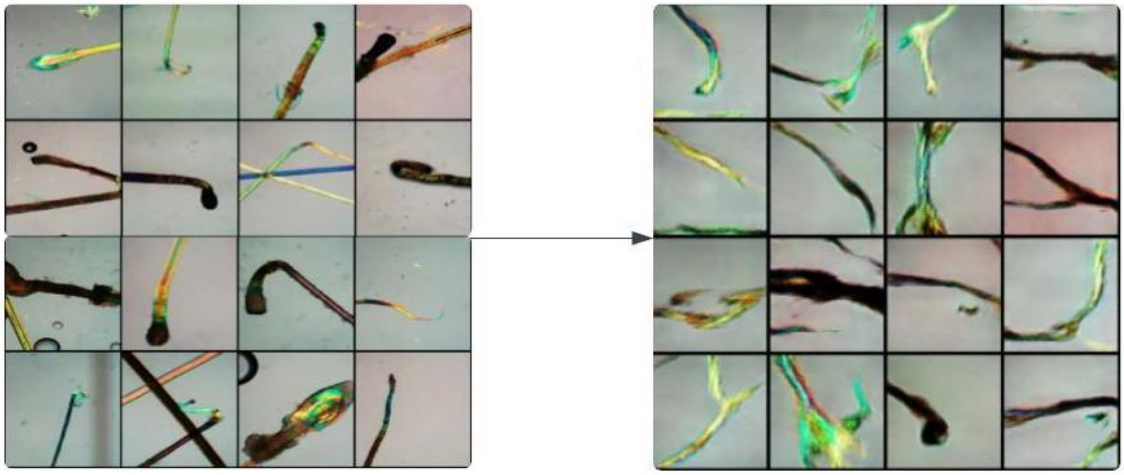
Genereeritud andmetega täiendatud treeningandmestikuga mudel saavutas kümne epohhi jooksul IoU 50% künnisega maksimumtäpsuse 0.296 mAP. Algandmete peal treenitud mudel saavutas sama epohhide arvuga maksimumtäpsuse 0.173 mAP. See tähendab, et genereeritud andmetega treenitud mudeli saavutatud täpsus sama aja jooksul oli ligikaudu 71% kõrgem algsete andmetega treenitud mudelist.

Saavutatud tulemus on heaks näitajaks, kui suur mõju on treeningandmestiku koguse suurendamisel ja ka treeningandmestiku väikeste muudatustega täiendamisel. Tuleb aga

arvestada, et genereeritud andmete näol on siiski tegemist algandmete erinevate versioonidega. Sellest tulenevalt on kontrollandmed ja treeningandmed väga sarnased ning saavutatud täpsus võib paremana näida kui see reaalses olukorras oleks. Lahenduseks võiks eraldada hilisemaks kordusvalideerimiseks andmed, mida ei kasutata ei genereerimisel ega mudeli treenimisel.

Loodud objektituvastusmudelit saab kasutada märgendamata juuksejuure mikrofotodelt treenitud märgendusklasside tuvastamiseks olukordades, kus saavutatud kõrgeim täpsus (hetkel loodud mudelite puhul 0.296 mAP) on rahuldav.

Teiseks generatiivseks meetodiks oli generatiivse masinõppemudeli GAN loomine. Esialgse GANi treeningu tulemuste näited on esitatud Joonisel 12.



Joonis 12. Näited generatiivse vastandvõrgu treeningandmetest (vasakul) ja genereeritud andmetest (paremal).

GAN-i genereeritud piltide kujul on esialgu küll tegu väga madala eraldusvõimega, 128x128 piksli suuruste piltidega, kuid juba on näha võrdlemisi häid tulemusi. Mitmed genereeritud pildid on Joonisel 12 vasakpoolsete reaalsete juuksejuurte kuju ja värviga. Kuna GAN-i treeniti olude sunnil väiksemate piltide peal, kui need tegelikult treeningandmetena võiksid olla, siis sellised tulemused on õigustatud. Edasistel üritustel ning ressursi olemasolul tuleks proovida luua treeningandmeteks sobivas suuruses pilte.

Selleks, et GAN-i poolt genereeritud pilte kasutada tuvastusmudeli treenimiseks, tuleks ka need ära märgendada, sest erinevalt transformatsioonidest ei ole võimalik algseid märgendeid genereeritud andmetele tõlgendada. Raskeks võib nende märgendamise teha asjaolu, et madala eraldusvõimega, genereeritud piltidelt ei pruugi olla kerge tuvastada

reaalse juuksejuure osi. Seega tuleb teha lähedalt koostööd spetsialistidega, et määratleda, millised genereeritud piltidest on päriselt kõige realistlikumad.

Peatükis 2.4 nimetati funktsionaalsed ja mittefunktsionaalsed nõuded, mida lahendus peaks täitma. Tabelis 2 on esitatud nimetatud nõuete tulemuste ja teostuse kokkuvõte.

Tabel 3. Ülevaade lahenduse nõuete teostustest ja tulemustest.

Nõue	**Teostus**	**Tulemus**
Funktsionaalsed nõuded		
Andmetele saab lisada märgendeid.	Märgenduskeskkonna Label Studio privaatselt hostitud teenus AWS abil.	Märgendajatel on ööpäevaringne ligipääs märgendatavatele piltidele ja varasematele märgendusandmetele.
Andmeid saab eksportida masinloetaval kujul.	Label Studioga tuleb kaasa märgendusandmete eksportimise funktsioon.	Andmeid saab eksportida erinevates üldlevinud formaatides, nt. json, CSV, XML
Treeningandmeid kogust saab suurendada generatiivsetel meetmetel.	Piltide transformeerimise skript ja generatiivne vastandvõrk.	GAN genereerib andmeid, millest mõned on rohkem juuksejuure moodi kui teised.
Võimaldab säilitada olemasolevaid märgendeid genereeritud piltidel.	Piltide transformatsioonidel säilitatakse vastavad märgendid, kuid GANi puhul mitte.	Piltide transformatsioonid pakuvad võimaluse ilma lisatööta treeningandmete kogust suurendada. GANide puhul see võimalik ei ole, seega peab GANi genereeritud pilte eraldi märgendama.
Mittefunktsionaalsed nõuded		
Märgendid on turvaliselt säilitatud.	S3 hoidla ja RDS PostgreSQL andmebaasi ühendus Label Studioga.	Märgendusandmed ei kao, kui Label Studio keskkonnaga on probleeme. Andmebaasile ja -hoidlale on juurdepääs AWS-is vaid selleks autoriseeritud isikutel.
Märgenduskeskkonnale on ligipääs vaid tuvastatavatel kasutajatel.	Label Studioga tuleb kaasa autentimis-funktsioon, kusjuures kasutajate andmed säilitatakse andmebaasis.	Privaatselt hostitud Label Studiosse saavad kasutajaid luua vaid inimesed, kellel on selleks ettenähtud link. Märgendusprojektid on nähtavad vaid sisselogitud kasutajatele.

7.1 Edasiarendused

Loodud lahenduse erinevatel komponentidel on rohkelt võimalikke edasiarendusi. Siinkohal pakutakse välja mõned valikud taristu, märgenduskeskkonna, generatiivse masinõppe mudeli ja pildituvastusmudeliga seotud parendused.

Taristu rajamise saab viia üle IaC (*Infrastructure as Code,* taristu-kui-kood) lahendusele, kasutades näiteks selleks eesmärgiks laialt kasutatavat tööriista Terraform. Seda on kasulik teha, sest kui peaks ette tulema olukord, kus keskkonna peab uuesti üles seadma, on manuaalne protsess aeganõudev, veaaldis ning inimesena võib alati midagi tähtsat ununeda. Lisaks on lihtsam jälgida, kui taristus on tehtud ootamatuid muutuseid.

Label Studio võimalike edasiarenduste ja parenduste hulka kuulub näiteks masinõppemudeli lisamine, mis õpiks aegamööda uutele andmetele ise automaatselt märgendeid pakkuma. Sel juhul, mida paremaks mudel treenitud saab, seda kiiremaks läheb märgendusprotsess, sest inimesel on vaja vaid pakutud märgendid kiirelt üle käia ning parandusi teha, kus tarvis.

Lõputöös loodud GAN-il on palju võimalikke edasiarendusi. Esiteks on võimalik hakata genereerima kõrgema eraldusvõimega pilte. Teiseks võib selle arendada cGAN-iks ehk tingimuslikuks GAN-iks. Nii saaks genereerida andmeid etteantud märgenditega ja aidata veelgi parandada andmete tasakaaslustamatuse probleemi.

Biomarkerite tuvastusmudeli loomulikuks edasiarenduseks on teiste biomarkerite lisamine tuvastatavate märgendite hulka. Seejuures oleks huvitav pöörata tähelepanu biomarkeritele, mida esineb palju vähem kui teisi, ning vaadelda kuidas generatiivsete meetodite rakendamine nende kui vähem tasakaalustatud märgendusklasside tuvastustäpsust parandab.

8 Kokkuvõte

Käesolevas lõputöös uuriti lahendusi masinõppe mudelite treeningandmete koguse, kvaliteedi ja tasakaalustatuse probleemile. Seejuures vaadeldi, kuidas generatiivsed meetodid võivad olla abiks ühe kindla masinõppe mudeli, täpsemalt juuksejuure biomarkerite pildituvastusmudeli, loomisel ja parendamisel.

Teoreetilises osas analüüsiti, kuidas treeningandmete kogus, kvaliteet ja tasakaalustatus võivad mõjutada masinõppe mudelite treeningprotsessi ja tulemusi. Esitati nõuded lõputöös loodavale lahendusele, mille raames tehakse algust juuksejuure biomarkerite tuvastusmudeli ehitamisega ning katsetatakse erinevaid meetodeid treeningandmete juurde genereerimiseks mudeli täpsuse tõstmise eesmärgil. Seejuures analüüsiti ja tehti valik kasutatavate generatiivsete meetodite osas.

Praktilises osas seati üles meeskonna poolt ligipääsetav märgenduskeskkond, juuksejuurte biomarkerite tuvastuse algne mudel ning loodi piltide klassikalise transformeerimise lahendus ja generatiivne vastandvõrk.

Selgus, et generatiivsed meetodid on sobiv lahendus masinõppe mudelite treeningandmete juurde genereerimiseks, kui reaalsete andmete kogumine on keeruline. Sel viisil treeningandmete koguse suurendamine andis häid tulemusi objektituvastusmudeli saavutatud täpsusel. Näiteks piltide transformatsioonidega suudeti sama arvu treeningtsüklite jooksul mudeli täpsust parandada 71%.

Generatiivsed masinõppemudelid on tänapäeval väga levinud uurimisteema. Mistahes masinõppega seonduvate projektide korral leiab aina rohkem viise, kuidas generatiivsed meetodid enda kasuks tööle panna. Töös käsitletud generatiivsed meetodid ja nende teostus on rakendatavad mistahes masinõppe mudelitele, kus piltvormis treeningandmete juurde genereerimine on mudeli eesmärgi vaates asjalik.

Kasutatud kirjandus

[1] G. Varun, L. Peng jt., „Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs," *JAMA,* kd. 316, nr 22, lk. 2402-2410, 2016.

[2] A. L. Beam ja I. S. Kohane, „Big data and machine learning in health care," *Jama,* kd. 319, nr 13, lk. 1317-1318, 2018.

[3] Y. Roh, G. Heo ja S. E. Whang, „A survey on data collection for machine learning: a big data-ai integration perspective," *IEEE Transactions on Knowledge and Data Engineering,* kd. 33, nr 4, lk. 1328-1347, 2019.

[4] D. Srihith ja I. V. Sai, „Training Data Alchemy: Balancing Quality and Quantity in Machine Learning Training," *Journal of Network Security and Data Mining,* kd. 6, nr 3, lk. 7-10, 2023.

[5] A. Halevy, P. Norvig ja F. Pereira, „The unreasonable effectiveness of data," *IEEE intelligent systems,* kd. 24, nr 2, lk. 8-12, 2009.

[6] E. Vali-Betts, K. J. Krause jt., „Effects of Image Quantity and Image Source Variation on Machine Learning Histology Differential Diagnosis Models," *Journal of Pathology Informatics,* kd. 12, nr 1, lk. 5, 2021.

[7] V. Gudivada, A. Apon ja J. Ding, „Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations," *International Journal on Advances in Software,* kd. 10, nr 1, lk. 1-20, 2017.

[8] C. Charrier, O. Lezoray ja G. Lebrun, „Machine learning to design full+reference image quality assessment algorithm," *Signal processing: Image communication,* kd. 27, nr 3, lk. 209-219, 2012.

[9] N. Japkowicz, „Learning from imbalanced data sets: a comparison of various strategies," *AAAI workshop on learning from imbalanced data sets,* kd. 68, lk. 10-15, 2000.

[10] J. M. Johnson ja T. M. Khoshgoftaar, „Survey on deep learning with class imbalance," *Journal of Big Data,* kd. 6, nr 1, lk. 1-54, 2019.

[11] B. Krawczyk ja M. Wozniak, „Cost-sensitive neural network with roc-based moving threshold for imbalanced classification," *Intelligent Data Engineering and Automated Learning–IDEAL 2015: 16th International Conference, Wroclaw, Poland, October 14-16, 2015, Proceedings 16*, Springer International Publishing, 2015, lk. 45-52.

[12] A. Krizhevsky, I. Sutskever ja G. E. Hinton, „Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems,* kd. 25, 2012.

[13] L. Perez ja J. Wang, „The effectiveness of data augmentation on image classification using deep learning," *Convolutional Neural Networks Vis. Recognit,* kd. 11, lk. 1-8, 2017.

[14] J. Ding, B. Chen, H. Liu ja M. Huang, „Convolutional Neural Network With Data Augmentation for SAR Target Recognition," *IEEE Geoscience and Remote Sensing Letters,* kd. 13, nr 3, lk. 364-368, 2016.

[15] X. Hao, L. Liu, R. Yang, L. Yin, L. Zhang ja X. Li, „A Review of Data Augmentation Methods of Remote Sensing Image Target Recognition," *Remote Sensing,* kd. 15, nr 3, lk. 827, 2023.

[16] A. Oussidi ja A. Elhassouny, „Deep generative models: Survey," *2018 International conference on intelligent systems and computer vision (ISCV)*, 2018.

[17] C. Bowles jt., „GAN Augmentation: Augmenting Training Data using Generative Adversarial Networks," *arXiv preprint arXiv,* kd. 1810, nr 10863, 2018.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville ja Y. Bengio, „Generative adversarial nets," *Advances in neural information processing systems,* kd. 27, 2014.

[19] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta ja A. A. Bharath, „Generative adversarial networks: An overview," *IEEE signal processing magazine,* kd. 35, nr 1, lk. 53-65, 2018.

[20] „The Discriminator," Google, 2022. [Võrgumaterjal]. Allikas: https://developers.google.com/machine-learning/gan/discriminator. Kasutatud 19.11.2023

[21] A. Shrivastava jt., „Learning from simulated and unsupervised images through adversarial training," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[22] M. Frid-Adar jt., „Synthetic data augmentation using GAN for improved liver lesion classification," *2018 IEEE 15h international symposium on biomedical imaging (ISBI 2018)*, 2018.

[23] M. Mirza ja S. Osindero, „Conditional generative adversarial nets," *arXiv preprint arXiv,* kd. 1411, nr 1784, 2014.

[24] T. Karras jt., „Progressive Growing of GANs for Improved Quality, Stability and Variation," *arXiv preprint arXiv,* kd. 1710, nr 10196, 2017.

[25] P. Isola jt., „Image-to-Image Translation with Conditional Adversarial Networks," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[26] I. Shafkat, „Intuitively Understanding Variational Autoencoders," Towards Data Science, 4.2.2018. [Võrgumaterjal]. Allikas: https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf. Kasutatud 28.11.2023.

[27] J. Altosaar, „Tutorial - What is a Variational Autoencoder?," Jaan Altosaar, 16.08.2016. [Võrgumaterjal]. Allikas: https://jaan.io/what-is-variational-autoencoder-vae-tutorial/. Kasutatud 28.11.2023.

[28] A. Parakatta, „VAE v/s GAN - A case study," Medium, 15.06.2023. [Võrgumaterjal]. Allikas: https://medium.com/@parakatta/vae-v-s-gan-a-case-study-b09c7169ac02. Kasutatud 30.11.2023.

[29] K. Ahirwar, „A Very Short Introduction to Diffusion Models," Medium, 26.09.2023. [Võrgumaterjal]. Allikas: https://kailashahirwar.medium.com/a-very-short-introduction-to-diffusion-models-a84235e4e9ae. Kasutatud 30.11.2023.

[30] J. Betker jt., „Improving Image Generation with Better Captions," OpenAI, 2023.

[31] A. Gainetdinov, „Diffusion Models vs. GANs vs. VAEs: Comparison of Deep Generative Models," Medium, 12.05.2023. [Võrgumaterjal]. Allikas: https://pub.towardsai.net/diffusion-models-vs-gans-vs-vaes-comparison-of-deep-generative-models-67ab93e0d9ae. Kasutatud 01.12.2023.

[32] K. Kaewsanmua, „Data Labeling Software: Best Tools for Data Labeling," Neptune AI, 22 8 2023. [Võrgumaterjal]. Allikas: https://neptune.ai/blog/data-labeling-software. Kasutatud 30.11.2023.

[33] „Python vs. Matlab: Which language is right for you?," IONOS, 18.10.2023. [Võrgumaterjal]. Allikas: https://www.ionos.com/digitalguide/websites/web-development/python-vs-matlab/. Kasutatud 29.11.2023.

[34] „imgaug," 2020. [Võrgumaterjal]. Allikas: https://imgaug.readthedocs.io/en/latest/. Kasutatud 16.10.2023.

[35] „Custom Object Detection: Training and Inference,“ ImageAI, 2022. [Võrgumaterjal]. Allikas: https://imageai.readthedocs.io/en/latest/customdetection/index.html. Kasutatud 02.11.2023.

[36] S. Paniagua, „TensorFlow vs. PyTorch: A Pragmatic Approach to Deep Learning Framework Selection,“ LinkedIn, 23 10 2023. [Võrgumaterjal]. Allikas: https://www.linkedin.com/pulse/tensorflow-vs-pytorch-pragmatic-approach-deep-sam-paniagua/. Kasutatud 03.01.2024.

[37] „Boto3 documentation,“ AWS, 2024. [Võrgumaterjal]. Allikas: https://boto3.amazonaws.com/v1/documentation/api/latest/index.html. Kasutatud 03.01.2024.

[38] H. Park, „How to Host Label Studio on AWS Elastic Beanstalk with Persistent Storage and RDS PostgreSQL,“ Letr, 03.11.2022. [Võrgumaterjal]. Allikas: https://www.letr.ai/blog/tech-20220706. Kasutatud 05.09.2023.

[39] N. Inkawhich, „DCGAN Tutorial,“ PyTorch, 2023. [Võrgumaterjal]. Allikas: https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html. Kasutatud 15.06.2023.

[40] D. Shah, „Mean Average Precision (mAP) Explained: Everything You Need to Know,“ V7, 07.03.2022. [Võrgumaterjal]. Allikas: https://www.v7labs.com/blog/mean-average-precision. Kasutatud 30.12.2023.

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

Mina, Marion Claudia Strižˇ

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Generatiivsete meetodite kasutamine masinõppe mudeli treeningandmete koguse suurendamiseks juuksejuure mikrofotode näitel“, mille juhendaja on Toomas Lepikult, PhD
   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.

04.01.2024

1 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 – Koodinäidete GitHub-i link

https://github.com/marionstriz/generative-methods-for-ml/tree/main

Lisa 3 – IAM poliis S3 hoidlast esemete loendamiseks, lisamiseks ja kustutamiseks

```
{
	"Version": "2012-10-17",
	"Statement": [
		{
			"Sid": "VisualEditor1",
			"Effect": "Allow",
			"Action": [
				"s3:ListBucket",
				"s3:GetObject",
				"s3:PutObject",
				"s3:DeleteObject"
			],
			"Resource": [
				"arn:aws:s3:::hair-root-images",
				"arn:aws:s3:::hair-root-images/*"
			]
		}
	]
}
```

Lisa 4 – Näide CSV-formaadis Label Studiost eksporditud märgendusandmetest tabeli kujul

annotation_id	annotator	created_at	create	image	label	lead_tir	updated_at
17534	7	2023-09-27T09:11:	69415	s3://hair-root-images/RAW/100723/Untitled 19535.jpg	[{"x":84.56082108150 72,"y":50.54704595185996,"width":9.231400437636765,"height":32.6039387308533	69.585	2023-09-27T09:11:36.695
17538	7	2023-09-27T09:16:	69418	s3://hair-root-images/RAW/100723/Untitled 19538.jpg		7.019	2023-09-27T09:16:45.908
17334	7	2023-09-25T13:37:	69420	s3://hair-root-images/RAW/100723/Untitled 19540.jpg	[{"x":28.43271334792123,"y":29.978118161925604,"width":27.201859956236326,"height":70.0218818380	34.8	2023-09-25T13:37:07.392
17335	7	2023-09-26T08:09:	69421	s3://hair-root-images/RAW/100723/Untitled 19541.jpg	[{"x":40.00213869338207,"y":65.86433260393873,"width":5.415754923413566,"height":10.9409190371 99	66738.8	2023-09-26T08:09:28.261
17339	7	2023-09-26T08:10:	69425	s3://hair-root-images/RAW/100723/Untitled 19545.jpg		4.657	2023-09-26T08:10:58.368
17340	7	2023-09-26T08:11:	69426	s3://hair-root-images/RAW/100723/Untitled 19546.jpg	[{"x":48.49701555059867,"y":69.36542669584244,"width":5.661925601750539,"height":13.347921225382	37.276	2023-09-26T08:11:36.423
17344	7	2023-09-26T08:13:	69430	s3://hair-root-images/RAW/100723/Untitled 19550.jpg	[{"x":40.12582056892779,"y":12.910284463894966,"width":17.355032822751 11,"height":32.82275711159	38.867	2023-09-26T08:13:32.028
17347	7	2023-09-26T08:19:	69433	s3://hair-root-images/RAW/100723/Untitled 19553.jpg	[{"x":10.955324286012 24,"y":72.42888402625822,"width":7.508205689277 9026,"height":12.2538292166	240.312	2023-09-26T08:19:15.786
17348	7	2023-09-26T08:20:	69434	s3://hair-root-images/RAW/100723/Untitled 19554.jpg	[{"x":84.43607869684 64,"y":49.234135667396 06,"width":4.923413566739613,"height":10.7221006564551	55.933	2023-09-26T08:20:12.545
17352	7	2023-09-26T08:22:	69438	s3://hair-root-images/RAW/100723/Untitled 19558.jpg	[{"x":25.60354015617644,"y":65.64551422319474,"width":17.355032822757114,"height":26.9146608315	14.191	2023-09-26T08:22:21.467
17354	7	2023-09-26T08:24:	69440	s3://hair-root-images/RAW/100723/Untitled 19560.jpg	[{"x":38.03376749398161 4,"y":33.47921225382 9324,"width":9.477571115973745,"height":43.7636761487	34.222	2023-09-26T08:24:09.230
17356	7	2023-09-26T08:26:	69442	s3://hair-root-images/RAW/100723/Untitled 19562.jpg	[{"x":30.89236539488057,"y":9.846827133479213,"width":6.277352297592998,"height":62.144420131291	50.466	2023-09-26T08:26:04.091
17357	7	2023-09-26T08:26:	69443	s3://hair-root-images/RAW/100723/Untitled 19563.jpg	[{"x":50.83212405835628 5,"y":55.36105032822757,"width":8.492888402625816,"height":22.9759299781 1	15.949	2023-09-26T08:26:20.960
17360	7	2023-09-26T08:28:	69446	s3://hair-root-images/RAW/100723/Untitled 19566.jpg	[{"x":22.39934442861537,"y":34.57330415754923,"width":16.001094091903 72,"height":16.192560175054	40.966	2023-09-26T08:28:23.020
17361	7	2023-09-26T08:28:	69447	s3://hair-root-images/RAW/100723/Untitled 19567.jpg	[{"x":0.9846827133479212,"y":11.816192560175056,"width":20.678336980306348,"height":26.695842450	14.843	2023-09-26T08:28:38.942
17362	7	2023-09-26T08:28:	69448	s3://hair-root-images/RAW/100723/Untitled 19568.jpg		3.252	2023-09-26T08:28:43.085
17363	7	2023-09-26T08:29:	69449	s3://hair-root-images/RAW/100723/Untitled 19569.jpg	[{"x":34.70999936369454,"y":24.288840262582056,"width":14.524070021881 84,"height":20.5689277899 3	18.678	2023-09-26T08:29:02.700
17364	7	2023-09-26T08:29:	69450	s3://hair-root-images/RAW/100723/Untitled 19570.jpg		4.059	2023-09-26T08:29:07.595
17367	7	2023-09-26T08:30:	69453	s3://hair-root-images/RAW/100723/Untitled 19573.jpg	[{"x":16.61778014232032 4,"y":51.64113785557986,"width":24.493982494529543,"height":35.2297592997	27.947	2023-09-26T08:30:58.736
17370	7	2023-09-26T08:31:	69456	s3://hair-root-images/RAW/100723/Untitled 19576.jpg		12.488	2023-09-26T08:31:56.246
17372	7	2023-09-26T08:32:	69458	s3://hair-root-images/RAW/100723/Untitled 19578.jpg	[{"x":26.34138937299167,"y":38.07439824945296,"width":16.124179431072 21,"height":16.630196936542	19.382	2023-09-26T08:32:28.301
17374	7	2023-09-26T08:33:	69460	s3://hair-root-images/RAW/100723/Untitled 19580.jpg	[{"x":9.228351473930 92,"y":35.667396061269145,"width":27.940371991247265,"height":30.63457330415	29.841	2023-09-26T08:33:47.401
17377	7	2023-09-26T08:34:	69463	s3://hair-root-images/RAW/100723/Untitled 19583.jpg	[{"x":50.58807439824946,"y":47.26477024070022,"width":28.432713347921222,"height":23.6323851203 5	13.137	2023-09-26T08:34:54.961
17381	7	2023-09-26T08:37:	69467	s3://hair-root-images/RAW/100723/Untitled 19587.jpg	[{"x":17.72428884026258 4,"y":59.737417943107 22,"width":28.678884026258206,"height":24.0700218818	43.253	2023-09-26T08:37:48.427
17382	7	2023-09-26T08:38:	69468	s3://hair-root-images/RAW/100723/Untitled 19588.jpg	[{"x":37.66424634919737,"y":2.4070021881838075,"width":49.84956236323851,"height":34.1356673960 6	17.029	2023-09-26T08:38:06.352
17386	7	2023-09-26T08:39:	69472	s3://hair-root-images/RAW/100723/Untitled 19592.jpg	[{"x":32.1256712138940 9,"y":41.57549234135667,"width":27.201859956236323,"height":26.0393873085 3	25.885	2023-09-26T08:39:18.205
17389	7	2023-09-26T08:41:	69475	s3://hair-root-images/RAW/100723/Untitled 19595.jpg	[{"x":38.03734671224 5,"y":50.32822757111598,"width":11.323851203501086,"height":45.9518599562363	42.277	2023-09-26T08:41:13.373
17391	7	2023-09-26T08:41:	69477	s3://hair-root-images/RAW/100723/Untitled 19597.jpg		3.753	2023-09-26T08:41:51.130
17392	7	2023-09-26T08:41:	69478	s3://hair-root-images/RAW/100723/Untitled 19598.jpg		3.866	2023-09-26T08:41:55.886
17393	7	2023-09-26T08:42:	69479	s3://hair-root-images/RAW/100723/Untitled 19599.jpg		19.857	2023-09-26T08:42:16.724
17394	7	2023-09-26T08:42:	69480	s3://hair-root-images/RAW/100723/Untitled 19600.jpg	[{"x":38.52663910521311 4,"y":36.761487964989 06,"width":29.78665207877463,"height":19.4748358662 1	17.055	2023-09-26T08:42:34.662
17398	7	2023-09-26T08:44:	69484	s3://hair-root-images/RAW/100723/Untitled 19604.jpg	[{"x":7.508205688927 79,"y":32.822757111597376,"width":61.05032822757113,"height":31.7286652078774	21.581	2023-09-26T08:44:11.327
17400	7	2023-09-26T08:45:	69486	s3://hair-root-images/RAW/100723/Untitled 19606.jpg	[{"x":49.84956236323852,"y":54.48577680525164,"width":17.355032822757 11,"height":10.5032822757 11	48.392	2023-09-26T08:45:35.877
17401	7	2023-09-26T08:46:	69487	s3://hair-root-images/RAW/100723/Untitled 19607.jpg	[{"x":72.12800875273524,"y":20.568927789934357,"width":5.292669584245 0755,"height":8.75273522975	30.452	2023-09-26T08:46:07.293
17402	7	2023-09-26T08:46:	69488	s3://hair-root-images/RAW/100723/Untitled 19608.jpg	[{"x":34.21984530707042,"y":7.002188188380744,"width":10.708424507658641,"height":21.2253829321 66	33.677	2023-09-26T08:46:41.963
17403	7	2023-09-26T08:47:	69489	s3://hair-root-images/RAW/100723/Untitled 19609.jpg	[{"x":54.28355097337062,"y":6.345733041575492 5,"width":10.831509846827132,"height":19.9124726477	49.298	2023-09-26T08:47:32.465
17404	7	2023-09-26T08:48:	69490	s3://hair-root-images/RAW/100723/Untitled 19610.jpg	[{"x":23.01795265180305,"y":65.20787746170677,"width":11.816192560175057,"height":31.7286652078 7	42.779	2023-09-26T08:48:16.105
17409	7	2023-09-26T08:50:	69495	s3://hair-root-images/RAW/100723/Untitled 19615.jpg	[{"x":35.94231095541266 5,"y":30.415754923413573,"width":18.585886214442006,"height":13.3479212225	32.132	2023-09-26T08:50:48.464
17411	7	2023-09-26T08:52:	69497	s3://hair-root-images/RAW/100723/Untitled 19617.jpg	[{"x":40.49660106828618,"y":40.04376367614 88,"width":10.708424507658648,"height":11.1597374179 43	41.676	2023-09-26T08:52:04.920
17412	7	2023-09-26T08:52:	69498	s3://hair-root-images/RAW/100723/Untitled 19618.jpg	[{"x":57.60526436724724 4,"y":47.04595185995623,"width":11.077680525164 12,"height":13.5667396061 2	39.743	2023-09-26T08:52:45.577

Lisa 5 – Näide CSV-formaadis transformeeritud märgendusandmetest tabeli kujul

image	label
GEN/resized/RAW/210823/WIN_20220329_15_17_28_Pro.jpg	[{"x1": 135.63333129882812, "y1": 115.05000114440918, "x2": 144.73333740234375, "y2": 132.16666793823242, "label": "MARKER1"}, {"x1": 182.0, "y1": 94.90000009536743, "x2": 192.18333435058594, "y2": 136.28333282470703, "label": "MARKER1"}, {"x1": 120.68334197998047, "y1": 137.3666648864746, "x2": 132.81666564941406, "y2": 175.06666564941406, "label": "MARKER1"}]
GEN/resized/RAW/210823/WIN_20220329_17_04_42_Pro.jpg	[{"x1": 348.3999938964844, "y1": 120.68333435058594, "x2": 407.3333435058594, "y2": 190.45000457763672, "label": "MARKER2"}, {"x1": 148.84999084472656, "y1": 186.5500030517578, "x2": 289.4666748046875, "y2": 241.36666870117188, "label": "MARKER6"}, {"x1": 15.383332252502441, "y1": 239.1999969482422, "x2": 122.41667175292969, "y2": 300.9499969482422, "label": "MARKER3"}]
GEN/resized/RAW/210823/WIN_20220329_19_30_45_Pro.jpg	[{"x1": 308.3166809082031, "y1": 272.34999084472656, "x2": 338.6499938964844, "y2": 292.28334045410156, "label": "MARKER4"}, {"x1": 316.3333435058594, "y1": 213.41665649414062, "x2": 377.6499938964844, "y2": 273.43333435058594, "label": "MARKER2"}, {"x1": 93.81666564941406, "y1": 113.31666564941406, "x2": 141.26666259765625, "y2": 161.8499984741211, "label": "MARKER3"}]
GEN/resized/RAW/210823/WIN_20220329_19_33_37_Pro.jpg	[{"x1": 192.8333282470703, "y1": 168.13333129882812, "x2": 266.0666809082031, "y2": 220.34999084472656, "label": "MARKER7"}, {"x1": 160.76666259765625, "y1": 131.08333206176758, "x2": 192.8333282470703, "y2": 159.03333282470703, "label": "MARKER4"}]
GEN/resized/RAW/210823/WIN_20220329_19_33_55_Pro.jpg	[{"x1": 282.5333251953125, "y1": 141.26666641235352, "x2": 299.8666687011719, "y2": 176.8000030517578, "label": "MARKER1"}, {"x1": 223.60000610351562, "y1": 182.0, "x2": 259.1333312988281, "y2": 209.9499969482422, "label": "MARKER4"}]
GEN/resized/RAW/210823/WIN_20220329_19_34_07_Pro.jpg	[{"x1": 366.8166809082031, "y1": 211.68333435058594, "x2": 403.8666687011719, "y2": 267.8000030517578, "label": "MARKER2"}, {"x1": 208.86668395996094, "y1": 126.96666717529297, "x2": 253.5, "y2": 194.56666564941406, "label": "MARKER3"}]
GEN/resized/RAW/210823/WIN_20220329_19_35_11_Pro.jpg	[{"x1": 127.61666870117188, "y1": 160.76667022705078, "x2": 215.8000030517578, "y2": 233.56666564941406, "label": "MARKER4"}, {"x1": 204.31666564941406, "y1": 130.0, "x2": 251.76666259765625, "y2": 186.5500030517578, "label": "MARKER4"}, {"x1": 147.11666870117188, "y1": 119.60000038146973, "x2": 237.46665954589844, "y2": 209.9499969482422, "label": "MARKER3"}, {"x1": 186.5500030517578, "y1": 139.10000228881836, "x2": 316.98333740234375, "y2": 321.53334045410156, "label": "MARKER5"}]
GEN/resized/RAW/210823/WIN_20220329_19_35_47_Pro.jpg	[{"x1": 130.43333435058594, "y1": 227.7166748046875, "x2": 160.11666870117188, "y2": 255.23333740234375, "label": "MARKER4"}, {"x1": 153.8333282470703, "y1": 187.20000457763672, "x2": 177.88333129882812, "y2": 208.86666870117188, "label": "MARKER4"}, {"x1": 190.01666259765625, "y1": 159.68333435058594, "x2": 279.7166748046875, "y2": 195.21666717529297, "label": "MARKER4"}, {"x1": 171.1666717529297, "y1": 214.06666564941406, "x2": 182.0, "y2": 246.56666564941406, "label": "MARKER1"}]

Lisa 6 – YOLO märgendusformaadi selgitus ning näide

Iga rida esindab ühte märgenduskasti. Andmed on eraldatud tühikuga ning on esitatud järjekorras:

1. Märgendusklassi indeks listis, mis YOLO mudelile ette antakse
2. Märgenduskasti keskpunkti x-koordinaadi suhe pildi laiusest
3. Märgenduskasti keskpunkti y-koordinaadi suhe pildi kõrgusest
4. Märgenduskasti laiuse suhe pildi laiusest
5. Märgenduskasti kõrguse suhe pildi kõrgusest

Näitena on esitatud pildile nimega *photo-1234* vastavate märgendite tekstifail YOLO-formaadis:

photo-1234.txt

```
1 0.37865 0.54245 0.15104 0.20781
0 0.76923 0.21233 0.09538 0.13442
```